



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

**ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
CAMPUS ARAGÓN**

**DESARROLLO DE UN SISTEMA DE
VISUALIZACIÓN DE DINÁMICA
MOLECULAR
USANDO UN ENFOQUE
ORIENTADO A OBJETOS**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

P R E S E N T A N :

**ERNESTO SALAS RODRÍGUEZ
Y**

RICARDO SAÚL RUGERIO GUTIÉRREZ

ASESOR: DR. LUIS JAVIER ÁLVAREZ NOGUERA

MÉXICO

2005

m. 344382





Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

DEDICATORIAS

Dedico este trabajo principalmente a **DIOS**, por permitirme finalizar felizmente esta Tesis, que representa un gran esfuerzo y dedicación, gracias a él por alentarme en los momentos difíciles y cuidar de mis seres queridos mientras ocupaba tiempo en este proyecto.

A mis Padres:

Por todo el apoyo que durante tantos años de estudio me otorgaron, por sus desvelos y sus angustias, por su gran confianza, por su cariño, y por el tiempo en que con sus cuidados y consejos me hicieron una mejor persona; infinitamente gracias por todo su amor.

A mi madre: Dedico especialmente este trabajo a ella, por todo lo maravillosa que ha sido, por enseñarme sus ideales y sus sueños, por enseñarme a tener fe y por despertar en mi el deseo de ser Ingeniero.

A mi padre: Por su compañía y su amistad, por todo lo bueno que me ha enseñado y he aprendido de sus largas charlas.

A mis Hermanos:

Por compartir tantos momentos de alegría y de tristeza, por la compañía de cada uno y sus divertidas ocurrencias, por su apoyo, sus ideas, por ser como son y estar siempre a mi lado y sobre todo por su amor y su cariño.

A Lety: Por ser la hermana mas linda del mundo, por apoyarme tanto en las buenas y en las malas, por comprenderme y por toda la felicidad que me ha dado al compartir su vida y la de su familia conmigo, por los regaños y los buenos consejos.

A Max: Por ser mi hermano y mi amigo, por su gran cariño y por mostrarme sus ideas y puntos de vista siempre esperando aprender de mi, por ser como es, un gran hermano, inteligente y amable.

A Manuel: Mi hermano pequeño, por su inteligencia y su cariño, le dedico este trabajo por su esfuerzo en sus estudios para que siga adelante y que al igual que yo termine todo lo que se proponga.

A mi Abuelita:

Por sus preocupaciones y sus bendiciones cada que salgo de casa, por sus consejos y por todo el amor que me ha dado.

A mis Sobrinos y mi Cuñado:

Por su bonita familia y todas las alegrías que me han dado.

A Tania: Por ser tan linda, latosa e inteligente, y que con su presencia y alegría me hace feliz.

A Jorgito: Por ser un bebe muy lindo y tierno.

A Jorge: Por cuidar de su familia y con empeño seguir siempre adelante, por su amistad.

DEDICATORIAS

- A Lucero:** Por su amor incondicional, por las alegrías y tristezas que hemos compartido, por su apoyo, buenos consejos y comprensión, así como la gran felicidad que me da con su amistad, su sinceridad y su compañía.
- A Gaspar:** Por ser mi mejor amigo a lo largo de tanto tiempo, por todo lo que he aprendido a su lado, por la confianza y lealtad, por todos los buenos y malos tiempos y porque siempre ha estado en los momentos importantes de mi vida.
- A Ricardo:** Por ser un magnífico amigo y compañero. Por haber decidido emprender junto conmigo la elaboración de este proyecto de tesis tan importante para ambos y concluirlo, y por todos los buenos momentos pasados y por venir.
- A Caro:** Por su maravillosa amistad, y apoyo en todo momento, ya que siempre estuvo al tanto de nuestro progreso durante el desarrollo de este trabajo y por lo amable que ha sido con su inestimable ayuda.
- A mi sensei
Sergio Cabrales :** Por sus buenos consejos y enseñanzas.
- A mis amigos:** Rosalia, Edgar, Ramón, Elva, Ady, Enrique, Raúl, Roxana, Atenea, MaryCruz y Adry.
Por su amistad, compañía y buenos momentos.

Ernesto Salas Rodríguez.

AGRADECIMIENTOS

Agradezco sinceramente al Dr. Luis Javier Álvarez Noguera por todo el apoyo que me brindo durante el tiempo que fui becario en el Laboratorio de Simulación de Materiales de la UNAM bajo su dirección, así como por el apoyo, asesoría y ayuda en la realización de este trabajo de tesis.

A María del Carmen Valencia Vivanco por su asesoría en mi estancia en el laboratorio en el uso de el equipo de computo y a Jorge Enrique Sánchez Sánchez por su amabilidad y valiosa ayuda.

Agradezco a la Dirección General de Servicios de Cómputo Académico (DGSCA) y a la Dirección de Cómputo para la Administración Académica (DCAA) por las oportunidades que me dieron al formar parte de su plan de becarios durante 1996-1997 y que me condujeron al Laboratorio de Simulación de Materiales donde surgió el tema de la presente tesis.

Agradezco profundamente a la Universidad Nacional Autónoma de México y a la Escuela Nacional de Estudios Profesionales Campus Aragón por todo el conocimiento y apoyo recibidos.

A mis profesores de todos los niveles, básico, medio y superior, que han contribuido a mi formación profesional gracias por sus enseñanzas.

Ernesto Salas Rodríguez.

DEDICATORIAS:

Dedico este trabajo a mis **padres**:

Alberto y Guadalupe

Que han sabido guiarme por el buen camino y apoyarme en las situaciones difíciles, así como también han compartido conmigo mis triunfos y mis fracasos y por darme su cariño y una educación universitaria, que es el mejor regalo que cualquiera pueda recibir. Por que me enseñaron a obtener lo que quiero por mi mismo y con mi trabajo y no a solo extender las manos.

A mis abuelitos (q.e.p.d):

Abuelita Felicitas
Abuelito Pedro
Abuelito Leonardo

Por todo el cariño que me dieron, por que fue una enorme alegría tenerlos conmigo y por que estén donde estén ahora, estoy seguro de que comparten este éxito conmigo.

Abuelita Josefina

Gracias a DIOS que me dejó tenerla conmigo para compartir este gran logro con usted.

Ustedes son también parte de esto

A todos ustedes, los quiero mucho.

Y gracias por quererme

Ricardo.

AGRADECIMIENTOS:

A DIOS:

Porque sin EL... ni siquiera estaría aquí.

A mis padres:

Porque a ellos les debo todo, gracias a ellos pude estudiar y soy lo que soy....

A Caro:

Porque gracias a tu apoyo y a tus regaños..., porque gracias a ti... finalmente terminé... y porque a pesar de todo.... siempre has creído en mí y crees en mi.

A la Universidad Nacional Autónoma de México:

Porque gracias a ella tengo todo lo que tengo y soy todo lo que soy y por que digan lo que digan es la MEJOR.

Al Doctor Luis Javier Álvarez:

Por todo su apoyo, por su paciencia y por las muchas facilidades que nos prestó en la realización de nuestro trabajo.

A Carmen Valencia:

Por tu ayuda y el apoyo que nos brindaste sin los cuales hubiera sido aún más difícil.

Al Doctor Jorge Enrique Sánchez:

Por su apoyo y ayuda en la realización de nuestro trabajo.

A mis verdaderos profesores:

Porque son pocos los que en realidad merecen ser llamados así....

Profesora: Esperanza

Profesora: Magdalena

Profesor: Fernando

Profesor: Luis Ramírez

A mis verdaderos amigos: César, Ernesto, Raúl y a todos aquellos que por falta de memoria pueda estar omitiendo.

A todos ellos:

GRACIAS por todo.

INDICE

ÍNDICE

1.	INTRODUCCIÓN.....	1
2.	DEFINICIÓN DEL PROBLEMA.....	2
3.	OBJETIVO.....	2
4.	JUSTIFICACIÓN.....	2
5.	CAPÍTULO I VISUALIZACIÓN DE SISTEMAS Y PROCESOS SIMULADOS.....	5
5.1	Simulación científica.....	5
5.1.1	Definición de simulación.....	6
5.1.1.1	Modelo.....	6
5.1.1.2	Historia de la simulación.....	8
5.1.1.3	Monte Carlo.....	8
5.1.1.4	Ventajas de la simulación.....	10
5.1.1.5	Etapas de un proceso de simulación.....	11
5.1.1.6	Aplicaciones.....	12
5.2	Visualización científica.....	12
5.2.1	Definición de visualización.....	13
5.2.2	Por qué visualizar.....	13
5.2.3	Historia de la visualización.....	13
5.2.4	Ventajas de la visualización.....	15
5.2.5	Aplicaciones.....	15
5.2.6	Visualización molecular.....	16
5.3	Dinámica molecular.....	17
5.3.1	El método de dinámica molecular.....	18
5.3.2	Integración de las ecuaciones del movimiento.....	19
5.3.3	Condiciones de periodicidad.....	19
5.4	Visualización de dinámica molecular.....	19
5.4.1	Visualización molecular para dinámica molecular.....	19
5.4.2	Edición de estructuras.....	19
5.4.3	Técnicas gráficas.....	20
6.	CAPÍTULO II HERRAMIENTAS DE DESARROLLO PARA EL SISTEMA DE VISUALIZACIÓN DE DINÁMICA MOLECULAR.....	23
6.1	Lenguaje C++.....	23
6.2	Lenguaje Fortran.....	23
6.3	Sistema operativo UNIX.....	24
6.3.1	El "kernel" o núcleo.....	24
6.3.2	El "shell" (Interprete de comandos).....	24
6.3.3	El sistema de archivos.....	25
6.3.3.1	Distintos tipos de archivos.....	25
6.3.3.2	Estructura del sistema de archivos.....	25
6.3.4	Sistema operativo Linux.....	26
6.3.4.1	Historia.....	26
6.3.4.2	Ventajas.....	27
6.3.4.3	Desventajas.....	27
6.3.4.4	Diferencias entre Windows y Linux.....	27
6.4	Supercómputo: Cómputo de alto desempeño.....	28
6.5	Bibliotecas gráficas: La integración de ventanas y gráficos en 3D.....	28
6.5.1	Una mirada a la historia.....	28
6.5.2	IRIS GL.....	29
6.5.3	OpenGL.....	30
6.5.3.1	Características.....	30
6.5.3.2	Lo que no es OpenGL.....	30
6.5.4	Biblioteca gráfica 3D Mesa.....	31
6.5.4.1	Sistemas soportados.....	31
6.5.5	Otras Herramientas.....	31
6.5.5.1	Gtkgl.....	31
6.5.5.2	GLE.....	31
6.5.5.3	GLUT (OpenGL Utility Toolkit).....	31
6.5.5.4	Magician.....	32
6.5.5.5	Pmesa(Parallel Mesa).....	32

7.	CAPÍTULO III INTERFACES DE INTERACCIÓN ENTRE EL SISTEMA Y EL USUARIO.....	35
7.1	Antecedentes de interfaces gráficas.....	35
7.2	Interacción con el usuario.....	35
7.3	Conceptos básicos de interfaces gráficas.....	36
7.3.1	Definición de interfaz grafica.....	36
7.4	Herramientas para la creación de interfaces gráficas.....	38
7.4.1	VDK y VDKBuilder.....	38
7.4.2	PerlTk.....	38
7.4.3	Xforms.....	39
7.4.4	Gtoolkit, GTK y GTK+.....	40
7.4.5	Lesstif.....	40
7.4.6	Tcl/tk.....	40
7.4.7	Otras.....	41
7.4.7.1	motif.....	41
7.4.7.2	xaw3d.....	42
7.4.7.3	blt.....	42
7.4.7.4	fwf.....	42
7.4.7.5	gtk--.....	42
7.4.7.6	gtk.....	42
7.4.7.7	gtkstep.....	42
7.4.7.8	iv.....	42
7.4.7.9	p5-Gtk.....	42
7.4.7.10	pt-Tcl-Tk.....	42
8.	CAPÍTULO IV ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS.....	45
8.1	Conceptos básicos del análisis orientado a objetos.....	45
8.1.1	Definición de objetos.....	45
8.1.2	Definición de clases.....	45
8.1.3	Mensajes: Activación de objetos.....	46
8.1.4	Programa orientado a objetos.....	46
8.1.5	Herencia.....	46
8.1.6	Polimorfismo.....	47
8.1.7	Reutilización.....	47
8.2	Metodología.....	47
8.2.1	Comparación entre las diferentes metodologías orientadas a objetos.....	48
8.2.2	Técnica de Modelado de Objetos (Object Modeling Technique OMT).....	50
9.	CAPÍTULO V ETAPAS DE DESARROLLO DEL SISTEMA DE VISUALIZACIÓN DE DINÁMICA MOLECULAR.....	53
9.1	Análisis orientado a objetos.....	53
9.1.1	Modelo de objetos.....	53
9.1.1.1	Planteamiento.....	53
9.1.1.2	Identificar objetos y clases.....	55
9.1.1.2.1	Seleccionar nombres en los requisitos.....	55
9.1.1.2.2	Añadir clases adicionales de nuestro conocimiento del tema.....	56
9.1.1.2.3	Eliminar redundancias.....	56
9.1.1.2.4	Eliminar clases irrelevantes.....	57
9.1.1.2.5	Eliminar clases vagas.....	58
9.1.1.2.6	Separar atributos.....	59
9.1.1.2.7	Separar clases y métodos.....	60
9.1.1.3	Identificar y depurar relaciones.....	62
9.1.1.3.1	Seleccionar verbos en relacionales en los requisitos.....	62
9.1.1.3.2	Añadir relaciones adicionales procedentes de nuestro conocimiento.....	63
9.1.1.3.3	Eliminar relaciones de diseño o entre clases eliminadas.....	63
9.1.1.3.4	Eliminar eventos transitorios.....	64
9.1.1.3.5	Eliminar relaciones redundantes o derivadas.....	64
9.1.1.3.6	Añadir relaciones olvidadas.....	64
9.1.1.3.7	Definir la multiplicidad de cada relación.....	65
9.1.1.4	Clases abstractas y terminales.....	65
9.1.1.5	Diccionario de clases.....	66
9.1.1.6	Identificar atributos de objetos y relaciones.....	69
9.1.1.6.1	Clases con atributos.....	69
9.1.1.7	Añadir herencia, comprobar casos de uso y modularizar.....	74
9.1.1.7.1	Jerarquía de clases gráficas.....	74
9.1.1.7.2	Jerarquía de clases cálculos.....	76
9.1.1.7.3	Jerarquía de clases contenedores.....	76
9.1.1.7.4	Jerarquía de clases monitor de procesos.....	77
9.1.1.7.5	Jerarquía de clases impresiones.....	77
9.1.1.7.6	Jerarquía de clases editor molecular.....	77
9.1.1.8	Añadir y simplificar métodos.....	78
9.1.1.8.1	Clases con atributos y métodos.....	78
9.1.1.8.2	Diagramas de relación entre objetos.....	87

9.1.2	Modelo dinámico.....	91
9.1.2.1	Escenarios de secuencias típicas de iteraciones.....	91
9.1.2.1.1	Manejo de archivos.....	91
9.1.2.1.1.1	CASO DE USO A1: Obtención de datos de configuración.....	91
9.1.2.1.1.2	CASO DE USO A2: Renombrar archivo de configuración.....	94
9.1.2.1.1.3	CASO DE USO A3: Guardar archivo de configuración.....	95
9.1.2.1.1.4	CASO DE USO A4: Guardar con otro nombre archivo de configuración.....	96
9.1.2.1.2	Visualización y editor molecular.....	97
9.1.2.1.2.1	CASO DE USO B1: Iniciar escena gráfica.....	97
9.1.2.1.2.2	CASO DE USO B2: Selección de objetos gráficos.....	99
9.1.2.1.2.3	CASO DE USO B3: Conexión de átomos.....	102
9.1.2.1.2.4	CASO DE USO B4: Transformación espacial.....	103
9.1.2.1.2.5	CASO DE USO B5 : Animaciones.....	104
9.1.2.1.2.6	CASO DE USO B6: Modificar propiedades.....	106
9.1.2.1.2.7	CASO DE USO B7: Editar molécula.....	108
9.1.2.1.2.8	CASO DE USO B8: Crear estructura cristalina.....	111
9.1.2.1.3	Monitor de procesos de cálculo.....	111
9.1.2.1.3.1	CASO DE USO C1: Monitoreo y activación de procesos.....	111
9.1.2.1.4	Impresiones.....	114
9.1.2.1.4.1	CASO DE USO D1: Impresión de documentos.....	114
9.1.2.2	Identificar eventos entre objetos (identificar sucesos = señales, entradas, decisiones, interrupciones, transiciones, acciones externas, condiciones de error).....	116
9.1.2.2.1	ESCENARIO A1.1: Obtención de datos de partículas.....	116
9.1.2.2.2	ESCENARIO A2.1: Renombrar archivo de partículas.....	116
9.1.2.2.3	ESCENARIO A3.1: Guardar archivo de partículas.....	116
9.1.2.2.4	ESCENARIO A4.1: Guardar con otro nombre archivo de partículas.....	117
9.1.2.2.5	ESCENARIO B1.1: Iniciar escena gráfica ejemplo 1.....	117
9.1.2.2.6	ESCENARIO B1.2: Iniciar escena gráfica ejemplo 2.....	118
9.1.2.2.7	ESCENARIO B2.1: Seleccionar átomos por especie.....	118
9.1.2.2.8	ESCENARIO B2.3: Seleccionar elementos de todos los tipos por grupo con esfera.....	118
9.1.2.2.9	ESCENARIO B3.1: Conexión de átomos entre especies desde un archivo.....	119
9.1.2.2.10	ESCENARIO B4.2: Rotación con ratón sobre el eje Y.....	119
9.1.2.2.11	ESCENARIO B5.1: Animar molécula.....	119
9.1.2.2.12	ESCENARIO B5.5: Animar molécula y sistemas de cavidades en sentido contrario.....	120
9.1.2.2.13	ESCENARIO B6.2: Modificar color a átomos con selección previa.....	120
9.1.2.2.14	ESCENARIO B7.1: Agregar átomos.....	120
9.1.2.2.15	ESCENARIO C1.3: Activación del proceso de cálculo generar imagen.....	121
9.1.2.2.16	ESCENARIO D1.1: Impresión de imagen de dinámica.....	121
9.1.2.3	Diagrama de eventos para cada escenario.....	122
9.1.2.3.1	Manejo de archivos.....	122
9.1.2.3.2	Visualizar y editar molécula.....	128
9.1.2.3.3	Monitor de procesos de cálculo.....	144
9.1.2.3.4	Impresiones.....	147
9.1.2.4	Diagramas de estados.....	148
9.1.2.4.1	Diagrama de estados de contenedor de datos.....	148
9.1.2.4.2	Diagrama de estados de la clase átomo.....	149
9.1.2.4.3	Diagrama de estados de la clase hoyo.....	151
9.1.2.4.4	Diagrama de estados de la clase dipolo.....	153
9.1.2.4.5	Diagrama de estados de la clase conexión.....	155
9.1.2.4.6	Diagrama de estados de la clase molécula.....	157
9.1.2.4.7	Diagrama de estados de la clase sistema de cavidades.....	159
9.1.2.4.8	Diagrama de estados de la clase superficie de potencial.....	161
9.1.2.4.9	Diagrama de estados de la clase escena gráfica.....	163
9.1.2.4.10	Diagrama de estados de la clase celda.....	165
9.1.2.4.11	Diagrama de estados de la clase proceso.....	167
9.1.2.4.12	Diagrama de estados de la clase monitor de procesos.....	168
9.1.2.4.13	Diagrama de estados de la clase editor molecular.....	169
9.1.2.4.14	Diagrama de estados de la clase modificador de propiedades.....	170
9.1.2.4.15	Diagrama de estados de la clase impresiones.....	171
9.1.3	Modelo Funcional.....	172
9.1.3.1	Identificar valores de entrada / salida.....	172
9.1.3.2	Depurar entradas y salidas.....	173
9.1.3.3	Construir diagramas de flujo de datos.....	174
9.1.3.3.1	Diagrama de contexto.....	175
9.1.3.3.2	DFD 0.....	176
9.1.3.3.3	DFD 1 : Proporcionar datos de configuración.....	177
9.1.3.3.3.1	DFD 1.2 : Verificar archivo.....	178
9.1.3.3.3.2	DFD 1.6 : Generar vector de datos configuración actual.....	179
9.1.3.3.4	DFD 2 : Preparar escena gráfica.....	180
9.1.3.3.4.1	DFD 2.5 : Visualizar escena gráfica.....	181
9.1.3.3.5	DFD 3 : Monitorear procesos.....	182
9.1.3.3.5.1	DFD 3.2 : Generar proceso asociado a cálculo.....	183
9.1.3.3.6	DFD 4 : Generar procesos de cálculo.....	184
9.1.3.3.7	DFD 5 : Indicar selección.....	185
9.1.3.3.8	DFD 6 : Editor molecular.....	186
9.1.3.3.8.1	DFD 6.3 : Ejecutar acción.....	187
9.1.3.3.9	DFD 7 : Modificar propiedades.....	188
9.1.3.3.9.1	DFD 7.1 : Cambiar propiedades.....	189
9.1.3.3.10	DFD 8 : Imprimir.....	190

9.1.3.4	Describir los procesos y funciones.....	191
9.1.3.4.1	DFD 1 Proporcionar datos de configuración.....	191
9.1.3.4.2	DFD 2 Preparar escena gráfica.....	192
9.1.3.4.3	DFD 3 Monitorear procesos.....	193
9.1.3.4.4	DFD 4 Generar procesos de cálculos.....	194
9.1.3.4.5	DFD 5 Indicar selección.....	195
9.1.3.4.6	DFD 6 Editar molécula.....	195
9.1.3.4.7	DFD 7 Modificar propiedades.....	196
9.1.3.4.8	DFD 8 Imprimir.....	196
9.1.3.5	Identificar las restricciones.....	197
9.1.3.5.1	DFD 1 Proporcionar datos de configuración.....	197
9.1.3.5.2	DFD 2 Preparar escena gráfica.....	197
9.1.3.5.3	DFD 3 Monitorear procesos.....	198
9.1.3.5.4	DFD 4 Generar procesos de cálculos.....	198
9.1.3.5.5	DFD 5 Indicar selección.....	198
9.1.3.5.6	DFD 6 Editar molécula.....	199
9.1.3.5.7	DFD 7 Modificar propiedades.....	199
9.1.3.5.8	DFD 8 Imprimir.....	199
9.1.3.6	Actualizar el diccionario de datos (diagramas de flujo).....	200
9.1.3.7	Comparar los métodos.....	208
9.2	Diseño orientado a objetos.....	211
9.2.1	Diseño de sistema.....	211
9.2.1.1	División del sistema en subsistemas.....	211
9.2.1.1.1	Relación entre subsistemas.....	211
9.2.1.1.2	Descomposición del sistema.....	214
9.2.1.1.3	Topología del sistema.....	214
9.2.1.2	Identificar concurrencia.....	215
9.2.1.2.1	Concurrencia de objetos.....	215
9.2.1.2.2	Objetos secuenciales.....	215
9.2.1.2.3	Independencia de objetos.....	215
9.2.1.2.4	Hilos de control.....	217
9.2.1.3	Asignar subsistemas a procesadores y tareas.....	217
9.2.1.3.1	Estimar los requisitos de recursos.....	217
9.2.1.3.2	Balancear entre hardware y software.....	217
9.2.1.3.3	Asignar las tareas a los procesadores.....	218
9.2.1.3.4	Determinar la conectividad física.....	219
9.2.1.4	Manejo de almacenamiento de datos.....	219
9.2.1.5	Escoger la implementación de control en software.....	219
9.2.1.6	Manejo de condiciones de borde.....	220
9.2.1.7	Decidir entre distintas prioridades.....	220
9.2.1.8	Arquitecturas.....	220
9.2.2	Diseño de objetos.....	222
9.2.2.1	Combinar los tres modelos.....	222
9.2.2.2	Diseñar los algoritmos.....	223
9.2.2.2.1	Escoger los algoritmos.....	223
9.2.2.2.1.1	Algoritmos de visualización.....	223
9.2.2.2.1.1.1	Dibujar átomos.....	223
9.2.2.2.1.1.2	Dibujar conexiones.....	225
9.2.2.2.1.1.3	Dibujar hoyos.....	225
9.2.2.2.1.1.4	Dibujar curvas de potencial.....	225
9.2.2.2.1.2	Algoritmos de realización de cálculos.....	226
9.2.2.2.1.2.1	Buscar vecinos.....	226
9.2.2.2.1.2.2	Buscar hoyos.....	226
9.2.2.2.1.2.3	Calcular volumen y densidad.....	227
9.2.2.2.1.3	Algoritmos de manejo de datos.....	228
9.2.2.2.1.3.1	Obtención de datos.....	228
9.2.2.2.1.3.2	Lectura de archivos.....	228
9.2.2.2.1.3.3	Almacenamiento de datos en memoria.....	228
9.2.2.2.1.3.4	Renombrar archivo de configuración.....	228
9.2.2.2.1.3.5	Guardar archivo de configuración.....	229
9.2.2.2.1.3.6	Guardar con otro nombre archivo de configuración.....	229
9.2.2.2.1.4	Algoritmos de impresiones.....	229
9.2.2.2.2	Escoger estructuras de datos.....	229
9.2.2.2.3	Definir elases internas y operaciones.....	230
9.2.2.2.4	Asignar responsabilidades para operaciones.....	230
9.2.2.2.4.1	Módulo de visualización (Jerarquía de clases gráficas).....	230
9.2.2.2.4.2	Módulo de clases de cálculos.....	238
9.2.2.2.4.3	Módulo de clases de monitor de procesos.....	241
9.2.2.2.4.4	Módulo de clases de impresiones.....	242
9.2.2.2.4.5	Módulo de clases de editor molecular.....	244
9.2.2.2.4.6	Módulo de clases de contenedores.....	245
9.2.2.3	Optimizar el diseño.....	253
9.2.2.3.1	Añadir asociaciones redundantes.....	253
9.2.2.3.2	Reorganizar el orden de ejecución.....	253
9.2.2.3.3	Guardar atributos derivados.....	254
9.2.2.4	Implementar el control.....	254
9.2.2.5	Ajustar la herencia.....	254
9.2.2.5.1	Reorganizar las clases.....	254
9.2.2.5.2	Abstractar comportamiento común.....	254
9.2.2.5.3	Delegación.....	255

9.2.2.6	Diseñar asociaciones.....	255
9.2.2.6.1	Revisar las asociaciones entre clases.....	256
9.2.2.6.1.1	Asociaciones de las clases gráficas.....	256
9.2.2.6.1.2	Asociaciones de las clases de cálculos.....	259
9.2.2.6.1.3	Asociaciones de las clases de monitor de procesos.....	261
9.2.2.6.1.4	Asociaciones de las clases de impresiones.....	262
9.2.2.6.1.5	Asociaciones de las clases de editor molecular.....	262
9.2.2.6.1.6	Asociaciones de las clases de contenedores.....	263
9.2.2.7	Representar objetos.....	265
9.2.2.7.1	Módulo de visualización.....	265
9.2.2.7.2	Módulo de cálculos.....	271
9.2.2.7.3	Módulo de editor molecular.....	273
9.2.2.7.4	Módulo de monitor de procesos.....	274
9.2.2.7.5	Módulo de contenedores.....	275
9.2.2.7.6	Módulo de impresiones.....	280
9.2.2.8	Empacar en módulos.....	281
9.2.2.8.1	Ocultar información.....	281
9.2.2.8.2	Coherencia de entidades.....	281
9.2.2.8.3	Construir módulos.....	281
9.2.2.9	Consideraciones finales.....	281
10.	CAPÍTULO VI FUNCIONAMIENTO DEL SISTEMA DE VISUALIZACIÓN DE DINÁMICA MOLECULAR.....	283
10.1	Especificaciones y requerimientos.....	285
10.2	Alcances y limitaciones.....	285
10.3	Descripción general y funcionamiento.....	286
10.3.1	Elementos de la interfaz de usuario.....	287
10.3.2	Ejecución del sistema.....	290
10.3.3	Descripción de menús, submenús y opciones del sistema.....	293
10.3.3.1	Menú Archivo.....	293
10.3.3.2	Menú Ver.....	294
10.3.3.3	Menú Selección.....	295
10.3.3.4	Menú Edición.....	296
10.3.3.5	Menú Propiedades.....	297
10.3.3.6	Menú Transformación.....	298
10.3.3.7	Menú Cálculos.....	299
10.3.3.8	Menú Ayuda.....	300
10.3.3.9	Submenú de activación de menú principal y pestañas de módulos.....	301
10.3.3.10	Submenú contextual sobre la escena gráfica.....	303
10.3.4	Descripción de los módulos principales.....	304
10.3.4.1	Escena gráfica.....	304
10.3.4.2	Tabla periódica.....	305
10.3.4.3	Opciones.....	309
10.3.4.4	Propiedades.....	311
10.3.4.4.1	Propiedad Radio.....	311
10.3.4.4.2	Propiedad Transparencia.....	312
10.3.4.4.3	Propiedad Calidad.....	313
10.3.4.4.4	Propiedad Estilo.....	314
10.3.4.4.5	Propiedad Color.....	315
10.3.4.5	Ayuda.....	317
10.3.5	Funcionamiento del sistema.....	318
10.3.5.1	Ver objetos de la escena gráfica.....	318
10.3.5.2	Visualización de propiedades en objetos gráficos.....	325
10.3.5.2.1	Radio de átomos.....	325
10.3.5.2.2	Radio de conexiones.....	328
10.3.5.2.3	Calidad de átomos.....	330
10.3.5.2.4	Calidad de conexiones.....	332
10.3.5.2.5	Transparencia de hoyos.....	332
10.3.5.2.6	Transparencia de átomos.....	334
10.3.5.2.7	Transparencia de conexiones.....	335
10.3.6	Visualización de configuraciones complejas.....	338
10.4	Perspectivas de ampliación funcional.....	346
11.	CONCLUSIONES.....	353
12.	BIBLIOGRAFÍA.....	357

INTRODUCCION

1. INTRODUCCIÓN

Anteriormente, en los diferentes campos de la investigación, el estudio de los diversos fenómenos se efectuaba utilizando métodos abstractos, debido a que, para analizar algún suceso, prácticamente solo se realizaban observaciones no participativas, esto es, se llevaba a cabo una observación pasiva, respecto a dichos fenómenos y esto representaba una gran pérdida de tiempo, ya que en el estudio de algunos procesos en determinadas situaciones, era necesario esperar hasta que se dieran las condiciones deseadas.

Con el paso de los años, se pudo apreciar que gran parte de los fenómenos responden a ecuaciones matemáticas específicas y es así como fue posible realizar predicciones en ciertos fenómenos, lo cual facilitaba en gran medida el estudio de los mismos.

Con el surgimiento de las computadoras, el hombre fue capaz de realizar los cálculos que caracterizaban a situaciones específicas y fenómenos complejos, sin embargo, los datos obtenidos continuaban siendo únicamente información numérica, la cual era muy difícil de interpretar a simple vista. Al generarse un mayor avance tecnológico fue posible la creación de computadoras cada vez más poderosas, que permitían una mejor interpretación de los resultados a través de ambientes gráficos, facilitando, una mejor percepción de dichos resultados, haciendo uso de la simulación.

La simulación está basada en el hecho de que una gran cantidad de fenómenos físicos y químicos se pueden “modelar”. Esto quiere decir que a partir de una teoría es posible construir un modelo matemático que describe sus consecuencias, es decir, predice y explica los fenómenos en los cuales la teoría en cuestión está basada.

Por otro lado y de no menor importancia, el concepto formal de visualización nos dice que es la ciencia computacional que puede brindar enormes niveles de soporte para producción científica, así como potenciar el mayor progreso científico con un nivel de influencia sólo comparable con las supercomputadoras.

Además de lo anterior, la falta de herramientas accesibles y sin restrictivas licencias comerciales, necesarias para la visualización de sistemas moleculares en el Laboratorio de Simulación de Materiales de la UNAM, originó que fuera necesario crear un sistema propio que resolviera las necesidades que se presentaban, a medida que estas fueron aumentando el sistema creado anteriormente se fue quedando anticuado ante los requerimientos tan exigentes de la dinámica molecular.

Como respuesta a este hecho se decidió utilizar mejores herramientas y técnicas para generar un nuevo sistema de visualización, para esto fue necesario investigar las nuevas tecnologías que han aparecido y evaluar su capacidad de desarrollo que hizo posible construir un sistema mejor y más útil.

En el presente trabajo se describe como se realizó un nuevo sistema de visualización capaz de interpretar los resultados numéricos que se obtienen de simulaciones de cálculos de dinámica molecular.

Además se explica cómo para esto fue necesario evaluar y utilizar nuevas metodologías y herramientas de desarrollo que hicieron posible que el sistema fuera portable y pudiera ser utilizado también en computadoras personales.

El sistema es en realidad un esfuerzo generado por varias personas a través de un largo periodo de tiempo, en cuanto a diseño, programación y mantenimiento que sin embargo, y debido a sus limitaciones y a la falta de posibilidad de agregar nuevas y más complejas características que los investigadores requerían para realizar mejores y más completos estudios, fue necesario llevar a cabo una completa reestructuración. Fue por eso que nosotros escogimos la metodología orientada a objetos, para diseñar y reconstruir el sistema original. La metodología orientada a objetos, se escogió ya que hace posible que un sistema se pueda mejorar posteriormente, llevando mas allá los alcances y posibilidades de éste.

El primer capítulo trata acerca de los conceptos generales sobre la visualización de procesos moleculares simulados con los métodos de la física estadística, la historia de la simulación científica, la visualización científica y la dinámica molecular.

En el segundo capítulo se mencionan y describen brevemente algunas de las herramientas de desarrollo usadas para la elaboración de programas de simulación de procesos moleculares.

En el tercer capítulo trata acerca de la importancia de las interfaces gráficas de usuario para facilitar el uso de las aplicaciones y programas, así como los conceptos básicos concernientes a la creación y uso de las interfaces gráficas.

El cuarto capítulo trata acerca del análisis y del diseño orientado a objetos que se utilizó para realizar el programa de visualización.

En el quinto capítulo se indican claramente cada una de las etapas que se siguieron durante la realización del sistema de visualización.

El último capítulo describe los requerimientos, especificaciones, perspectivas y funcionamiento del sistema.

2. DEFINICIÓN DEL PROBLEMA

Se requiere rediseñar y ampliar las funciones del sistema de visualización de dinámica molecular, debido a sus limitaciones, construyendo otro sistema, a partir de un nuevo análisis y un nuevo diseño, haciendo uso de una metodología orientada a objetos, e implementando características más complejas, para aumentar las posibilidades y desempeño y crecimiento del mismo.

3. OBJETIVO

El objetivo final de este trabajo es el de desarrollar un nuevo sistema que facilite el análisis de resultados de simulaciones de dinámica molecular, agregando funciones mejores y más complejas de visualización y estudio de los materiales simulados.

4. JUSTIFICACIÓN

Una de las corrientes actuales en la teoría de comunicación se basa principalmente en el manejo de las gráficas, se estima que 50% de las neuronas está dedicado a la visión, ya que la densidad de información por unidad de área es notablemente mayor a la de un texto. Esta no es la única ventaja que ofrecen las gráficas, además está comprobado que la retención de información en el cerebro humano es mayor cuando el conocimiento adquirido se encuentra respaldado por una imagen. Por otro lado la visualización nos permite ver lo que no es posible ver, es posible reconocer patrones de comportamiento de los datos, ver en una sola imagen o en una secuencia de imágenes (animación) una gran cantidad de datos y nos facilita la comprensión de algunos conceptos, sobre todo de tipo abstracto. Por ejemplo, supongamos una serie de datos, obtenidos de un proceso industrial, al mostrar la tabla no se distingue alguna relación entre los datos, al hacer una gráfica de los valores vemos que siguen cierto patrón de comportamiento.

El desarrollo del presente trabajo se dio a partir de la necesidad de aumentar las funciones del sistema original de visualización, así como el grado de visualización de un sistema de simulación de dinámica molecular requerido por investigadores del Laboratorio de Simulación de Materiales de La UNAM, con el propósito de permitir la observación de modelos tridimensionales de conjuntos de átomos en estructuras moleculares que les permitan a los investigadores realizar sus estudios en una manera más eficiente de la que hasta el momento han utilizado.

El nuevo sistema a desarrollar, planteado a través del análisis de las diferentes técnicas y herramientas estudiadas en este trabajo permitirá tener una biblioteca de objetos que pueden ser reutilizables y un programa que tenga contemplada la posibilidad de ampliar aun más sus capacidades de visualización y sus funciones generales.

Este trabajo se enfoca también al estudio de esas herramientas que nos permitirán un mejor desarrollo del sistema de simulación que se requiere y que contribuyen a hacerlo más portable, proporcionándole así un mayor crecimiento a futuro.

CAPITULO I

VISUALIZACIÓN DE SISTEMAS Y PROCESOS SIMULADOS

1. CAPÍTULO I VISUALIZACIÓN DE SISTEMAS Y PROCESOS SIMULADOS

En los últimos años el uso de tecnologías cada vez más avanzadas ha propiciado grandes adelantos en diversos campos de la ciencia, las artes y las comunicaciones; la investigación es uno de los motores de esta revolución y la computación, la herramienta necesaria en todos los ámbitos donde se requiere de una gran capacidad de alto rendimiento en cálculos, procesos e interpretación de resultados.

El uso de las computadoras en la solución de problemas reales como el tratamiento, almacenamiento y distribución de la información, con la ayuda de las bases de datos, o el control digital de sistemas mecánicos y electrónicos desde una computadora, o la comunicación a través de las redes permitiendo obtener y compartir información a lo largo de Internet, son las muestras más comunes del uso actual de las computadoras; en el ámbito científico el uso de la computación se vuelve más exigente, más especializado y más caro, se requiere de máquinas y programas diseñados para usos especiales, que le permiten al investigador o científico generar los datos requeridos, y les ayudan a interpretar de una manera más fácil los resultados de sus trabajos.

En diversas áreas de la ciencia, el investigador debe entender y estudiar el comportamiento de los fenómenos físicos, químicos, sociales y naturales que interactúan en el mundo real, una de las formas en que esto es llevado a cabo es reproduciendo a pequeña escala dichos fenómenos en un laboratorio controlado, modificar las variables que intervienen en el fenómeno de estudio y observar los efectos y reacciones que se generan. Es en esta parte de la investigación donde la computadora juega un papel muy importante, pues el uso de ésta para generar datos que representan procesos y sistemas reales por medio de la simulación, es la manera en que se pueden resolver problemas cuya complejidad es tal que no pueden ser reproducidos en un laboratorio o están fuera del alcance de una observación directa, así, se pueden realizar observaciones sobre fenómenos reales con modelos y conceptos matemáticos abstractos del pensamiento al representar los datos numéricos o estadísticos visualmente, de manera que el investigador pueda entender mejor lo que ocurre en el sistema que se simula.

El cómputo avanzado o supercómputo es requerido en muchos sistemas que simulan procesos reales, ya que se necesita de un gran potencial de cálculo para resolver esos sistemas, que la mayoría de las veces están representados por fórmulas matemáticas complejas o necesitan de elevados ciclos de procesamiento. Ahora, si lo que se desea es tener más que números y datos para su interpretación final, ya sea gráficas, imágenes o representaciones más complejas tales como modelos tridimensionales o animaciones que dan una idea más real de lo que se observa, el costo computacional crece aún más, pues son necesarias computadoras especiales cuyas características les permiten manejar grandes volúmenes de información en imágenes, sonidos, gráficas y vídeo con una muy alta calidad, muy adecuada para la visualización.

Resumiendo, cuando un investigador hace una simulación numérica de un sistema real, utiliza en gran medida ecuaciones o fórmulas matemáticas que para su solución requieren de cálculos muy extensos, y para la representación de los resultados obtenidos necesita de un equipo y programas adecuados que permitan generar una visualización gráfica para la observación, pues dependiendo de las necesidades del sistema los recursos de cómputo pueden ser también muy elevados. De esta manera puede obtener conclusiones y tomar decisiones en el campo de estudio al que pertenece; la ventaja de esta técnica radica en que puede modificar las variables del sistema o proceso, realizar nuevamente la simulación y ver los resultados haciendo uso de potentes imágenes y gráficas que en todos los casos pueden ser más relevantes que sólo números o letras, y con la flexibilidad de un método interactivo.

5.1 Simulación científica

En el desarrollo del presente trabajo de tesis, el concepto de simulación es la parte que complementa y da significado al sistema de visualización que hemos realizado, pues son los resultados de simulaciones de dinámica molecular, aquellos que los investigadores del Laboratorio de Simulación de Materiales de la UNAM necesitan visualizar para el análisis de sus investigaciones. La simulación científica es una herramienta muy poderosa en el campo de estudio de la física y la química, en lo que concierne a los investigadores de este laboratorio, sus estudios sobre las características estructurales tanto físicas y químicas de los materiales que ellos analizan está basado en análisis de simulaciones numéricas, en los cuales aplican el método de dinámica molecular, explicado más adelante, y cuyos resultados de las simulaciones son desplegados visualmente por un sistema de visualización, permitiendo observar a los expertos características microscópicas de materiales que de otra manera sería imposible observar.

Por la importancia de la simulación como parte principal en el presente trabajo (la visualización final de los datos con el sistema que se pretende generar), su definición es necesaria en este punto para su mejor comprensión.

5.1.1 Definición de simulación

*"H. Maisel y G. Gnugnoli: Simulación es una técnica numérica para realizar experimentos en una computadora digital. Estos experimentos involucran ciertos tipos de modelos matemáticos y lógicos que describen el comportamiento de sistemas de negocios, económicos, sociales, biológicos, físicos o químicos a través de largos periodos de tiempo."*¹

Una simulación también permite observar y comprobar nuevas teorías de comportamiento de los procesos que intervienen en el sistema; al evaluar diferentes condiciones en un programa computacional que simula el sistema real, el investigador puede obtener conclusiones muy alentadoras si el programa genera resultados que corresponden con los observados en la realidad o con las estimaciones esperadas. De una manera más explícita tenemos una segunda definición:

*"Robert E. Shannon: Simulación es el proceso de diseñar y desarrollar un modelo computarizado de un sistema o proceso y conducir experimentos con este modelo con el propósito de entender el comportamiento del sistema o evaluar varias estrategias con las cuales se puede operar el sistema."*²

El investigador que hace uso de la simulación tiene con esta una gran herramienta que le permite representar situaciones alternas del sistema real sin afectarlo, evitando casos donde dichas situaciones se toman complejas e implican costos sumamente altos, en caso de llevarlas a cabo en el mundo real. Sin embargo, la simulación no representa a la realidad tal cual, y los resultados que de ella se obtienen sólo son una aproximación que debe tenerse presente, pues las limitaciones de los métodos y modelos utilizados son inherentes a la simulación. Por lo tanto:

*"Simulación es el proceso de imitar un fenómeno real con un conjunto de fórmulas matemáticas. Los programas avanzados de computación pueden simular condiciones climáticas, reacciones químicas, reacciones atómicas, algunos procesos biológicos, etc. En teoría, cualquier fenómeno que pueda ser reducido a datos y ecuaciones matemáticas puede ser simulado en una computadora. En la práctica, la simulación es extremadamente difícil porque muchos fenómenos naturales están sujetos a un número casi infinito de influencias. Uno de los trucos para desarrollar simulaciones útiles, antes que nada, es determinar cuáles son los factores más importantes."*³

De esta manera, en la simulación se pueden seleccionar sólo las condiciones significativas, y las que permiten delimitar el problema de estudio para observar así los resultados de interés en el trabajo de los investigadores; tal como se menciono anteriormente, imitar un sistema real con todas las variables y situaciones que lo afectan, puede llegar a complicarse por la misma naturaleza del sistema que se simula, y sólo se intenta en algunos casos obtener una aproximación lo más cercana al proceso real en la medida que el modelo o los modelos matemáticos lo puedan representar. Ahora bien ¿qué es un modelo?

5.1.2 Modelo

Un modelo es la representación de un objeto, sistema, o idea, de forma diferente a la de la entidad misma. Usualmente, su propósito es ayudarnos a explicar, a entender o mejorar un sistema.

*"Un modelo es una descripción lógica de cómo un sistema, un proceso, o un componente se comporta."*⁴

Imaginemos que tenemos un avión de juguete, este avión representa de una manera limitada al objeto real, puede tener varios detalles pero el fin no es imitar de manera exacta al avión real, sino de tener sólo las características mínimas representativas (por ejemplo: el tipo de avión, el número de motores, el color, etc.) para el uso del juguete no es necesario

^{1, 2} *Simulación un enfoque práctico*

Raúl Coss Bú.

México 1985. Editorial LIMUSA, S.A de C.V

pp 12.

³ <http://webopedia.internet.com/TERM/s/simulation.html>

⁴ http://www.imaginehatinc.com/frame_simulation.html

simular el comportamiento real del avión que modela. En el caso de una representación más realista donde se requiriera tal vez tener un objeto a escala y simular el comportamiento real (resistencia a flujos de aire, despegue, aterrizaje, sistema eléctrico, interiores, etc.) para observar el comportamiento de un avión experimental, un modelo sólo serviría para ayudar a explicar y entender mejor el objeto real y simular su comportamiento.

La utilidad del modelo esta en el sentido de que ayuda al investigador a organizar conceptos, secuencia de pasos, y las necesidades requeridas en el modelado. Un modelo es una herramienta que permite generalmente predecir características que se requieren observar en el comportamiento de una entidad modelada. En el caso de la experimentación los modelos se usan para representar situaciones controladas en donde un experimento directo resultaría impracticable o muy costoso.

En la simulación numérica de materiales que llevan a cabo los investigadores de este laboratorio los modelos que se utilizan para representar estructuras de sistemas vítreos⁵ o catalizadores⁶ permiten obtener resultados experimentales sobre los cuales se tiene un gran interés en observar propiedades y fenómenos físico-químicos.

Los modelos matemáticos que se utilizan sirven para tener una representación estática de las estructuras moleculares de los materiales en donde se pueden observar detalles y configuraciones mínimas del sistema en estudio, como por ejemplo, el modelo de una zeolita⁷ describe y muestra las bases para poder simular en detalle las características y el comportamiento que el sistema real tiene o puede tener. El modelo pues, es en otras palabras una descripción de una entidad o sistema para su mejor entendimiento, siendo necesario para la simulación científica de sistemas o procesos. Por lo anterior:

“La simulación implica el diseño de un modelo de un sistema, de un proceso, o de un componente y la realización de experimentos a través de ella. El propósito de estos experimentos es determinar cómo el sistema verdadero se realiza y predecir el efecto de cambios al sistema mientras que progresa el tiempo.”⁸

Un modelo adecuado da la oportunidad de llevar a cabo simulaciones en las cuales se pueden observar grupos de variables de interés que afectan el comportamiento del sistema, al cambiar algunos parámetros y dejar fijos otros, se modifican las respuestas y las reacciones estructurales de los compuestos, permitiendo tener ciertas conclusiones sobre

⁵ Una estructura vítrea es aquella que tiene un arreglo atómico desordenado a diferencia de un sistema cristalino que presenta una estructura ordenada o periódica. Algunos sólidos tienen patrones regulares ordenadamente para el arreglo de átomos, moléculas e iones, pero los materiales de vidrio son altamente desordenados. Hay algún orden de corto alcance en el vidrio, pero más allá de uno o dos átomos o iones, el orden puede ser descrito como aleatorio. Esto es, con un submicroscopio básico, los sólidos de vidrio se ven más parecidos a líquidos que a sólidos.

Van Nostrand's Scientific Encyclopedia

Edited by Douglas M. 1976.

Pp 1183.

⁶ Substancias con las cuales se genera una reacción para provocar cambios en el sistema, formalmente sin provocar un cambio químico permanente. Los catalizadores pueden ser homogéneos o heterogéneos, pero todos completamente con el siguiente criterio:

1. A través de los catalizadores se afecta el rango de reacción, estos no pueden afectar la posición de equilibrio en una reacción reversible.
2. En teoría, los catalizadores pueden ser recuperados sin cambios químicos al final de una reacción, a través de esta pueden ser cambiados físicamente.

The Penguin Dictionary of Chemistry

D. W. A. Sharp.

Pp 84.

⁷ Aluminosilicato conteniendo estructuras de $(\text{SiAl})_n\text{O}_{2n}$ con un cambio negativo el cual es balanceado por cationes presentes en las cavidades. Los cationes son fácilmente cambiados, el agua y los gases pueden ser selectivamente adsorbidos dentro de las cavidades. Varios tipos de zeolitas son conocidos incluyendo chabazita, faujasita, zeolita sintética A y natrolita. El sodio esta contenido en zeolitas que son usadas para ablandar agua, reemplazando Ca^{2+} por Na^+ , la zeolita es regenerada con una solución concentrada de NaCl . Las zeolitas son usadas para remover moléculas de tamaños específicos por absorción de los poros de otra molécula, para manejo de solventes y absorción de gases.

The Penguin Dictionary of Chemistry

D. W. A. Sharp.

Pp 432.

⁸ http://www.imagethatinc.com/frame_simulation.html

cómo afectan al sistema los diferentes estímulos y condiciones a las que se somete el experimento, por lo que al final de una simulación se genera un conocimiento mayor del sistema modelado.

“Los resultados de simulaciones numéricas también se pueden comparar con los de experimentos de laboratorio. La simulación constituye, en primer lugar, una prueba del modelo que la sustenta. Si el modelo es bueno, es posible obtener nuevos conocimientos acerca del sistema bajo estudio y proveer elementos de juicio para la interpretación de resultados experimentales, o inclusive, para el diseño de nuevos experimentos.”⁹

En conclusión, si no existe un modelo que represente a un sistema adecuadamente, los resultados de la simulación numérica no representan la verdad de lo que ocurre en el sistema real. Esto es muy común en algunos casos donde los fenómenos representados son tan complejos como para ser reducidos a ecuaciones y fórmulas matemáticas o las variables que intervienen y sus relaciones aun no pueden ser explicadas en un cien por ciento. Por otra parte el caso contrario y uno de los principales fines de la simulación, es dar con ese porcentaje faltante y obtener la solución a los problemas no resueltos y teorías que de otra manera no podrían ser explicados.

5.1.3 Historia de la simulación

La construcción de modelos ha sido usada desde la época del Renacimiento, en la época moderna la palabra simulación aparece en 1940, con el trabajo de los científicos Von Neumann y Stanislaw Ulam en el proyecto Monte Carlo. En el transcurso de la Segunda Guerra Mundial, con la simulación encontraron la solución a las reacciones nucleares para la creación de las bombas atómicas, en donde las soluciones experimentales eran demasiado costosas y su modelado matemático demasiado complejo.

Los años siguientes al fin de la Segunda Guerra Mundial vieron una nueva forma de generar conocimiento científico sobre la cual, científicos como Stanislaw Ulam, Enrico Fermi, John Von Neumann, entre otros, compartieron en el estudio de dominios y campos divergentes una manera común de coordinar temas altamente diversos. El uso de la nueva máquina calculadora (la ENIAC¹⁰) en los experimentos de simulación con pesados cálculos numéricos de ecuaciones integrodiferenciales y sus resultados exitosos, pusieron a los investigadores al borde de una nueva “revolución industrial”. Para ellos esta manera de experimentación prometía una “zona de estudio” en la cual actividades radicalmente diversas podrían estar coordinadas localmente. Actividades como la estadística, el muestreo, la codificación o la teoría del juego fueron pronto los objetivos de las simulaciones, y la máquina de cálculo tomó un papel muy importante para estos científicos.

“Su actividad común se centró alrededor de la computadora. Más precisamente, los teóricos de las armas nucleares transformaron la máquina calculadora naciente y en el proceso crearon realidades alternativas con las cuales descubrieron interrelaciones inquietantes entre la teoría y el experimento.”¹¹

5.1.4 Monte Carlo

Como se dijo anteriormente, los creadores de las armas nucleares crearon la manera de tratar aquellos problemas demasiado complejos para la teoría, careciendo de los materiales indispensables para la experimentación. A este método le llamaron “Monte Carlo” en honor a la Meca de los juego de azar. Con esta herramienta pasaron por la creación de números aleatorios hasta la simulación de procesos analíticos demasiado complejos; muy pronto los físicos e ingenieros convirtieron a Monte Carlo en una realidad alternativa más allá del mero esquema numérico, con la cual podría ser conducida la

⁹ <http://i-it-ma/cuer.unam.mx>
SIMULACIÓN NUMÉRICA DE MATERIALES
Luis Javier Álvarez

Laboratorio de Simulación de Materiales.
¹⁰ (Electronic Numerical Integrator and Calculator).

¹¹ The disunity of science
Boundaries, Context, and Power
Stanford, California 1996
Computer Simulations and the Trading Zone
Peter Galison Pp 119.

experimentación y lo probaron con el problema más complejo de la historia científica en ese momento: el diseño de la primera bomba de hidrógeno.

Durante la guerra los científicos trabajaban en el problema de la fisión nuclear y su principal obstáculo era entender el proceso por el cual la fisión de los neutrones, la dispersión, y el ensamblado de los núcleos de uranio formaban la base de un arma nuclear. Con la experimentación era insuficiente probar los detalles de la masa crítica necesaria para la fisión, ya que la teoría conducía a ecuaciones integrodiferenciales irresolubles. *“Con tales problemas, la realidad artificial del Monte Carlo era la única solución - el método de muestreo podría reconstruir tales procesos modelando una secuencia de dispersiones al azar en una computadora.”*¹²

*“Una vez en la máquina, la simulación podría producir una representación gráfica de la onda expansiva mientras se propagaba.”*¹³

En sus investigaciones, Von Neumann redefinió varias veces su modelo para poder procesarlo en la computadora y que ésta pudiera simular un sistema físico, el cual no se podía representar usando las técnicas analíticas. El problema requería de miles de cálculos que incluso la ENIAC era incapaz de procesar, pero con un nuevo enfoque y gracias a la manipulación de algunas ecuaciones, bastó para obtener los resultados necesarios y se tuvo éxito en la representación “visual” de ecuaciones diferenciales. Sin embargo, la excesiva complejidad del problema rebasó las capacidades de modelado de la computadora y tanto Von Neumann como Ulam se dedicaron entonces a buscar una nueva manera de explotar las capacidades de la nueva máquina.

Para poner en marcha el método de Monte Carlo, Ulam y Von Neumann requirieron de un conjunto inmenso de números aleatorios que se generaron en la computadora, con el fin de poder tener muestras de dispersiones, fisiones y fusiones. De la manera anterior hubieran tenido que usar números aleatorios generados de procesos físicos y que se hallaban en las tablas de los libros, lo cual era inadecuado para sus experimentos, por lo que crearon sus propios números aleatorios.

En las primeras aplicaciones de Monte Carlo como la difusión de gases, los rayos cósmicos o la dispersión de neutrones se tiene una muestra de la simulación en los procesos de la naturaleza, en donde otros métodos o herramientas no pueden tener éxito usando sólo relaciones teóricas.

*“Históricamente, los primeros cálculos a gran escala para hacer uso del método de Monte Carlo eran estudios de dispersión del neutrón y absorción, los procesos aleatorios para los cuales era absolutamente natural emplear números aleatorios. Tales cálculos, un subconjunto de los cálculos de Monte Carlo, se conocen como simulación directa, “ desde la población hipotética... ” correspondiendo directamente a la población verdadera que es estudiada.”*¹⁴

Como todos los adelantos científicos de la edad moderna en que vivimos, el naciente uso de la computadora y del método de Monte Carlo que dieron inicio a la técnica de la simulación, se vieron impulsados por los intereses de la guerra. En un mundo donde se estaba dando una espantosa crisis, la necesidad de ganar la guerra fue el motivo que llevó a estos científicos a crear mejores herramientas para la investigación y que con el tiempo sirvieron para fines más nobles e impulsaron el estudio de la naturaleza misma.

*“... el temprano mundo de las simulaciones fue empapado en consideraciones de las armas; las bombas nucleares saturaron cada aspecto de estas discusiones tempranas, desde el lenguaje a la misma representación de los simuladores.”*¹⁵

¹² Peter Galison *op. cit.* Pp 120.

¹³ Peter Galison *op. cit.* Pp 125.

¹⁴ The disunity of science
Boundaries, Context, and Power
Stanford, California 1996
Computer Simulations and the Trading Zone
Peter Galison Pp 150.

¹⁵ Peter Galison *op. cit.* Pp 156.

5.1.5 Ventajas de la simulación

- 1) Se usa cuando el procedimiento matemático es muy complejo y difícil.

En este caso el uso de la simulación ayuda a los investigadores a resolver difíciles series de ecuaciones y otras fórmulas matemáticas que representan sistemas reales complejos, y que son altamente complicadas para ser resueltas de una manera tradicional y sin el apoyo de alguna máquina calculadora; debido a que algunos sistemas requieren de grados de precisión muy altos, los cálculos se complican al ir aumentando los factores que intervienen en el comportamiento del sistema en estudio.

- 2) Se pueden probar hipótesis en menos tiempo y a un menor costo que si se realiza el experimento en un laboratorio real.

Si se requiere reproducir las condiciones de un determinado sistema, para investigar alguna situación en particular o una serie de éstas, lo más común es instalar un laboratorio en el cual se puedan manipular los factores que intervienen en el sistema. En la práctica hacer lo anterior para probar teorías e hipótesis sobre experimentos de procesos resulta mucho más caro o en ocasiones innecesario que el hacer simulaciones de los sistemas de manera artificial en una computadora, esto reduce en cierta medida el tiempo de respuesta desde el inicio de la simulación hasta la obtención de resultados, haciéndolo mucho menor que el invertido en la experimentación y representación en un ambiente real del fenómeno.

- 3) Se puede tener una observación interactiva del sistema bajo estudio al alterar el modelo y ver los comportamientos ocurridos.

Al tener un sistema simulado, el investigador puede tener una representación de una manera más accesible, de diversas situaciones generadas por los cambios en las relaciones de los agentes que intervienen en el sistema. Con la simulación, el científico puede controlar las variables que más le interesen en el proceso, ver los cambios ocurridos en el sistema para estudiar los efectos externos e internos y volver a repetir el ciclo hasta encontrar los resultados que busca.

- 4) Al tener características detalladas en el sistema simulado se puede tener un mejor entendimiento del sistema real para tomar mejores decisiones basándose en los resultados obtenidos.

Se puede entender mejor cómo opera el sistema al tener identificadas las variables más importantes y las interrelaciones entre éstas y así evaluar situaciones específicas en las que se cambien las condiciones iniciales.

- 5) Se puede experimentar con situaciones nuevas en donde no se tiene la información suficiente para saber qué sucede.

A través de esta experimentación se puede estar mejor preparado ante situaciones o posibles resultados que no se tengan previstos en el desarrollo teórico. *“Mientras que la experimentación con los modelos de procesos existentes es común a la mayoría del campo, la simulación también se utiliza para explorar lo desconocido y no probado. Además, el proceso que modela en si mismo es beneficioso: se reconoce generalmente que cerca del 50 por ciento de las ventajas de un proyecto que modela son ganadas por los esfuerzos realizados antes de comenzar a modelar (recopilando datos, presentando preguntas, el entender procesos, etc.)”*¹⁶

- 6) Sirve para observar el comportamiento de fenómenos que no pueden ser reproducidos en un laboratorio.

En la investigación de algunos fenómenos la simulación es una herramienta indispensable, ejemplos de estos sistemas son aquellos que tienen que ver con la física de partículas, el estudio de procesos químicos, cuyos aspectos microscópicos impiden a los investigadores una observación directa de lo que ocurre en la realidad, de la misma manera sirve para reproducir el comportamiento de eventos cósmicos y naturales sumamente complejos y lejanos de los aparatos, laboratorios e investigadores.

¹⁶ http://www.imagethatinc.com/frame_simulation.html

- 7) Para ver cómo se comportan elementos extraños introducidos al sistema y como lo afectan.

En este caso, algunas situaciones requieren ser evaluadas probando los efectos causados por agentes extraños en el comportamiento del sistema para tener datos precisos de las reacciones favorables o negativas según el tipo de experimento y realizar después pruebas sobre los sistemas reales una vez conocidas las consecuencias. Un ejemplo es el desarrollo de fertilizantes, plaguicidas, vacunas, sustancias químicas o nuevos materiales.

Como puede verse por las ventajas anteriores, la simulación se ha vuelto indispensable para los investigadores y científicos de esta época, que además cuentan con computadoras y programas de visualización cada vez más avanzados, que les permiten imitar y ver de una manera más realista, sistemas complejos de la naturaleza y la ciencia que antes eran más difíciles o imposibles de analizar y estudiar.

5.1.6 Etapas de un proceso de simulación

Un experimento de simulación de un sistema implica las siguientes etapas:

- a) Definir el sistema: Identificar cuales son los alcances y limitaciones, las restricciones, la interacción con otros sistemas así como las variables que intervienen y las relaciones entre estas, para definir las metas.
- b) Determinar el modelo: Establecer las relaciones lógicas y variables que intervienen en el proceso, los diagramas o esquemas, fórmulas o algoritmos que describan en forma completa y correcta al modelo.
- c) Conjunto de datos: Es importante definir claramente los datos requeridos por el modelo.
- d) Implantar el modelo en la computadora : Definir la metodología para el análisis y el diseño, elegir el lenguaje de programación para construir el modelo en la computadora y poder obtener los resultados de la simulación.
- e) Validación del sistema: En este punto se pueden encontrar algunos errores en la determinación o formulación del modelo usado en la simulación, así como los datos que alimentan a éste. Las siguientes son algunas de las maneras más comunes de validar un modelo:

“La opinión de expertos sobre los resultados de la simulación.

La exactitud con que se predicen datos históricos.

La exactitud en la predicción del futuro.

La comprobación de falla del modelo de simulación al utilizar datos que hacen fallar al sistema real.

La aceptación y confianza en el modelo de la persona que hará uso de los resultados que arroje el experimento de simulación.”¹⁷

- f) Experimentación: Una vez que el modelo ha sido validado, revisado e implementado adecuadamente el investigador puede tener las simulaciones del sistema para su observación y estudio.
- g) Interpretación: En esta fase, el investigador tiene ya los datos necesarios surgidos de la simulación, con los que puede tomar decisiones o formular nuevas investigaciones.
- h) Documentar el modelo: Por ultimo es recomendable hacer los manuales tanto técnico y de usuario del modelo, así como del programa de cómputo que se encarga de la simulación y visualización del sistema o proceso simulado.

¹⁷ *Simulación un enfoque practico*

Raúl Coss Bú

México 1985, Editorial LIMUSA, S.A de C.V

pp. 13.

5.1.7 Aplicaciones

La simulación se usa en diferentes áreas de la ciencia, y los problemas que ayuda a resolver son muy variados, los siguientes son sólo una muestra:

- a) En medicina: simulación de órganos internos expuestos a agentes externos. Representación de regiones cerebrales, señales vitales y flujos sanguíneos en órganos dañados, etc.
- b) Biología: simulación de la evolución de especies animales y de plantas.
- c) Entretenimiento: simulación de ambientes virtuales. Animación.
- d) Ingeniería: simulación de estructuras y superficies expuestas a diferentes agentes como temperaturas, deformaciones o campos electromagnéticos. Visión artificial y robótica, inteligencia artificial.
- e) Física: simulación de fluidos dinámicos, mecánica de partículas.
- f) Matemáticas: Modelado de objetos naturales y fractales.
- g) Química: Representación de modelos moleculares, dinámica molecular. Estudio de materiales y compuestos.

Como se definió antes, la simulación permite analizar sistemas complejos para la observación de situaciones reales en donde se desea acortar el tiempo de diseño, reducir costos u obtener nuevo conocimiento.

También puede aplicarse en disciplinas como la administración, la política, la economía o la educación. En las ciencias sociales como las ciencias del comportamiento, la psicología o las relaciones internacionales, en estudios urbanos, la industria o el gobierno.

Debido a la gran importancia de la simulación en la investigación, las técnicas usadas para representar la información generada en las simulaciones, son hoy en día, el complemento que permite interpretar de una manera más natural los datos y resultados obtenidos, por esto el avance en el campo de la visualización gráfica es uno de los temas de la actualidad que más ayuda al investigador en el proceso de la simulación numérica. Simulación y visualización son dos herramientas que van de la mano en la búsqueda actual de soluciones de sistemas y procesos complejos.

5.2 Visualización científica

Con el gran desarrollo de la computación y en especial de la computación gráfica, cada día los avances en el campo de la visualización aumentan la capacidad de interpretación de datos y de ideas, los equipos cada vez más poderosos, las técnicas avanzadas de graficación, los algoritmos, los programas especializados y las interfaces visuales efectivas, nos habilitan para observar, manipular, buscar, explorar, navegar, filtrar, descubrir, entender e interactuar más rápida, fácil y efectivamente con una cantidad enorme de datos e información y poder así descubrir patrones ocultos que generen nuevo conocimiento.

La visualización es el puente de unión entre el procesamiento de información de la computadora y el que se da en la mente humana. La transformación de datos, la información y el conocimiento que se obtienen de una manera visual, le permiten a la gente explorar de una manera natural el mundo que le rodea con el simple hecho del reconocimiento rápido de patrones visuales, es por eso que la representación visual computarizada se ha convertido en un lenguaje muy importante en diversas áreas de la vida humana.

Con el aumento de información cada vez más completa, los investigadores y desarrolladores en el campo de la visualización cambiaron fundamentalmente la forma en que presentamos y entendemos los conjuntos muy grandes de datos complejos; el impacto ha sido fundamental en la forma en que se comunican nuevas ideas para tomar decisiones eficientes. La comunidad científica se vio en la necesidad de utilizar la visualización para arreglárselas con los grandes volúmenes de datos que recolectaban de instrumentos científicos o que eran generados por masivas simulaciones hechas en supercomputadoras. En contraste otro tipo de usuarios no técnicos necesitaron también de interfaces visuales para manipular y desplegar su información generándose una variedad de campos de aplicación para la visualización gráfica.

La visualización especializada o técnica, mejor llamada visualización científica, es el complemento de la simulación en los trabajos de investigación hoy en día. Una definición detallada nos mostrará la importancia de ella en la investigación, y nos dará una suficiente justificación para el desarrollo de nuestra aplicación y trabajo de tesis.

5.2.1 Definición de visualización

Visualizar es crear una representación visible de algo, ya sea un concepto, idea, un grupo de datos o de algún objeto que por pequeño, enorme o distante, no lo podemos abarcar o alcanzar a ver por métodos comunes. Visualizar es representar de manera gráfica un fenómeno, ya sea estáticamente (como por ejemplo, con una gráfica de barras) o dinámicamente, (por ejemplo, el cubo de Rubick, que fue ideado para representar las soluciones a problemas espaciales) haciendo uso de medios artificiales para representar uno o más comportamientos.

Visualización

Usa la computadora para transformar datos en imágenes. La visualización en su forma más básica es aquella que permite convertir un conjunto de datos o de información ordenada en diagramas y gráficas, otra forma de la visualización es usada en el diseño asistido por computadora (CAD) en el cual se pueden obtener imágenes desplegadas de modelos tridimensionales mismos que pueden ser vistos desde cualquier ángulo e incluso ser animados.

Visualización científica

La visualización científica es una nueva aproximación en el área de la simulación numérica. Esta le ayuda a los investigadores a observar el resultado de simulaciones numéricas usando representaciones gráficas complejas.

Usa la computadora para representar objetos del mundo real que no pueden ser vistos normalmente, tales como las formas de las moléculas, el aire y la dinámica de fluidos, conducción del calor, flujos, plasmas, mecanismos de terremotos. Las aplicaciones de la visualización científica requieren de enormes recursos de cómputo, y los centros de súper cómputo, así como los laboratorios de universidades a través del mundo, son siempre los encargados de poner en marcha proyectos en los cuales se investiga haciendo uso de esta actividad. Las representaciones gráficas se hacen mediante líneas, ejes coordenados figuras básicas o modelos tridimensionales de variada complejidad que interpretan los resultados numéricos de algún cálculo científico o modelo que explique la realidad.

Así pues la visualización científica se basa en el uso de tecnologías para la creación de imágenes computacionales que hacen posible entender fenómenos complejos simulados.

5.2.2 Por qué visualizar

El objetivo de visualizar, es lograr obtener un mayor entendimiento de los datos, así como información nueva, antes no evidente u oculta.

Hoy en día muchos científicos utilizan técnicas básicas de visualización para analizar y entender los datos empíricos, de simulación o de un modelo teórico, entre éstas incluyen líneas, contornos, vectores de puntos, superficies de malla 3D, figuras en 2D, gráficas de barras o combinaciones de los anteriores elementos, que les permiten manipular y obtener mejores conclusiones de manera visual.

Al poder representar una enorme cantidad de datos con una sola imagen o con una secuencia de ellas en una animación, podemos reconocer ciertos patrones en el comportamiento de los datos numéricos, y algunos conceptos, en especial los de tipo abstracto son más fáciles de comprender.

5.2.3 Historia de la visualización

En el afán de contar con mejores medios y técnicas para representar la gran cantidad de información, generada por el surgimiento de la computadora, hizo su aparición el complemento más impactante (por ser más natural al ser humano) creado para la nueva tecnología: la computación gráfica.

Al ser la computadora una herramienta de propósito general, cada rama del conocimiento se vio beneficiada por las múltiples aplicaciones y usos que estaban a su alcance. De manera inversa, técnicas y conceptos de diversas áreas fueron poco a poco integrándose al gran campo de las aplicaciones computarizadas. Así, tenemos las bases de la

computación gráfica en los inicios mismos de la fotografía y la animación tradicional, ya que fueron varios los conceptos de estas dos áreas los que se incorporaron a los avances tecnológicos para la creación de imágenes computarizadas, y posteriormente al procesamiento digital de imágenes. Una vez ya teniendo la posibilidad de manejar estas imágenes dentro de la máquina, el siguiente paso dentro de la computación fue tener una secuencia de estas imágenes, cuadro por cuadro, como en las películas del cine o la televisión.

La visualización depende de varias técnicas para la representación de datos, una de éstas es la animación, la cual no es más que una secuencia de imágenes de algún objeto o conjunto de objetos en un determinado periodo de tiempo. Dentro de la animación existen conceptos variados que sirven para generar mejores animaciones, pero sólo mencionaremos que dependiendo de la calidad y precisión que se requieran para entender los datos, en este caso, de visualización científica, se eligen apropiadamente cada una de estas técnicas.

Desde los años 60, cuando los científicos de los laboratorios Bell, Zajak y Knowlton, desarrollaron las primeras animaciones por computadora en el mundo, se establecieron los cimientos para la realización de las animaciones posteriores de alta tecnología. Estos dos científicos trabajaban en configuraciones abstractas y texturizadas, es decir, modelando texturas en la pantalla de la computadora. Con el avance del despliegue de gráficos la animación comenzó a ser usada por importantes centros de investigación en aplicaciones tales como visualización de simulaciones del flujo de fluidos viscosos (Los Alamos), la propagación de las ondas de choque en un sólido (Lawrence Livermore National Laboratory), y la vibración y el aterrizaje de un avión (Boeing Aircraft). (David Fox y Mitchell Waite)¹⁸

A la par al desarrollo de la tecnología, las técnicas de visualización comenzaron a usarse, y los programadores proporcionaron sistemas computacionales especializados y cada vez más completos. Fue la computación gráfica generada en los centros de investigación la que se benefició de técnicas simples como la visualización de datos escalares y vectoriales por medio de la unión de puntos y gráficas de barras, contornos de mapas, vectores de tramas y superficies de malla. Esto permitió a los investigadores escribir sus propios programas que les proporcionaban las funciones más elementales, como por ejemplo dibujar una línea, un círculo, y que en algunos casos permitió desarrollar programas a muy bajo costo. Conforme las técnicas se fueron haciendo más avanzadas, los sistemas se volvieron también más complejos, y aparecieron en el mercado de cómputo paquetes comerciales muy caros y bibliotecas de rutinas gráficas con algoritmos más eficaces, algunas de las cuales manejaban ya métodos como tramas paramétricas, sistemas de partículas y visualización de funciones continuas.

Al evolucionar la creación de gráficas y modelos en dos dimensiones a modelos de tres dimensiones, se tuvo una interpretación todavía más representativa de la realidad. Los modelos tridimensionales en un principio se construyeron sólo como objetos de líneas o alambre, después usando una técnica de despliegue para la creación de superficies sólidas por medio de puntos a lo largo de una red, se les dio volumen a los objetos. Las técnicas continuaron avanzando hasta permitir manipular, modificar y animar modelos tridimensionales en ambientes artificiales. Con la complejidad de los cálculos realizados en aumento, las grandes supercomputadoras se encargaron de generar grandes cantidades de datos dejando la presentación gráfica de los resultados a estaciones de trabajo diseñadas para la visualización.

En la actualidad los avances en la computación gráfica y la tecnología de visualización, permiten tener representaciones visuales interactivas cada vez más complejas, como el uso de potentes sistemas de realidad virtual. La realidad virtual es la manera de representar tridimensionalmente objetos y lugares de manera casi real, con objetos, sonidos, estímulos visuales, en ambientes en los cuales nuestros sentidos se encuentran aislados del mundo exterior, por medio de dispositivos especiales que una computadora y una pantalla normal no podrían hacer. Un ejemplo del uso de realidad virtual en la visualización es la observación de órganos o sistemas biológicos dentro de ellos mismos, análisis de reacciones y estructuras químicas, simuladores de vuelo, etc.

¹⁸ *Gráficos animados por Computadora*

David Fox y Mitchell Waite
McGraw-Hill 1983.

pp 17

5.2.4 Ventajas de la visualización

Las ventajas de la visualización son evidentes, podemos representar datos de varias dimensiones. Asociando una dimensión a una variable, tenemos sistemas de más de las tres dimensiones conocidas, por ejemplo el plano cartesiano cuenta con dos dimensiones, una en el eje x y otra en el eje y, si agregamos un plano más, un eje z, tendremos 3 dimensiones o variables, usando colores para la representación de temperatura tendremos cuatro variables, y si se hace la animación de la gráfica podríamos ver una quinta variable.

Interpretación de datos: Examinar datos de algún estudio, simulación, recopilación de instrumentos, etc. sin la ayuda de una representación visual puede ser algo complicado y tedioso, y aun más cuando se tienen conjuntos enormes de datos numéricos como resultado de extensos cálculos. Al tener una visualización gráfica es más fácil comprender lo que se estudia, pues con una imagen es posible identificar puntos de interés para su análisis e interpretación detallada.

Encontrar tendencias: Con la visualización se puede observar el comportamiento de los datos, si se comportan de manera diferente a los datos de una visualización anterior, y si es así encontrar las posibles causas, con esto se pueden obtener estadísticas alrededor de valores característicos de interés para el investigador.

Localizar datos erróneos: al observar una representación visual de los resultados es posible detectar más fácilmente alguna anomalía, interpretarla e identificar la fuente de error, incluso tal vez se llegue a la conclusión de que el modelo mismo podría estar mal planteado o que el error o errores provengan de los dispositivos externos (sensores, periféricos) o sean sólo propios del usuario.

En ambientes interactivos la ventaja es que: *“...se facilita la rápida experimentación porque ésta muestra una retroalimentación visual inmediata sobre el impacto de un cambio en la entrada, la cual puede ser seguida por un análisis estadístico. La retroalimentación rápida permite algunas veces tener experimentos sin un análisis estadístico completo.”*¹⁹

La computación gráfica tridimensional puede ser extremadamente expresiva. Con el acercamiento correcto al diseño visual de una disposición, la gente puede comprender rápida y fácilmente enormes cantidades de información. Además es más cómodo manipular más información en 3D que con las gráficas de barras 2D, o las filas y las columnas de números.

En la comunicación de los trabajos e ideas, la visualización permite presentar los materiales visuales en conferencias, congresos, publicaciones, así mismo la divulgación de estos trabajos a gente no especializada en temas complejos puede ser mas clara y concisa con imágenes visuales que con meros números o fórmulas.

5.2.5 Aplicaciones

La visualización es más que una sola representación gráfica, sin aplicaciones que justifiquen su uso para evaluar diversos métodos, modelos o técnicas, tal vez no tendría mucho valor excepto por su estética o impacto visual. Sin embargo, hoy en día se ha incrementado y diversificado el uso de un gran conjunto de fabulosas técnicas visuales provenientes desde áreas tan tradicionales como la medicina, la meteorología, la aeronáutica o la química, hasta las más novedosas áreas como la bioinformática, la inteligencia artificial o la visualización del WWW (con espacios tridimensionales como mundos virtuales).

La enorme variedad de áreas de aplicación contribuye a renovar y perfeccionar el conjunto actual de técnicas o algoritmos de visualización, algunos ejemplos son: las isosuperficies y el despliegue de volúmenes que se originaron en la medicina y el concepto de líneas de flujo originado en la aeronáutica.

¹⁹ *Multilevel Visualization of Spinal Reflex Circuit Simulations*

Kalpathi R. Subramanian, David P. Bashor,

IEEE Computers Graphics

May-June 1997

Pp 6

El uso de la visualización se ha expandido por absorción, adaptación y aplicación de ideas y técnicas desde una gran variedad de campos, gracias a esto. incluso se ha podido extender a áreas de aplicación las cuales no eran el objetivo original como es el caso en el análisis exploratorio y confirmativo de información, la simulación, la educación, la física, etc.

En el análisis, las aplicaciones se usan para control de calidad, proyecciones financieras; en simulación se incluye representación visual de fenómenos ópticos, modelos atmosféricos, reacciones químicas o estudios de dinámica molecular. En educación desde demostraciones matemáticas hasta modelos de física cuántica y planetarios.

En áreas fuera de la investigación como en los servicios. El manejo de bases de datos por medio de interacciones visuales, o en la industria, haciendo uso de integración de procesos a sistemas de realidad virtual. También existen técnicas de representación visual altamente especializadas como la detección de centros de vórtices, o la visualización de simulaciones de choque de automóviles.

Aplicaciones como las anteriores son llevadas a cabo por científicos e investigadores en universidades. mientras que otras robustecen proyectos de investigación a gran escala en empresas privadas o en la industria.

Gracias a variadas técnicas provenientes principalmente de la computación gráfica (como el sombreado y el manejo de texturas), del procesamiento de imágenes y visión (como la detección de gradiente y operaciones de suavizado) o de las matemáticas (cuaternios y splines), entre otras muchas áreas (algunas ya mencionadas anteriormente), el campo de influencia de la visualización crece constantemente dando lugar al surgimiento de programas comerciales muy sofisticados. Por ejemplo en las simulaciones numéricas generalmente se usan paquetes específicos de visualización de propósito general como AVS (Advanced Visual System), los cuales son en su mayor parte adaptados para algunas aplicaciones, en otros casos los investigadores prefieren desarrollar sus propios programas ajustados a sus muy particulares necesidades. En la actualidad todos estos sistemas de cómputo visual son orientados al escritorio gráfico del usuario (con interfaces gráficas muy accesibles), es decir, dependen de un ratón, un teclado, un cursor tridimensional (un puntero para obtener coordenadas [x, y, z]) y eventualmente de un modo estéreo que permite generar imágenes para ser observadas con gafas especiales. Todo lo anterior para facilitar una interacción más cómoda y directa al usuario con el sistema.

Con el arribo de la multimedia, también llegó un gran cambio en los sistemas educativos y de documentación. Mientras que la visualización fue el medio ideal para llevar la fascinación de la ciencia y la tecnología a la gente común, y justificar así las grandes cantidades de dinero gastadas en ella, en el campo de la investigación los expertos, científicos, ingenieros y especialistas, encontraron una de las herramientas más esenciales para entender y acceder a sus propios datos.

*"La visualización puede proporcionarnos un acceso confortable a un mundo en el cual las estructuras abstractas, las redes, los datos, y la información juegan un papel esencial en los negocios de todos los días, en la vida privada, la educación y el entretenimiento"*²⁰

5.2.6 Visualización molecular

Los biólogos moleculares, los químicos y los ingenieros químicos, algunas veces necesitan conocer la estructura y dinámicas de moléculas existentes o hipotéticas. Actualmente el uso de una estación de trabajo gráfica puede en principio proporcionar el control de la adquisición experimental de datos, para procesar e interpretar las observaciones, para derivar y refinar modelos estructurales así como para planear y realizar una amplia gama de cálculos teóricos.

El desarrollo de la visualización molecular como disciplina ha sido encabezado principalmente por la cristalografía de proteínas, en donde la relación de estructuras y funciones es un principio fundamental. Así mismo las mejoras en la geometría, en datos cuantitativos, o simulación de vibraciones moleculares son convenientemente realizadas usando un mecanismo de descripción molecular, por ejemplo, una configuración molecular dada puede obtenerse al computar la desviación de las coordenadas moleculares (longitudes de los enlaces, ángulos de torsión, separaciones donde no hay enlaces, etc.) de sus valores de equilibrio. Para una coordenada molecular interna dada, una función de potencial

²⁰ *Visualization for Everyone*
Alex Pang, Hans-Georg Pagendam
IEEE Computers Graphics
July/August 1998
Pp 47.48

describe como la contribución para la energía potencial de la molécula varía con el valor de esa coordenada, que es el grado de acortamiento o alargamiento del enlace. Así las fuerzas atómicas y masas son usadas para hacer cálculos de la evolución de posiciones atómicas con tiempo y temperatura finitos que son trayectorias generadas en una simulación de dinámica molecular.

La visualización molecular se extiende a sistemas complejos como proteínas, DNA, enzimas, polímeros y más recientemente, sistemas inorgánicos como zeolitas. Generalmente los investigadores utilizan estaciones de trabajo para gráficas 3D, o computadoras personales que utilizan algún protocolo de comunicación gráfica como X-windows, en éstas, las imágenes son generadas de alguna estructura, la cual es manipulada en la pantalla (trasladada, rotada, escalada). La manipulación se puede dar a través del uso de periféricos en forma de perillas o por el control del propio ratón. Lo más importante es que las imágenes tridimensionales creadas son aumentadas y mejoradas por el uso de técnicas de sombreado, iluminado, coloreado, perspectiva y eliminado de superficies ocultas.

Cabe mencionar que la visualización molecular puede llegar a ser muy complicada, de acuerdo al grado de precisión requerida en la obtención de imágenes, así como en el uso de técnicas cada vez más avanzadas que permiten aumentar el nivel de estudio de los investigadores. Un ejemplo de lo anterior, es el poder de las estaciones de trabajo que puede llegar a incrementarse por enlaces de alta velocidad, que permiten tener comunicación en tiempo real donde las aplicaciones locales interactúan con supercomputadoras que ejecutan los cálculos pesados, o en arquitecturas paralelas que mejoran el desempeño de los cálculos en una misma máquina con el poder de la visualización gráfica 3D. (C.M. Freeman y S.M. Levine).²¹

En el siguiente capítulo y los subsecuentes, veremos la definición de dinámica molecular así como la definición de algunas herramientas básicas que usaremos en el análisis, desarrollo y programación del sistema de visualización de dinámica molecular, que constituye el principal objetivo de este trabajo de tesis.

5.3 Dinámica molecular

Uno de los problemas más significativos dentro de la química, es entender y predecir la cinética de las reacciones químicas. La cinética formal nos permite determinar la velocidad de una reacción, sin embargo, ésta únicamente nos proporciona poca información sobre los detalles de aquellas reacciones que se dan a través de una única colisión simultánea entre los reactivos que intervienen y que se conocen como reacciones elementales.

La cinética formal o macroscópica trabaja con moléculas de reacción que se encuentran presentes en una gran variedad de estados y por tanto a partir de mediciones de la constante de velocidad se obtiene información indirecta sobre los sucesos individuales que tienen lugar durante una reacción química.

Las técnicas de simulación de sólidos, permiten obtener propiedades estáticas y dinámicas a partir de modelos microscópicos de interacciones atómicas. Entre dichas técnicas tenemos el método de la dinámica molecular y el de Monte Carlo, que tienen aplicación en sistemas clásicos y cuánticos.

Las aplicaciones de estas técnicas, incluyen el estudio de propiedades de gases adsorbidos en superficies, problemas orden-desorden en zeolitas y el estudio de impurezas aisladas en semiconductores.

La dinámica molecular se ocupa de lo que sucede en el nivel molecular durante una reacción química elemental. Los resultados de la dinámica molecular enriquecen los conocimientos que tenemos sobre los aspectos macroscópicos de la cinética. La dinámica molecular nos permite comprender, a nivel microscópico, el mecanismo de las reacciones químicas elementales e interpretar la cinética macroscópica en términos moleculares.

Para complementar e interpretar los estudios experimentales se realizan cálculos dinámicos del paso del sistema reactivo sobre la superficie de energía potencial.

²¹ Modelling of Structure and Reactivity in Zeolites
Academic Press 1992.
Zeolite Computer Graphics
C.M. Freeman, S.M. Levine
Pp 133-135.

La energía potencial de un sistema es el valor de la energía de interacción que se obtiene de la resolución de la ecuación de Schrödinger para una configuración fija de los núcleos, después de haber realizado la aproximación de Born-Oppenheimer del movimiento electrónico y nuclear.

La superficie de energía potencial es la función que nos proporciona el valor de la energía potencial para cada configuración nuclear. Es representada por $V(R)$, donde R son todas las coordenadas internas del sistema. En una molécula atómica esta función depende de una única coordenada que es la distancia entre los dos núcleos. En una molécula de 3 o más átomos esta función constituye una superficie o hipersuperficie.

Los estudios teóricos se realizan de manera paralela a las investigaciones experimentales y son de gran ayuda para interpretar los resultados experimentales. Por otro lado, los resultados experimentales son importantes para sugerir nuevas líneas de investigación teórica. Es ahí donde, los estudios teóricos y experimentales se complementan para darnos una visión más completa de lo que sucede en el nivel molecular.

De los estudios experimentales y teóricos de dinámica molecular, es posible extraer información, entre otras cosas, sobre:

- Influencia de la distribución energética de las reacciones en la eficiencia de las colisiones.
- Distribución angular y energética de los productos que se forman.
- Características moleculares de las colisiones.
- Cálculo de la función de opacidad, de la sección eficaz de la constante de velocidad.
- La validez de la teoría del estado de transición.

5.3.1 El método de dinámica molecular

“El método de Dinámica Molecular (DM) está basado en la solución de las ecuaciones clásicas de movimiento de un conjunto de N partículas colocadas en una región del espacio de volumen, V , fijo. Las partículas representan átomos, iones o moléculas y se les asignan posiciones y velocidades. Las primeras, de acuerdo con la estructura que se simula, y las segundas, al azar, con una distribución estadística de Maxwell-Boltzmann cuyas magnitudes se establecen de acuerdo con la temperatura a la que se pretende realizar la simulación. Con la especificación del estado dinámico del sistema, dada por los N vectores de posición de las partículas, $r=(x,y,z)$, y los $3N$ momentos, $p=mv$, se pueden encontrar las posiciones de las partículas en cualquier instante posterior, resolviendo la segunda ley de Newton ($F=ma$) para cada una de las N partículas. Para encontrar la solución de este problema es necesario resolver numéricamente un conjunto de $3N$ ecuaciones diferenciales acopladas de segundo orden.”²²

Las ecuaciones de Newton son las más frecuentemente utilizadas, sin embargo, algunas veces las ecuaciones de Hamilton se prefieren.

Este tipo de estudios trabaja principalmente, con sistemas moleculares nanoscópicos, es decir, sistemas que miden 10^{-12} de metro y que pueden interactuar de manera individual o colectiva.

Para resolver las ecuaciones clásicas del movimiento, es necesario conocer cómo interactúan las partículas del sistema a través de los llamados potenciales de interacción. Estos se pueden obtener empíricamente, o a partir de cálculos mecánico-cuánticos que constituyen una representación de la naturaleza de la interacción de las especies que componen el sistema.

El Método de Monte Carlo es muy útil para probar el espacio de búsqueda de configuraciones posibles de un sistema molecular, pero para obtener información sobre su dependencia en el tiempo se utiliza el método de dinámica molecular.

²² <http://icla.matcuern.unam.mx/>

5.3.2 Integración de las ecuaciones del movimiento

En la simulación de dinámica molecular se trata de obtener las posiciones y velocidades de las partículas y otras propiedades dinámicas del sistema, como su comportamiento en función del tiempo. Es por esto que el esquema más apropiado para resolver las ecuaciones de movimiento es el de diferencias finitas. Así, se obtienen primero estas cantidades en el tiempo t y después, en el tiempo $t + \Delta t$. El valor de Δt se escoge en función de características del sistema tales como la temperatura y las frecuencias típicas de vibración de las partículas.

Al comenzar la simulación se cuenta solamente con una configuración. Dada la fuerza total sobre cada partícula se encuentra una nueva configuración usando aproximaciones para posiciones y velocidades a partir de series de Taylor alrededor de un tiempo t .

5.3.3 Condiciones de periodicidad

Las condiciones de periodicidad se utilizan para simular un sistema mucho más grande que el que verdaderamente se tiene, evitando así efectos de superficie. La idea es repetir el cubo computacional en el espacio en todas direcciones, para formar un sistema infinito. En cada cubo imagen, las partículas se mueven exactamente de la misma manera que en el original.

5.4 Visualización de dinámica molecular

5.4.1 Visualización molecular para dinámica molecular

En general, el procedimiento consiste en lo siguiente: los datos precalculados, y que tienen un significado (obtenidos luego de la realización de los procesos de dinámica mencionados), son descargados de la base y son tomados con un formato predeterminado por un programa especializado, es decir el programa lee datos numéricos, que por lo regular consisten en coordenadas y propiedades de las partículas, y por medio de funciones especiales y rutinas gráficas adecuadas, propias del equipo utilizado para la visualización, la plataforma de desarrollo, y el lenguaje usado, se lleva a cabo la representación de las partículas, generalmente dibujando esferas, aunque puede haber diversas representaciones, dentro de un espacio tridimensional, esta gráfica puede ser modificada y estudiadas sus propiedades según lo requieran los investigadores.

5.4.2 Edición de estructuras

Más allá de la visualización de una estructura determinada, por ejemplo, a través de un método cristalográfico, las herramientas de computación gráficas permiten obtener datos geométricos, tales como distancias, ángulos o dimensiones de los poros para ser calculados de una manera sencilla. La velocidad de los sistemas gráficos permite el monitoreo de datos como: distancias y ángulos interactivamente, posibles coincidencias de elementos y aun simples estimaciones de fuerzas de interacción como ajustes que son hechos a una estructura.

La manipulación geométrica incluye edición de estructuras en las cuales un usuario puede rápidamente construir, adicionar, o cortar a lo largo de un sistema químico dado. Por ejemplo, se puede hacer uso de una plantilla interactiva definible al seleccionar una pequeña cantidad de átomos para un estudio posterior.

Las herramientas eficientes para construir estructuras moleculares y extendidas están disponibles. Los átomos pueden ser movidos o borrados a partir de la estructura a la simetría del grupo del espacio, o la simetría por sí misma puede ser ajustada. En este contexto la importancia de un enlace cerrado entre gráficos y herramientas de simulación es evidente. La minimización de las mallas de energía o métodos de mecánica molecular pueden ser usados para optimizar un modelo estructural construido por estos métodos gráficos.

5.4.3 Técnicas gráficas

Las técnicas gráficas permiten al investigador examinar rápidamente la copiosa salida de un cálculo de dinámica molecular, examinando una trayectoria para migración de eventos significantes. Los beneficios de animar un cálculo de dinámica molecular pueden ser mejorados volviendo a representar la evolución de los datos de la dinámica en sincronización con la animación. Los sistemas moleculares gráficos pueden aportar datos no sobresalientes detrás de los eventos clave en la simulación. Las herramientas gráficas pueden entonces ayudar al extrapolar a partir de resultados de simulaciones atómicas, hacia una apreciación de su objetivo en determinar propiedades macroscópicas.

CAPITULO II

**HERRAMIENTAS DE DESARROLLO PARA EL
SISTEMA DE VISUALIZACIÓN DE DINÁMICA
MOLECULAR**

6. CAPÍTULO II HERRAMIENTAS DE DESARROLLO PARA EL SISTEMA DE VISUALIZACIÓN DE DINÁMICA MOLECULAR

6.1. Lenguaje C++

El lenguaje de programación C, es un lenguaje de nivel medio, ya que proporciona las características de los lenguajes de bajo nivel, así como las ventajas que proporcionan los lenguajes de alto nivel. El lenguaje de programación C++, es un superconjunto de C, diseñado por Bjarne Stroustrup a principios de los años 80's, en AT&T Bell Laboratories, y quién, además de mejorar las características de C, le añadió recursos para la programación orientada a objetos. En realidad, el C++, no es más que una evolución lógica del lenguaje C con nuevas características que hacen a C++, un mejor y más poderoso C, al que se le han incluido los conceptos de soporte para crear y utilizar la abstracción de datos, la división de la programación en módulos y la ocultación de datos, junto con las propiedades específicas de la programación orientada a objetos.

Una gran ventaja de usar C++ es la gran compatibilidad que proporciona con los programas en C, ya que si bien es preciso el adiestramiento en las nuevas características del lenguaje, y naturalmente, formación en las técnicas orientadas a objetos, el programador podrá comenzar a utilizar C++ desde el comienzo de su aprendizaje. Por otra parte, los programadores pueden hacer uso de prácticamente todo el código C existente sin ninguna modificación, enlazándolo con los programas C++ sin requerir siquiera una recompilación del citado código en C.

Además de los recursos propios de C, C++ proporciona nuevos recursos relativos a orientación a objetos, como son:

- Abstracción de datos.
- Verificación estricta de tipos.
- Paso de argumentos por referencia.
- Limpieza automática de memoria.
- Inicialización garantizada.
- Sobrecarga de operadores y funciones.
- Tipos genéricos o parametrizados ("*templates*" o plantillas basadas en el tipo de dato recibido).
- Tratamiento de excepciones.
- Conversiones de tipos.

6.2. Lenguaje Fortran

El lenguaje de programación FORTRAN, fue introducido en 1956, como el primer lenguaje de alto nivel. Desde entonces ha sido revisado varias veces. Actualmente, aunque FORTRAN 77, no es la última versión, es ampliamente compatible con las versiones anteriores.

Es el primer lenguaje de programación usado para aplicaciones científicas y numéricas, y es también el mas ampliamente utilizado alrededor del mundo. A las versiones originales le hacían falta procedimientos recursivos y estructuras de bloques, tiene una sintaxis orientada a líneas en la cual ciertas columnas tienen una importancia especial.

Su nombre es un acrónimo de FORMula TRANslator, FORTRAN es el lenguaje de programación de alto nivel más antiguo, es un lenguaje de programación de 3ra generación (3GL), diseñado por John Backus para IBM en 1954. Tiene una sintaxis muy breve, y fue creado para uso de ingenieros, matemáticos y otros usuarios de algoritmos científicos. Aún es popular actualmente, particularmente para aplicaciones científicas que requieren de computaciones matemáticas extensas. Un código Fortran fue el primer programa en ser portado de uno de los vendedores de hardware a otro. Este fue también el primer lenguaje de computadora que fue objeto de un estándar internacional.

Las dos versiones más comunes de FORTRAN son FORTRAN IV y FORTRAN 77. FORTRAN IV fue aprobado como un estándar USASI en 1966. FORTRAN 77 es una versión de FORTRAN que fue aprobada por ANSI en 1977 (lo aprobaron en 1977, de aquí el nombre). FORTRAN 77 incluye un número de características no disponibles en versiones

más antiguas de FORTRAN. Un nuevo estándar ISO y ANSI para FORTRAN, llamado FORTRAN-90, fue desarrollado a principios de los 90s.

FORTRAN siempre ha sido el lenguaje preferido por científicos e ingenieros. Este lenguaje ha visto muchos desarrollos:

- 1.FORTRAN II
- 2.FORTRAN IV
- 3.FORTRAN 66
- 4.FORTRAN 77
- 5.Fortran 90
- 6.Fortran 95
- 7.Fortran 2000

6.3. Sistema operativo UNIX

Un sistema operativo es el programa o conjunto de programas que controlan todas los componentes de un sistema de computadora tanto a nivel *hardware* como *software*. La característica más importante de un sistema operativo, es que permite usar las facilidades proveídas por el sistema. Todas las computadoras tienen un sistema operativo.

El sistema operativo UNIX tiene tres importantes componentes, el “kernel”, el “shell” y el sistema de archivos.

6.3.1. El “kernel” o núcleo

Como su nombre lo indica es el núcleo de cada sistema operativo UNIX y es cargado siempre que el sistema se inicia.

Este maneja todos los recursos del sistema, presentándolos a cada usuario como un sistema coherente. No es necesario saber nada acerca del núcleo para poder utilizar un sistema UNIX.

Las siguientes son algunas de las funciones realizadas por el *kernel*:

- Administrar la memoria de la computadora y asignarla para cada proceso.
- Programar el trabajo destinado al CPU de modo que los procesos de cada usuario sean realizados tan eficientemente como sea posible.
- Organizar la transferencia de datos desde cada una de las partes de la máquina a otra.
- Aceptar instrucciones desde el *shell* y realizarlas.
- Hacer cumplir los permisos de acceso que tienen los archivos en el sistema de archivos.

6.3.2. El “shell” (Interprete de comandos)

Siempre que se accede a un sistema UNIX, éste coloca al usuario en un programa llamado “shell”. El *shell* recibe este nombre por ser la capa externa del sistema operativo. Se puede ver un indicador de sistema operativo, llamado “propmt”, en la parte inferior izquierda de la pantalla. Para realizar algún trabajo, es necesario introducir comandos en este indicador de sistema.

El *shell* actúa como un intérprete de comandos, toma cada comando y lo pasa al núcleo del sistema operativo, para que actúe sobre ellos y posteriormente despliega los resultados de la operación realizada en la pantalla, a través del indicador del sistema.

6.3.3. El sistema de archivos

Un sistema de archivos es un método lógico de organizar y almacenar una gran cantidad de información de manera que ésta sea fácil de manejar. Un archivo es la unidad más pequeña en la cual la información es almacenada. El sistema de archivos UNIX tiene algunas características importantes.

6.3.3.1. Distintos tipos de archivos

Los sistemas de archivos UNIX, contienen varios tipos de archivos para el manejo de información.

- Archivos ordinarios

Este tipo de archivos es usado para almacenar la información de usuario, tal como textos, o imágenes. Este es el tipo de archivos con los que usualmente se trabaja.

- Directorios

Un directorio es un archivo que contiene a otros archivos y otros directorios. Se pueden crear directorios dentro del directorio de algún usuario, para guardar archivos y otros subdirectorios

El tener una estructura de directorios propia, proporciona a los usuarios un lugar para trabajar con posibilidades de personalización y permite dar una estructura a la información, de manera que tenga un mejor sentido para cada usuario.

Los directorios son propiedad de los usuarios, y ellos pueden fijar permisos de acceso para controlar qué usuarios pueden tener acceso a la información que estos contienen.

- Archivos especiales

Este tipo de archivos es utilizado para representar un dispositivo físico real tal como una impresora, una unidad de cinta o una terminal.

Puede parecer inusual pensar en un dispositivo físico como un archivo, pero esto permite enviar la salida de algún comando a un dispositivo en la misma forma en que se enviaría a un archivo, esto es manejarlo como si fuera un archivo en realidad.

El directorio `/dev` contiene los archivos especiales que son usados para representar dispositivos en un sistema UNIX.

- “Pipes” (Entubamientos)

UNIX permite enlazar comandos juntos, utilizando un “pipe” (una tubería). El *pipe*, actúa como un archivo temporal que sólo existe para mantener datos desde un comando hasta que son leídos por otro.

6.3.3.2. Estructura del sistema de archivos

El sistema de archivos de UNIX, está organizado como una jerarquía de directorios comenzando desde un simple directorio llamado “root” el cual es representado por una diagonal “/”.

Inmediatamente abajo del directorio *root* hay varios sistemas de directorios que contienen información requerida para el sistema operativo. El archivo que contiene el *kernel* de UNIX está también aquí.

- Directorios hogar (directorios de inicio, llamados directorios *home*)

Cualquier sistema UNIX, puede tener muchos usuarios al mismo tiempo. A todos los usuarios, se les proporciona un directorio de inicio de sesión. Los directorios de inicio son usualmente agrupados juntos bajo un directorio de sistema tal como */home*. Un gran sistema UNIX, puede tener varios cientos de usuarios, con sus directorios de inicio agrupados en subdirectorios de acuerdo a algún esquema similar a su departamento organizacional.

- Permisos de acceso

Para controlar el acceso a sus archivos, los usuarios pueden protegerlos, o hacerlos accesibles a los otros usuarios, cambiando sus permisos de acceso.

Un usuario sólo puede cambiar los permisos para los archivos y directorios de su propiedad.

6.3.4. Sistema operativo Linux

Linux, es una opción relativamente nueva en sistemas operativos, y es una versión de distribución libremente e independiente, del sistema operativo UNIX, y hay para diferentes plataformas como son: máquinas x86, Motorola 68k, Digital Alpha, Sparc, Mips y Motorola Power PC, y últimamente estaciones de trabajo SUN y SGI.

En la actualidad, este sistema operativo es utilizado por miles de usuarios para desarrollo de software, redes y para plataformas de usuarios finales.

Linux, entre los miles de sistemas operativos alternos que existen, se ha convertido en una opción interesante, independientemente de que estas vengan de UNIX o de las más conocidas donde se encuentra Windows y NT.

Es una implantación de la especificación POSIX con la cual cumplen todas las verdaderas versiones de UNIX.

El núcleo de Linux no usa código de AT&T o de cualquier otra fuente propietaria, la mayoría de los programas disponibles para Linux son desarrollados por el proyecto GNU de la *Free Software Foundation*.

Este sistema operativo soporta una amplia gama de aplicaciones o paquetes de programación tales como X Window, Emacs, redes de datos bajo protocolos TCP/IP (incluyendo SLIP, PPP, ISDN).

Linux está disponible en Internet en cientos de servidores ftp y en distribuidores en discos CD-ROM de revendedores que lo ofrecen empacado con manuales e información que es realmente la del costo, pues el programa es gratuito.

Algunos de estos son: Caldera, Debian, Slackware, Red Hat, etc. El núcleo del Linux está legalmente protegido por la licencia pública GNU (GPL).

Linux incluye compiladores, ensambladores, depuradores, editores de texto, paquetes de correo electrónico, lectores de noticias, navegadores, servidores y programas para la creación y edición gráfica.

6.3.4.1. Historia

Linux fue creado originalmente por Linus Benedict Torvalds en la Universidad de Helsinki en Finlandia. Este ha sido desarrollado con la ayuda de muchos programadores a través de Internet. Linus originalmente inició la modificación del núcleo como su proyecto favorito, inspirado por su interés en MINIX, un pequeño sistema Unix. El se propuso a crear lo que en sus propias palabras sería un mejor Minix que el Minix.

El 5 de octubre de 1991, Linus anunció su primera versión "oficial" de linux, versión 0.02. Desde entonces, muchos programadoras han respondido a su llamado, y han ayudado a construir Linux como el sistema operativo completamente funcional que es hoy. Linux soporta muchos periféricos, desde procesadores hasta sintonizadores de televisión, discos compactos no ATAPI y reconoce buena cantidad de tarjetas de sonido. Incluye también soporte para tipos de archivos para Macintosh HFS, Unix UFS y en modo de lectura, HPFS de OS/2 y NTFS, de NT.

Además tiene todas las características esperadas en una versión moderna y completa de UNIX, incluyendo multitarea real, memoria virtual, bibliotecas compartidas, redes TCP/IP en entre otras. La última versión estable es la versión 2.6.

6.3.4.2. Ventajas

- Preciõ.
- Estabilidad, no se traba a cada rato.
- Seguridad, es mucho más seguro que otros sistemas operativos.
- Compatibilidad, reconoce la mayoría de los otros sistemas operativos en una red.
- Velocidad, es mucho más veloz para realizar las tareas.
- Posee el apoyo de miles de programadores y usuarios a nivel mundial.
- El paquete incluye el código fuente, lo que permite modificarlo de acuerdo a las necesidades del usuario.
- Ideal para la programación, ya que se puede programar tanto en Linux para distintas plataformas, como para Windows.
- Un sistema de crecimiento rápido.
- Se puede usar en casi cualquier computadora, desde una 386, hasta una estación de trabajo.
- Puede manejar múltiples procesadores.
- Es menos vulnerable a los virus.
- Maneja discos duros de hasta 16 TeraBytes.
- Se consiguen parches con facilidad, además de ser gratuitos.
- Los fabricantes de Hardware le están dando su apoyo, como IBM y Compaq, y últimamente compañías como SUN y SGI.
- Vendedores y desarrolladores implementan un sistema de certificación para Linux.
- La corporación DATA Internacional predice que el crecimiento de este programa será del orden de un 25 por ciento anual en el nuevo milenio.

6.3.4.3. Desventajas

- Linux no cuenta con una empresa que lo respalde, por lo que no existe un verdadero soporte como el de otros sistemas operativos, aunque hay distribuciones muy fuertes como Red Hat y empresas que ha empezado a impulsarlo como IBM e Intel.
- Linux corre el riesgo de llegar a fragmentarse como fue el caso de UNIX.
- Algunas empresas pueden llegar a ayudar a Linux con la intención de mejorar sus relaciones públicas, aunque en el fondo no tengan ninguna intención de utilizarlo fielmente.

6.3.4.4. Diferencias entre Windows y Linux

Linux es técnicamente superior a cualquier ambiente basado en DOS, como Windows 9x y aún Windows NT. Estas son las principales diferencias entre DOS/Windows y Linux:

- Windows corre Microsoft Office y una gran cantidad de juegos.
 - Es visto para ser fácil de instalar y configurar.
 - Es notoriamente inestable.
 - Se desempeña pobremente.
 - Las caídas del sistema son frecuentes.
- Linux corre StarOffice (entre algunas aplicaciones suite de oficina, como Koffice), software técnico y menos juegos;
 - Puede ser difícil de instalar y configurar;
 - Es uno de los sistemas operativos mas seguros que hay actualmente.
 - Se desempeña impecablemente.
 - Las caídas del sistema son extremadamente raras.

Linux da más poder al usuario, pero toma algo de tiempo aprender como manejarlo. Debido a que representa una buena cantidad de dificultades iniciales.

Actualmente, las personas que desarrollan Linux, alrededor del mundo, trabajan para hacerlo más simple de usar, sin embargo, siempre es necesario tener una gran cantidad de documentación y usarla al menos unos cuantos meses.

Una cosa importante, es que Linux y DOS/Win pueden coexistir satisfactoriamente en una misma computadora. Pero se debe recordar, que Windows y Linux son totalmente diferentes, tanto como lo son DOS y UNIX, ya que el sistema operativo DOS tiene una muy pobre o escasa relación con el sistema operativo UNIX.

6.4. Supercómputo : Cómputo de alto desempeño

El supercómputo puede ser concebido como una clasificación de disciplinas científicas e industriales que intentan computar con procesos de alto desempeño teniendo en cuenta los actuales requerimientos de funcionamiento. Su lugar está dentro de la ciencia de la computación debido a que su mayor objetivo es tomar ventaja de todo el potencial de desempeño disponible que puede ser ofrecido a los diferentes niveles de los procesos computacionales. Sin embargo, es fuertemente impactado por otras ciencias contribuyentes, asociadas con aplicaciones científicas e industriales de alto desempeño, es por ello que aquellos que lo utilizan ven al supercómputo como la manera de alcanzar sus propios objetivos de desempeño.

Los propósitos del supercómputo son:

- Proporcionar soluciones a importantes problemas en ciencia e ingeniería proveyendo recursos computacionales de primer nivel para la comunidad científica e industrial.
- Avance en ciencia computacional, y técnicas computacionales.
- Asistir al sector privado en la explotación del cómputo de alto desempeño debido a sus ventajas competitivas.
- Aumentar la velocidad en la computación distribuida.

Investigaciones específicas incluyen integrar avances en optimizadores y compiladores paralelos, nuevas arquitecturas paralelas, y algoritmos paralelos.

De esta forma, el supercómputo se ha convertido en una verdadera mezcla de diversos componentes científicos e industriales dirigidos a permitir procesos de aplicaciones de desempeño sostenido y superior o alto desempeño, basándose en el procesamiento paralelo de las diferentes tareas, aprovechando todos los recursos con los que cuenta una supercomputadora.

Actualmente las investigaciones por medio de algoritmos que simulan el comportamiento de la naturaleza, han cobrado gran importancia con el uso de las supercomputadoras.

6.5. Bibliotecas gráficas : La integración de ventanas y gráficos en 3D

6.5.1. Una mirada a la historia

Hace algunos años, muchos científicos, ingenieros y otros investigadores y desarrolladores en diferentes campos de la computación, descubrieron el poder y los beneficios de los gráficos en tres dimensiones. También se dieron cuenta, después de gastar una gran cantidad de tiempo y esfuerzo, que el lenguaje usado para aprovechar el sistema gráfico de desplegado, a menudo no era muy conveniente para la integración con otras tareas a las que se enfrentaron.

No sucedió así en el caso de los gráficos en tres dimensiones y los ambientes de ventanas. De hecho, hace unos cuantos años que un vendedor en la conferencia anual SIGGRAPH declaró que no había gráficos disponibles dentro de un ambiente de ventanas. El se equivocó, Tektroniks creó un primer intento de tener una coexistencia de gráficos en tres dimensiones con sistema de ventanas X-Window, pero este primer intento fue muy crudo, no había concepto de integración entre las dos ideas, llamadas gráficos 3D y ambientes de ventana.

Hoy en día la mera aparición de una de interfaz de usuario de caracteres es una rareza, ya que virtualmente todas las computadoras (desde computadoras personales baratas hasta el mundo de las estaciones de trabajo gráficas) tienen algún ambiente de ventanas. Algunos sistemas satisfacen al usuario que quiere utilizar los gráficos en 3D, pero la característica de usar los componentes de aceleradores gráficos 3D y el ambiente de ventanas plantea algunos intereses únicos. Durante algún tiempo, los usuarios estuvieron contentos de vivir con los dos ambientes diferentes y a menudo enteramente incompatibles. Hoy, los usuarios demandan interoperabilidad, portabilidad y que no haya pérdidas en el desempeño.

Dentro de todo este contexto, es necesario tener en cuenta el concepto de biblioteca gráfica, una biblioteca gráfica consiste en un conjunto de subrutinas o funciones, escritas en algún determinado lenguaje de programación, que permiten al programador o desarrollador, representar y “dibujar” gráficamente los resultados de un proceso, generalmente, en una estación de trabajo, con grandes capacidades gráficas.

Entre algunos de estos grupos de subrutinas gráficas tenemos los siguientes:

6.5.2. IRIS GL

IRIS GL, es un conjunto de bibliotecas gráficas, propiedad de Silicon Graphics, las cuales fueron desarrolladas de 1982 a 1992, por esta empresa fabricante de estaciones de trabajo especializadas en gráficas de alta calidad. IRIS GL es una biblioteca de subrutinas para crear gráficos y animaciones con color en dos y tres dimensiones.

IRIS GL es una biblioteca de subrutinas que pueden ser llamadas desde un programa en lenguaje C, para dibujar y animar escenas gráficas con color en dos y tres dimensiones. Las aplicaciones desarrolladas con estas bibliotecas, son construidas utilizando comandos GL, dentro del marco de trabajo del lenguaje de programación. El lenguaje de programación provee la estructura lógica, en tanto que los comandos GL proporcionan la interfaz para el *software* y *hardware* gráficos. Las bibliotecas GL, son transparentes a la red, es decir, se pueden desplegar los resultados a través de la red, en una estación de trabajo remota, compartiendo el proceso de tareas con otros sistemas y desplegando gráficos en múltiples pantallas.

El lenguaje de programación C provee un área de trabajo para desarrollar programas GL. Idealmente los programas GL son independientes de cualquier plataforma particular IRIS (IRIS es el sistema operativo propiedad de Silicon Graphics y es el que tienen instalado las estaciones de trabajo de esta compañía), y del *hardware* gráfico en que este corriendo la aplicación.

Los pasos que debe seguir un programa GL son los siguientes:

- Consultar al sistema sobre la disponibilidad de recursos gráficos.
- Iniciar las bibliotecas gráficas GL.
- Llamar a las subrutinas GL para fijar y obtener el estado global de los atributos y crear el cálculo de iluminaciones y sombreado (“render”) gráfico, es decir realizar los cálculos de luminosidad de una determinada escena.
- Salir de las gráficas.

Este conjunto de bibliotecas gráficas fueron, a su vez, el punto de inicio para el posterior desarrollo de las bibliotecas gráficas OpenGL.

6.5.3. OpenGL

6.5.3.1. Características

Aquí, el ambiente de interfaces gráficas OpenGL, es discutido en términos de sus capacidades para proveer un lenguaje de programación de grandes características gráficas en una forma neutral, y su compatibilidad para ser utilizado con diversos sistemas de ventanas.

OpenGL como una interfaz de programación de aplicaciones (API)

Mientras el sistema de ventanas X ha llegado a ser el estándar preestablecido en el mundo del mercado de estaciones de trabajo UNIX, OpenGL fue diseñado para ser independiente de los diferentes sistemas de ventanas existentes. Esto quiere decir que los otros ambientes de ventanas, como Windows/NT, son también capaces de soportar OpenGL. Esto es un muy importante beneficio, y es posible ya que la especificación OpenGL es independiente de :

- Sistema de ventanas
- Sistema operativo
- Red.

Además, todas las implementaciones de OpenGL, independientemente del vendedor, deben ser completamente apegadas a los estándares básicos, así el usuario/programador esta seguro de que todas las características y funciones básicas estarán disponibles en todas las diferentes plataformas. Cada proveedor esta requerido de pasar por una serie de exámenes para asegurarse de la compatibilidad de código entre todas las implementaciones OpenGL.

Como un ejemplo de neutralidad de OpenGL, a continuación hay una lista de vendedores que soportan OpenGL (diciembre de 1995):

•Silicon Graphics •A T & T •Barco Chromatics •Cirrus Logic (que compro la tecnología A1060 de Austek Microsystems)
•Cray Research •Daikin •Digital Equipment •3Dlabs •Fujitsu •Evans & Sutherland •Harris Computer •Hewlett-Packard
•Hitachi •Hummingbird Communications •IBM •The Institute for Information Industry •Intel •Intergraph •Kubota Pacific
•Media Vision •Microsoft •Miro •Mitsubishi •NEC •Peritek •Portable Graphics (Formalmente Nth Portable Graphics; que soporta Sun and HP) •PsiTech •RasterOps •SPEA •Samsung •Siemens-Nixdorf •Sony •Sun •Template Graphics Software
•Univel

OpenGL por si mismo es controlado por un consorcio industrial, OpenGL *Architectural Review Board* (Comité de Revisión Arquitectural). Los miembros del mayor vendedor de *hardware* y *software* unidos, guiaron y el crecimiento y desarrollo de OpenGL, de este modo se asegura que ningún único vendedor indica la dirección sobre OpenGL.

En suma, el ARB aprobó la especificación OpenGL 1.1 en diciembre (1995). Esta fue la primera actualización a la especificación desde la primera versión. La ultima especificación sobre OpenGL que ha sido liberada es la OpenGL 1.5

6.5.3.2. Lo que no es OpenGL

OpenGL no es:

- Un conjunto de herramientas o una interfaz de programación de aplicaciones de alto nivel.
- Un sistema de ventanas.
- Un sistema gráfico descriptivo.
- No es orientado a objetos.

Esto no es tampoco un sistema de ventanas, y es dependiente de ventanas para hacer todas las tareas relativas a ventanas (como crear el lienzo o base “canvas” sobre la cual se dibujan las figuras en 2D y 3D), tratando con entrada del usuario, etc.). OpenGL no es descriptivo, esto es, el programador no fija un modelo de la escena a ser desplegada con cálculos de iluminación y sombras “render”, y enseguida deja al manejador de sistema gráfico las tareas para hacer el

dibujo. Finalmente, OpenGL no es orientado a objetos (pero hay un conjunto de herramientas gráficas 3D, orientadas a objetos que están actualmente disponibles, como Open Inventor, por ejemplo).

6.5.4. Biblioteca gráfica 3D Mesa

Mesa es una biblioteca gráfica en tres dimensiones, con un API, el cual es muy similar al de OpenGL. Al grado de que Mesa utiliza la sintaxis de los comandos OpenGL o su máquina de estados, y es para ser usado con autorización de la compañía Silicon Graphics (SGI). Sin embargo, el autor no posee una licencia OpenGL de SGI, y afirma que Mesa no es de algún modo un reemplazo compatible de OpenGL o asociados con SGI. Mesa es distribuido bajo los términos de biblioteca de Licencia Pública General GNU. La mayoría de las aplicaciones escritas para OpenGL pueden usar Mesa sin cambiar el código fuente.

6.5.4.1. Sistemas soportados

Mesa fue diseñado originalmente para sistemas UNIX/X11 y es por lo tanto mejor soportado sobre dichos sistemas. Todo lo que se necesita es un compilador ANSI C y el ambiente de desarrollo de X-Window para poder usar Mesa. Otros desarrolladores han contribuido con manejadores para plataformas Amiga, Apple Macintosh, BeOS, NeXT, OS/2, MS-DOS, VMS, y Windows 95/NT.

6.5.5. Otras Herramientas

Existen algunas otras bibliotecas gráficas para desarrollo, propias de cada sistema UNIX, aunque la mayoría de ellas están basadas en OpenGL o Mesa, lo cual hace que las aplicaciones desarrolladas con estas bibliotecas sean compatibles y por lo tanto portables entre diferentes sistemas UNIX.

Entre estas tenemos las siguientes:

6.5.5.1. Gtkgl

Se trata de un componente OpenGL para GTK+GUI toolkit. Las funciones de bajo nivel gdkgl hacen más fácil el cálculo de iluminaciones sobre cualquier componente que tiene capacidad visual OpenGL.

6.5.5.2. GLE

Tubing and Extrusion Library, Biblioteca de Entubamiento y Extrusión, GLE es un paquete de bibliotecas de funciones en C que dibuja superficies extrudidas, incluyendo superficies de revolución, curvas, tubos, polígonos, policilindros y helicoides. Genéricamente, las superficies extrudidas son especificadas con un polilínea en dos dimensiones que es extrudida a lo largo de una trayectoria en tres dimensiones. Un sistema de coordenadas local permite flexibilidad adicional en las primitivas dibujadas. La extrusión puede ser una textura mapeada en una variedad de maneras. La biblioteca GLE genera coordenadas triangulares 3D, vectores de iluminación normal y coordenadas de textura como salida. GLE usa el API de GL o de OpenGL para ejecutar el cálculo de iluminación actual.

6.5.5.3. GLUT (OpenGL Utility Toolkit).

Es un agradable conjunto de bibliotecas que permiten escribir aplicaciones sin tener preocuparse sobre el sistema de ventanas y su interfaz para OpenGL. Es mucho más amigable que aux y tk, inicialmente usados por SGI. Trabaja sobre Mesa también.

6.5.5.4. Magician

La interfaz Magician es una implementación de OpenGL para Java. Utilizando Magician, los programadores pueden escribir código Java portable que sin modificaciones usa bibliotecas nativas existentes para proveer cálculo de iluminación sobre una variedad de plataformas, todas basadas en UNIX y Win 32. Magician también puede proveer componentes extensibles AWT para incrustar cálculo de iluminación de superficies tridimensionales dentro de las aplicaciones. Magician tiene muchas características poderosas para permitir a los desarrolladores la posibilidad de escribir magníficas aplicaciones basadas en OpenGL rápidamente.

6.5.5.5. PMesa (*Parallel Mesa*)

Es una versión modificada de Mesa, una biblioteca gráfica con un "API" muy similar al de OpenGL. PMesa es una actualización del anterior Mesa 2.6. PMesa proporciona un mayor beneficio a partir de las computadoras SMP (*Single Multy Procesor*, equipos cuya característica es tener diversos procesadores conectados en el interior de una máquina paralela, compartiendo recursos como memoria y buses en una sola máquina) para las transformaciones geométricas de Mesa. La primera razón para esta paralelización es que los aceleradores actuales pueden **rasterizar**¹ más triángulos de los que la geometría puede transformar (hecha en el CPU). Esta es la razón por la cual algunos fabricantes de microprocesadores introducen nuevos conjuntos de instrucciones de punto flotante SIMD. Otras maneras de acelerar las transformaciones geométricas es usar computadoras SMP con hilos de ejecución. PMesa usa hilos de ejecución POSIX. Este sólo puede ser examinado sobre Linux, pero debería trabajar posteriormente, en otros sistemas UNIX.

¹ **rasterizar** El proceso de creación de datos de píxeles a partir de elementos vectoriales.
<http://www.telecable.es/personales/charlyg/glosario.htm>

CAPITULO III

**INTERFACES DE INTERACCIÓN ENTRE EL
SISTEMA Y EL USUARIO**

7. CAPÍTULO III INTERFACES DE INTERACCIÓN ENTRE EL SISTEMA Y EL USUARIO

7.1 Antecedentes de interfaces gráficas

La actual disponibilidad de computadoras rápidas y baratas así como de aplicaciones que aumentan la productividad han convertido en usuarios de computadoras a una parte importante de la población. Los avances en facilidad de uso (hechos posibles por las interfaces con el usuario actualmente existentes), han ayudado mucho en este proceso. Estas interfaces son posibles gracias a la proliferación de los dispositivos gráficos y el ratón, y dependen de las contribuciones de diseñadores de interfaces que han creado toda una nueva disciplina de diseño (con sus propias herramientas y métodos).

Los avances de hardware y reducciones continuas en el factor precio/rendimiento, la generalización de multimedia y *hardware* para gráficas tridimensionales en tiempo real con iluminación y sombreado en las estaciones de trabajo más baratas y computadoras personales llevarán a un cambio de paradigma parecido al que surgió del trabajo pionero en Xerox PARC sobre estaciones de trabajo con dispositivos gráficos a principio de los setentas.

Uno de los principales cambios en computación que están próximos es la introducción de gráficas tridimensionales en tiempo real a las aplicaciones de uso diario y la aparición de nuevas aplicaciones tridimensionales. En los años ochentas las computadoras personales fueron lo suficientemente rápidas para correr aplicaciones interactivas bidimensionales como las aplicaciones de dibujo y pintura, procesadores de palabra y publicación.

En tanto que en los noventas se han ido desarrollando muchas aplicaciones tridimensionales que van desde las que están enfocadas para especialistas (CAD/CAM en 3D, visualización científica) como aplicaciones de utilidad general (programas de ilustración y animación 3D, diseño de interiores).

Pero inclusive en otras aplicaciones ahora consideradas bidimensionales (hojas de cálculo, CASE, documentación interactiva), se han ido logrando los mismos beneficios de las gráficas tridimensionales en tiempo real ahora obtenidos por los científicos e ingenieros por medio del uso de la visualización científica.

A pesar de que las aplicaciones tridimensionales han sido comunes por años en campos como CAD/CAM y visualización, las interfaces han sido en su mayoría bidimensionales basadas en menús, botones, etc. Hay pocas formas de interacción tridimensional más allá de cursores (3D) y simulaciones de bastones de mando usando esferas virtuales.

Esto se debe en parte a que pocas plataformas tenían gráficas tridimensionales en tiempo real, y también a que el diseño de interfaces tridimensionales es mucho más complejo. Además se ha investigado poco en cuanto a nuevas metáforas y paradigmas de interacción tridimensionales. No solo deben mejorar los dispositivos de entrada y salida, sino que deben diseñarse nuevas formas de interactuar con objetos o datos y sus relaciones.

7.2 Interacción con el usuario

Hay que considerar los pros y los contras de la interacción directa contra la interacción indirecta a través de menús, botones, y otros controles. Se debe analizar si la actual separación entre los objetos y sus controles (los que inclusive se diseñan con herramientas propias) es adecuada. Las gráficas tridimensionales pueden servir también para crear algunos de esos controles. En este caso las gráficas tridimensionales no son únicamente una tecnología o una aplicación, sino también una herramienta para crear interfaces de usuario útiles e interesantes.

Desde la perspectiva del usuario, el hecho de poder aprovechar las ventajas de la interacción indirecta, sin tener que interesarse por problemas como "cómo funciona internamente tal o cual programa", y ocuparse mejor por cómo utilizarlo es mucho más cómodo para ellos.

Mientras que por el lado del desarrollador, es mejor saber que los usuarios no se adentraran en cuestiones más profundas con respecto al funcionamiento de los sistemas y que están en un nivel más abstracto, esto es las interfaces gráficas pueden hacer que un sistema sea transparente a los usuarios, ya que ellos no se preocupan por lo que no ven.

7.3 Conceptos básicos de interfaces gráficas

7.3.1 Definición de interfaz gráfica

Lewis y Rieman en 1993 definieron a las interfaces hombre-computadora como:

“Las interfaces básicas de usuario son aquellas que incluyen cosas como menús, ventanas, teclado, ratón, los "beeps" y algunos otros sonidos que la computadora hace, en general, todos aquellos canales por los cuales se permite la comunicación entre el hombre y la computadora”.

En base a la tesis de Sjoerd Michels [1995] esta definición de interfaz hombre-computadora, no es muy completa debido a que solo menciona que existe una comunicación entre ellos, pero no menciona ningún objetivo o meta. Aparentemente la comunicación se da a través de la selección de alguna opción dentro de algún menú o mediante alguna instrucción que es introducida a la computadora por medio del teclado. Pero, ¿Cómo es que esto responde?, ¿Cómo es que esta información es comunicada al usuario?, ¿Que pasa cuando el usuario ejecuta una acción?

Una interfaz gráfica debe contemplar los siguientes aspectos:

- Es la parte de cualquier programa o aplicación la cual esta enfocada básicamente a la interacción entre el usuario y la computadora.
- La aplicación es realizada en ambiente gráfico.
- Es centrada en el usuario (orientado a las tareas). Esto quiere decir que esta basado principalmente en las necesidades que el usuario tenga, es decir lo que él debe de realizar en la aplicación. Por ejemplo, en una aplicación el usuario necesita una herramienta para rellenar algún área con un color que él seleccionó. Esta necesidad es una tarea.

Dentro del modelo del diseño de una interfaz centrada en el usuario, como se observa en el diagrama 7.a , el análisis de las tareas es un punto muy importante para obtener información.

Se aplica la idea de "common ground" (bases para hacer algo en común) y se trata de utilizar esta metodología durante todo el proceso de interacción. "common ground" es un interesante modelo de comunicación creado por Clark y Brennan en 1990. Esto es coordinar el contenido, cada uno de los componentes con los que interactúa la interfaz gráfica, y que conforman el sistema. Además se deberán de sincronizar sus entradas y salidas, en otras palabras ajustarse los componentes de uno al otro en tiempo y dinámica. Esto es coordinar el proceso. Para coordinar el contenido es necesario que los componentes tengan mucha información compartida, tal como: conocimiento, creencias (reglas), suposiciones etc. Para coordinar el proceso es necesario ir actualizando el "common ground" momento a momento.

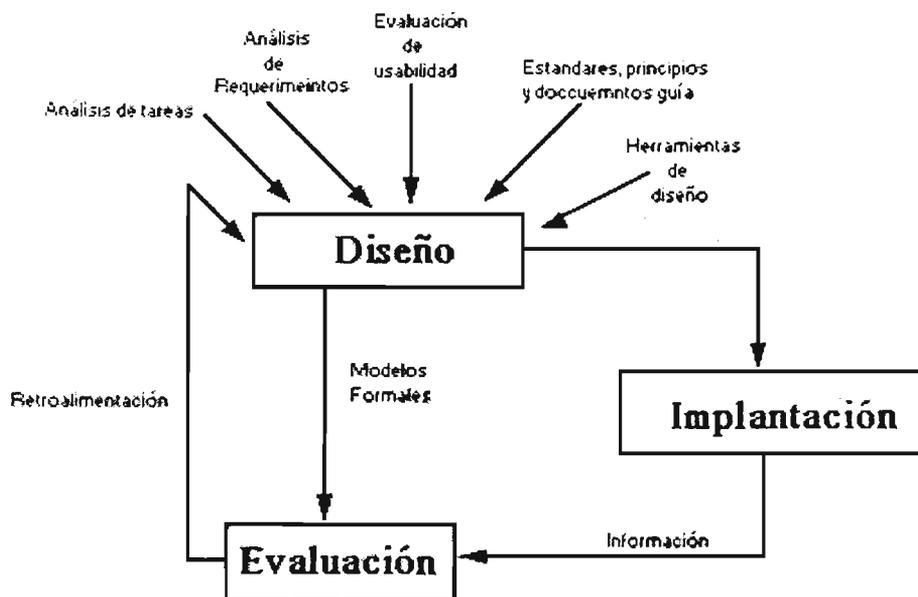


Diagrama 7.a Modelo del diseño de una interfaz

- Se utilizan los tres principales estilos de interacción para las interfaces gráficas hombre-computadora.
 1. Manipulación directa. Los usuarios sienten que están a cargo de las actividades de la computadora. Cuando realizan alguna actividad física en ella (clic en algún objeto por ejemplo), esta les proporciona un resultado. Continuamente están recibiendo una retroalimentación por parte del equipo, esto es una respuesta.
 2. Lo que tú puedes ver es lo que puedes conseguir (WYSIWYG- *What you see is what you get*). No debe de haber secretos para el usuario, ni comandos abstractos que prometan resultados futuros, así como significantes diferencias entre lo que ellos ven en la pantalla y lo que pueden conseguir en la impresora.
 3. Interfaces de usuario basadas en iconos. Un icono es una representación gráfica de un objeto, una acción, una propiedad o algún otro concepto. Por ejemplo, un clic en un icono que represente la verificación de ortografía en un procesador de texto deberá necesariamente, realizar esta acción.
- El usuario esta familiarizado con los objetos utilizados con la interfaz. Esto es relacionado con el uso de metáforas dentro de la interfaz para darle mayor facilidad de uso a la misma. La metáfora es utilizada para presentar nueva información haciendo comparaciones y similitudes con información que es muy familiar. Por ejemplo, si en la interfaz es necesario poner una calculadora para realizar algunas operaciones, esta deberá tener la forma en que normalmente tienen todas las calculadoras que utilizamos en la vida diaria. Por lo tanto, si presentamos objetos del mundo real (por ejemplo, metáforas del mundo real), los usuarios, de una forma natural, sabrán qué hacer con ellos.

En base a todo lo anterior, la definición de interfaz gráfica hombre-computadora sería:

Es la parte de la aplicación gráfica por computadora encargada de establecer las bases comunes de funcionamiento del sistema con un usuario en particular. La tarea es ir actualizando y manteniendo dichas a través del proceso de interacción con la aplicación. Siempre es posible utilizar los tres principales estilos de interacción (Manipulación directa, WYSIWYG, Interfaces de usuario basados en iconos) mezclados y utilizando metáforas del mundo real.

7.4 Herramientas para la creación de interfaces gráficas

7.4.1 VDK y VDKBuilder

VDK fue diseñado para ser un armazón robusto y confiable en C++, basado en la biblioteca de componentes Gtk+ y no únicamente una simple envoltura de estos componentes. Esto es debido a que VDK oculta tanto como sea posible en lugar de “envolver” o sólo disfrazar las llamadas a funciones Gtk+. Al utilizar VDK uno descubre que muchas de las declaraciones uno a uno con Gtk+ NO son realmente cubiertas y es necesario que el usuario modifique el componente directamente desde el código. Esto no sucede con VDK debido a que es posible usar VDK sin conocer Gtk+, así mismo, los usuarios mas experimentados sacan ventaja de conocer las llamadas y convenciones de Gtk. Con respecto a este punto, VDK proporciona un método para acceder y utilizar los componentes Gtk+ que se encuentran debajo.

VDK es usado como biblioteca base de la herramienta RAD VDKBuilder, usar VDK con VDKbuilder es altamente recomendable debido a que de ese modo el usuario puede concentrar su atención en la aplicación en lugar de perder tiempo en la construcción de la interfaz. También en este caso un buen conocimiento de VDK es una buena manera de sacar ventaja de usar VDKBuilder.

VDKBuilder es una herramienta de desarrollo rápido (RAD) basada en VDK, que ayuda al programador en la construcción de interfaces graficas (GUI), edición, compilación, enlazado, y depuración dentro de un ambiente integrado.

VDk consiste de 2 módulos separados:

La biblioteca núcleo VDK (*VDK core Library*) que contiene el *kernel* y los componentes envueltos Gtk+. Los componentes VDK (*VDK components*) que contienen componentes totalmente nuevos y aquellos que aun no son lo suficientemente estables para ser incluidos en la biblioteca núcleo.

Para la implementación del sistema de visualización se eligió el conjunto de componentes VDK por su facilidad de uso al programar en C++ y VDKBuilder como herramienta de desarrollo rápido de aplicaciones para elaborar la interfaz gráfica para el sistema de una manera amigable y dinámica. Es importante aclarar que en el momento de elegir VDKBuilder no existe otra herramienta que facilite la creación de interfaces gráficas con las ventajas que esta proporciona.

7.4.2 PerlTk

Perl/ Tk (también conocido como pTk o ptk), es una extensión del lenguaje de programación PERL, es una colección de módulos y código que intenta conjuntar el fácilmente configurado conjunto de componentes (*widgets*) Tk 4 a las poderosas capacidades lexicográficas, de memoria dinámica, de entrada y salida y orientadas a objetos de Perl 5. En otras palabras es un lenguaje interpretado de guiones (“scripts”) para hacer componentes y programas con interfaz gráfica de usuario.

La versión actual de Perl/Tk esta basada en "Tk 4.0p3", el conjunto de componentes originalmente asociado con el lenguaje Tcl. Sin embargo, Perl/Tk no requiere de ninguna de las características de léxico ni de idiosincrasia de Tcl. Perl/Tk usa la sintaxis, la gramática, y las estructuras de datos de Perl 5.

El código propio de pTk es conjunto de herramientas que puede ser llamado externamente. Ptk puede ser llamado desde Perl 5 vía Tk.pm y todos los módulos “encolados” de Perl (*modules glue*). De aquí, ptk no necesariamente se refiere a Perl Tk, pero puede ser tomado como un Tk portable – dando un paquete encolado para otro lenguaje. El propósito declarado del código Ptk es tener una biblioteca utilizable desde perl, Tcl, LISP, C++, python, etc.

El lenguaje Perl/Tk es posteriormente extensible asimismo a través del mecanismo estándar de módulos de Perl 5. Un número de componentes compuestos y extensiones especiales al lenguaje han sido escritos usando módulos de Perl.

Además, no es necesario tener instalado Tcl/Tk antes de instalar Perl/Tk, ya que este último es completamente independiente de Tcl/Tk.

7.4.3 Xforms

Xforms, es una de las bibliotecas gráficas disponible para sistemas UNIX, orientada a la implementación de interfaces de usuario gráficos (*GUI's*) y distribuida libremente para su uso, por lo menos para su uso no comercial.

Se trata de una biblioteca disponible prácticamente para la totalidad de sistemas UNIX actuales basada en Xlib, el API de X-Window a bajo nivel. Además se distribuye con la biblioteca una aplicación gráfica denominada *fdesign* (*Form Designer*) la cual permite desarrollar interactivamente las interfaces de usuario de una forma visual e interactiva. Esta aplicación es capaz de generar código C una vez terminado el diseño del interfaz que podemos utilizar en nuestra aplicación. Esta característica acelera enormemente el proceso de creación de aplicaciones con interfaz gráfico de usuario, ya que de otra forma es necesario escribir código en lenguaje C, y no tenemos la posibilidad de observar los resultados de nuestro trabajo hasta que este se haya terminado, y si alguna característica de nuestra interfaz no nos agrada, es necesario modificar el código y volver a observarla al finalizar.

El código fuente de la biblioteca no se distribuye. Únicamente son accesibles binarios compilados en forma de biblioteca estática o dinámica, así como el binario *fdesign*, la aplicación para desarrollo visual de interfaces gráficos de usuario.

Las claves del diseño de XForms, según uno de sus autores T. C. Zhao, fueron crear un paquete que fuese intuitivo, y simple de usar, potente, con un buen estilo gráfico de los objetos y fácilmente ampliable.

El principal concepto de XForms (también conocida como *Forms Library*) es la forma "Form". Una forma es una ventana en la cual pueden colocarse diferentes objetos. estas ventanas pueden mostrarse en la pantalla y el usuario puede interactuar con los diferentes objetos de la ventana para indicar lo que desea realizar en cada momento. En cuanto a objetos, existen diferentes clases por ejemplo, botones (con diferentes características), barras deslizantes con los que el usuario puede especificar un valor dentro de un rango, campos de entrada en los que el usuario puede escribir, menús en los que el usuario puede realizar elecciones, etc. Cuando el usuario cambia el estado de un objeto en particular, la aplicación propietaria recibe una señal para que pueda actuar ante esa situación. La mejor manera de diseñar la aplicación implica eventos y llamadas internas que realiza la interfaz para notificar algún cambio "callbacks".

La aplicación que este siendo diseñada dispone de control sobre los objetos de una ventana. Es posible elegir el color, fuente de letra, tamaño, color del texto, forma externa, etc.

XForms está formada por un gran número de rutinas en lenguaje C que permiten construir los interfaces de forma sencilla y con muy pocas líneas de código. Las rutinas pueden ser usadas desde programas en lenguaje C y C++. La biblioteca usa únicamente los servicios de Xlib, por lo que funciona en cualquier estación de trabajo u ordenador con el sistema X instalado. La versión actual de XForms necesita 4 bits de color en el sistema gráfico o un sistema de escala de grises con esa profundidad (es decir, un sistema capaz de mostrar 16 tonos distintos aunque sea dentro de un solo color) para disponer de una apariencia agradable. Sin embargo XForms funciona correctamente sobre terminales X en blanco y negro, aunque la apariencia de las ventanas pierde mucha vistosidad.

La biblioteca es fácil de usar. Es posible definir una ventana con objetos en unas pocas líneas de código. Existen rutinas que aportan ventanas completas cuya utilización es muy común. Como ejemplo se dispone de una rutina que permite con una sola línea de código mostrar en pantalla una ventana que permite al usuario seleccionar un archivo del disco duro navegando a través de su sistema de archivos y así mismo devuelve a la aplicación el nombre completo de dicho archivo, el cual puede ser utilizado posteriormente por el programa.

Para hacer más fácil el diseño de las interfaces el sistema lleva incluido un programa denominado *Form Designer*, que permite diseñarlas visualmente y genera el correspondiente código C para incluir en la aplicación.

XForms oculta completamente las ventanas X. Sin embargo en algunos casos puede ser conveniente conocer este sistema gráfico a bajo nivel. Esta disponible un objeto que es una simple ventana en la que se puede realizar cualquier acción X de bajo nivel.

Según la documentación aportada por T. C. Zhao, XForms ha sido probado satisfactoriamente bajo X11 R4, R5 y R6 en una gran cantidad de plataformas Unix: SGI, SUN, HP, IBM RS6000/AIX, Dec Alpha/SF1, así como Linux/i386, FreeBSD/i386, NetBSD/i386, SCO y Unixware.

7.4.4 Gtoolkit, GTK y GTK+

GTK es una biblioteca que se utiliza para crear interfaces gráficas de usuario, similares a la apariencia y características de Motif. Esta diseñada para ser pequeña y eficiente, pero aún así es lo suficientemente flexible para permitir al programador libertad en las interfaces creadas. GTK permite al programador usar una variedad de componentes denominados “widgets”, tales como botones simples (*push buttons*), botones de opciones (*radio buttons*) y botones de encendido y apagado (*check buttons*), así como menús, listas y componentes base llamados “frames”. Este también provee varios componentes contenedores que pueden ser usados para controlar la disposición de los elementos de la interfaz de usuario, así como componentes comunes y algunos más complejos tales como un selector de archivos y un selector de color.

GTK provee algunas características únicas. Por ejemplo, un botón de etiqueta no contiene una etiqueta, contiene un componente hijo, el cual en la mayoría de las instancias será una etiqueta. Sin embargo, el componente hijo puede ser también un mapa de bits, una imagen, o una combinación posible de los deseos del programador. Esta flexibilidad es adicionada a toda la biblioteca completa.

Para hacerlo más fácil, GTK presenta esta flexibilidad en un panorama uniforme. Específicamente, implementa su propio soporte para programación orientada a objetos que es también adaptada a los propósitos de un conjunto de herramientas de interfaz de usuario e intenta proveer una interfaz de programación razonable y disciplinada. Esta uniformidad y disciplina es proyectada para hacer fácil y confiable acceder a GTK de otros lenguajes que no son C. Especialmente lenguajes más dinámicos como *Perl*, *Python* o *Scheme* encontrarán soporte holgado, y de hecho, enlaces para estos lenguajes ya existen.

Es importante mencionar que el proyecto GNOME de distribución de Linux, esta utilizando GTK+ para construir un escritorio libre para Linux

La diferencia entre GTK y GTK+ consiste en que originalmente GTK fue escrito por Peter Mattis, incluyendo las tres bibliotecas *libglib*, *libgdk* and *libgtk*, esto presentó una jerarquía limitada de componentes, esto es, no es posible derivar un nuevo componente de un componente existente. Y este contenía mas mecanismos de llamada en lugar de los mecanismos de señal ahora presentes en GTK+. El símbolo + fue adicionado para distinguir entre la versión original de GTK y la nueva versión. Esto se puede pensar como una especie de perfeccionamiento al original GTK que adiciona características orientadas a objetos.

7.4.5 Lesstif

Se trata de un clon API compatible de Motif, es un reemplazo libre de OSF/Motif, esto quiere decir que el mismo código debe compilarse con ambos y trabajar igual. Como con Motif, Lesstif consiste de un conjunto de componentes, un manejador de ventanas, y una implementación UIL (*libMrm* y el compilador *UIL*). Aunque, el área de principal atención es el conjunto de componentes.

Lesstif tiene licencia bajo LGPL, *Library Gnu Public License*, biblioteca de licencia GNU publica. La licencia por si misma acompaña a lesstif en el archivo *COPYING.LIB*.

7.4.6 Tcl/tk

Tcl/tk es un lenguaje de guiones “scripts” interpretado, creado por la compañía Sun para su sistema operativo Solaris (para ser usado como lenguaje de *scripts* nativo del sistema). Está diseñado para combinarse fácilmente con cualquier aplicación. Tcl se entiende por “Tool Command Language”, lenguaje de Herramientas de Comandos. Tcl es más conveniente para el control de flujo de datos entre pequeños programas, muy parecidos a los guiones escritos en PERL. Tk se entiende por “Toolkit”. Tk es la porción gráfica de Tcl/tk y proporciona la funcionalidad de agregar una interfaz gráfica de usuario (GUI) al programa. Tk es la interfaz de usuario para Tcl.

Tanto Tcl como Tk son *software* libre que se distribuye con ficheros fuente y ejecutables sin ningún cargo. Son completamente gratis, y se pueden distribuir a terceros sin tener que pagar cuotas ni regalías de derechos de autor. Existen versiones de Tcl para UNIX, DOS, Windows, Windows NT, OS/2 y Macintosh. De Tk hay versiones para X-Windows (UNIX), Windows y Macintosh.

Siendo un lenguaje de guiones o procesos por lotes “scripting”, es similar a otros lenguajes de ese tipo característicos de UNIX, tales como *Bourne shell* (sh) o *Korn shell* (ksh) o Perl. Todos ellos pueden funcionar como la capa externa de una aplicación, facilitando en gran medida la automatización y el control de procesos y opciones que sin ellos serían tediosos.

Sin embargo, Tcl/Tk es la combinación precisa que hace que usar este lenguaje sea sencillo y permita hacerlo interactuar con una aplicación de manera trivial. Tcl permite ejecutar aplicaciones, definir nuevos comandos Tcl y crear extensiones para este lenguaje. O sea, que si necesita idear un lenguaje para una aplicación, o desea implementar una interfaz de usuario sin el tedioso trabajo que ello resulta ya no necesita pensar más, Tcl/Tk es una buena elección.

Tcl/Tk es ideal para combinarse con cualquier aplicación porque:

- * Proporciona acceso a las bibliotecas de C más utilizadas.
- * Con él se pueden manejar todo tipo de archivos fácilmente.
- * Permite manipular cadenas con todo tipo de funciones y operadores y realizar operaciones matemáticas.
- * Usa listas y arreglos de forma intuitiva.
- * Es posible crear ventanas y llenarlas de todo tipo de controles.
- * Permite la creación de nuevos comandos Tcl, definidos desde una aplicación.

La interfaz de usuario ahora queda totalmente aislada; así el programador se puede concentrar en el programa y luego añadir de manera fácil la interfaz de usuario. Además se pueden crear controles de usuario a medida muy fácilmente, ya que las bibliotecas en C de Tcl/Tk son completamente transparentes.

Además, ya no tiene que preocuparse con los nuevos entornos de Internet, existe un *plug-in* que permite ejecutar aplicaciones tcl/tk dentro de los navegadores más populares.

Tcl/Tk incluye dos intérpretes, uno para Tcl (tclsh) y otro para Tcl/Tk (wish) que permiten interpretar cualquier archivo Tcl/Tk o bien el funcionamiento interactivo.

7.4.7 Otras

Además de las herramientas anteriores, existen otras igual de útiles, algunas basadas en X-window, que es el estándar de ventanas para sistemas UNIX, y otras específicas de cada sistema. Entre estas otras herramientas se encuentran:

7.4.7.1 motif

En principio Motif es un conjunto de normas propuesto por OSF que especifican un modo de presentación y comportamiento de los interfaces de usuario gráficos. El modelo elegido está basado en MS-Windows y *Presentation Manager*.

Un programador podría construir una aplicación que cumpla las “guías de estilo” de Motif pero, normalmente, usará una biblioteca de objetos de diálogo que cumplan con los requisitos de la especificación, como el conjunto de componentes “widgets” definido por OSF.

Además de la guía de estilo y la biblioteca de componentes, en Motif también se incluyen un gestor de ventanas (mwm) y un lenguaje de especificación de interfaces de usuario (UIL).

En general, mwm permitirá ejecutar cualquier aplicación realizada con un API Xwindow, y de igual manera, una aplicación creada usando el conjunto de herramientas Motif será compatible en cualquier manejador de ventanas. Esto debido a que Motif es el estándar de sistemas de ventanas preestablecido para todos los sistemas de Xwindow.

7.4.7.2 xaw3d

Es un conjunto de componentes 3-D *Athena* (*Athena Widget set*) que es parecido a Motif. Xaw3d es un Xaw (*Athena Widget Set*, Conjunto de Componentes *Athena*) que reemplazan a los que parecen de tres dimensiones.

7.4.7.3 blt

Una extensión Tk (con bibliotecas compartidas). Esta es versión 2.4f de la biblioteca BLT. BLT es una extensión al conjunto de herramientas Tk *toolkit*, adicionando nuevos componentes, manejadores de geometría, y comandos diversos. No requiere ningún parche o actualización de Tcl o Tk, aunque hay que notar que necesita como un pre-requisito los binarios de Tcl y Tk.

7.4.7.4 fwf

La Fundación “Free Widget Foundation” (FWF) fue la primera organizada en el verano de 1990, como un esfuerzo voluntario para generar una fuente públicamente accesible de componentes X. El objetivo de FWF fue recoger un conjunto de componentes poderosos, flexibles, de libre acceso, y asistir a los programadores a construir interfaces de usuario complicadas.

7.4.7.5 gtk--

Cubierta C++ para gtk, una biblioteca gráfica x11.

7.4.7.6 gtk

General Toolkit para X11 GUI.

7.4.7.7 gtkstep

Un módulo para hacer que GTK+ parezca como la interfaz NeXTSTEP(tm).

7.4.7.8 iv

InterViews: Un *toolkit* de la Universidad de Stanford y Silicon Graphics.

7.4.7.9 p5-Gtk

Una interfaz perl5 para bibliotecas Gráfica Gtk.

7.4.7.10 p5-Tcl-Tk

Un módulo de acceso a Tk vía la extensión de Tcl.

En general, la mayoría de las herramientas de desarrollo para interfaces gráficas, están basadas en X-Window, cuando se trata de plataformas UNIX, o en ambientes basados en adaptaciones similares a herramientas como Motif, cuando es el caso de otros sistemas, como Windows, Linux, etc.

CAPITULO IV

ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS

8. CAPÍTULO IV ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS

8.1 Conceptos básicos del análisis orientado a objetos

La programación orientada a objetos, se puede definir como una técnica o estilo de programación que utiliza objetos como bloque esencial de construcción.

Los objetos son en realidad como los tipos abstractos de datos, ampliamente utilizados por los programadores en la década de los setenta y ochenta. Un tipo abstracto de datos es un tipo de datos definido por el programador junto con un conjunto de operaciones que se pueden realizar sobre ellos. Se denominan abstractos para diferenciarlos de los tipos de datos fundamentales o básicos definidos por el lenguaje, por ejemplo en lenguaje C, algunos de los tipos de datos estándar existentes son entero (*int*), carácter (*char*), y doble (*double*).

Al igual que los tipos de datos definidos por el usuario, un objeto es una colección de datos, junto con las funciones asociadas, utilizadas para operar sobre estos. Sin embargo el alcance real de los objetos reside en las propiedades que poseen: herencia, encapsulamiento y polimorfismo junto con los conceptos de objetos, clases, métodos y mensajes.

8.1.1 Definición de objetos

Los objetos son cualquier entidad del mundo real que sea posible imaginar.

Un objeto, en programación orientada a objetos (POO), es una unidad que contiene datos y las funciones que operan sobre estos datos. A los elementos de un objeto se les conoce como miembros; las funciones que operan sobre los objetos se denominan métodos y los datos se denominan datos miembros. En C++ un programa consta de objetos. Los objetos de un programa se comunican entre sí, mediante el paso o envío de mensajes (acciones que debe ejecutar el objeto), así como de datos si es necesario.

Los datos y las funciones se encapsulan en una única entidad. Los datos están ocultos y sólo mediante las funciones miembro es posible acceder a ellos. Los conceptos de encapsulamiento y ocultamiento de datos son términos claves empleados en los lenguajes de programación orientados a objetos.

En un objeto, hay campos y métodos que están completamente en el interior del objeto y que son ocultos para el exterior, lo que significa que estas características están encapsuladas. Por otro lado hay campos o funciones que se extienden fuera del objeto, y son accesibles desde el exterior y actúan de interfaz. El acceso a las características miembro (campos y métodos) es posible a través de la interfaz del objeto.

Un programa, generalmente, consta de un número de objetos que se comunican entre sí mediante métodos que son a su vez, llamados funciones miembro del objeto invocado. En C++, las funciones se denominan funciones miembro y en otros lenguajes orientados a objetos, son llamadas métodos.

8.1.2 Definición de clases

En POO, se suele decir que los objetos son miembros de clases. Una clase es un tipo definido por el usuario que determina las estructuras y las operaciones asociadas con ese tipo. Las clases son como plantillas o modelos que describen cómo se construyen ciertos tipos de objetos. Cada vez que se construye un objeto de una clase, se crea una instancia de esa clase. Por consiguiente, los objetos son instancias de clases. En general, los términos objetos e instancias de una clase se pueden utilizar indistintamente. Una clase es una colección de objetos similares y un objeto es una instancia de una definición de una clase. Una clase puede tener muchas instancias y cada una es un objeto independiente.

Una clase es simplemente un modelo que se utiliza para describir uno o más objetos del mismo tipo.

La comunicación del objeto se realiza a través del paso de mensajes. El envío de un mensaje a una instancia de una clase produce la ejecución de un método. El paso de mensajes es el término empleado para referirse a la invocación o llamada de una función miembro de un objeto. La noción del paso de mensajes es fundamental en todos los lenguajes de programación orientada a objetos.

8.1.3 Mensajes: Activación de objetos

A diferencia de los métodos tradicionales, donde los datos son elementos pasivos, los objetos pueden ser activados mediante la recepción de mensajes. Un mensaje es simplemente una petición de un objeto a otro objeto para que éste se comporte de una manera determinada, ejecutando uno de sus métodos. La técnica de enviar mensajes se conoce como paso de mensajes.

Estructuralmente, un mensaje consta de tres partes: La identidad del objeto receptor, el método (función miembro) del receptor cuya ejecución se ha solicitado y cualquier otra información adicional que el receptor pueda necesitar para ejecutar el método requerido. Esta información suele darse en forma de parámetros.

Los mensajes juegan un papel vital en POO; sin ellos, los objetos que se definan no se podrían comunicar con otros objetos. Desde un punto de vista convencional, el paso de mensajes no es más que el sinónimo de llamada a una función. De hecho, esto es lo que sucede cuando un mensaje se envía a un programa en C++.

8.1.4 Programa orientado a objetos

Un programa orientado a objetos es una colección de clases. Generalmente, necesitará una función principal que cree objetos y comience la ejecución mediante la invocación de sus funciones miembro o métodos.

Esta organización conduce a separar las diferentes partes de una aplicación en distintos archivos. La idea consiste en poner la descripción de la clase para cada una de ellas en un archivo independiente. El compilador ensamblará, entonces, el programa completo a partir de los archivos independientes en una única unidad. Naturalmente, las clases que están en archivos independientes se pueden utilizar en más de una aplicación.

En realidad, cuando se ejecuta un programa orientado a objetos, ocurren tres acciones o sucesos. En primer lugar, se crean los objetos cuando se necesitan. Segundo, los mensajes se envían desde unos objetos y se reciben en otros, a medida que el programa procesa internamente información o responde a la entrada del usuario. Por último, se borran los objetos cuando ya no son necesarios y se recupera la memoria ocupada por ellos.

8.1.5 Herencia

Una característica muy importante de los objetos y las clases es la herencia. Esta es la propiedad que permite a los objetos construirse a partir de las características de otros objetos. El concepto de herencia está presente en nuestras vidas diarias, donde las clases se dividen en subclases. Así por ejemplo, las clases de animales se dividen en mamíferos, anfibios, insectos, aves, etc. La clase de vehículos se divide en automóviles (coches o carros), autobuses, camiones y motocicletas.

El principio de este tipo de división es que cada subclase comparte características comunes con la clase de la cual deriva. Aunque además de las características compartidas con otros miembros de la clase, cada subclase tiene las suyas propias.

La idea de herencia consiste generalmente en lo siguiente: las características que pertenecen a la clase base, también son comunes a todas las clases derivadas, y a su vez estas clases derivadas tienen sus propias características.

En C++ la clase original se denomina clase base; las clases que se definen a partir de la clase base, compartiendo sus características y añadiendo otras nuevas, se denominan clases derivadas.

Las clases derivadas pueden heredar código y datos de su clase base, añadiendo su propio código especial y nuevos datos a la misma.

La herencia permite definir nuevas clases a partir de clases ya existentes. Impone una relación jerárquica entre clases en la cual una clase hija hereda de su clase padre. Cuando una clase sólo puede recibir las características de otra clase base, la herencia se denomina herencia simple.

Si una clase recibe propiedades de más de una clase base, la herencia se denomina herencia múltiple. Muchos lenguajes orientados a objetos no soportan herencia múltiple, sin embargo, C++ sí incorpora esta propiedad.

8.1.6 Polimorfismo

Polimorfismo significa la cualidad de tener más de una forma. En el contexto de POO, el polimorfismo se refiere al hecho de que una misma operación puede tener diferente comportamiento en diferentes objetos, es decir, diferentes objetos reaccionan al mismo mensaje de modo diferente. Como ejemplo, se puede considerar la operación de sumar: ya que no es lo mismo realizar la operación con dos o más números, que sumar (concatenar) dos datos de tipo cadena, debido a que es necesaria una implementación diferente para cada operación.

De modo similar, suponiendo un número determinado de figuras geométricas que responden todas al mensaje, dibujar. Cada objeto reacciona a este mensaje al desplegar su figura en una pantalla de visualización. Obviamente, el mecanismo real para visualizar los objetos difiere de una figura a otra, pero todas las figuras realizan esta tarea en respuesta al mismo mensaje.

8.1.7 Reutilización

Una vez que una clase se ha escrito, creado, depurado y utilizado, se puede difundir entre otros programadores para que puedan utilizarla en sus propios programas. Esta propiedad se denomina de reutilización o reutilizabilidad. Es similar a la forma en que una biblioteca de funciones de un lenguaje procedimental se puede incorporar en programas diferentes.

En POO, el concepto de herencia proporciona una importante extensión a la idea de la reutilizabilidad. Un programador puede tomar una clase existente, y añadirle nuevas características y posibilidades adicionales. Estas operaciones se realizan por derivación de una clase nueva a partir de una clase existente.

Los objetos en POO son fácilmente reutilizables y es una de las razones principales para justificar la utilización de la metodología orientada a objetos en la mayoría de los casos. Por esta razón, cualquier programa en POO suele estar formado de un conjunto de clases predefinidas, que permitirá ahorrar tiempo en el desarrollo de las aplicaciones más complejas.

8.2 Metodología

Una metodología es un conjunto de pasos a seguir para llegar a un resultado, resolver un problema o encontrar una solución.

Existen diversas maneras de implementar los conceptos y propiedades de la programación orientada a objetos anteriormente mencionados, al desarrollar un sistema. A cada una de estas se les denomina metodología orientada a objetos.

Generalmente todas las metodologías orientadas a objetos logran alcanzar el objetivo, el cual es realizar un sistema que posea las características ya mencionadas.

8.2.1 Comparación entre las diferentes metodologías orientadas a objetos

En general, las metodologías difieren entre sí en el número de pasos que hay que seguir. Sin embargo, básicamente abarcan los mismos aspectos y todas o la mayoría tienen como punto en común las fases de análisis, y diseño, así como la parte de análisis y determinación de requerimientos.

Entre las metodologías orientadas a objetos están:

- Metodología OMT (*Object Modeling Technique*), Técnica de Modelado de Objetos:

Esta metodología se divide en cuatro fases:

1. Fase de análisis de objetos.
2. Fase de diseño del sistema.
3. Fase de diseño de objetos.
4. Fase de implementación.

- Metodología de Booch.

El método de Booch es un método orientado a objetos ampliamente usado, que ayuda al diseño de sistemas utilizando el paradigma de objetos.

El método de Booch cubre las fases de análisis y diseño de un sistema orientado a objetos. Booch define un conjunto de símbolos para documentar casi todas las decisiones de diseño. Se inicia con clases y diagramas de objetos en la fase del análisis y se va refinando estos durante los siguientes pasos. Únicamente cuando se va a generar código, se adhieren algunos símbolos, y esto realmente es útil para documentar el código orientado a objetos.

- Método de fusión.

Este provee una aproximación sistemática (método) para el desarrollo de software orientado a objetos, además de un proceso de desarrollo, y se divide en las siguientes fases:

1. Análisis.
2. Diseño.
3. Implementación.

Proporciona una guía sobre el orden en el cual las cosas deben ser realizadas dentro de cada fase, así como la notación para los modelos que describen varios aspectos de software y el manejo de herramientas a través de la definición de modelos cuya producción puede ser manejada.

- Metodología *Overview of the object-oriented information engineering* (Martin/Odell): Metodología de la ingeniería de la información orientada a objetos.
- Metodología EROOS: *Entity-Relationship Object-Oriented Specifications* (Katholieke Universiteit Leuven). Especificaciones de entidad relación orientadas a objetos.
- *RealTime Object-Oriented Modeling* (ROOM). Modelado orientado a objetos de tiempo real.
- B.O.N. : *Business Object Notation*.
- UML. Lenguaje unificado de modelado.
- Método de Yourdon. *The Object-Oriented Analysis and Design (OOA/OOD) method by Coad & Yourdon*

Estas son sólo algunas de las metodologías y conjuntos de convenciones creadas para la realización de sistemas orientados a objetos.

Para la realización de la mayoría de los sistemas, a veces es mejor hacer la creación de un supermétodo, es decir, crear una metodología propia a partir de los puntos que establecen las diferentes metodologías, para esto es necesario realizar lo siguiente:

Se debe formar una lista inicial que contenga todas las actividades de más bajo nivel de los métodos de modelado de procesos, para luego agrupar estas actividades y así crear una más pequeña como común denominador.

En este punto sólo miraremos a las actividades de más bajo nivel, las cuales son las más importantes y tienen descripción detallada, de una manera de "abajo hacia arriba" se crearán las actividades de más alto nivel para el supermétodo que deseamos crear.

Luego de que este pequeño común denominador es revisado, y las actividades son agregadas y eliminadas, obtenemos como resultado un llamado supermétodo que incluye todas las actividades de los diferentes métodos que son importantes para el análisis y diseño orientado a objetos de acuerdo a nuestra visión particular.

Para la comparación de actividades es importante considerar que hay una diferencia entre los conceptos de los métodos proveídos, y el soporte para este concepto es una actividad común en los métodos. Algunos métodos pueden tener conceptos los cuales no son soportados por otras actividades.

Los conceptos que debe tomar en cuenta una buena metodología orientada a objetos, para el desarrollo de un sistema, son los siguientes:

1. Clases.
2. Objetos.
3. Atributos de los objetos.
4. Operaciones (funciones miembro).
5. Subsistemas.
6. Coacción sobre asociación.
7. Mensajes.
8. Estados.
9. Transición.
10. Eventos.
11. Actores.
12. Casos de uso.
13. Almacenes de datos.
14. Procesos.
15. Módulos.

Las actividades a considerar:

1. Análisis.
2. Diseño.
3. Implementación.
4. Pruebas.

En el caso del presente trabajo, nosotros nos basamos principalmente en la metodología OMT.

8.2.2 Técnica de Modelado de Objetos (*Object Modeling Technique OMT*)

OMT es una metodología para desarrollo de sistemas orientados a objetos con una notación especial . El modelo OMT está formado por los siguientes tres componentes que describen de forma completa un sistema:

- a) Modelo de objetos
- b) Modelo dinámico
- c) Modelo funcional

En el modelo de objetos se describen las estructuras estáticas (clases y objetos), la relación entre las estructuras estáticas, se representa el modelo de objetos mediante diagramas de clases. En los diagramas de clases los nodos corresponden a las clases y los arcos a las relaciones entre las clases y los diagramas de clases se acompañan de diagramas de objetos. Este modelo describe los componentes del sistema.

En el modelo dinámico se describen los aspectos de control (eventos externos del sistema), aspectos que varían con el tiempo (secuencias de operaciones), se describen las interacciones entre los objetos y los estados de los objetos. La representación del modelo dinámico se hace por medio de diagramas de estado, los nodos corresponden a los estados de los objetos y los arcos corresponden a las transiciones entre estados. Este modelo describe en que momento o cuando suceden los eventos en el sistema.

En el modelo funcional se describen las dependencias de datos en el sistema así como sus transformaciones. La representación se hace por medio de diagramas de flujo de datos, en donde los nodos corresponden a los procesos de los objetos y los arcos a los flujos de datos entre los objetos. Este modelo describe la manera como se llevan a cabo los procesos y operaciones en el sistema.

La metodología OMT consiste en tres etapas de desarrollo

- 1) Análisis.
- 2) Diseño.
- 3) Implementación.

Durante el análisis se muestra los aspectos más importantes del sistema sin importar su implementación final. De esta manera el análisis se utiliza para comunicarse con los expertos en el área, sin necesidad de tener un conocimientos en computación.

En el análisis se refleja lo “que” el sistema va a realizar pero no “como” . Durante esta etapa se trabaja para obtener una descripción del problema la cual puede originalmente ser incompleta.

La etapa de diseño se compone de dos fases:

- Diseño de sistemas y
- Diseño de objetos.

Durante el diseño de sistemas se toman decisiones de alto nivel sobre la arquitectura del sistema a desarrollar. Se toman decisiones sobre el *hardware* y el *software*. Se divide el sistema en subsistemas, se identifica la concurrencia, se escoge el manejo de almacenamiento de datos y se deciden las prioridades durante el diseño como : rendimiento, memoria, protocolos de comunicación, etc.

En el diseño de objetos se añaden detalles que corresponden a la implementación final como descripción final de algoritmos. También se combinan los tres modelos para determinar a detalle las operaciones en cada clase, se diseñan algoritmos para implementar los métodos, se añaden clases, se ajusta la herencia y se hacen optimizaciones al sistema. Se empacan las clases y las operaciones en módulos.

Durante la implementación se traduce el diseño a un lenguaje de programación específico, de acuerdo a los requerimientos de cada aplicación como bases de datos o *hardware* especial.

CAPITULO V

**ETAPAS DE DESARROLLO DEL
SISTEMA DE VISUALIZACIÓN DE DINÁMICA
MOLECULAR**

9. CAPÍTULO V ETAPAS DE DESARROLLO DEL SISTEMA DE VISUALIZACIÓN DE DINÁMICA MOLECULAR

9.1. Análisis orientado a objetos

9.1.1. Modelo de objetos

9.1.1.1. Planteamiento:

Se requiere diseñar un sistema de visualización de dinámica molecular para realizar simulaciones de procesos a nivel molecular. Se pretende que, en versiones posteriores, dicho sistema permita un manejo aun más versátil del programa de Dinámica Molecular, desde la elaboración de archivos de entrada del programa (proporcionando datos a través de ventanas), en la ejecución del mismo y para manejar la gran variedad de posibilidades involucradas en la visualización e interpretación de los cálculos realizados.

La simulación se realiza en la supercomputadora Cray, pero los resultados de los cálculos, es decir, los datos numéricos, difíciles de interpretar a simple vista, se procesan y analizan en una estación de trabajo, por lo cual es necesario tener una interfaz que nos permita observar más directamente y entender mejor lo que sucede a escala molecular, a través de una representación gráfica, que haga posible visualizar los cambios y el comportamiento de una molécula en determinadas condiciones.

De esta forma, el sistema recibe los datos, leyendo el o los archivos de datos de entrada, los interpreta y visualiza como salida a pantalla de manera que se puedan observar situaciones difíciles o inclusive imposibles de observar, en condiciones reales y de este modo se puedan estudiar y comprender mejor. A su vez, el sistema entregará resultados proporcionando uno o más archivos de datos de salida.

Aquí llamamos "molécula" para abreviar, refiriéndonos a lo que en realidad consiste de un agregado de átomos o estructura molecular, con sus conexiones y dipolos.

El sistema efectuará la representación de las moléculas, dibujando esferas de tamaños y colores adecuados que representan los átomos con sus cargas relativas y cilindros que simulan las conexiones o enlaces entre ellas. Además de esto, dentro de una molécula existen partes en las cuales hay ausencia de partículas y conexiones, estos sitios son denominados cavidades u hoyos (los cuales serán también representados por esferas translúcidas, luego de leer un archivo de configuración que será generado por una subrutina capaz de buscar estas porosidades en la estructura).

Todos los elementos anteriormente mencionados se situarán dentro de una celda computacional, esta consiste en una especie de caja que delimitará el espacio de la molécula inicial. Dicha celda computacional será utilizada posteriormente, también para multiplicar el sistema de trabajo, esto quiere decir, hacer replicas de la molécula original y situarlas junto a esta, de manera uniforme, generando una estructura que denominaremos estructura cristalina.

Los componentes de la molécula, se encuentran delimitados, por la caja computacional que es el elemento que será utilizado para contener una configuración básica de una estructura

Además de la caja, también hay un elemento muy importante, que son los ejes, que nos permitirán una mejor ubicación visual de los elementos de la molécula en el espacio.

Dentro de la parte de pintar conexiones, o mejor dicho antes de que se dibujen, deberá haber una manera de validar conexiones entre elementos, es decir, se debe de verificar que las conexiones sean válidas según se defina en una tabla de conectividad, la cual servirá para indicar los elementos con los cuales es posible que una determinada partícula pueda tener enlaces.

El sistema será capaz de representar también, una animación por cuadros de las moléculas, dicha animación es generada con la evolución de las posiciones de las partículas dentro de la estructura en estudio al transcurrir el tiempo de la simulación. Como parte de esta característica, el sistema será capaz de manipular la animación, es decir, permitirá detenerla en algún cuadro que el usuario requiera e invertir el sentido de la misma.

Es importante mencionar, que para la representación de los elementos de la molécula, es necesario tener en cuenta las luces que van a dar luminosidad a estos, así como los materiales usados para la representación adecuada de su consistencia, esto es la calidad que deben tener los materiales que van a usarse para representar cada partícula y elemento dentro de la molécula. Esto se hará al leer el archivo de configuración y antes de dibujar, asignando el material correcto a un elemento específico.

El sistema deberá de realizar la manipulación de los elementos representados, a través de operaciones como son: rotaciones, traslaciones y escalamientos de la molécula.

Será posible también el seleccionar el tipo de caja si se requiriera ver una molécula específica con una simetría diferente a la cúbica convencional (romboidal, hexagonal, etc.). Por lo tanto el sistema deberá permitir asignar la configuración del tipo de caja.

Todos y cada uno de los elementos se deben poder identificar de una manera única y exclusiva, para poder realizar una mejor manipulación, y esto se hace por medio de una asignación de etiquetas.

Es necesario también que el sistema represente, además de los componentes de las moléculas (partículas, conexiones, etc.) otro tipo de información característica de ésta, para esto hay que dibujar las superficies de potencial y los interferogramas (Gráficas) de rayos x, a través de una rutina que sea capaz de calcular el potencial eléctrico y otra que pueda generar los patrones de difracción de rayos x a partir de sus archivos de configuración.

Se requiere además, como otra forma de proporcionar información al usuario, que se pueda obtener una impresión de una imagen de la molécula visualizada, en cualquier momento y en cualquier disposición de partículas dentro de la escena gráfica. Lo anterior, es muy importante porque al tener una visión de la configuración y disposición correcta de una molécula en la pantalla, es posible y muy útil tener, si así se desea, una copia impresa del trabajo realizado y así poder observar y reportar mejor algunos detalles más relevantes.

Se requiere también realizar una impresión de los archivos de configuración de la estructura molecular estudiada (partículas, dipolos (momentos eléctricos), o conexiones), así como su correspondiente archivo de configuración de hoyos (cavidades)).

En el sistema será posible seleccionar, en versiones posteriores, los objetos visualizados. El tipo de selección podrá ser una selección por especie, una selección individual y una selección por grupo, y por lo tanto, para visualizar y manejar los elementos de la molécula, será posible agrupar por especie y agrupar por conjunto, esto quiere decir que si se desea, se podrá observar sólo una región o conjunto, seleccionar todos los átomos de determinado tipo o individualmente haciendo uso del puntero del ratón, esto permitirá actuar sobre los objetos de la molécula, modificar sus propiedades y trabajar de una mejor manera con cada elemento que la compone.

Las operaciones que se pueden realizar con el sistema sobre las moléculas, para conocer mejor su comportamiento son: obtener las distancias entre partículas, obtener ángulos entre conexiones, además de difuminar la profundidad de una molécula, esto quiere decir hacerla más transparente de modo que sea posible observar las partículas que están detrás.

Será posible minimizar el radio de todas las partículas, para poder tener una mejor perspectiva de observación, poder manipular mejor la molécula y así estudiarla mas fácilmente.

Otra de las características planeadas para el sistema en futuras versiones, es que pueda leer archivos de configuración de otros formatos aparte del formato específico del mismo. Para lo cual, será necesario que se identifique el tipo del archivo y posteriormente se lleve a cabo una interpretación del formato y una subsecuente traducción al formato manejado por definición. Por lo anterior, es necesario que el usuario pueda guardar el archivo de datos, en caso de que durante el uso del sistema haya realizado algunos cambios. Así mismo, será posible renombrarlo en caso de que se desee conservar el archivo de configuración original.

Debido a que hay cálculos necesarios para el funcionamiento del sistema que no se realizan localmente por su complejidad, se propone que se tenga además un monitor de procesos que sea capaz de enviar cálculos a máquinas remotas y verificar su estado para informar al usuario en dado caso de que dicho cálculo haya finalizado (esto después de recibir el

archivo de datos), o algún otro estado de interés sobre el proceso remoto, que le indique al usuario si pueden o no usar los resultados de los cálculos para poder observarlos.

Otra de las características importantes del sistema es que la posibilidad de crear nuevas moléculas de manera "manual", insertando átomos y conexiones, a través del editor molecular, con el cual, el usuario podrá crear nuevas moléculas, así como modificar el número de partículas que hay dentro de las ya existentes, quitando y agregando además de poder modificar sus atributos como radio, carga eléctrica, interacción, color, calidad, transparencia, etc. Sin embargo en el presente trabajo, se realiza únicamente el análisis necesario para la implementación del editor en versiones posteriores

Este editor molecular requerirá de una tabla periódica, a través de la cual será posible que el usuario seleccione el tipo de elemento que va a insertar dentro de la caja que delimita el área de trabajo.

Los colores de los objetos, serán predefinidos para cada objeto dentro de la molécula de trabajo, pero además de eso será posible que el usuario los modifique según le parezca conveniente, a través de la interfaz gráfica de usuario.

La interfaz gráfica, podrá ser utilizada también por medio de los dispositivos, como son: teclado, ratón, y perillas (para las cuales, será necesario realizar algunas adecuaciones posteriores al sistema), para activar varios eventos que permitan al usuario manipular la escena gráfica de la manera mas adecuada en la observación de la visualización de la dinámica molecular.

9.1.1.2. Identificar objetos y clases

9.1.1.2.1. Seleccionar nombres en los requisitos

1. Esfera	23. Salida en impresora
2. Dipolo	24. Imagen
3. Átomo	25. Cálculos remotos
4. Hoyo	26. Animación por cuadros
5. Conexión	27. Minimizar radio
6. Celda computacional	28. Validar conexiones
7. Superficie equipotencial	29. Leer archivo de datos
8. Ejes	30. Seleccionar partículas
9. Molécula	31. Seleccionar hoyos
10. Rayos X	32. Seleccionar conexiones
11. Luces	33. Seleccionar dipolos
12. Materiales	34. Agrupar por especie
13. Rotaciones	35. Agrupar por conjunto
14. Traslaciones	36. Obtener distancias entre partículas
15. Etiquetas	37. Obtener ángulos entre conexiones
16. Tabla periódica	38. Puntero 3D
17. Colores de objetos	39. Pinta partículas
18. Dispositivos	40. Pinta hoyos
19. Interfaz gráfica	41. Pinta superficies de potencial
20. Archivo de datos de entrada	42. Pinta dipolos
21. Archivo de datos de salida	43. Pinta conexiones
22. Salida a pantalla	44. Pinta rayos x
45. Pinta caja	62. Impresión de imagen
46. Pinta ejes	63. Impresión de archivo de configuración
47. Guardar archivo de datos	64. Mostrar propiedades de elementos
48. Identificar tipo de archivo	65. Deshacer cambios
49. Renombrar archivo de datos	66. Difuminar la profundidad de la molécula
50. Editor molecular	67. Asignar material
51. Asignar propiedades a objetos	68. Seleccionar tipo de caja
52. Insertar átomo	69. Ajustar configuración al tipo de caja

53. Pintar átomo	70. Detener animación en cualquier cuadro
54. Fondo	71. Invertir sentido de la animación
55. Distancia de conexión	72. Ejecutar proceso remoto
56. Buscar hoyos	73. Verificar estado del proceso
57. Generar difractograma de rayos x	74. Enviar avisos del proceso
58. Multiplicar sistema	75. Recibir archivo de datos
59. Generar potencial electrostático	76. Copiar objetos de escena
60. Estructura cristalina	77. Cortar objetos de escena
61. Generar estructura cristalina	78. Proceso remoto

9.1.1.2.2. Añadir clases adicionales procedentes de nuestro conocimiento del tema

79. Escena gráfica	90. Interacción
80. Impresiones	91. Parámetros de la interacción
81. Cálculo local	92. Número de interacciones
82. Proceso local	93. Controles de la dinámica molecular
83. Monitor de procesos	94. Masa de los átomos
84. Sistema de cavidades (hoyos)	95. Carga eléctrica de los iones.
85. Elemento de tabla periódica	96. Ajustar parámetros de interacción
86. Pegar objetos en escena	97. Generar archivos de datos de entrada para la dinámica
87. Generar vecinos	98. Generar archivos de datos de salida
88. Calcular análisis de coordinación	99. Pintar cortes de la estructura.
89. Generar dipolos	

9.1.1.2.3. Eliminar redundancias

En este paso se eliminó 61) Generar estructura cristalina, ya que es lo mismo que 58) Multiplicar sistema.

1. Esfera	14. Traslaciones
2. Dipolo	15. Etiquetas
3. Átomo	16. Tabla Periódica
4. Hoyo	17. Colores de objetos
5. Conexión	18. Dispositivos
6. Celda computacional	19. Interfaz gráfica
7. Superficie equipotencial	20. Archivo de datos de entrada
8. Ejes	21. Archivo de datos de salida
9. Molécula	22. Salida a pantalla
10. Rayos X	23. Salida en impresora
11. Luces	24. Imagen
12. Materiales	25. Cálculos remotos
13. Rotaciones	26. Animación por cuadros
27. Minimizar radio	53. Pintar átomo
28. Validar conexiones	54. Fondo
29. Leer archivo de datos	55. Distancia de conexión
30. Seleccionar partículas	56. Buscar hoyos
31. Seleccionar hoyos	57. Generar difractograma de rayos x
32. Seleccionar conexiones	58. Multiplicar sistema
33. Seleccionar dipolos	59. Generar potencial electrostático
34. Agrupar por especie	60. Estructura cristalina
35. Agrupar por conjunto	62. Impresión de imagen

36. Obtener distancias entre partículas	63. Impresión de archivo de configuración
37. Obtener ángulos entre conexiones	64. Mostrar propiedades de elementos
38. Puntero 3D	65. Deshacer cambios
39. Pinta partículas	66. Difuminar la profundidad de la molécula
40. Pinta hoyos	67. Asignar material
41. Pinta superficies de potencial	68. Seleccionar tipo de caja
42. Pinta dipolos	69. Ajustar configuración al tipo de caja
43. Pinta conexiones	70. Detener animación en cualquier cuadro
44. Pinta rayos x	71. Invertir sentido de la animación
45. Pinta caja	72. Ejecutar proceso remoto
46. Pinta ejes	73. Verificar estado del proceso
47. Guardar archivo de datos	74. Enviar avisos del proceso
48. Identificar tipo de archivo	75. Recibir archivo de datos
49. Renombrar archivo de datos	76. Copiar objetos de escena
50. Editor molecular	77. Cortar objetos de escena
51. Asignar propiedades a objetos	78. Proceso remoto
52. Insertar átomo	

9.1.1.2.4. Eliminar clases irrelevantes

Las siguientes resultaron ser clases irrelevantes, de modo que fueron eliminadas:

- 18. Dispositivos
- 19. Interfaz gráfica
- 22. Salida a pantalla

1. Esfera	16. Tabla periódica
2. Dipolo	17. Colores de objetos
3. Átomo	
4. Hoyo	20. Archivo de datos de entrada
5. Conexión	21. Archivo de datos de salida
6. Celda computacional	
7. Superficie equipotencial	23. Salida en impresora
8. Ejes	24. Imagen
9. Molécula	25. Cálculos remotos
10. Rayos X	26. Animación por cuadros
11. Luces	27. Minimizar radio
12. Materiales	28. Validar conexiones
13. Rotaciones	29. Leer archivo de datos
14. Traslaciones	30. Seleccionar partículas
15. Etiquetas	31. Seleccionar hoyos
32. Seleccionar conexiones	56. Buscar hoyos
33. Seleccionar dipolos	57. Generar difractograma de rayos x
34. Agrupar por especie	58. Multiplicar sistema
35. Agrupar por conjunto	59. Generar potencial electrostático
36. Obtener distancias entre partículas	60. Estructura cristalina
37. Obtener ángulos entre conexiones	
38. Puntero 3D	62. Impresión de imagen
39. Pinta partículas	63. Impresión de archivo de configuración
40. Pinta hoyos	64. Mostrar propiedades de elementos
41. Pinta superficies de potencial	65. Deshacer cambios
42. Pinta dipolos	66. Difuminar la profundidad de la molécula

43. Pinta conexiones	67. Asignar material
44. Pinta rayos x	68. Seleccionar tipo de caja
45. Pinta caja	69. Ajustar configuración al tipo de caja
46. Pinta ejes	70. Detener animación en cualquier cuadro
47. Guardar archivo de datos	71. Invertir sentido de la animación
48. Identificar tipo de archivo	72. Ejecutar proceso remoto
49. Renombrar archivo de datos	73. Verificar estado del proceso
50. Editor molecular	74. Enviar avisos del proceso
51. Asignar propiedades a objetos	75. Recibir archivo de datos
52. Insertar átomo	76. Copiar objetos de escena
53. Pintar átomo	77. Cortar objetos de escena
54. Fondo	78. Proceso remoto
55. Distancia de conexión	

9.1.1.2.5. Eliminar clases vagas

Se cambio clase 24. Imagen por Cálculo de imagen, se elimino 23. Salida en impresora. Así mismo, se cambiaron: 26. Animación por cuadros por Objeto animado y 20. Archivo de datos de entrada. 21. Archivo de datos de salida por 20. Archivo de configuración.

1. Esfera	25. Cálculos remotos
2. Dipolo	26. Objeto animado
3. Átomo	27. Minimizar radio
4. Hoyo	28. Validar conexiones
5. Conexión	29. Leer archivo de datos
6. Celda computacional	30. Seleccionar partículas
7. Superficie equipotencial	31. Seleccionar hoyos
8. Ejes	32. Seleccionar conexiones
9. Molécula	33. Seleccionar dipolos
10. Rayos X	34. Agrupar por especie
11. Luces	35. Agrupar por conjunto
12. Materiales	36. Obtener distancias entre partículas
13. Rotaciones	37. Obtener ángulos entre conexiones
14. Traslaciones	38. Puntero 3D
15. Etiquetas	39. Pinta partículas
16. Tabla periódica	40. Pinta hoyos
17. Colores de objetos	41. Pinta superficies de potencial
20. Archivo de configuración	42. Pinta dipolos
24. Cálculo de imagen	43. Pinta conexiones
	44. Pinta rayos x
45. Pinta caja	63. Impresión de archivo de configuración
46. Pinta ejes	64. Mostrar propiedades de elementos
47. Guardar archivo de datos	65. Deshacer cambios
48. Identificar tipo de archivo	66. Difuminar la profundidad de la molécula
49. Renombrar archivo de datos	67. Asignar material
50. Editor molecular	
51. Asignar propiedades a objetos	68. Seleccionar tipo de caja
52. Insertar átomo	69. Ajustar configuración al tipo de caja
53. Pintar átomo	70. Detener animación en cualquier cuadro
54. Fondo	71. Invertir sentido de la animación
	72. Ejecutar proceso remoto
55. Distancia de conexión	73. Verificar estado del proceso
56. Buscar hoyos	74. Enviar avisos del proceso

57. Generar difractograma de rayos x 58. Multiplicar sistema 59. Generar potencial electrostático 60. Estructura cristalina 62. Impresión de imagen	75. Recibir archivo de datos 76. Copiar objetos de escena 77. Cortar objetos de escena 78. Proceso remoto
-----------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------

9.1.1.2.6. Separar atributos

ATRIBUTOS	
11. Luces 12. Materiales 15. Etiquetas 17. Colores de objetos 54. Fondo	90. Interacción 91. Parámetros de la interacción 92. Número de interacciones 93. Controles de la dinámica molecular 94. Masa de los átomos 95. Carga eléctrica de los iones.

1. Esfera 2. Dipolo 3. Átomo 4. Hoyo 5. Conexión 6. Celda computacional 7. Superficie equipotencial 8. Ejes 9. Molécula 10. Rayos X 13. Rotaciones 14. Traslaciones 16. Tabla periódica 20. Archivo de configuración 23. Salida en impresora 43. Pinta conexiones 44. Pinta rayos x 45. Pinta caja 46. Pinta ejes 47. Guardar archivo de datos 48. Identificar tipo de archivo 49. Renombrar archivo de datos 50. Editor molecular 51. Asignar propiedades a objetos 52. Insertar átomo 53. Pintar átomo	24. Cálculo de imagen 25. Cálculos remotos 26. Objeto animado 27. Minimizar radio 28. Validar conexiones 29. Leer archivo de datos 30. Seleccionar partículas 31. Seleccionar hoyos 32. Seleccionar conexiones 33. Seleccionar dipolos 34. Agrupar por especie 35. Agrupar por conjunto 36. Obtener distancias entre partículas 37. Obtener ángulos entre conexiones 38. Puntero 3D 39. Pinta partículas 40. Pinta hoyos 41. Pinta superficies de potencial 42. Pinta dipolos 61. Impresión de imagen 62. Impresión de archivo de configuración 63. Mostrar propiedades de elementos 64. Deshacer cambios 65. Difuminar la profundidad de la molécula 66. Asignar material 67. Seleccionar tipo de caja 68. Ajustar configuración al tipo de caja 69. Detener animación en cualquier cuadro 70. Invertir sentido de la animación 71. Ejecutar proceso remoto 72. Verificar estado del proceso
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

56. Buscar hoyos 57. Generar difractograma de rayos x 58. Multiplicar sistema 59. Generar potencial electrostático 60. Estructura cristalina	73. Enviar avisos del proceso 74. Recibir archivo de datos 75. Copiar objetos de escena 76. Cortar objetos de escena 77. Proceso remoto
----------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------

9.1.1.2.7. Separar clases y métodos

CLASES

1. Esfera 2. Dipolo 3. Átomo 4. Hoyo 5. Conexión 6. Plano de corte 7. Caja 8. Celda computacional 9. Superficie equipotencial	10. Ejes 11. Molécula 12. Rayos X 13. Archivo de datos 14. Cálculo remoto 15. Objeto animado 16. Puntero 3D 17. Editor molecular 18. Estructura cristalina	19. Proceso remoto 20. Escena gráfica 21. Impresiones 22. Cálculo local 23. Proceso local 24. Monitor de procesos 25. Sistema de cavidades (hoyos) 26. Tabla periódica
-------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

ATRIBUTOS

1. Luces 2. Materiales 3. Interacción 4. Etiquetas	5. Colores de objetos 6. Interacción 7. Parámetros de la interacción 8. Número de interacciones	9. Masa de los átomos 10. Carga eléctrica de los iones 11. Fondo 12. Controles de la dinámica molecular
-------------------------------------------------------------	----------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------

METODOS

<p>En escena gráfica:</p> 1. Rotaciones 2. Traslaciones 3. Detener animación en cualquier cuadro 4. Invertir sentido de la animación 5. Agrupar por especie 6. Agrupar por conjunto 7. Mostrar propiedades de elementos 8. Asignar propiedades a objetos 9. Deshacer cambios 10. Difuminar la profundidad de la molécula 11. Ajustar configuración al tipo de caja	<p>En Cálculos:</p> <p>a) Cálculos remotos:</p> 20. Buscar hoyos 21. Generar potencial electrostático 22. Generar dipolos 23. Generar dinámica 24. Generar archivos de salida
<p>En impresiones:</p> 12. Impresión de imagen 13. Impresión de archivo de configuración	<p>b) Cálculos locales:</p> 25. Cálculo Imagen 26. Generar difractograma de rayos x

En archivos:

14. Leer archivo de datos
15. Guardar archivo de datos
16. Identificar tipo de archivo
17. Renombrar archivo de datos
18. Generar archivos de datos de entrada para la dinámica.
19. Ajustar parámetros de interacción

En objetos gráficos :

32. Minimizar radio
33. Validar conexiones
34. Seleccionar partículas
35. Seleccionar hoyos
36. Seleccionar conexiones
37. Seleccionar dipolos
38. Obtener distancias entre partículas
39. Obtener ángulos entre conexiones
40. Pinta partículas
41. Pinta hoyos
42. Pinta superficies de potencial
43. Pinta dipolos
44. Pinta conexiones
45. Pinta rayos x
46. Pinta caja
47. Pinta ejes
48. Pinta cortes de la estructura
49. Asignar material
50. Seleccionar tipo de caja

27. Calcular análisis de coordinación
28. Multiplicar sistema
29. Generar vecinos
30. Determinar átomos visibles de un corte

En editor molecular:

51. Insertar átomo
52. Pintar átomo
53. Copiar objetos de escena
54. Cortar objetos de escena
55. Pegar objetos en escena
56. Agrupar objetos en escena

En monitor de procesos:

57. Ejecutar proceso remoto
58. Verificar estado del proceso
59. Enviar avisos del proceso
60. Recibir archivo de datos

9.1.1.3. Identificar y depurar relaciones

9.1.1.3.1. Seleccionar verbos relacionales en los requisitos

- | | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>1.) Un átomo es una esfera</p> <p>2.) Un dipolo es una esfera con excentricidad $\neq 1$</p> <p>3.) Un hoyo es una esfera</p> <p>4.) Una molécula esta formada de átomos (partículas),</p> <p>5.) Una molécula puede o no tener dipolos, hoyos, conexiones.</p> <p>6.) Una conexión une a dos átomos.</p> <p>7.) Los hoyos forman parte de una molécula.</p> <p>8.) Una escena gráfica tiene tres ejes coordenados.</p> <p>9.) Una escena gráfica tiene una celda unitaria.</p> <p>10.) Una molécula genera potencial.</p> <p>11.) Las luces forman parte de la escena gráfica.</p> <p>12.) Los materiales definen las características de los átomos.</p> <p>13.) Los materiales definen las características de las conexiones.</p> <p>14.) Un material y un modelo de iluminación (luz) determinan si se realizan cálculos de iluminación o no.</p> <p>15.) Cada partícula posee una única etiqueta que la identifica.</p> <p>16.) La molécula experimenta transformaciones espaciales dentro de la escena gráfica</p> <p>17.) Una rotación es una transformación espacial</p> <p>18.) Una traslación es una transformación espacial</p> <p>19.) La tabla periódica requiere de los datos del archivo de definición de elementos</p> <p>20.) Cada objeto posee colores</p> <p>21.) La interfaz gráfica permite al usuario interactuar con la molécula visualizada</p> <p>22.) El archivo de datos de entrada es necesario para la visualización de las moléculas</p> <p>23.) El programa devuelve un archivo de datos de salida.</p> <p>24.) El monitor de procesos recibe y envía datos para realizar los cálculos remotos</p> <p>25.) El método de salida a impresora entrega una impresión de archivo de configuración</p> <p>26.) El método de salida a impresora entrega una impresión de imagen.</p> <p>27.) El puntero 3D nos permite seleccionar partículas.</p> <p>28.) El puntero 3D nos permite seleccionar hoyos.</p> <p>29.) El puntero 3D nos permite seleccionar conexiones.</p> <p>30.) El puntero 3D nos permite seleccionar dipolos.</p> <p>31.) Una escena gráfica tiene una o más moléculas</p> <p>32.) Una estructura cristalina es una repetición de una molécula</p> <p>33.) El editor molecular sirve para insertar un átomo en la escena gráfica y formar una nueva molécula o modificar alguna existente.</p> <p>34.) El editor molecular permite al usuario pintar átomos.</p> <p>35.) El método de copiar objetos de escena permite copiar objetos de la escena gráfica a través del editor molecular.</p> <p>36.) El editor molecular permite pegar objetos en la escena gráfica.</p> <p>37.) El editor molecular permite cortar objetos en la escena gráfica.</p> <p>38.) Con el editor molecular podemos llamar a la tabla periódica para asignar las propiedades para pintar un átomo.</p> <p>39.) La tabla periódica permite asignar las propiedades para mostrar elementos sobre una ventana.</p> <p>40.) La escena gráfica se puede animar por cuadros.</p> <p>41.) La escena gráfica se puede agrupar por especie a los elementos de la molécula.</p> <p>42.) La escena gráfica se puede agrupar por conjunto a los elementos de la molécula.</p> <p><u>43.) La escena gráfica se puede construir por cortes sobre la estructura original.</u></p> <p>44.) Se pueden mostrar las propiedades de los elementos de la escena gráfica.</p> <p><u>45.) Se pueden construir o leer los archivos de entrada de la dinámica molecular.</u></p> <p>46.) Se pueden asignar las propiedades a los objetos de la escena gráfica.</p> <p>47.) Es posible deshacer los cambios hechos a la escena gráfica.</p> | <p>48.) En la escena gráfica se puede ajustar la configuración del tipo de la caja.</p> <p>49.) Una caja define los límites de la celda computacional.</p> <p>50.) En la escena es posible difuminar la profundidad de la molécula.</p> <p>51.) El monitor de procesos verifica el estado de algún proceso remoto.</p> <p>52.) El monitor de procesos permite ejecutar un proceso remoto.</p> <p>53.) Con el monitor de procesos recibimos avisos del proceso.</p> <p>54.) El monitor de procesos recibe los archivos de datos del proceso remoto.</p> <p>55.) Es posible minimizar el radio de las conexiones y partículas en la escena gráfica.</p> <p>56.) Se pueden validar las conexiones en la escena gráfica.</p> <p>57.) El método obtener ángulos entre conexiones permite conocer el ángulo que hay entre tres partículas.</p> <p>58.) El método obtener distancias entre partículas permite conocer la distancia que existe entre dos partículas.</p> <p>59.) Con el método asignar material, se puede asignar un material específico a una partícula o a una conexión.</p> <p>60.) Se dibujan partículas con el método pinta partículas en la escena gráfica.</p> <p>61.) El método pinta hoyos, dibuja hoyos en la escena gráfica.</p> <p>62.) El método pinta campos escalares, dibuja potenciales en la escena.</p> <p>63.) El método pinta dipolos, dibuja dipolos en la escena gráfica.</p> <p>64.) El método pinta conexiones, dibuja conexiones en la escena gráfica.</p> <p>65.) El método pinta rayos X, dibuja la gráfica de rayos X de la molécula.</p> <p><u>66.) El método pinta corte, dibuja los pedazos de estructura resultantes de un corte.</u></p> <p>67.) El método pinta caja, dibuja la caja en la escena gráfica.</p> <p>68.) El método pinta ejes, dibuja los ejes en la escena gráfica.</p> <p>69.) Con el método leer archivo de datos podemos obtener los datos para dibujar la molécula en la escena gráfica.</p> <p>70.) Se puede guardar el archivo de datos de la molécula visualizada con el método guardar archivo de datos.</p> <p>71.) Es necesario identificar el tipo de archivo de datos para dibujar la molécula.</p> <p>72.) Se puede renombrar el archivo de datos de la molécula visualizada.</p> <p>73.) Con el método guardar imagen visualizada, es posible guardar una imagen de la molécula.</p> <p>74.) El método cálculos remotos envía y recibe los cálculos realizados por el proceso remoto.</p> <p>75.) Es posible buscar hoyos para dibujarlos en la escena gráfica.</p> <p><u>76.) Se puede hacer el ajuste de los parámetros de los potenciales.</u></p> <p>77.) El método generar rayos X, permite realizar cálculos necesarios para dibujar la gráfica de rayos X.</p> <p><u>78.) Son necesarios los métodos para graficar las salidas: $g(r)$, $T(t)$, $P(t)$, $En(t)$, $Etot(t)$</u></p> <p>79.) Podemos multiplicar el sistema y aumentar el número de moléculas, con el método de multiplicar sistema.</p> <p>80.) El método generar campo escalar, realiza los cálculos para generar los planos de potencial (campos escalares).</p> <p>81.) El método generar estructura cristalina, realiza los cálculos para generar una estructura de molécula repetida.</p> <p>82.) Los objetos gráficos (partículas, dipolos, hoyos, conexiones) requieren de una definición de un material para ser dibujadas en la escena gráfica.</p> <p>83.) Los objetos gráficos tienen etiquetas.</p> <p>84.) Los objetos gráficos tienen colores.</p> <p>85.) Los objetos gráficos tienen un tamaño (al modificar el radio)</p> <p>86.) La escena gráfica tiene un fondo.</p> <p>87.) Las partículas, los dipolos, los hoyos, las conexiones, la caja, los ejes, el fondo y el puntero del ratón; son objetos de la escena gráfica.</p> <p>88.) El método deshacer cambios, deshace lo que se ha realizado sobre los objetos de la escena gráfica.</p> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

9.1.1.3.2. Añadir relaciones adicionales procedentes de nuestro conocimiento

- 89.) La celda tiene una molécula.
- 90.) La celda tiene una caja.
- 91.) La celda tiene planos de potencial.
- 92.) La celda tiene un sistema de cavidades.
- 93.) La escena gráfica tiene una o más moléculas de diferente tipo.
- 94.) La molécula experimenta transformaciones espaciales dentro de la escena gráfica.
- 95.) Un reescalamiento es una transformación espacial.
- 96.) La estructura también puede sufrir cortes.

9.1.1.3.3. Eliminar relaciones de diseño o entre clases eliminadas

- | | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> 1.) Un átomo es una esfera 2.) Un dipolo es una esfera con excentricidad $\neq 1$ 3.) Un hoyo es una esfera 4.) Una molécula esta formada de átomos (partículas), 5.) Una molécula puede o no tener dipolos, hoyos, conexiones. 6.) Una conexión une a dos átomos. 7.) El método deshacer cambios, deshace lo que se ha realizado sobre los objetos de la escena gráfica. 8.) Una escena gráfica tiene tres ejes coordenados. 9.) Una escena gráfica tiene una celda unitaria. 10.) Una molécula genera potencial. 11.) La molécula experimenta transformaciones espaciales dentro de la escena gráfica 12.) Una rotación es una transformación espacial 13.) Una traslación es una transformación espacial 14.) El archivo de datos de entrada es necesario para la visualización de las moléculas 15.) El monitor de procesos recibe y envía datos para realizar los cálculos remotos 16.) El método de salida a impresora entrega una impresión de archivo de configuración 17.) Impresiones entrega una impresión de imagen. 18.) El puntero 3D nos permite seleccionar partículas. 19.) El puntero 3D nos permite seleccionar hoyos. 20.) El puntero 3D nos permite seleccionar conexiones. 21.) El puntero 3D nos permite seleccionar dipolos. 22.) Una escena gráfica tiene una o más moléculas 23.) Una estructura cristalina es una repetición de una molécula 24.) El editor molecular permite al usuario pintar átomos. 25.) El método de copiar objetos de escena permite copiar objetos de la escena gráfica a través del editor molecular. 26.) El editor molecular permite pegar objetos en la escena gráfica. 27.) El editor molecular permite cortar objetos en la escena gráfica. <u>28.) La escena gráfica se puede animar por cuadros.</u> 29.) La escena gráfica se puede agrupar por especie a los elementos de la molécula. 30.) La escena gráfica se puede agrupar por conjunto a los elementos de la molécula. <u>31.) La escena gráfica se puede construir por cortes sobre la estructura original.</u> 32.) Es posible deshacer los cambios hechos a la escena gráfica. 33.) En la escena gráfica se puede ajustar la configuración del tipo de la caja. <u>34.) Se pueden construir o leer los archivos de entrada de la dinámica molecular.</u> | <ul style="list-style-type: none"> 35.) Una caja define los límites de la celda computacional. 36.) En la escena es posible difuminar la profundidad de la molécula. 37.) El monitor de procesos verifica el estado de algún proceso remoto. 38.) El monitor de procesos permite ejecutar un proceso remoto. 39.) Con el monitor de procesos recibimos avisos del proceso. 40.) El monitor de procesos recibe los archivos de datos del proceso remoto. 41.) El método obtener ángulos entre conexiones permite conocer el ángulo que hay entre tres partículas. 42.) El método obtener distancias entre partículas permite conocer la distancia que existe entre dos partículas. 43.) Se dibujan partículas con el método pinta partículas en la escena gráfica. 44.) El método pinta hoyos, dibuja hoyos en la escena gráfica. 45.) El método pinta campos escalares, dibuja potenciales en la escena. 46.) El método pinta dipolos, dibuja dipolos en la escena gráfica. 47.) El método pinta conexiones, dibuja conexiones en la escena gráfica. 48.) El método pinta rayos X, dibuja la gráfica de rayos X de la molécula. <u>49.) El método pinta corte, dibuja los pedazos de estructura resultantes de un corte.</u> 50.) El método pinta caja, dibuja la caja en la escena gráfica. 51.) El método pinta ejes, dibuja los ejes en la escena gráfica. <u>52.) Con el método leer archivo de datos podemos obtener los datos para dibujar la molécula en la escena gráfica.</u> 53.) Se puede guardar el archivo de datos de la molécula visualizada con el método guardar archivo de datos. 54.) Se puede renombrar el archivo de datos de la molécula visualizada. 55.) Con el método guardar imagen visualizada, es posible guardar una imagen de la molécula. 56.) El método cálculos remotos envía y recibe los cálculos realizados por el proceso remoto. 57.) Es posible buscar hoyos para dibujarlos en la escena gráfica. 58.) El método generar rayos X, permite realizar cálculos necesarios para dibujar la gráfica de rayos X. <u>59.) Se puede hacer el ajuste de los parámetros de los potenciales.</u> <u>60.) Son necesarios los métodos para graficar las salidas: $g(r)$, $T(t)$, $P(t)$, $En(t)$, $Etot(t)$.</u> 61.) Podemos multiplicar el sistema y aumentar el número de moléculas, con el método de multiplicar sistema. 62.) El método generar campo escalar, realiza los cálculos para generar los planos de potencial (campos escalares). 63.) El método generar estructura cristalina, realiza los cálculos para generar una estructura de molécula repetida. 64.) Las partículas, los dipolos, los hoyos, las conexiones, la caja, los ejes, el fondo y el puntero del ratón; son objetos de la escena gráfica. |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

9.1.1.3.4. Eliminar eventos transitorios

<p>1.) Un átomo es una esfera 2.) Un dipolo es una esfera con excentricidad $\neq 1$ 3.) Un hoyo es una esfera 4.) Una molécula genera planos de potencial. 5.) La molécula experimenta transformaciones espaciales dentro de la escena gráfica 6.) Las transformaciones espaciales modifican la escena gráfica 7.) Una rotación es una transformación espacial 8.) Una traslación es una transformación espacial 9.) Un reescalamiento es una transformación espacial</p> <p>10.) La celda se puede ajustar la configuración del tipo de la caja. 11.) Impresiones genera una impresión de archivo de configuración 12.) Impresiones genera una impresión de imagen. 13.) El puntero 3D marca partículas 14.) El puntero 3D marca hoyos 15.) El puntero 3D marca conexiones 16.) El puntero 3D marca dipolos 17.) Una estructura cristalina es una repetición de una celda unitaria</p> <p>*18.) El editor molecular permite al usuario insertar átomos a la molécula</p> <p>19) El editor molecular permite copiar objetos gráficos de una o varias moléculas</p>	<p>20.) El editor molecular permite pegar objetos gráficos de una o varias moléculas 21.) El editor molecular permite cortar objetos gráficos de una o varias moléculas</p> <p>22.) El editor molecular permite deshacer los cambios hechos a las moléculas 23.) El editor molecular permite rehacer los cambios hechos a las moléculas 24.) El monitor de procesos verifica el estado de los procesos remotos. 25.) El monitor de procesos verifica el estado de los procesos locales. 26.) El monitor de procesos permite ejecutar procesos remotos. 27.) El monitor de procesos permite ejecutar procesos locales. 28.) El monitor de procesos emite mensajes de estatus del proceso 29.) El proceso remoto recibe el archivo de datos de cálculo remoto. 30.) El proceso remoto envía el archivo de datos de partículas a cálculo remoto. 31.) El proceso local recibe el vector de datos de cálculo local. 32.) El proceso local envía el vector de datos de partículas a cálculo local.</p> <p>*33.) Las partículas, los dipolos, los hoyos, las conexiones, la caja, los ejes, el fondo y el puntero del ratón; son objetos de la escena gráfica.</p> <p>34.) Una caja define los límites de la celda .</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

9.1.1.3.5. Eliminar relaciones redundantes o derivadas

En el cuadro anterior *33 es una relación redundante
 *18 es lo mismo que pegarlo en la molécula.

9.1.1.3.6. Añadir relaciones olvidadas

<p>1.) Impresiones recibe un vector de datos de archivo de configuración 2.) Impresiones recibe un vector de datos de cálculo de imagen 3.) El Editor molecular permite conectar átomos 4.) El Editor molecular permite conectar moléculas 5.) El Editor molecular usa un puntero 3D 6.) Contenedor de partículas proporciona configuración para representar una molécula 7.) Contenedor de hoyos proporciona configuración para representar el sistema de cavidades 8.) Contenedor de dipolos proporciona configuración para dibujar dipolos 9.) Contenedor de potencial proporciona configuración para representar el campo potencial 10.) Cálculo imagen genera un vector de imagen de la escena gráfica 11.) Cálculo de imagen coloca un vector de imagen en archivo de imagen 12.) El sistema de cavidades se forma dentro de la molécula 13.) El indicador de selecciones asiste al editor molecular para seleccionar objetos gráficos</p>	<p>14.) El indicador de selecciones asiste al modificador de propiedades para seleccionar objetos gráficos 15.) El indicador de selecciones asiste al cálculo de magnitudes y ángulos para seleccionar objetos gráficos 16.) El modificador de propiedades cambia propiedades a los objetos gráficos 17.) El modificador de propiedades notifica los cambios al editor molecular</p> <p><i>18.) La escena gráfica se puede construir por cortes sobre la estructura original.</i> <i>19.) Contenedor de dinámica proporciona configuración de entrada para generar la dinámica molecular.</i> <i>20.) Plano de corte, dibuja los pedazos de estructura resultantes de un corte sobre una molécula.</i> <i>21.) Rayos X genera las salidas: $g(r)$, $T(t)$, $P(t)$, $En(t)$, $Etot(t)$.</i></p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

9.1.1.3.7. Definir la multiplicidad de cada relación

- 1.) Una conexión une a dos átomos.
- 2.) Una escena gráfica tiene tres ejes coordenados.
- 3.) La escena gráfica tiene una estructura cristalina
- 4.) La escena gráfica tiene un puntero 3D
- 5.) La celda tiene una o más moléculas
- 6.) La celda tiene una caja
- 7.) La celda tiene planos de potencial
- 8.) La celda tiene un sistema de cavidades

- 9.) Una molécula esta formada de muchos átomos (partículas)
- 10.) Una molécula puede o no tener muchos dipolos
- 11.) Una molécula puede o no tener muchas conexiones
- 12.) Un dipolo esta asociado a un sólo átomo
- 13.) El sistema de cavidades esta formado por varios hoyos
- 14.) El campo potencial esta formado por varios planos de potencial
- 15.) La estructura cristalina esta formada de una o varias celdas

9.1.1.4. Clases abstractas y terminales

CLASES

1. Objeto gráfico A
2. Objeto estático A
3. Caja
4. Eje
5. Plano de corte
6. Objeto dinámico A
7. Esfera A
8. Dipolo
9. Átomo
10. Hoyo
11. Conexión
12. Plano potencial
13. Objeto animado A
14. Sistema de cavidades (hoyos)
15. Molécula
16. Superficie potencial
17. Indicador de selección

18. Modificador de propiedades
19. Escena gráfica
20. Estructura cristalina
21. Celda
22. Proceso cálculo A
23. Cálculo remoto
24. Cálculo local
25. Cálculo generado por partículas A
26. Cálculo de imagen
27. Cálculo de magnitudes y ángulos
28. Cálculo de dinámica
29. Contenedor de datos A
30. Contenedor de partículas
31. Contenedor de hoyos
32. Contenedor de potencial
33. Contenedor de dipolos
34. Contenedor de imagen

35. Monitor de procesos
36. Proceso A
37. Proceso local
38. Proceso remoto
39. Impresiones A
40. Impresión de archivo de configuración
41. Impresión de imagen
42. Impresión de ayuda
43. Puntero 3D
44. Editor molecular
45. Contenedor de conexiones
46. Contenedor de rayos x
47. Contenedor de configuración de dinámica
48. Contenedor de inicia dinámica

A = Clase abstracta

9.1.1.5. Diccionario de datos (definición de clases)

1. Caja:

La caja "computacional" delimita el volumen en el cual se representa y manipula la molécula, permite contener un sistema básico, que constituye una unidad de la estructura molecular, para tener una mejor referencia de sus características. Dicha caja puede tomar diferentes formas dependiendo del tipo de estructura que se requiera analizar.

2. Eje:

Un eje es una referencia de la estructura contenida en la caja, que permite tener la orientación de la estructura durante las transformaciones espaciales que ésta experimenta.

3. Dipolo:

Es la representación del momento dipolar eléctrico de una determinada partícula. El momento dipolar eléctrico es un vector que señala el eje a lo largo del cual se han separado las cargas + y - del átomo y su sentido va de - a +. Se le representa por un elipsoide alargado en la dirección de dicho eje.

4. Átomo:

Es la representación de una partícula o elemento químico, los átomos se representan gráficamente con esferas cuyos radios y colores son determinados usando las características propias de la tabla periódica tales como peso atómico, carga, radio iónico, etc. Los átomos son la clase representativa que constituye a las moléculas.

5. Hoyo:

Un hoyo es una cavidad o espacio vacío dentro de la molécula, representado como una esfera.

6. Conexión:

Es la unión de dos partículas (átomos), mediante la representación gráfica de un cilindro o un prisma, a un determinado radio de alcance, en función de la distancia del enlace químico.

7. Plano de potencial:

Permite representar el potencial electrostático generado por la estructura molecular, un plano contiene regiones que indican valores potenciales positivos o negativos presentes en todos los puntos de una partición tridimensional contenida en el sistema de cavidades.

8. Sistema de cavidades (hoyos):

Los hoyos representan los espacios vacíos que constituyen la porosidad de una estructura molecular y se visualizan como esferas dentro de las cavidades de la molécula. El sistema de cavidades es el conjunto de todos los hoyos en la estructura.

9. Molécula:

Es la estructura formada por la agrupación de átomos, que puede también tener conexiones que los unan, así como dipolos que indiquen el momento eléctrico de cada átomo dentro de la molécula. La molécula representa la estructura de algún compuesto químico de interés para su visualización.

10. Superficie potencial:

Es el conjunto de valores que representan los niveles de potencial eléctrico de la molécula en todos los puntos del espacio dentro de la celda computacional, y que unidos forman una estructura tridimensional.

11. Indicador de selección:

Es un auxiliar del editor molecular, de modificador de propiedades y de cálculo de magnitudes y ángulos. Selecciona, marca e identifica objetos de molécula y hoyos. Proporciona a los objetos que auxilia, una lista de los valores de identificación (etiquetas) y posiciones en sus respectivos vectores de datos, lo que va a permitir manipular a los objetos seleccionados de manera individual, en grupo (un conjunto) o por especie en el caso de las partículas.

12. Modificador de propiedades:

Es un auxiliar del editor molecular, contiene una lista de los objetos gráficos que fueron seleccionados dentro de la escena, así como los mecanismos para indicar a cada uno de ellos que propiedades deben cambiar y que valores deben tomar. Maneja un vector de historia de cambios realizados. Además notifica cualquier cambio al editor molecular.

13. Escena gráfica:

Es la agrupación de todos los elementos y área de trabajo que se visualiza, contiene a los ejes, a la estructura cristalina y al puntero del ratón 3D, igualmente el color del fondo del área de trabajo es una característica particular de la escena gráfica, así como los mecanismos que permiten realizar las transformaciones espaciales dentro de ella a todos los objetos que la conforman.

14. Estructura cristalina:

Es la formación que se crea a partir de una representación de la celda unitaria en una repetición uniforme, dentro de la escena gráfica.

15. Celda

Es la agrupación de moléculas (puede ser sólo una) junto con el sistema de cavidades y la caja computacional, que sirve para el estudio y visualización de agregados de átomos, cuando la configuración dentro de la caja es una base, se denomina "celda unitaria", la cual representa la unidad básica de una estructura cristalina. Por medio de una repetición de la celda unitaria se crea una estructura cristalina compleja.

16. Cálculo remoto:

Es un tipo de cálculo que debido a su complejidad o elevado tiempo de cómputo, es necesario que se envíe a una máquina remota para que sea realizado, con los resultados obtenidos el sistema realiza funciones importantes en el esquema de visualización de la dinámica molecular, por ejemplo: Generar dipolos, hoyos y campo escalar (potencial).

17. Cálculo local:

Es un tipo de cálculo que se realiza en la máquina local (en la cual se ejecuta el sistema de visualización), y que no consume elevados recursos de cómputo para generar resultados necesarios en la visualización, por ejemplo: Generar gráfica de rayos x, generar vecinos o multiplicar el sistema (para obtener la estructura cristalina).

18. Cálculo de imagen:

Es el cálculo que se realiza para generar la imagen de la estructura visualizada en la escena gráfica, es decir, es el cálculo que permite obtener una imagen de la zona seleccionada por el usuario y ponerla en un vector de datos (para mandarla a imprimir) o en un archivo.

19. Cálculo de magnitudes y ángulos

Este es el cálculo que se encarga de obtener distancias y ángulos entre átomos y conexiones.

20. Cálculo de dinámica

Este es el cálculo que se realiza para generar un archivo de configuración de partículas.

21. Contenedor de partículas:

Representa un tipo de objeto que puede contener la configuración de partículas que formarán a una molécula a partir de la información obtenida de un archivo físico. Si el archivo está en un formato diferente al establecido para el sistema éste objeto puede ser capaz de leer sus datos y traducirlos al formato propio.

22. Contenedor de hoyos:

Contiene la salida generada por el programa busca hoyos en el que se encuentra la configuración de hoyos y coloca los datos necesarios como son las coordenadas de los centros y su radio correspondiente en un arreglo.

23. Contenedor de potencial:

Contiene la salida generada por el programa "busca potenciales" en el que se encuentra la configuración de las superficies de potencial proporcionando los datos necesarios, como son las coordenadas de los puntos con mayor y menor potencial, el color de su representación, en un arreglo.

24. Contenedor de dipolos:

Representa un tipo de objeto que se encarga de leer un archivo de configuración de dipolos y devuelve un vector de datos de dipolos asociados a los datos de un vector de partículas.

25. Contenedor de imagen:

Representa un tipo de objeto que contiene la imagen generada por un objeto de tipo cálculo de imagen, y que puede guardar un vector de datos de imagen en un archivo físico en un formato gráfico preseleccionado, por ejemplo BMP, GIF, RGB, JPG o Postscript.

26. Monitor de procesos:

Esta clase se encarga del monitoreo de los procesos que se están realizando en la máquina local, y en la máquina remota generando mensajes del estado actual de los procesos solicitados, identificándolos y desplegando información importante sobre su situación.

27. Proceso local:

Representa un tipo de objeto que se encarga de ejecutar cálculos locales, es decir, en la máquina donde se encuentra trabajando el sistema, identifica el tipo de cálculo y le proporciona los datos que requiere para su ejecución.

28. Proceso remoto:

Representa un tipo de objeto que se encarga de la ejecución de cálculos en una máquina remota, por lo que además de identificar el tipo de cálculo a ejecutarse y los datos que requiere, se encarga de la conexión para establecer la comunicación con el cálculo remoto y poder obtener los resultados necesarios en la visualización de la dinámica molecular.

29. Impresión de archivo de configuración:

Es la clase que se encarga de imprimir los datos contenidos en los archivos de configuración del sistema de dinámica molecular, ya sea en los archivos físicos, o en los objetos de tipo archivo de datos, tanto de partículas, hoyos, dipolos o potenciales, mostrando el formato usado en el sistema.

30. Impresión de imagen:

Es la clase que se encarga de la impresión de las imágenes generadas por la clase cálculo de imagen.

31. Puntero 3D:

Representa a un objeto que permite seleccionar a otros objetos desplazándose en las tres coordenadas del espacio dentro de la escena gráfica, para posteriormente realizar acciones sobre dichos objetos.

32. Editor molecular:

Representa a un editor que permite modificar las moléculas existentes en la escena gráfica, al agregar o quitar átomos y conexiones. además de permitir copiar, cortar, pegar y unir objetos, seleccionándolos de manera individual o en grupo, o modificar atributos como radio y color y permitiendo también deshacer cambios realizados en alguna acción previa.

33. Plano de corte:

Es un corte transversal en diagonal de extremo a extremo en cualquier dirección sobre el cubo o caja

34. Contenedor de conexiones:

Contiene los datos necesarios para dibujar las conexiones existentes entre los diferentes átomos en un momento determinado

35. Contenedor de rayos x:

Contiene los datos necesarios para generar las gráficas de rayos x

36. Contenedor de configuración de dinámica:

Contiene los datos necesarios para iniciar el cálculo de una dinámica molecular que originará los archivos de datos requeridos para la visualización.

37. Contenedor de inicia dinámica:

Contiene los parámetros iniciales que son también necesarios para el cálculo de la dinámica molecular.

9.1.1.6. Identificar atributos de objetos y relaciones

Pasos para identificar los posibles objetos y sus atributos.

- a) Distinguir los objetos de los atributos
- b) Distinguir entre los atributos de objetos y de relaciones
- c) El identificador del objeto es siempre un atributo implícito
- d) Eliminar atributos privados (de diseño)
- e) Eliminar atributos de detalle fino
- f) Localizar atributos discordantes (dividir clase)

9.1.1.6.1. Clases con atributos

1) Objeto gráfico (A)	2) Objeto estático (A)	3) Caja
<ul style="list-style-type: none"> • Color • Etiqueta • Origen x • Origen y • Origen z 	<ul style="list-style-type: none"> • Grosor de línea • Tipo de línea 	<ul style="list-style-type: none"> • Tipo de caja • Lados • Angulo a • Angulo b • Angulo g • Longitud de lado • Longitud de ancho • Longitud de alto
4) Eje	5) Objeto dinámico (A)	6) Esfera
<ul style="list-style-type: none"> • Punto X (destino) • Punto Y (destino) • Punto Z (destino) • Longitud 	<ul style="list-style-type: none"> • Calidad • Material • Selección (grupo o individual) • Transparencia • Slices (divisiones alrededor del eje z) • Stacks (divisiones a lo largo del eje z) • Radio 	<ul style="list-style-type: none"> • Molécula asociada
7) Dipolo	8) Átomo	9) Hoyo
<ul style="list-style-type: none"> • Partícula asociada • Excentricidad • Angulo 	<ul style="list-style-type: none"> • Tipo (de elemento) • Carga 	<ul style="list-style-type: none"> • Lista de vecinos
10) Conexión	11) Plano potencial	12) Objeto animado (A)
<ul style="list-style-type: none"> • Longitud • Punto x (destino) • Punto y (destino) • Punto z (destino) 	<ul style="list-style-type: none"> • Valor constante del plano • Número de partículas por plano • Máximo potencial • Mínimo potencial • Rango de potenciales • Indicador de plano a graficar • Alto • Ancho • Largo 	<ul style="list-style-type: none"> • Lista de datos por cuadro • Número de cuadros • Sentido de animación (avance, pausa, retroceso) • número de cuadro actual

13) Sistema de cavidades (hoyos)	14) Molécula	15) Superficie potencial
<ul style="list-style-type: none"> • Molécula asociada • Número de hoyos • Lista de hoyos 	<ul style="list-style-type: none"> • Nombre • Descripción • Lista de átomos • Lista de dipolos • Lista de conexiones • Número de conexiones • Número de átomos • Número de dipolos 	<ul style="list-style-type: none"> • Lista de planos potenciales • Número de planos • Estructura cristalina asociada
16) Indicador de selección	17) Modificador de propiedades	18) Escena gráfica
<ul style="list-style-type: none"> • Tipo de objeto seleccionado • Lista de etiquetas de objetos seleccionados • Tipo de selección 	<ul style="list-style-type: none"> • Lista de etiquetas de objetos seleccionados • Tipo de selección • Propiedad a cambiar • Vector de historia 	<ul style="list-style-type: none"> • Fondo • Ejes • Estructura cristalina • Puntero 3D
19) Estructura cristalina	20) Celda unitaria	21) Proceso cálculo (A)
<ul style="list-style-type: none"> • Nombre • Lista de celdas unitarias • Número de celdas unitarias • Campo potencial 	<ul style="list-style-type: none"> • Número de moléculas • Lista de moléculas • Caja • Sistema de cavidades 	<ul style="list-style-type: none"> • Nombre • Tipo
22) Cálculo generado por partículas (A)	23) Cálculo local (A)	24) Cálculo remoto (A)
<ul style="list-style-type: none"> • Vector de datos de partículas • Número de partículas 	<ul style="list-style-type: none"> • Proceso local asociado 	<ul style="list-style-type: none"> • proceso remoto asociado
25) Cálculo de imagen	26) Contenedor de datos (A)	27) Contenedor de partículas
<ul style="list-style-type: none"> • Vector de datos de imagen • Tipo de imagen 	<ul style="list-style-type: none"> • Nombre del archivo • Tipo del archivo • Vector de datos • Número de elementos 	<ul style="list-style-type: none"> • Lado x • Lado y • Lado z • Ángulo x • Ángulo y • Ángulo z • Número de cuadros • Número de partículas • Vector de datos <ul style="list-style-type: none"> ○ Etiqueta ○ X ○ Y ○ Z ○ Carga ○ Tipo

28) Contenedor de hoyos <ul style="list-style-type: none"> • Número de cuadros • Número de hoyos • Archivo de partículas Asociado • Vector de datos <ul style="list-style-type: none"> ○ Etiqueta ○ X ○ Y ○ Z ○ Radio 	29) Contenedor de potencial <ul style="list-style-type: none"> • Número de cuadros • Número de potenciales • Archivo de partículas asociado • Máximo valor de potencial • Mínimo valor de potencial • Vector de datos <ul style="list-style-type: none"> ○ Etiqueta. ○ T_plan. ○ Número de partículas por plano. ○ Máximo Potencial. ○ Mínimo Potencial ○ Tsup. ○ Rango de Potenciales. ○ Alto. ○ Ancho. ○ Largo. 	30) Contenedor de dipolos <ul style="list-style-type: none"> • Número de cuadros • Número de dipolos • Archivo de partículas asociado • Vector de datos <ul style="list-style-type: none"> ○ Etiqueta ○ X1 ○ Y1 ○ Z1 ○ X2 ○ Y2 ○ Z2 ○ Carga
31) Contenedor de imagen <ul style="list-style-type: none"> • Formato de imagen • Calidad de imagen • Largo (píxeles) • Ancho(píxeles) 	32) Monitor de procesos <ul style="list-style-type: none"> • Acción sobre proceso de cálculo • Tabla de procesos • Tipo de mensaje. • Número de procesos locales • Lista de procesos locales • Número de procesos remotos • Lista de proceso remotos 	33) Proceso (A) <ul style="list-style-type: none"> • Estado del proceso • Nombre del proceso • Tipo de proceso • Fecha de inicio • Hora de inicio • Fecha de terminación • Hora de terminación • Número de proceso
34) Proceso local <ul style="list-style-type: none"> • Vector de datos de entrada • Vector de datos de salida • Tipo de cálculo • Tipo de proceso • Número de elementos 	35) Proceso remoto <ul style="list-style-type: none"> • Dirección IP • Login • Password • Puerto • Nombre del archivo de entrada • Nombre del archivo de salida • Tipo de cálculo • Tipo de proceso • Número de elementos 	36) Impresiones (A) <ul style="list-style-type: none"> • Nombre • Vector de datos • Tipo de dato a imprimir • Identificador de impresión • Calidad de Impresión
37) Impresión de archivo de configuración	38) Impresión de imagen	39) Puntero 3D
<ul style="list-style-type: none"> • Tipo de archivo de configuración 	<ul style="list-style-type: none"> • Formato de imagen 	<ul style="list-style-type: none"> • Coordenada en x • Coordenada en y • Coordenada en z • Tipo • Color
40) Editor molecular <ul style="list-style-type: none"> • Vector de partículas • Tipo de selección • Tipo de acción • Molécula • Vector de historia 	41) Cálculo de magnitudes y ángulos <ul style="list-style-type: none"> • Tipo (m o a) • Lista de objetos seleccionados 	42) Cálculo de dinámica <ul style="list-style-type: none"> • Vector de datos • Proceso remoto asociado

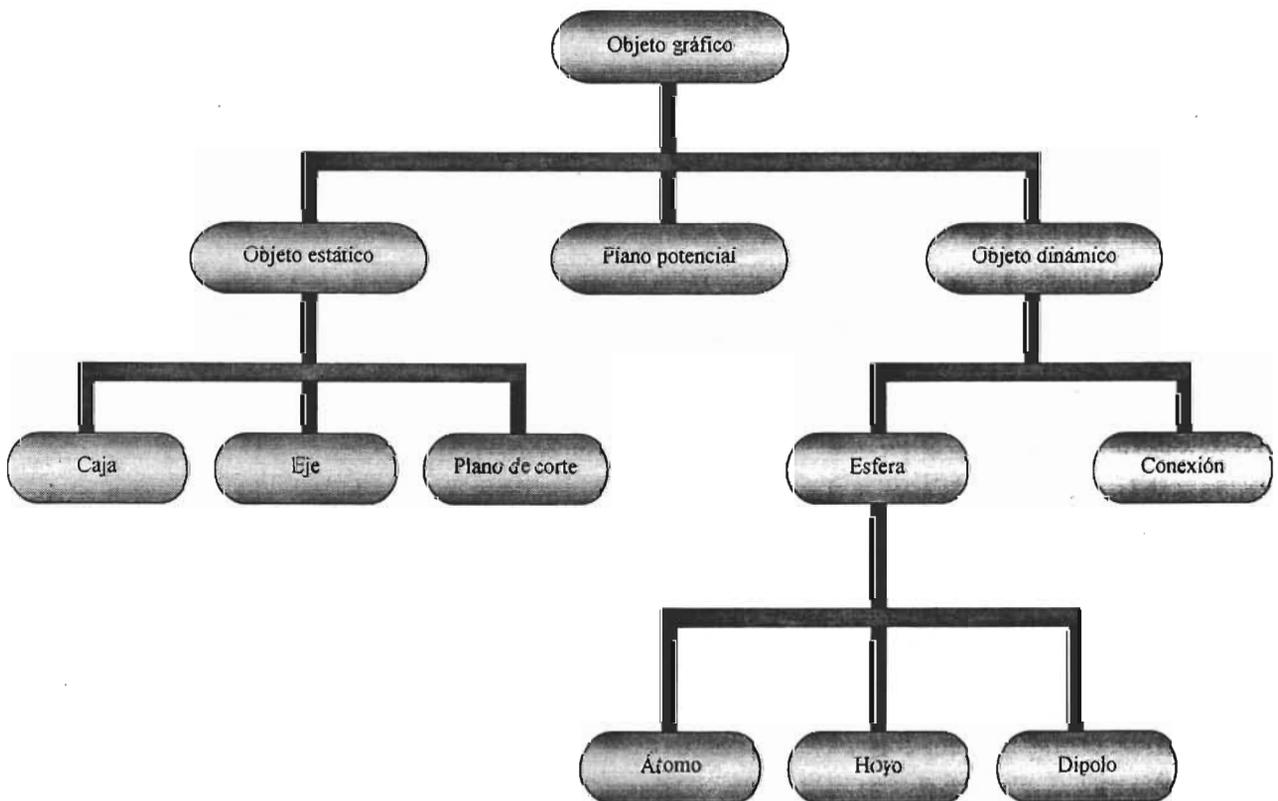
<p>43) Plano de corte</p> <ul style="list-style-type: none"> • Valor constante del plano • Número de partículas por plano • Indicador de plano a graficar • Alto • Ancho • Largo 	<p>44) Contenedor de conexiones</p> <ul style="list-style-type: none"> • Vector de datos <ul style="list-style-type: none"> ○ Etiqueta ○ X1 ○ Y1 ○ Z1 ○ X2 ○ Y2 ○ Z2 	<p>45) Contenedor de rayos</p> <ul style="list-style-type: none"> • Número de rayos • Archivo de partículas Asociado • Vector de datos <ul style="list-style-type: none"> ○ Etiqueta ○ X ○ Y
<p>46) Contenedor de configuración de dinámica t</p> <ul style="list-style-type: none"> • Densidad inicial • Temperatura inicial • Número de partículas • Vector de datos (Lista de configuración al tiempo t) <ul style="list-style-type: none"> ○ Número de partícula ○ RX(I) (Posición) ○ RY(I) ○ RZ(I) ○ VX(I) (Velocidad) ○ VY(I) ○ VZ(I) ○ ITYP(I) (Tipo partícula) 	<p>47) Contenedor de configuración de dinámica t -dt</p> <ul style="list-style-type: none"> • Densidad • Temperatura • RX(I) • RY(I) • RZ(I): • RX0 • RY0 • RZ0 • Q(I) • VX(I) • VY(I) • VZ(I) • ITYP(I) • V_itcan • V_itcan1 • Sedacan • Sedacan0 • Tinst • Eptpn • Eptn • Vol • Eptpn0 • Eptn0 • Vol0 • Pinst • Dens0 • Dens 	<p>48) Contenedor de inicia dinámica</p> <ul style="list-style-type: none"> • Vector de datos KINIT <ul style="list-style-type: none"> ○ ITMAX ○ IRE ○ IMV ○ IAN ○ IAF ○ TFIN ○ INAP ○ IFCHD ○ FDENS ○ IQST • Vector de datos B <ul style="list-style-type: none"> ○ IZERO ○ IBKUP ○ I3BF ○ ITCIN ○ IRENO ○ IINT ○ IRECH ○ IVEL ○ DT • Vector de datos C <ul style="list-style-type: none"> ○ JUMP ○ ITCAN • Vector de datos D <ul style="list-style-type: none"> ○ TQUE ○ PEXT ○ ICONI ○ ICONF • Vector de datos E <ul style="list-style-type: none"> ○ NTIPOS ○ Ptle ○ Rmtpcl(Ki) ○ Nformls • Vector de datos F <ul style="list-style-type: none"> ○ PTNCL ○ Npmax ○ RCP • Vector de datos G <ul style="list-style-type: none"> ○ NFOCOS ○ KNK

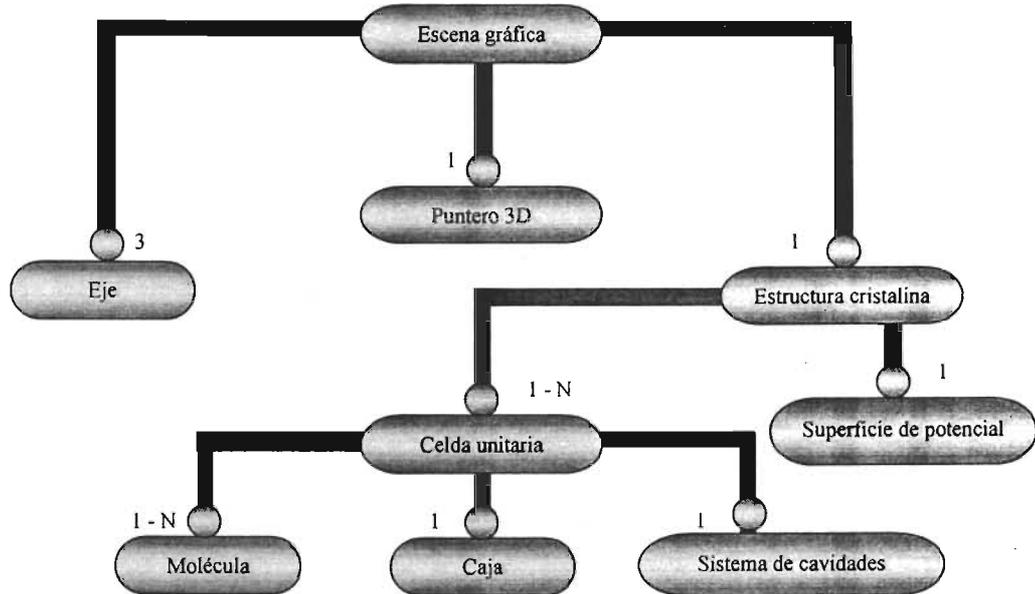
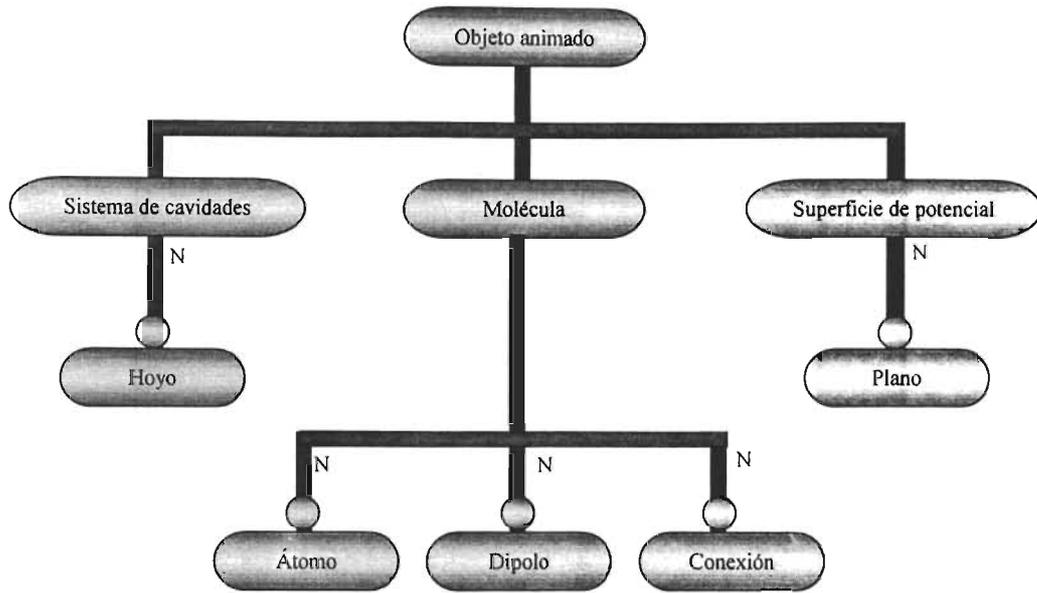
		<ul style="list-style-type: none">○ Taut (knf)○ Sedacan (knf)○ TauPY○ Eptpn● Vector de datos H:<ul style="list-style-type: none">○ lrdfnw○ mc (10)○ coef(4)
--	--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

9.1.1.7. Añadir herencia, comprobar casos de uso y modularizar

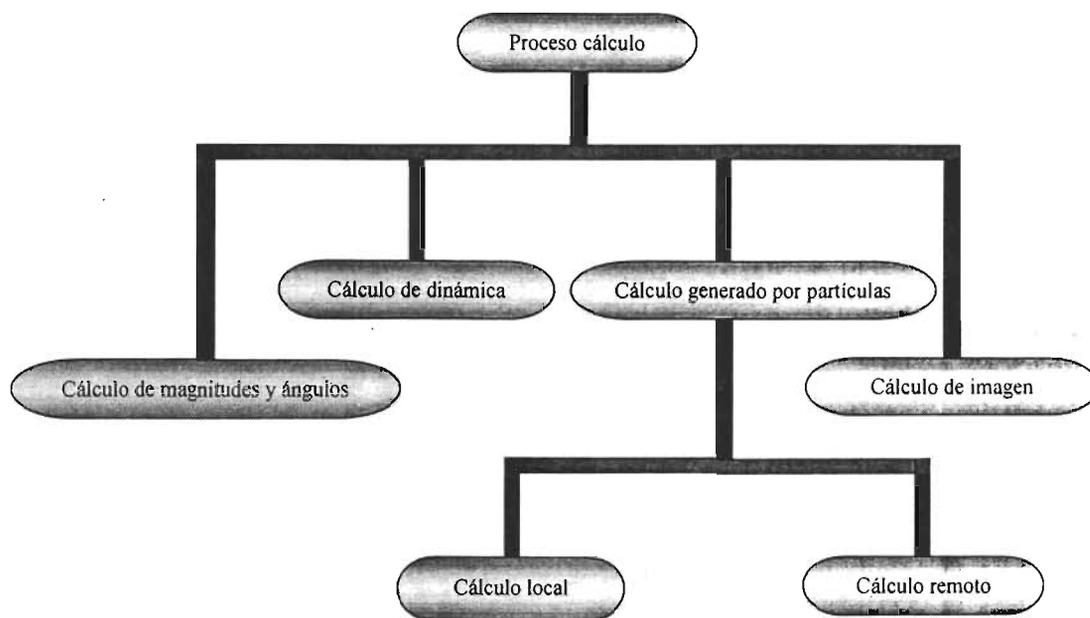
- 1) **Clases gráficas:** Objeto gráfico, objeto estático, objeto dinámico, plano potencial, plano de corte, caja, eje, plano de corte, esfera, átomo, hoyo, dipolo, conexión, objeto animado, sistema de cavidades, molécula, superficie potencial, escena gráfica, estructura cristalina, celda unitaria.
- 2) **Clases de cálculos:** Proceso cálculo, cálculo de imagen, cálculo de magnitudes y ángulos, cálculo generado por partículas, cálculo local, cálculo remoto, cálculo de dinámica.
- 3) **Clases de contenedores:** Contenedor de datos, contenedor de hoyos, contenedor de imagen, contenedor de potenciales, contenedor de dipolos, contenedor de conexiones, contenedor de rayos x, contenedor de configuración de dinámica t, configuración de dinámica t - dt, contenedor de inicia dinámica.
- 4) **Clases de monitor de procesos:** Monitor de procesos, proceso, proceso local, proceso remoto.
- 5) **Clases de impresiones:** Impresiones, impresión de archivo de configuración, impresión de imagen.
- 6) **Clases de editor molecular:** Editor molecular, indicador de selección, modificador de propiedades, puntero 3D.

9.1.1.7.1. Jerarquía de clases gráficas

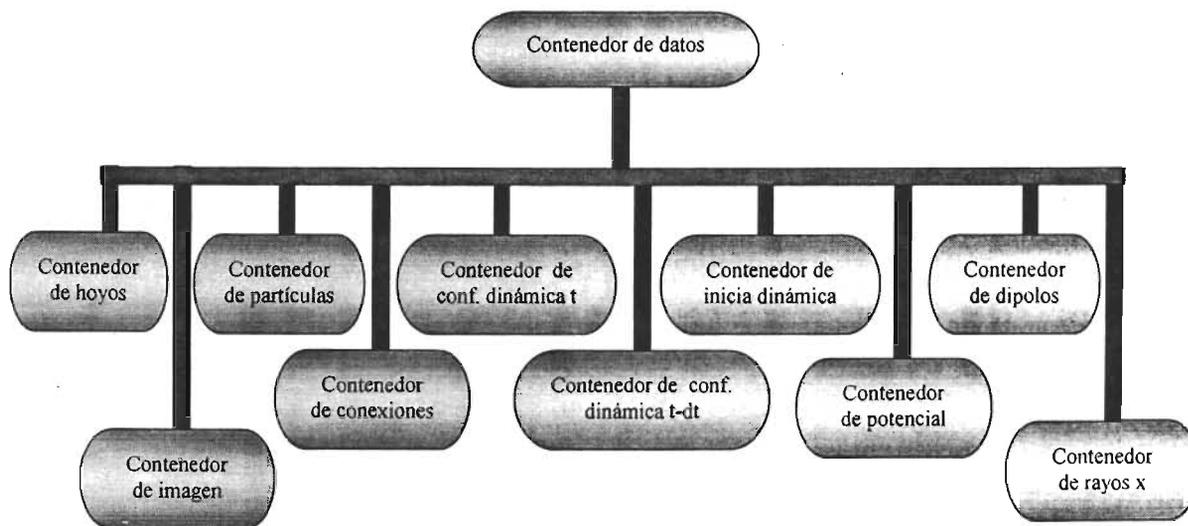




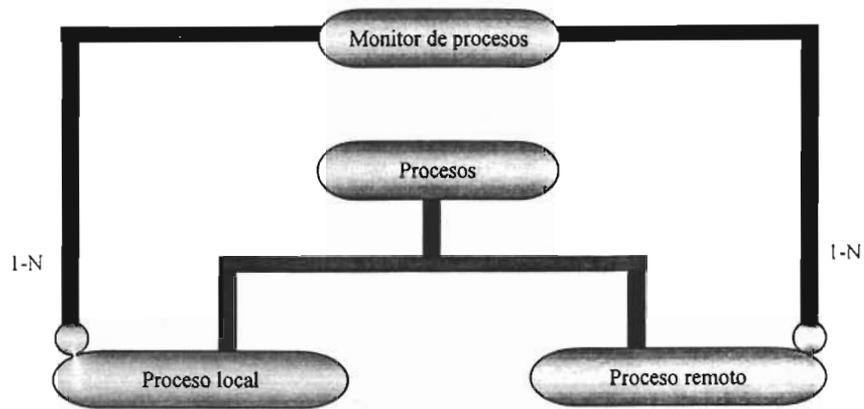
9.1.1.7.2. Jerarquía de clases de cálculos



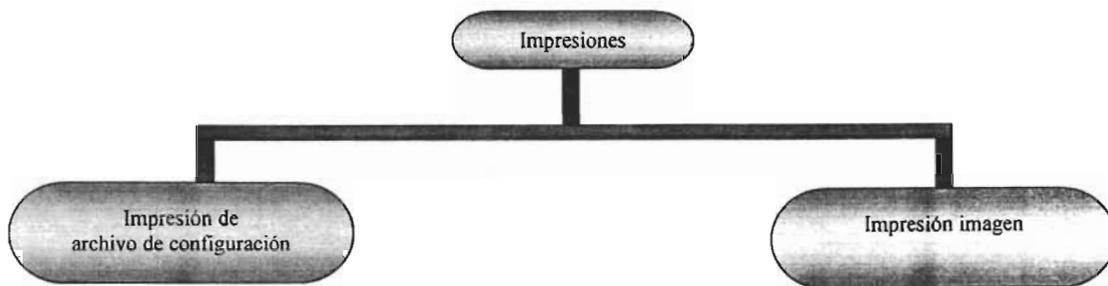
9.1.1.7.3. Jerarquía de clases contenedores



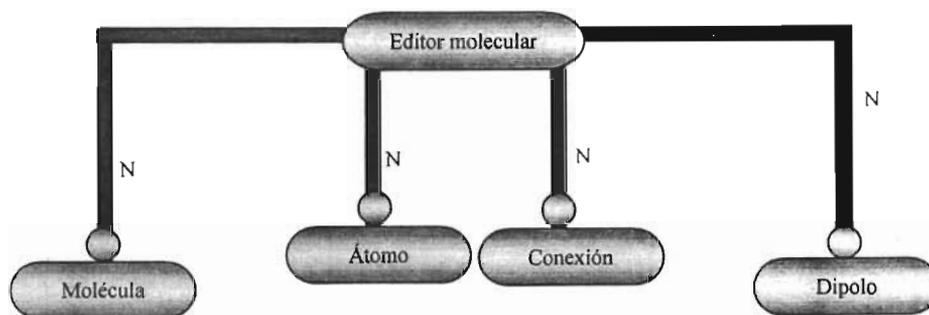
9.1.1.7.4. Jerarquía de clases de monitor de procesos



9.1.1.7.5. Jerarquía de clases de impresiones



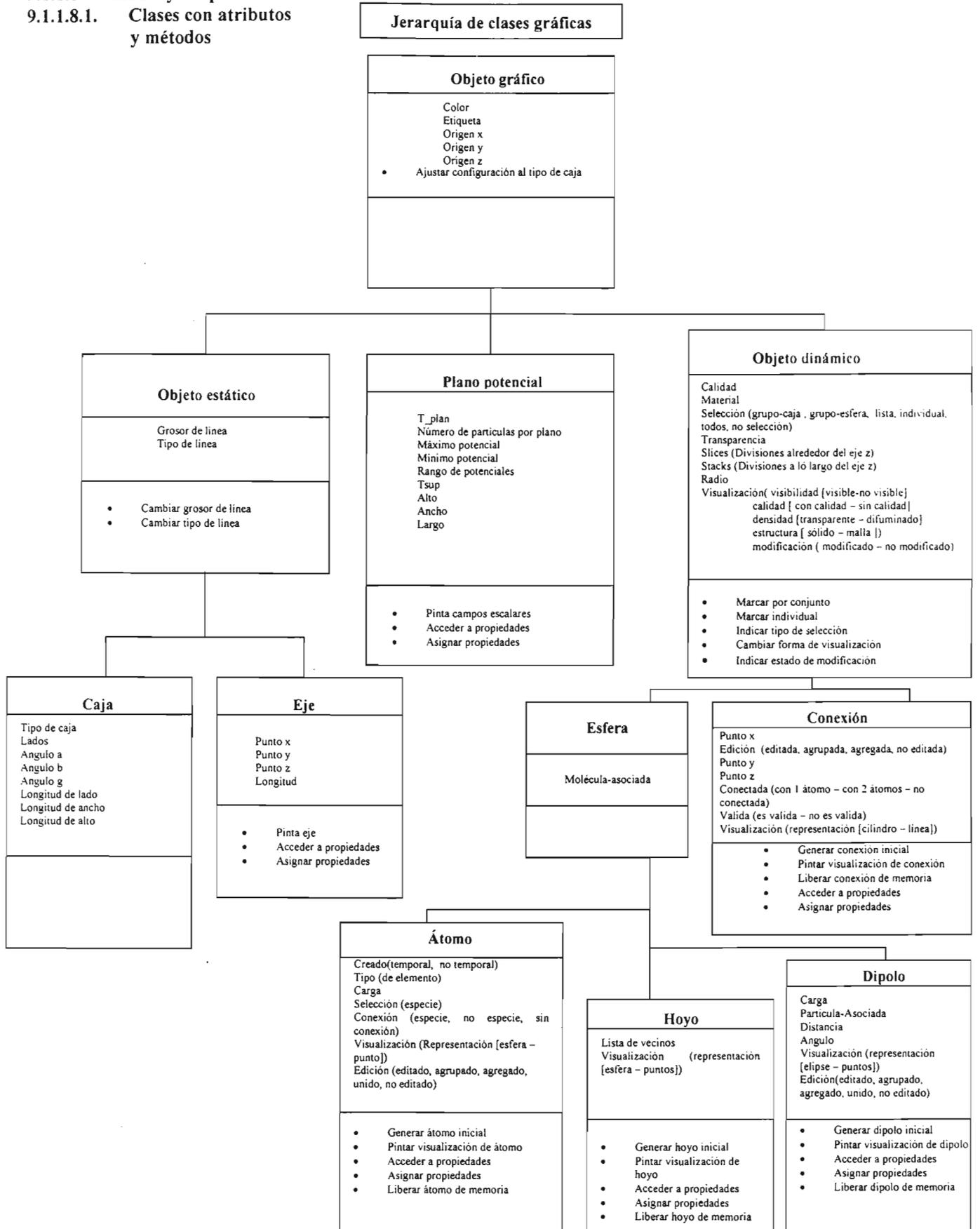
9.1.1.7.6. Jerarquía de clases de editor molecular



9.1.1.8. Añadir y simplificar métodos

9.1.1.8.1. Clases con atributos y métodos

Jerarquía de clases gráficas



Jerarquía de clases gráficas (continuación)

Objeto animado

Lista de datos por cuadro
 Número de cuadros
 Sentido de animación (avance, pausa, retroceso)
 Número de cuadro actual
 Visualización(visibilidad [visible-no visible], calidad [con calidad – sin calidad]
 estructura [sólido – malla])
 Modificación (modificado – no modificado)
 Selección(seleccionado – no seleccionado)

- Detener animación en cualquier cuadro
- Invertir sentido de la animación
- Encender y Apagar
- Recibir estado de animación
- Cambiar estado de animación
- Proporcionar estado de animación actual
- Indicar tipo de selección

Sistema de cavidades

Molécula asociada
 Número de hoyos
 Lista de hoyos (vector)

Hoyo

Visualización(representación [esfera – punto],
 Densidad [transparente – difuminado])

- Recibir vector de hoyos
- Generar sistema de cavidades inicial
- Pintar visualización de cavidades
- Acceder a propiedades
- Asignar propiedades
- Devolver vector de hoyos
- Liberar sistema de cavidades de memoria

Molécula

Nombre
 Descripción
 Lista de átomos (vector)

Átomo

Lista de dipolos (vector)

Dipolo

Lista de conexiones (vector)

Conexión

Número de conexiones
 Número de átomos
 Número de dipolos
 Conexión(por especie–no por especie–sin conexión)
 Edición(editada, agregada, unida, no editada)
 Visualización(representación [esferas- cilindros – puntos - líneas],
 Densidad [transparente – difuminado])

- Generar molécula inicial
- Pintar visualización de molécula
- Fraccionar molécula
- Mover molécula
- Conectar átomos
 - Lee tabla de conectividad
 - Valida conexiones
 - Establecer tipo de conexión
- Modificar lista de átomos
- Modificar lista de dipolos
- Modificar lista de conexiones
- Unir molécula
- Recibir vector de partículas
- Recibir vector de dipolos
- Recibir vector de conexiones
- Devolver vector de partículas
- Devolver vector de dipolos
- Devolver vector de conexiones
- Acceder a propiedades
- Cambiar propiedades
- Liberar molécula de memoria

Superficie potencial

Lista de planos
 Potenciales

Plano potencial

Número de planos
 Estructura cristalina asociada
 Visualización(representación [superficie – plano]
 Selección(plano – superficie – no seleccionada)

- Recibir vector de potenciales
- Generar superficie potencial inicial
- Pintar visualización de superficie Potencia
- Acceder a propiedades
- Asignar propiedades
- Liberar superficie potencial de memoria

**ESTA TESIS NO SALE
 DE LA BIBLIOTECA**

Jerarquía de clases gráficas (continuación)

Indicador de selección

Tipo de objeto seleccionado
 Modo de selección
 Lista de etiquetas de objetos seleccionados

- Marcar objetos seleccionados
- Identificar objetos seleccionados
- Obtener lista de objetos seleccionados
- Obtener lista de etiquetas de objetos seleccionados
- Regresar lista de objetos seleccionados
- Regresar lista de etiquetas de objetos seleccionados

Modificador de propiedades

Lista de etiquetas de objetos seleccionados
 Tipo de selección
 Propiedad a cambiar
 Vector de historia

- Identificar propiedad y objeto
- Cambiar propiedad
- Deshacer cambios
- Rehacer cambios
- Notificar a editor

Estructura cristalina

Nombre
 Lista de celdas (unitarias)

Celda

Número de celdas unitarias
 Lista de planos potenciales

Superficie potencial

- Multiplicar celda
- Crear estructura cristalina
- Pintar estructura cristalina
- Cambiar factores de multiplicación
- Liberar estructura de la memoria

Escena gráfica

Fondo
 Ejes (3)

Eje

Estructura cristalina

Puntero 3D

Transformación espacial
 (en rotación –en translación – en escalamiento)
 Perspectiva (activada- desactivada)
 Modificación(molécula- ejes – puntero 3D –
 fondo – celda)

- Generar escena gráfica inicial
- Modificar objetos de escena gráfica
 - Modificar molécula
 - Modificar ejes
 - Modificar puntero 3D
 - Modificar fondo
 - Modificar celda
- Deshacer cambios hechos en la escena
- Rehacer cambios
- Cambiar perspectiva
- Asignar estado de animación
- Cambiar transformación espacial
 - Rotación
 - Traslación
 - Escalamiento
- Visualizar ejes
- Visualizar estructura cristalina
- Visualizar puntero 3D
- Liberar escena gráfica de la memoria

Celda

Número de moléculas
 Lista de moléculas

Molécula

Caja

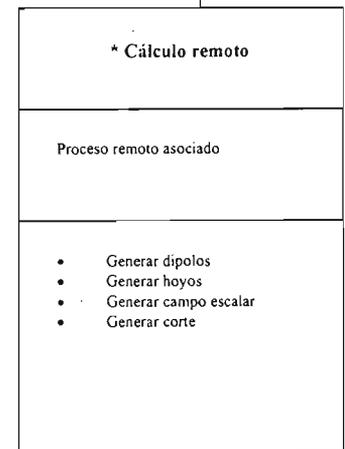
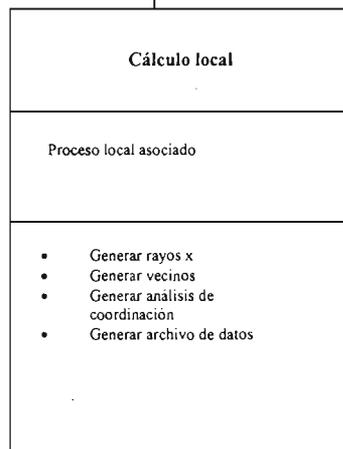
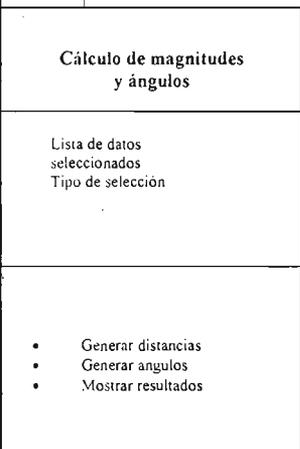
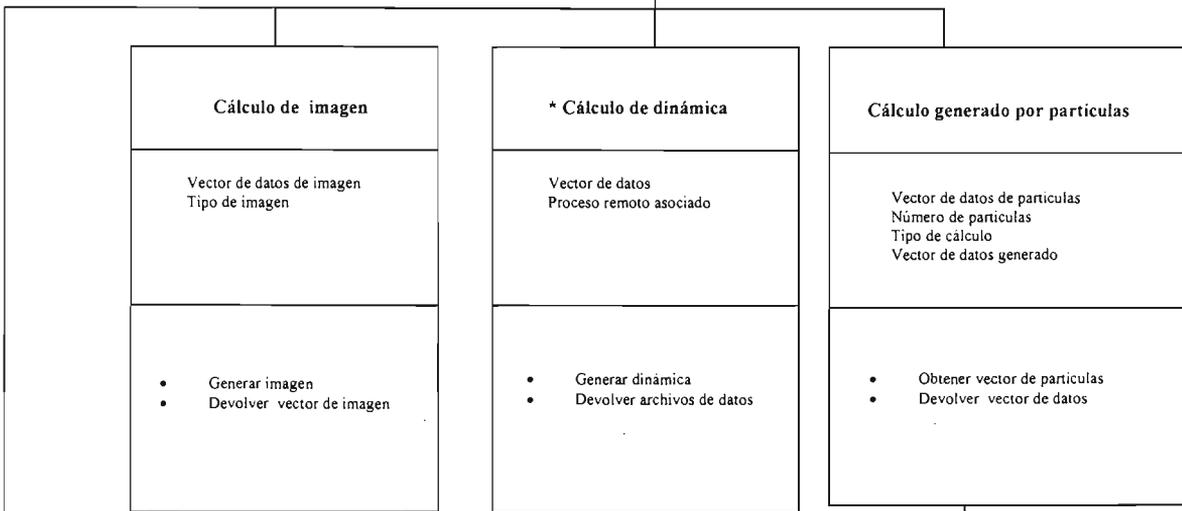
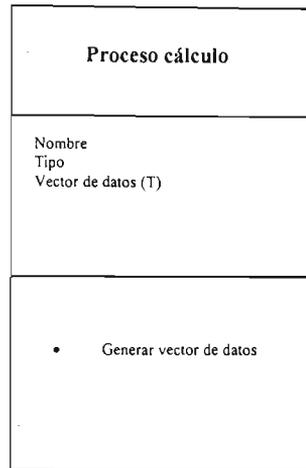
Sistema de cavidades

Visualización (celda -
 celda unitaria -
 estructura cristalina)

- Generar celda inicial
- Pintar celda
- Liberar celda de la memoria

Nota : Una celda es unitaria cuando se comprueba que es una celda base, sólo con celdas unitarias se puede generar una estructura cristalina.

Jerarquía de clases de cálculos



* Nota: Estos son programas externos al sistema

Jerarquía de clases de contenedores

Contenedor de datos

Nombre del archivo
 Tipo del archivo
 Vector de datos (T)
 Número de elementos

- Verificar tipo de archivo de datos
- Verificar nombre de archivo de datos
- Verificar formato de archivo de datos
- Leer archivo de datos
- Actualizar vector
- Traducir datos
- Renombrar archivo de datos
- Guardar archivo de datos
- Cerrar archivo de datos
- Generar vector de datos de configuración actual
- Devolver vector de datos
- Devolver archivo de datos
- Notificar mensajes de error

1

Contenedor de hoyos

Número de cuadros
 Número de hoyos
 Archivo de partículas asociado
 Vector de datos

Etiqueta
 X
 Y
 Z
 Radio

- Verificar tipo de archivo de hoyos
- Verificar nombre de archivo de hoyos
- Verificar formato de archivo de hoyos
- Leer archivo de hoyos
- Actualizar vector
- Guardar archivo de hoyos
- Generar vector de configuración actual de hoyos
- Devolver vector de hoyos
- Devolver archivo de hoyos

Contenedor de imagen

Formato de imagen
 Calidad de imagen
 Largo (pixeles)
 Ancho (pixeles)

- Actualizar vector
- Generar vector de configuración actual de imagen
- Guardar archivo de imagen
- Devolver vector de imagen
- Devolver archivo de imagen

Contenedor de dipolos

Número de cuadros
 Número de dipolos
 Archivo de partículas asociado
 Vector de datos

Etiqueta
 X1
 Y1
 Z1
 X2
 Y2
 Z2
 Carga

- Verificar tipo de archivo de dipolos
- Verificar nombre de archivo de dipolos
- Verificar formato de archivo de dipolos
- Leer archivo de dipolos
- Actualizar vector
- Guardar archivo de dipolos
- Generar vector de configuración actual de dipolos
- Devolver vector de dipolos
- Devolver archivo de dipolos

Contenedor de partículas

Lado x
 Lado y
 Lado z
 Ángulo x
 Ángulo y
 Ángulo z
 Número de cuadros
 Número de partículas
 Vector de datos

Etiqueta
 X
 Y
 Z
 Carga
 Tipo

- Verificar tipo de archivo de partículas
- Verificar nombre de archivo de partículas
- Verificar formato de archivo de partículas
- Leer archivo de partículas
- Actualizar vector
- Traducir datos de partículas
- Guardar archivo de partículas
- Generar vector de configuración actual de partículas
- Devolver vector de partículas
- Devolver archivo de partículas
- Proporcionar tipo de caja

Contenedor de potencial

Número de cuadros
 Número de potenciales
 Archivo de partículas asociado
 Potmax
 Potmin
 Vector de datos

Etiqueta
 T_plan
 Número de partículas por plano
 Máximo potencial
 Mínimo potencial
 Tsup
 Rango de potenciales
 Alto
 Ancho
 Largo

- Verificar tipo de archivo de potencial
- Verificar nombre de archivo de potencial
- Verificar formato de archivo de potencial
- Leer archivo de potencial
- Actualizar vector
- Guardar archivo de potencial
- Generar vector de configuración actual de potencial
- Devolver vector de potencial
- Devolver Archivo de potencial

Jerarquía de clases de contenedores (continuación)

I

Contenedor de conexiones

Vector de datos

Etiqueta
X1
Y1
Z1
X2
Y2
Z2

- Verificar tipo de archivo de conexiones
- Verificar nombre de archivo de conexiones
- Verificar formato de archivo de conexiones
- Leer archivo de conexiones
- Actualizar vector
- Guardar archivo de conexiones
- Generar vector de configuración actual de conexiones
- Devolver vector de conexiones
- Devolver archivo de conexiones

Contenedor de inicia dinámica

Vector de datos A

KINIT
ITMAX
IRE
IMV
IAN
IAF
TFIN
INAP
IFCHD
FDENS
IQST

Vector de datos B

IZERO
IBKLP
IBBF
ITCIN
IRENO
IINT
IRECH
IVEL
DT

Vector de datos C

JUMP
ITCAN

Vector de datos D

TQUE
PEXT
ICONI
ICONF

Vector de datos E

NTIPOS
Ptele
Rmpcl(Ki)
Nformls

Vector de datos F

PTNCL
Npmax
RCP

Vector de datos G

NFOCOS
KNK
Taut (knf)
Sedacan (knf)
TauPY
Eptpn

Vector de datos H

trdfnw
mc (10)
coeff(4)

- Verificar tipo de archivo de inicia dinamica
- Verificar nombre de archivo de inicia dinamica
- Verificar formato de archivo de inicia dinamica
- Leer archivo de inicia dinamica
- Actualizar vector
- Guardar archivo de inicia dinamica
- Generar vector de configuración actual de inicia dinamica.
- Devolver vector de inicia dinamica
- Devolver archivo de inicia dinamica

Contenedor de rayos x

Numero de rayos

Archivo de particulas asociado

Vector de datos

Etiqueta
X
Y

- Verificar tipo de archivo de rayos x
- Verificar nombre de archivo de rayos x
- Verificar formato de archivo de rayos x
- Leer archivo de rayos x
- Actualizar vector
- Guardar archivo de rayos x
- Generar vector de configuración actual de rayos x
- Devolver vector de rayos x

Contenedor de configuración de dinámica t

Densidad inicial

Temperatura inicial

Numero de particulas

Vector de datos (Lista de conf. al tiempo t)

Numero de particula
RX(I) (Posicion)
RY(I)
RZ(I)
VX(I) (Velocidad)
VY(I)
VZ(I)
ITYP(I) (Tipo particula)

- Verificar tipo de archivo de conf. de dinamica t
- Verificar nombre de archivo de conf. de dinamica t
- Verificar formato de archivo de conf. de dinamica t
- Leer archivo de conf. de dinamica t
- Actualizar vector
- Guardar archivo de conf. de dinamica t
- Generar vector actual de conf. de dinamica t
- Devolver vector de conf. de dinamica t
- Devolver archivo de conf. de dinamica t

Contenedor de configuración de dinámica t-dt

Densidad inicial

Temperatura inicial

RN(I)

RY(I)

RZ(I)

RX0

RY0

RZ0

Q(I)

VX(I)

VY(I)

VZ(I)

ITYP(I)

V_itcan

V_itcan1

Sedacan

Sedacan0

Tinst

Eptpn

Vol

Eptpn0

Eptn0

Vol0

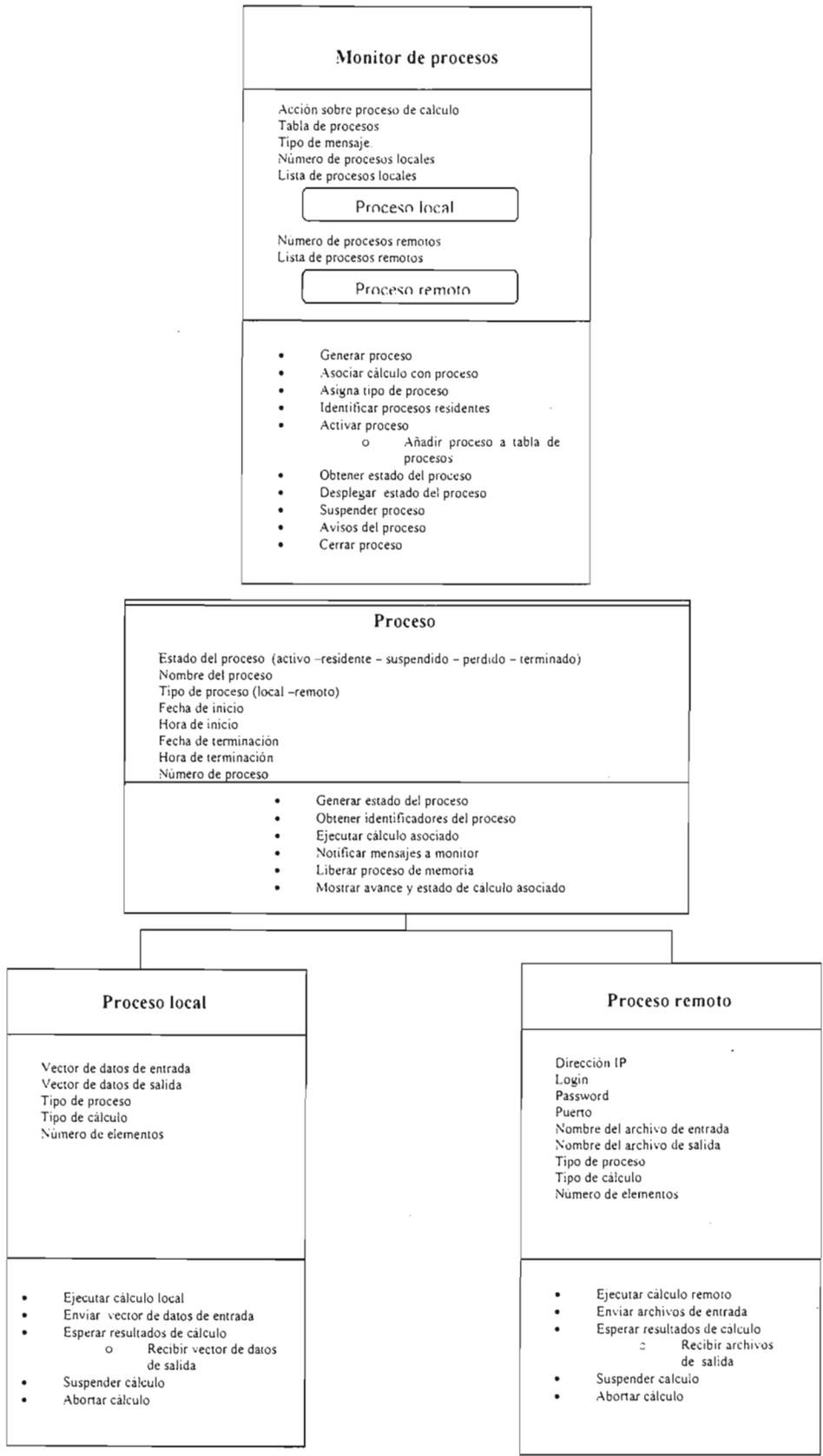
Pinst

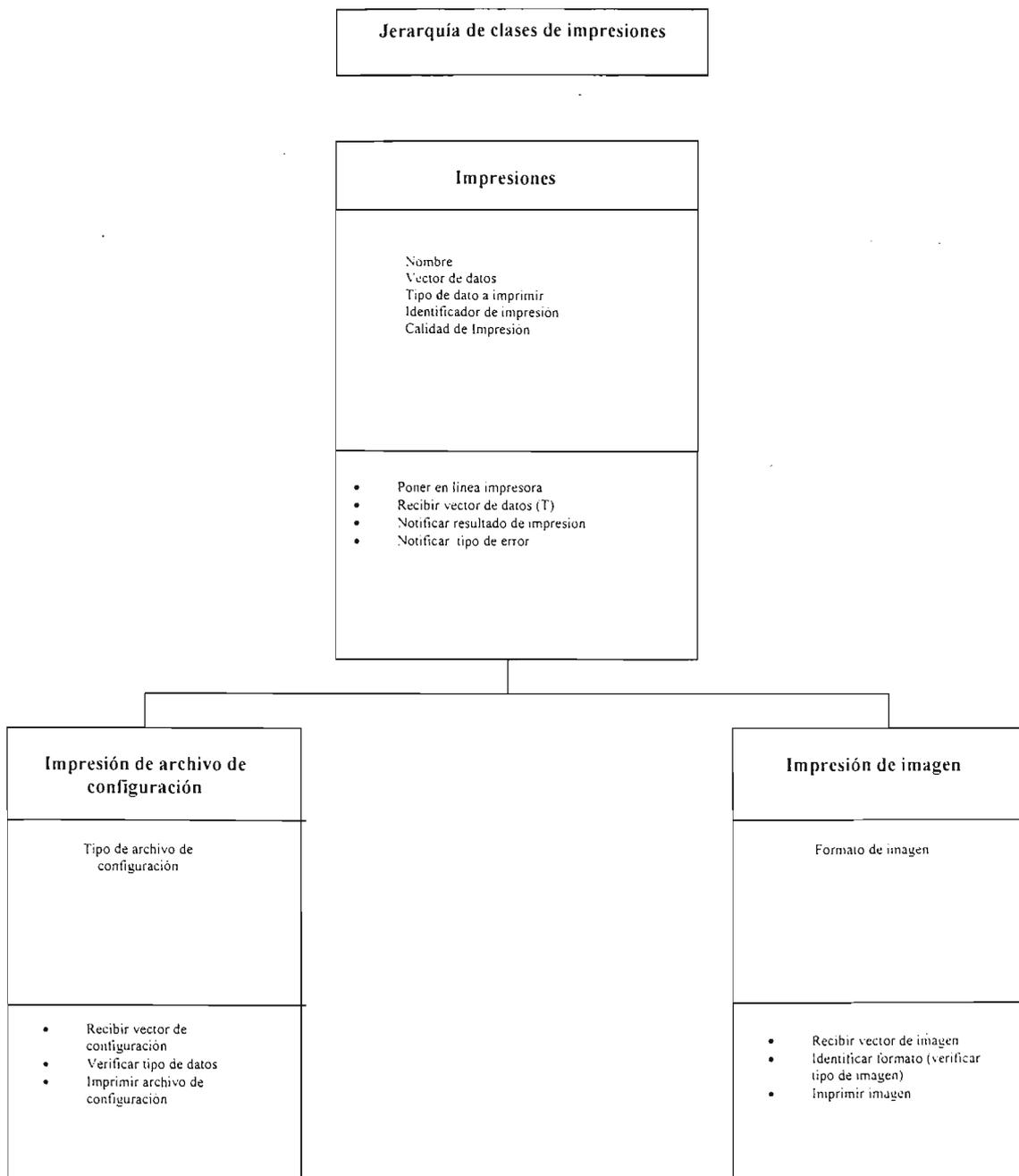
Dens0

Dens

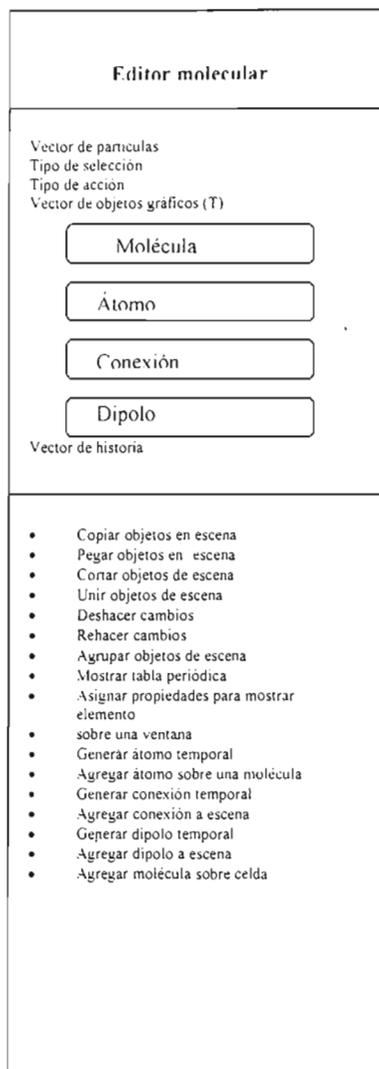
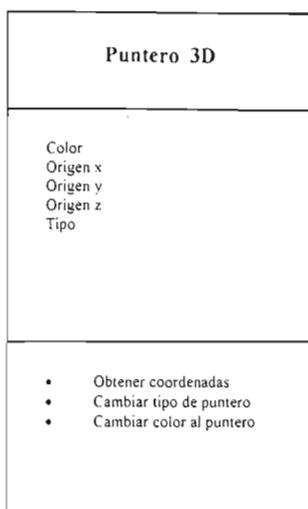
- Verificar tipo de archivo de conf. de dinamica t-dt
- Verificar nombre de archivo de conf. de dinamica t-dt
- Verificar formato de archivo de conf. de dinamica t-dt
- Leer archivo de conf. de dinamica t-dt
- Actualizar vector t-dt
- Guardar archivo de conf. de dinamica t-dt
- Generar vector actual de conf. de dinamica t-dt
- Devolver vector de conf. de dinamica t-dt
- Devolver archivo de conf. de dinamica t-dt

Jerarquía de clases de monitor de procesos

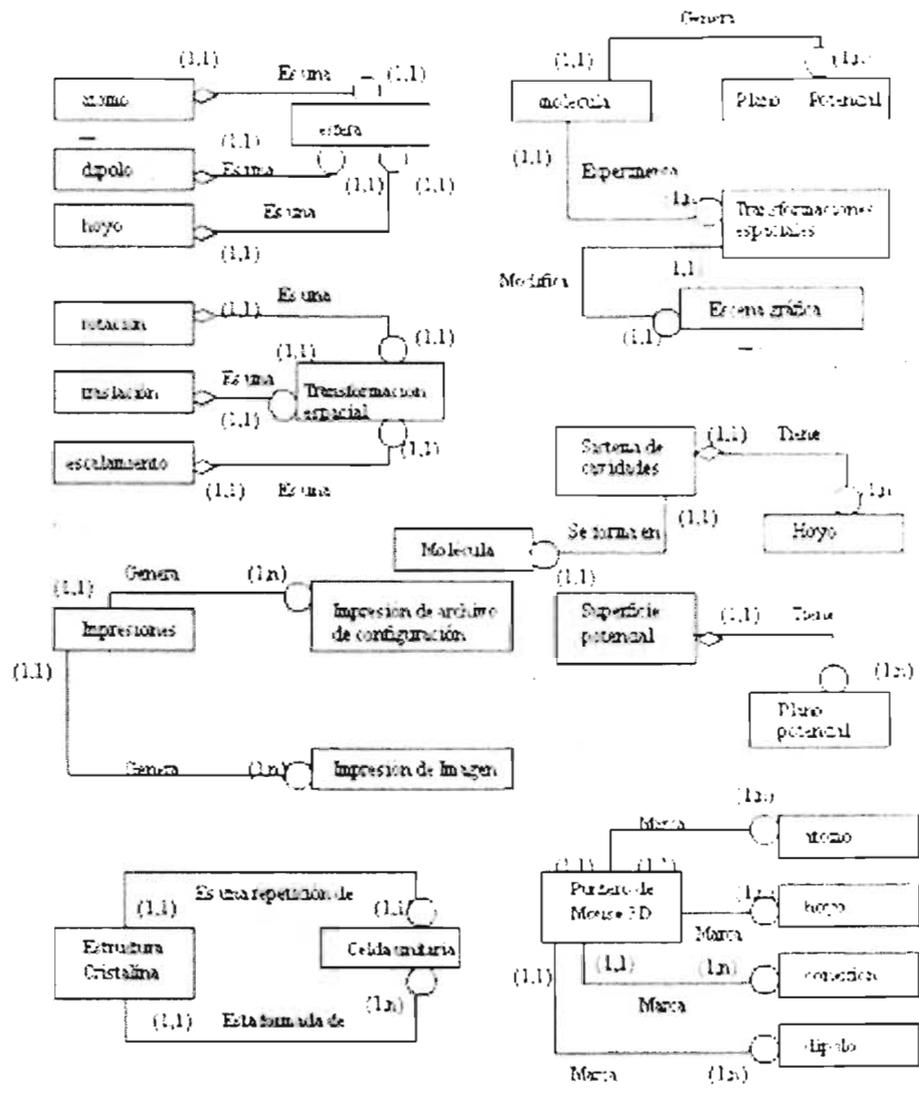


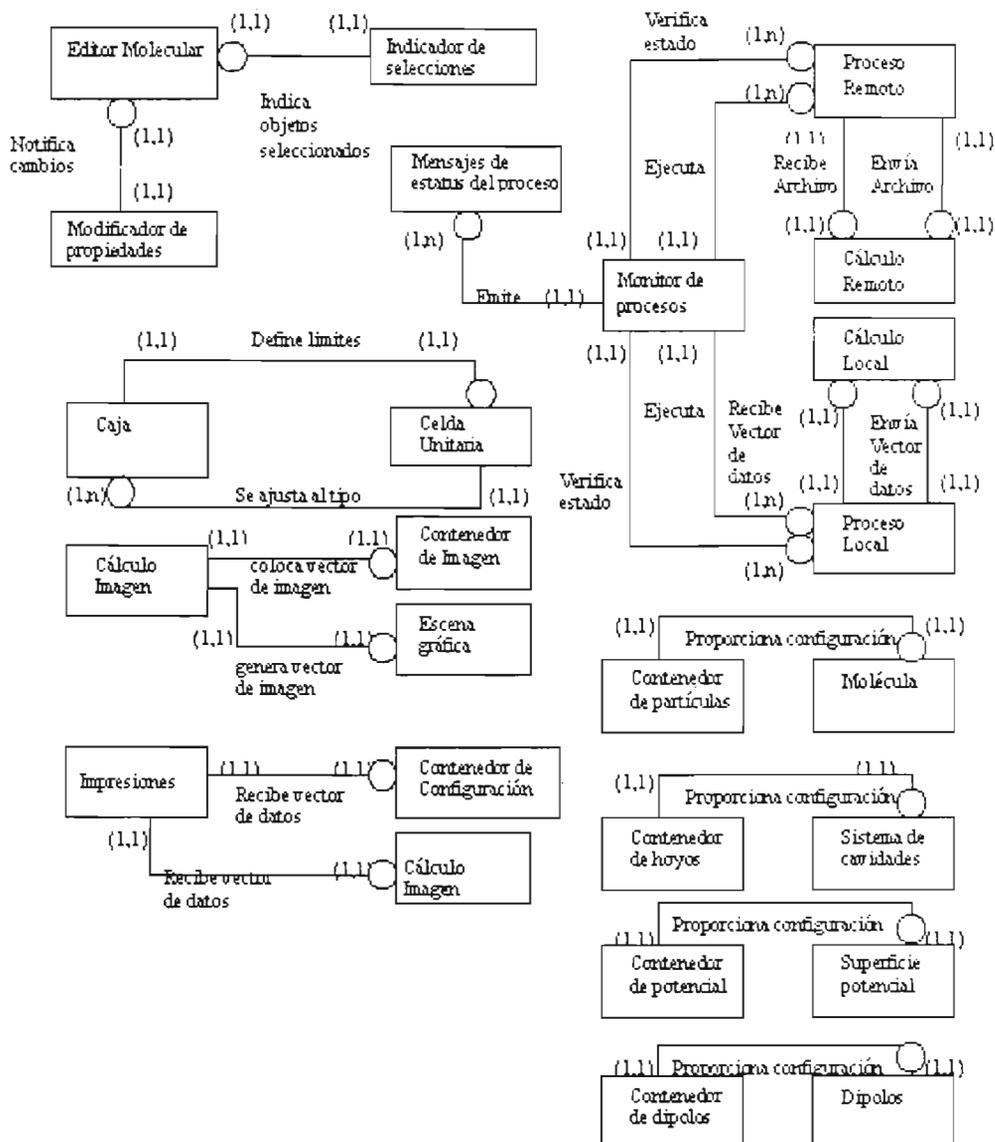


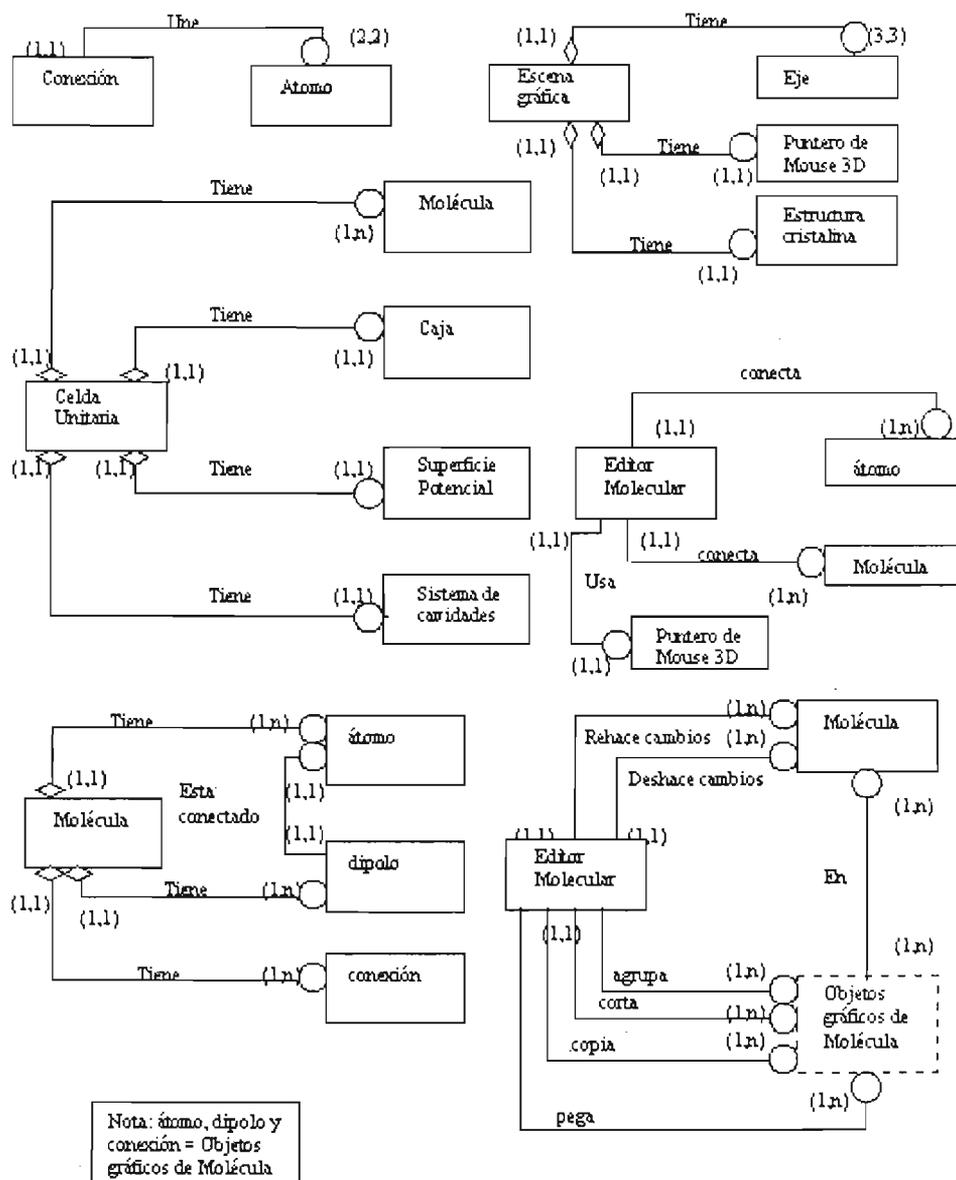
Jerarquía de clases de editor molecular



9.1.1.8.2. Diagramas de relación entre objetos







9.1.2. Modelo dinámico

1. Preparar escenarios de secuencias típicas de iteraciones (preparar escenarios detallados a partir de casos de uso). empatar eventos entre objetos para verificar consistencia).
2. Identificar eventos entre objetos (identificar sucesos = señales, entradas, decisiones, interrupciones, transiciones, acciones externas, condiciones de error).
3. Preparar un diagrama de eventos para cada escenario.
4. Construir diagramas de estado.
5. Comprobar consistencia
6. Añadir métodos.

9.1.2.1. Escenarios de secuencias típicas de iteraciones.

9.1.2.1.1. Manejo de archivos

9.1.2.1.1.1. CASO DE USO A1: OBTENCIÓN DE DATOS DE CONFIGURACIÓN

1. Si es necesario:
 - El SISTEMA obtiene el nombre del ARCHIVO DE CONFIGURACIÓN
 - ó El SISTEMA solicita al USUARIO el nombre del ARCHIVO DE CONFIGURACIÓN
 - El USUARIO selecciona el nombre del ARCHIVO DE CONFIGURACIÓN
2. Si es necesario:
 - El SISTEMA verifica el tipo del ARCHIVO DE CONFIGURACIÓN
 - El SISTEMA lee el ARCHIVO DE CONFIGURACIÓN
 - ó El SISTEMA obtiene los datos de configuración
 - Si es necesario:
 - El SISTEMA traduce los datos de configuración a su propio formato
3. El SISTEMA coloca los datos de configuración en su correspondiente vector de datos
4. Cuando se le solicita:
 - El SISTEMA devuelve un vector de datos de configuración a IMPRESIONES
 - ó El SISTEMA devuelve un vector de datos de configuración a objetos de VISUALIZACIÓN
 - ó El SISTEMA devuelve un vector de datos de configuración a MONITOR DE PROCESOS
 - ó El SISTEMA devuelve el ARCHIVO DE CONFIGURACIÓN al USUARIO
 - ó El SISTEMA devuelve el ARCHIVO DE CONFIGURACIÓN al MONITOR DE PROCESOS

ESCENARIO A1.1: OBTENCIÓN DE DATOS DE PARTÍCULAS

1. Si es necesario:
 - El SISTEMA obtiene el nombre del ARCHIVO DE PARTÍCULAS
 - ó El SISTEMA solicita al USUARIO el nombre del ARCHIVO DE PARTÍCULAS
 - El USUARIO selecciona el nombre del ARCHIVO DE PARTÍCULAS
2. Si es necesario:
 - El CONTENEDOR DE PARTÍCULAS verifica el tipo del ARCHIVO DE PARTÍCULAS
 - El CONTENEDOR DE PARTÍCULAS lee el ARCHIVO DE PARTÍCULAS
 - ó El CONTENEDOR DE PARTÍCULAS obtiene los datos de partículas
 - Si es necesario:
 - El CONTENEDOR DE PARTÍCULAS traduce los datos de partículas a su propio formato
3. El CONTENEDOR DE PARTÍCULAS coloca los datos de partículas en su correspondiente vector de datos.
4. Cuando se le solicita:
 - El CONTENEDOR DE PARTÍCULAS devuelve un vector de datos de partículas a IMPRESIONES
 - ó El CONTENEDOR DE PARTÍCULAS devuelve un vector de datos de partículas a MOLÉCULA
 - ó El CONTENEDOR DE PARTÍCULAS devuelve un vector de datos de partículas a MONITOR DE PROCESOS
 - ó El SISTEMA devuelve el ARCHIVO DE PARTÍCULAS al USUARIO
 - ó El CONTENEDOR DE PARTÍCULAS devuelve el ARCHIVO DE PARTÍCULAS al MONITOR DE PROCESOS

ESCENARIO A1.2: OBTENCIÓN DE DATOS DE HOYOS

1. Si es necesario:
 - El SISTEMA obtiene el nombre del ARCHIVO DE HOYOS
 - ó El SISTEMA solicita al USUARIO el nombre del ARCHIVO DE HOYOS
 - El USUARIO selecciona el nombre del ARCHIVO DE HOYOS
2. Si es necesario:
 - El CONTENEDOR DE HOYOS verifica el tipo del ARCHIVO DE HOYOS
 - El CONTENEDOR DE HOYOS lee el ARCHIVO DE HOYOS
 - ó El CONTENEDOR DE HOYOS obtiene los datos de hoyos
 - Si es necesario:
 - El CONTENEDOR DE HOYOS traduce los datos de hoyos a su propio formato
3. El CONTENEDOR DE HOYOS coloca los datos de hoyos en su correspondiente vector de datos.
4. Cuando se le solicita:
 - El CONTENEDOR DE HOYOS devuelve un vector de datos de hoyos a IMPRESIONES
 - ó El CONTENEDOR DE HOYOS devuelve un vector de datos de hoyos a SISTEMA DE CAVIDADES
 - ó El SISTEMA devuelve el ARCHIVO DE HOYOS al USUARIO

ESCENARIO A1.3: OBTENCIÓN DE DATOS DE POTENCIALES

1. Si es necesario:
 - El SISTEMA obtiene el nombre del ARCHIVO DE POTENCIALES
 - ó El SISTEMA solicita al USUARIO el nombre del ARCHIVO DE POTENCIALES
 - El USUARIO selecciona el nombre del ARCHIVO DE POTENCIALES
2. Si es necesario:
 - El CONTENEDOR DE POTENCIALES verifica el tipo del ARCHIVO DE POTENCIALES
 - El CONTENEDOR DE POTENCIALES lee el ARCHIVO DE POTENCIALES
 - ó El CONTENEDOR DE POTENCIALES obtiene los datos de potenciales
 - Si es necesario:
 - El CONTENEDOR DE POTENCIALES traduce los datos de potenciales a su propio formato
3. El CONTENEDOR DE POTENCIALES coloca los datos de potenciales en su correspondiente vector de datos.
4. Cuando se le solicita:
 - El CONTENEDOR DE POTENCIALES devuelve un vector de datos de potenciales a IMPRESIONES
 - ó El CONTENEDOR DE POTENCIALES devuelve un vector de datos de potenciales a CAMPO POTENCIAL
 - ó El SISTEMA devuelve el ARCHIVO DE POTENCIALES al USUARIO

ESCENARIO A1.4: OBTENCIÓN DE DATOS DE DIPOLOS

1. Si es necesario:
 - El SISTEMA obtiene el nombre del ARCHIVO DE DIPOLOS
 - ó El SISTEMA solicita al USUARIO el nombre del ARCHIVO DE DIPOLOS
 - El USUARIO selecciona el nombre del ARCHIVO DE DIPOLOS
2. Si es necesario:
 - El CONTENEDOR DE DIPOLOS verifica el tipo del ARCHIVO DE DIPOLOS
 - El CONTENEDOR DE DIPOLOS lee el ARCHIVO DE DIPOLOS
 - ó El CONTENEDOR DE DIPOLOS obtiene los datos de dipolos
 - Si es necesario:
 - El CONTENEDOR DE DIPOLOS traduce los datos de dipolos a su propio formato
3. El CONTENEDOR DE DIPOLOS coloca los datos de dipolos en su correspondiente vector de datos.
4. Cuando se le solicita:
 - El CONTENEDOR DE DIPOLOS devuelve un vector de datos de dipolos a IMPRESIONES
 - ó El CONTENEDOR DE DIPOLOS devuelve un vector de datos de dipolos a MOLÉCULA
 - ó El SISTEMA devuelve el ARCHIVO DE DIPOLOS al USUARIO

ESCENARIO A1.5: OBTENCIÓN DE DATOS DE CONEXIONES

1. Si es necesario:
 - El SISTEMA obtiene el nombre del ARCHIVO DE CONEXIONES
 - ó El SISTEMA solicita al USUARIO el nombre del ARCHIVO DE CONEXIONES
 - El USUARIO selecciona el nombre del ARCHIVO DE CONEXIONES
2. Si es necesario:
 - El CONTENEDOR DE CONEXIONES verifica el tipo del ARCHIVO DE CONEXIONES
 - El CONTENEDOR DE CONEXIONES lee el ARCHIVO DE CONEXIONES
 - ó El CONTENEDOR DE CONEXIONES obtiene los datos de conexiones
 - Si es necesario:
 - El CONTENEDOR DE CONEXIONES traduce los datos de conexiones a su propio formato

3. El **CONTENEDOR DE CONEXIONES** coloca los datos de conexiones en su correspondiente vector de datos.
4. Cuando se le solicita:
 - El **CONTENEDOR DE CONEXIONES** devuelve un vector de datos de conexiones a **IMPRESIONES**
 - ó El **CONTENEDOR DE CONEXIONES** devuelve un vector de datos de conexiones a **MOLÉCULA**
 - ó El **SISTEMA** devuelve el **ARCHIVO DE CONEXIONES** al **USUARIO**

ESCENARIO A1.6: OBTENCIÓN DE DATOS DE IMAGEN

1. Si es necesario:
 - El **SISTEMA** obtiene el nombre del **ARCHIVO DE IMAGEN**
 - ó El **SISTEMA** solicita al **USUARIO** el nombre del **ARCHIVO DE IMAGEN**
 - El **USUARIO** selecciona el nombre del **ARCHIVO DE IMAGEN**
2. Si es necesario:
 - El **CONTENEDOR DE IMAGEN** verifica el tipo del **ARCHIVO DE IMAGEN**
 - El **CONTENEDOR DE IMAGEN** lee el **ARCHIVO DE IMAGEN**
 - ó El **CONTENEDOR DE IMAGEN** obtiene los datos de imagen
 - Si es necesario:
 - El **CONTENEDOR DE IMAGEN** traduce los datos de imagen a su propio formato
3. El **CONTENEDOR DE IMAGEN** coloca los datos de imagen en su correspondiente vector de datos.
4. Cuando se le solicita:
 - El **CONTENEDOR DE IMAGEN** devuelve un vector de datos de imagen a **IMPRESIONES**
 - ó El **SISTEMA** devuelve el **ARCHIVO DE IMAGEN** al **USUARIO**

ESCENARIO A1.7: OBTENCIÓN DE DATOS DE RAYOS X

1. Si es necesario:
 - El **SISTEMA** obtiene el nombre del **ARCHIVO DE RAYOS X**
 - ó El **SISTEMA** solicita al **USUARIO** el nombre del **ARCHIVO DE RAYOS X**
 - El **USUARIO** selecciona el nombre del **ARCHIVO DE RAYOS X**
2. Si es necesario:
 - El **CONTENEDOR DE RAYOS X** verifica el tipo del **ARCHIVO DE RAYOS X**
 - El **CONTENEDOR DE RAYOS X** lee el **ARCHIVO DE RAYOS X**
 - ó El **CONTENEDOR DE RAYOS X** obtiene los datos de rayos x
 - Si es necesario:
 - El **CONTENEDOR DE RAYOS X** traduce los datos de rayos x a su propio formato
3. El **CONTENEDOR DE RAYOS X** coloca los datos de rayos x en su correspondiente vector de datos.
4. Cuando se le solicita:
 - El **CONTENEDOR DE RAYOS X** devuelve un vector de datos de rayos x a **IMPRESIONES**
 - ó El **SISTEMA** devuelve el **ARCHIVO DE RAYOS X** al **USUARIO**

ESCENARIO A1.8: OBTENCIÓN DE DATOS DE CONFIGURACIÓN PARA DINÁMICA

1. Si es necesario:
 - El **SISTEMA** obtiene el nombre del **ARCHIVO DE CONFIGURACIÓN PARA DINÁMICA**
 - ó El **SISTEMA** solicita al **USUARIO** el nombre del **ARCHIVO DE CONFIGURACIÓN PARA DINÁMICA**
 - El **USUARIO** selecciona el nombre del **ARCHIVO DE CONFIGURACIÓN PARA DINÁMICA**
2. Si es necesario:
 - El **CONTENEDOR DE CONFIGURACIÓN PARA DINÁMICA** verifica el tipo del **ARCHIVO DE CONFIGURACIÓN PARA DINÁMICA**
 - El **CONTENEDOR DE CONFIGURACIÓN PARA DINÁMICA** lee el **ARCHIVO DE CONFIGURACIÓN PARA DINÁMICA**
 - ó El **CONTENEDOR DE CONFIGURACIÓN PARA DINÁMICA** obtiene los datos de configuración para dinámica
 - Si es necesario:
 - El **CONTENEDOR DE CONFIGURACIÓN PARA DINÁMICA** traduce los datos de configuración para dinámica a su propio formato
3. El **CONTENEDOR DE CONFIGURACIÓN PARA DINÁMICA** coloca los datos de configuración para dinámica en su correspondiente vector de datos.
4. Cuando se le solicita:
 - El **CONTENEDOR DE CONFIGURACIÓN PARA DINÁMICA** devuelve un vector de datos de configuración para dinámica a **IMPRESIONES**
 - ó El **CONTENEDOR DE CONFIGURACIÓN PARA DINÁMICA** devuelve el **ARCHIVO DE CONFIGURACIÓN PARA DINÁMICA** a **MONITOR DE PROCESOS**
 - ó El **SISTEMA** devuelve el **ARCHIVO DE CONFIGURACIÓN PARA DINÁMICA** al **USUARIO**
 - ó El **SISTEMA** devuelve el vector de datos de configuración de dinámica al **USUARIO**

ESCENARIO A1.9: OBTENCIÓN DE DATOS DE INICIACIÓN DE DINÁMICA

1. Si es necesario:
 - El **SISTEMA** obtiene el nombre del **ARCHIVO DE INICIACIÓN DE DINÁMICA**
 - ó El **SISTEMA** solicita al **USUARIO** el nombre del **ARCHIVO DE INICIACIÓN DE DINÁMICA**
 - El **USUARIO** selecciona el nombre del **ARCHIVO DE INICIACIÓN DE DINÁMICA**
2. Si es necesario:
 - El **CONTENEDOR DE INICIACIÓN DE DINÁMICA** verifica el tipo de **ARCHIVO DE INICIACIÓN DE DINÁMICA**
 - El **CONTENEDOR DE INICIACIÓN DE DINÁMICA** lee el **ARCHIVO DE INICIACIÓN DE DINÁMICA**
 - ó El **CONTENEDOR DE INICIACIÓN DE DINÁMICA** obtiene los datos de iniciación de dinámica
 - Si es necesario:
 - El **CONTENEDOR DE INICIACIÓN DE DINÁMICA** traduce los datos de iniciación de dinámica a su propio formato
3. El **CONTENEDOR DE INICIACIÓN DE DINÁMICA** coloca los datos de iniciación de dinámica en su correspondiente vector de datos.
4. Cuando se le solicita:
 - El **CONTENEDOR DE INICIACIÓN DE DINÁMICA** devuelve un vector de datos de iniciación de dinámica a **IMPRESIONES**
 - ó El **SISTEMA** devuelve el **ARCHIVO DE INICIACIÓN DE DINÁMICA** al **USUARIO**
 - ó El **SISTEMA** devuelve el vector de datos de iniciación de dinámica al **USUARIO**

9.1.2.1.1.2. CASO DE USO A2: RENOMBRAR ARCHIVO DE CONFIGURACIÓN

- | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none">1. El USUARIO elige el ARCHIVO DE CONFIGURACIÓN a renombrar2. El SISTEMA solicita al USUARIO el nuevo nombre del ARCHIVO DE CONFIGURACIÓN3. El USUARIO introduce el nuevo nombre del ARCHIVO DE CONFIGURACIÓN4. El SISTEMA verifica el nuevo nombre del ARCHIVO DE CONFIGURACIÓN<ul style="list-style-type: none">Si es valido:<ul style="list-style-type: none">El SISTEMA cambia el nombre anterior por el nuevo nombrePaso 5.Si no:<ul style="list-style-type: none">Paso 5.5. El SISTEMA notifica al USUARIO sobre el resultado de la operación |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

ESCENARIO A2.1: RENOMBRAR ARCHIVO DE PARTÍCULAS

1. El **USUARIO** elige el **ARCHIVO DE PARTÍCULAS** a renombrar
2. El **SISTEMA** solicita al **USUARIO** el nuevo nombre del **ARCHIVO DE PARTÍCULAS**
3. El **USUARIO** introduce el nuevo nombre del **ARCHIVO DE PARTÍCULAS**
4. El **CONTENEDOR DE PARTÍCULAS** verifica el nuevo nombre del **ARCHIVO DE PARTÍCULAS**
 - Si es valido:
 - El **CONTENEDOR DE PARTÍCULAS** cambia el nombre anterior por el nuevo nombre
 - Paso 5.
 - Si no:
 - Paso 5.
5. El **SISTEMA** notifica al **USUARIO** sobre el resultado de la operación

ESCENARIO A2.2: RENOMBRAR ARCHIVO DE IMAGEN

1. El **USUARIO** elige el **ARCHIVO DE IMAGEN** a renombrar
2. El **SISTEMA** solicita al **USUARIO** el nuevo nombre del **ARCHIVO DE IMAGEN**
3. El **USUARIO** introduce el nuevo nombre del **ARCHIVO DE IMAGEN**
4. El **CONTENEDOR DE IMAGEN** verifica el nuevo nombre del **ARCHIVO DE IMAGEN**
 - Si es valido:
 - El **CONTENEDOR DE IMAGEN** cambia el nombre anterior por el nuevo nombre
 - Paso 5.
 - Si no:
 - Paso 5.
5. El **SISTEMA** notifica al **USUARIO** sobre el resultado de la operación

ESCENARIO A2.3: RENOMBRAR ARCHIVO DE INICIACIÓN DE DINÁMICA

1. El **USUARIO** elige el **ARCHIVO DE INICIACIÓN DE DINÁMICA** a renombrar
2. El **SISTEMA** solicita al **USUARIO** el nuevo nombre del **ARCHIVO DE INICIACIÓN DE DINÁMICA**
3. El **USUARIO** introduce el nuevo nombre del **ARCHIVO DE INICIACIÓN DE DINÁMICA**
4. El **CONTENEDOR DE INICIACIÓN DE DINÁMICA** verifica el nuevo nombre del **ARCHIVO DE INICIACIÓN DE DINÁMICA**
Si es valido:
El **CONTENEDOR DE INICIACIÓN DE DINÁMICA** cambia el nombre anterior por el nuevo nombre
Paso 5.
Si no:
Paso 5.
5. El **SISTEMA** notifica al **USUARIO** sobre el resultado de la operación

9.1.2.1.1.3. CASO DE USO A3: GUARDAR ARCHIVO DE CONFIGURACIÓN

1. El **USUARIO** elige el **ARCHIVO DE CONFIGURACIÓN** a guardar
2. El **SISTEMA** guarda los datos en el **ARCHIVO DE CONFIGURACIÓN**
3. El **SISTEMA** notifica al **USUARIO** sobre el resultado de la operación

ESCENARIO A3.1: GUARDAR ARCHIVO DE PARTÍCULAS

1. El **USUARIO** elige el **ARCHIVO DE PARTÍCULAS** a guardar
2. El **CONTENEDOR DE PARTÍCULAS** guarda los datos en el **ARCHIVO DE PARTÍCULAS**
3. El **SISTEMA** notifica al **USUARIO** sobre el resultado de la operación

ESCENARIO A3.2: GUARDAR ARCHIVO DE IMAGEN

1. El **USUARIO** elige el **ARCHIVO DE IMAGEN** a guardar
2. El **CONTENEDOR DE IMAGEN** guarda los datos en el **ARCHIVO DE IMAGEN**
3. El **SISTEMA** notifica al **USUARIO** sobre el resultado de la operación

9.1.2.1.1.4. CASO DE USO A4: GUARDAR CON OTRO NOMBRE ARCHIVO DE CONFIGURACIÓN

1. El USUARIO elige la configuración de datos para guardar la en un ARCHIVO DE CONFIGURACIÓN con otro nombre
2. El SISTEMA solicita al usuario que introduzca un nombre para el ARCHIVO DE CONFIGURACIÓN
3. El USUARIO introduce el nombre
4. El SISTEMA valida el nombre
5. El SISTEMA crea un nuevo CONTENEDOR DE DATOS
6. El SISTEMA copia la configuración de datos al nuevo CONTENEDOR DE DATOS
7. El SISTEMA le pasa el nombre al nuevo CONTENEDOR DE DATOS
8. El nuevo CONTENEDOR DE DATOS cambia el nombre a su ARCHIVO DE CONFIGURACIÓN
9. El nuevo CONTENEDOR DE DATOS guarda los datos en su ARCHIVO DE CONFIGURACIÓN
10. El antiguo CONTENEDOR DE DATOS cierra su ARCHIVO DE CONFIGURACIÓN sin guardar los cambios
11. El SISTEMA elimina el antiguo CONTENEDOR DE DATOS de la memoria
12. El SISTEMA indica al USUARIO que la operación de guardar con otro nombre se llevo a cabo.

ESCENARIO A4.1: GUARDAR CON OTRO NOMBRE ARCHIVO DE PARTÍCULAS

1. El USUARIO elige la configuración de particulas para guardarla en un ARCHIVO DE PARTÍCULAS con otro nombre
2. El SISTEMA solicita al usuario que introduzca un nombre para el ARCHIVO DE PARTÍCULAS
3. El USUARIO introduce el nombre
4. El SISTEMA valida el nombre
5. El SISTEMA crea un nuevo CONTENEDOR DE PARTÍCULAS
6. El SISTEMA copia la configuración de particulas al nuevo CONTENEDOR DE PARTÍCULAS
7. El SISTEMA le pasa el nombre al nuevo CONTENEDOR DE PARTÍCULAS
8. El nuevo CONTENEDOR DE PARTÍCULAS cambia el nombre a su ARCHIVO DE PARTÍCULAS
9. El nuevo CONTENEDOR DE PARTÍCULAS guarda los datos en su ARCHIVO DE PARTÍCULAS
10. El antiguo CONTENEDOR DE PARTÍCULAS cierra su ARCHIVO DE PARTÍCULAS sin guardar los cambios
11. El SISTEMA elimina el antiguo CONTENEDOR DE PARTÍCULAS de la memoria
12. El SISTEMA indica al USUARIO que la operación de guardar con otro nombre se llevo a cabo.

ESCENARIO A4.2: GUARDAR CON OTRO NOMBRE ARCHIVO DE DIPOLOS

1. El USUARIO elige la configuración de dipolos para guardarla en un ARCHIVO DE DIPOLOS con otro nombre
2. El SISTEMA solicita al usuario que introduzca un nombre para el ARCHIVO DE DIPOLOS
3. El USUARIO introduce el nombre
4. El SISTEMA valida el nombre
5. El SISTEMA crea un nuevo CONTENEDOR DE DIPOLOS
6. El SISTEMA copia la configuración de dipolos al nuevo CONTENEDOR DE DIPOLOS
7. El SISTEMA le pasa el nombre al nuevo CONTENEDOR DE DIPOLOS
8. El nuevo CONTENEDOR DE DIPOLOS cambia el nombre a su ARCHIVO DE DIPOLOS
9. El nuevo CONTENEDOR DE DIPOLOS guarda los datos en su ARCHIVO DE DIPOLOS
10. El antiguo CONTENEDOR DE DIPOLOS cierra su ARCHIVO DE DIPOLOS sin guardar los cambios
11. El SISTEMA elimina el antiguo CONTENEDOR DE DIPOLOS de la memoria
12. El SISTEMA indica al USUARIO que la operación de guardar con otro nombre se llevo a cabo.

ESCENARIO A4.3: GUARDAR CON OTRO NOMBRE ARCHIVO DE HOYOS

1. El USUARIO elige la configuración de hoyos para guardarla en un ARCHIVO DE HOYOS con otro nombre
2. El SISTEMA solicita al usuario que introduzca un nombre para el ARCHIVO DE HOYOS
3. El USUARIO introduce el nombre
4. El SISTEMA valida el nombre
5. El SISTEMA crea un nuevo CONTENEDOR DE HOYOS
6. El SISTEMA copia la configuración de hoyos al nuevo CONTENEDOR DE HOYOS
7. El SISTEMA le pasa el nombre al nuevo CONTENEDOR DE HOYOS
8. El nuevo CONTENEDOR DE HOYOS cambia el nombre a su ARCHIVO DE HOYOS
9. El nuevo CONTENEDOR DE HOYOS guarda los datos en su ARCHIVO DE HOYOS
10. El antiguo CONTENEDOR DE HOYOS cierra su ARCHIVO DE HOYOS sin guardar los cambios
11. El SISTEMA elimina el antiguo CONTENEDOR DE HOYOS de la memoria
12. El SISTEMA indica al USUARIO que la operación de guardar con otro nombre se llevo a cabo.

9.1.2.1.2. Visualización y Editor Molecular

9.1.2.1.2.1. CASO DE USO B1: INICIAR ESCENA GRÁFICA

1. El SISTEMA reconoce la configuración disponible
2. El SISTEMA obtiene el tipo y datos de caja del CONTENEDOR DE PARTÍCULAS
3. El SISTEMA obtiene los datos de configuración de cada CONTENEDOR DE DATOS
4. El SISTEMA crea una ESCENA GRÁFICA
El SISTEMA indica a la CELDA COMPUTACIONAL que la caja es cúbica, hexagonal o romboidal
El SISTEMA acomoda los datos en cada OBJETO DINÁMICO
5. El SISTEMA dibuja los resultados de la dinámica
El SISTEMA dibuja los OBJETOS ANIMADOS con la apariencia inicial

ESCENARIO B1.1: INICIAR ESCENA GRÁFICA EJEMPLO 1

1. El SISTEMA reconoce la configuración disponible:
 Hay un archivo de partículas
 Hay un archivo de hoyos
 Hay un archivo de potenciales
 Hay un archivo de dipolos
 Hay un archivo de conexiones
2. El SISTEMA obtiene el tipo y datos de caja del CONTENEDOR DE PARTÍCULAS:
 El tipo de caja es cúbica
 El tipo de caja es hexagonal
 El tipo de caja es romboidal
3. El SISTEMA obtiene los datos de configuración:
 Del CONTENEDOR DE PARTÍCULAS
 Del CONTENEDOR DE HOYOS
 Del CONTENEDOR DE POTENCIALES
 Del CONTENEDOR DE DIPOLOS
 Del CONTENEDOR DE CONEXIONES
4. El SISTEMA crea una ESCENA GRÁFICA
El SISTEMA indica a la CELDA COMPUTACIONAL que la caja es: cúbica
El SISTEMA acomoda los datos de PARTÍCULAS en MOLÉCULA
5. El SISTEMA dibuja los resultados de la dinámica:
 El SISTEMA dibuja la MOLÉCULA con la apariencia inicial
 El SISTEMA dibuja el SISTEMA DE CAVIDADES con la apariencia inicial
 El SISTEMA dibuja la SUPERFICIE DE POTENCIAL con la apariencia inicial

ESCENARIO B1.2: INICIAR ESCENA GRÁFICA EJEMPLO 2

1. El SISTEMA reconoce la configuración disponible:
 Hay un archivo de partículas
 Hay un archivo de hoyos
 Hay un archivo de potenciales
 Hay un archivo de dipolos
 Hay un archivo de conexiones
2. El SISTEMA obtiene el tipo y datos de caja del CONTENEDOR DE PARTÍCULAS:
 El tipo de caja es cúbica
 El tipo de caja es hexagonal
 El tipo de caja es romboidal
3. El SISTEMA obtiene los datos de configuración:
 Del CONTENEDOR DE PARTÍCULAS
 Del CONTENEDOR DE HOYOS
 Del CONTENEDOR DE POTENCIALES
 Del CONTENEDOR DE DIPOLOS
 Del CONTENEDOR DE CONEXIONES

4. El SISTEMA crea una ESCENA GRÁFICA
- El SISTEMA indica a la CELDA COMPUTACIONAL que la caja es: cúbica
 - El SISTEMA acomoda los datos de PARTÍCULAS en MOLÉCULA
 - El SISTEMA acomoda los datos de HOYOS en SISTEMA DE CAVIDADES
 - El SISTEMA acomoda los datos de POTENCIALES en SUPERFICIE POTENCIAL
 - El SISTEMA acomoda los datos de DIPOLOS en MOLÉCULA
5. El SISTEMA dibuja los resultados de la dinámica:
- El SISTEMA dibuja la MOLÉCULA con la apariencia inicial
 - El SISTEMA dibuja el SISTEMA DE CAVIDADES con la apariencia inicial
 - El SISTEMA dibuja la SUPERFICIE DE POTENCIAL con la apariencia inicial

ESCENARIO B1.3: INICIAR ESCENA GRÁFICA EJEMPLO 3

1. El SISTEMA reconoce la configuración disponible:
- Hay un archivo de partículas
 - Hay un archivo de hoyos
 - Hay un archivo de potenciales
 - Hay un archivo de dipolos
 - Hay un archivo de conexiones
2. El SISTEMA obtiene el tipo y datos de caja del CONTENEDOR DE PARTÍCULAS:
- El tipo de caja es cúbica
 - El tipo de caja es hexagonal
 - El tipo de caja es romboidal
3. El SISTEMA obtiene los datos de configuración:
- Del CONTENEDOR DE PARTÍCULAS y acomoda los datos en MOLÉCULA
 - Del CONTENEDOR DE HOYOS y acomoda los datos en SISTEMA DE CAVIDADES
 - Del CONTENEDOR DE POTENCIALES y acomoda los datos en SUPERFICIE POTENCIAL
 - Del CONTENEDOR DE DIPOLOS y acomoda los datos en MOLÉCULA
 - Del CONTENEDOR DE CONEXIONES y acomoda los datos en MOLÉCULA
4. El SISTEMA crea una ESCENA GRÁFICA
- El SISTEMA indica a la CELDA COMPUTACIONAL que la caja es: hexagonal
 - El SISTEMA acomoda los datos de PARTÍCULAS en MOLÉCULA
 - El SISTEMA acomoda los datos de HOYOS en SISTEMA DE CAVIDADES
 - El SISTEMA acomoda los datos de CONEXIONES en MOLÉCULA
5. El SISTEMA dibuja los resultados de la dinámica:
- El SISTEMA dibuja la MOLÉCULA con la apariencia inicial
 - El SISTEMA dibuja el SISTEMA DE CAVIDADES con la apariencia inicial
 - El SISTEMA dibuja la SUPERFICIE DE POTENCIAL con la apariencia inicial

ESCENARIO B1.4: INICIAR ESCENA GRÁFICA EJEMPLO 4

1. El SISTEMA reconoce la configuración disponible:
- Hay un archivo de partículas
 - Hay un archivo de hoyos
 - Hay un archivo de potenciales
 - Hay un archivo de dipolos
 - Hay un archivo de conexiones
2. El SISTEMA obtiene el tipo y datos de caja del CONTENEDOR DE PARTÍCULAS:
- El tipo de caja es cúbica
 - El tipo de caja es hexagonal
 - El tipo de caja es romboidal
3. El SISTEMA obtiene los datos de configuración:
- Del CONTENEDOR DE PARTÍCULAS y acomoda los datos en MOLÉCULA
 - Del CONTENEDOR DE HOYOS y acomoda los datos en SISTEMA DE CAVIDADES
 - Del CONTENEDOR DE POTENCIALES y acomoda los datos en SUPERFICIE POTENCIAL
 - Del CONTENEDOR DE DIPOLOS y acomoda los datos en MOLÉCULA
 - Del CONTENEDOR DE CONEXIONES y acomoda los datos en MOLÉCULA

4. El SISTEMA crea una **ESCENA GRÁFICA**
 El SISTEMA indica a la **CELDA COMPUTACIONAL** que la caja es: romboidal
 El SISTEMA acomoda los datos de **PARTÍCULAS** en **MOLÉCULA**
 El SISTEMA acomoda los datos de **DIPOLOS** en **MOLÉCULA**
5. El SISTEMA dibuja los resultados de la dinámica:
 El SISTEMA dibuja la **MOLÉCULA** con la apariencia inicial
 El SISTEMA dibuja el **SISTEMA DE CAVIDADES** con la apariencia inicial
 El SISTEMA dibuja la **SUPERFICIE DE POTENCIAL** con la apariencia inicial

9.1.2.1.2.2. CASO DE USO B2: SELECCIÓN DE OBJETOS GRÁFICOS

1. El USUARIO indica al SISTEMA el modo de selección y el tipo de objeto a seleccionar:

Modo		Tipo
<input type="checkbox"/> Especie	<input type="checkbox"/> Tipo de especie	<input type="checkbox"/> Átomos
<input type="checkbox"/> Individual	<input type="checkbox"/> P3D <input type="checkbox"/> P2D	<input type="checkbox"/> Hoyos
<input type="checkbox"/> Grupo		<input type="checkbox"/> Dipolos
<input type="checkbox"/> Caja		<input type="checkbox"/> Conexiones
<input type="checkbox"/> Esfera		<input type="checkbox"/> Elementos de todos los tipos
<input type="checkbox"/> Lista		<input type="checkbox"/> Molécula
2. El SISTEMA llama a **INDICADOR DE SELECCIÓN** y le proporciona el modo y el tipo:
 Modo: _____ Tipo: _____
 Tipo de especie: _____
3. El **INDICADOR DE SELECCIÓN** de acuerdo al tipo de selección marca los **OBJETOS GRÁFICOS** indicados
 Si es por grupo con caja:
 El **INDICADOR DE SELECCIÓN** usa una **CAJA**
 Si es por grupo con esfera:
 El **INDICADOR DE SELECCIÓN** usa una **ESFERA**
 Si es por especie:
 El **INDICADOR DE SELECCIÓN** marca sólo todos los átomos de una especie
 Si es individual:
 El **INDICADOR DE SELECCIÓN** usa un **PUNTERO 3D o 2D**
 Si es Molécula el tipo:
 La selección siempre es individual
4. El **INDICADOR DE SELECCIÓN** obtiene las etiquetas de los **OBJETOS GRÁFICOS** seleccionados
5. El **INDICADOR DE SELECCIÓN** devuelve las etiquetas al **SISTEMA**

ESCENARIO B2.1: SELECCIONAR ÁTOMOS POR ESPECIE

1. El USUARIO indica al SISTEMA el modo de selección y el tipo de objeto a seleccionar:

Modo		Tipo
<input checked="" type="checkbox"/> Especie <u>Oxigeno</u>	<input type="checkbox"/> Tipo de especie	<input checked="" type="checkbox"/> Átomos
<input type="checkbox"/> Individual	<input type="checkbox"/> P3D <input type="checkbox"/> P2D	<input type="checkbox"/> Hoyos
<input type="checkbox"/> Grupo		<input type="checkbox"/> Dipolos
<input type="checkbox"/> Caja		<input type="checkbox"/> Conexiones
<input type="checkbox"/> Esfera		<input type="checkbox"/> Elementos de todos los tipos
<input type="checkbox"/> Lista		<input type="checkbox"/> Molécula
2. El SISTEMA llama a **INDICADOR DE SELECCIÓN** y le proporciona el modo y el tipo:
 Modo: Especie Tipo: ÁTOMOS
 Tipo de especie: Oxigeno
3. El **INDICADOR DE SELECCIÓN** de acuerdo al tipo de selección marca los **ÁTOMOS** indicados
 El **INDICADOR DE SELECCIÓN** marca sólo todos los átomos de una especie
4. El **INDICADOR DE SELECCIÓN** obtiene las etiquetas de los **ÁTOMOS** seleccionados
5. El **INDICADOR DE SELECCIÓN** devuelve las etiquetas al **SISTEMA**

ESCENARIO B2.2: SELECCIONAR ÁTOMOS POR GRUPO CON CAJA

1. El USUARIO indica al SISTEMA el modo de selección y el tipo de objeto a seleccionar:
 Modo _____ Tipo _____

<input type="checkbox"/>	Especie	<input type="checkbox"/>	Tipo de especie	<input checked="" type="checkbox"/>	Átomos
<input type="checkbox"/>	Individual	<input type="checkbox"/>	P3D	<input type="checkbox"/>	Hoyos
<input checked="" type="checkbox"/>	Grupo	<input type="checkbox"/>	P2D	<input type="checkbox"/>	Dipolos
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Caja	<input type="checkbox"/>	Conexiones
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Esfera	<input type="checkbox"/>	Elementos de todos los tipos
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Lista	<input type="checkbox"/>	Molécula

- El **SISTEMA** llama a **INDICADOR DE SELECCIÓN** y le proporciona el modo y el tipo:
 Modo: Grupo - Caja Tipo: ÁTOMOS
 Tipo de especie: X
- El **INDICADOR DE SELECCIÓN** de acuerdo al tipo de selección marca los **OBJETOS GRÁFICOS** indicados
 El **INDICADOR DE SELECCIÓN** usa una **CAJA**
- El **INDICADOR DE SELECCIÓN** obtiene las etiquetas de los **ÁTOMOS** seleccionados
- El **INDICADOR DE SELECCIÓN** devuelve las etiquetas al **SISTEMA**

ESCENARIO B2.3: SELECCIONAR ELEMENTOS DE TODOS LOS TIPOS POR GRUPO CON ESFERA

- El **USUARIO** indica al **SISTEMA** el modo de selección y el tipo de objeto a seleccionar:

<input type="checkbox"/>	Especie	<input type="checkbox"/>	Tipo de especie	<input type="checkbox"/>	Átomos
<input type="checkbox"/>	Individual	<input type="checkbox"/>	P3D	<input type="checkbox"/>	Hoyos
<input checked="" type="checkbox"/>	Grupo	<input type="checkbox"/>	P2D	<input type="checkbox"/>	Dipolos
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Caja	<input type="checkbox"/>	Conexiones
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Esfera	<input checked="" type="checkbox"/>	Elementos de todos los tipos
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Lista	<input checked="" type="checkbox"/>	Molécula
- El **SISTEMA** llama a **INDICADOR DE SELECCIÓN** y le proporciona el modo y el tipo:
 Modo: Grupo - Esfera Tipo: Elementos de todos los tipos
 Tipo de especie X
- El **INDICADOR DE SELECCIÓN** de acuerdo al tipo de selección marca los **OBJETOS GRÁFICOS** indicados
 El **INDICADOR DE SELECCIÓN** usa una **ESFERA**
- El **INDICADOR DE SELECCIÓN** obtiene las etiquetas de los **ÁTOMOS, HOYOS, DIPOLOS** y **CONEXIONES**, seleccionados
- El **INDICADOR DE SELECCIÓN** devuelve las etiquetas al **SISTEMA**

ESCENARIO B2.4: SELECCIONAR MOLÉCULA

- El **USUARIO** indica al **SISTEMA** el modo de selección y el tipo de objeto a seleccionar:

<input type="checkbox"/>	Especie	<input type="checkbox"/>	Tipo de especie	<input type="checkbox"/>	Átomos
<input checked="" type="checkbox"/>	Individual	<input checked="" type="checkbox"/>	P3D	<input type="checkbox"/>	Hoyos
<input type="checkbox"/>	Grupo	<input type="checkbox"/>	P2D	<input type="checkbox"/>	Dipolos
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Caja	<input type="checkbox"/>	Conexiones
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Esfera	<input type="checkbox"/>	Elementos de todos los tipos
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Lista	<input checked="" type="checkbox"/>	Molécula
- El **SISTEMA** llama a **INDICADOR DE SELECCIÓN** y le proporciona el modo y el tipo:
 Modo: Individual Tipo: Molécula
 Tipo de especie: X
- El **INDICADOR DE SELECCIÓN** de acuerdo al tipo de selección marca las **MOLÉCULAS** indicadas
 El **INDICADOR DE SELECCIÓN** usa un **PUNTERO 3D o 2D**
- El **INDICADOR DE SELECCIÓN** obtiene las etiquetas de las **MOLÉCULAS** seleccionadas
- El **INDICADOR DE SELECCIÓN** devuelve las etiquetas al **SISTEMA**

ESCENARIO B2.5: SELECCIONAR CONEXIONES INDIVIDUALES

- El **USUARIO** indica al **SISTEMA** el modo de selección y el tipo de objeto a seleccionar:

<input type="checkbox"/> Modo	<input type="checkbox"/> Tipo	
<input type="checkbox"/> Especie	<input type="checkbox"/> Tipo de especie	<input type="checkbox"/> Átomos
<input checked="" type="checkbox"/> Individual	<input type="checkbox"/> P3D <input checked="" type="checkbox"/> P2D	<input type="checkbox"/> Hoyos
<input type="checkbox"/> Grupo		<input type="checkbox"/> Dipolos
<input type="checkbox"/> Caja		<input checked="" type="checkbox"/> Conexiones
<input type="checkbox"/> Esfera		<input type="checkbox"/> Elementos de todos los tipos
<input type="checkbox"/> Lista		<input type="checkbox"/> Molécula
- El **SISTEMA** llama a **INDICADOR DE SELECCIÓN** y le proporciona el modo y el tipo:

Modo: Individual Tipo: Conexiones

Tipo de especie: X
- El **INDICADOR DE SELECCIÓN** de acuerdo al tipo de selección marca las **CONEXIONES** indicadas
El **INDICADOR DE SELECCIÓN** usa un **PUNTERO 3D o 2D**
- El **INDICADOR DE SELECCIÓN** obtiene las etiquetas de las **CONEXIONES** seleccionadas
- El **INDICADOR DE SELECCIÓN** devuelve las etiquetas al **SISTEMA**

ESCENARIO B2.6: SELECCIONAR DIPOLOS POR GRUPO EN FORMA LIBRE

- El **USUARIO** indica al **SISTEMA** el modo de selección y el tipo de objeto a seleccionar:

<input type="checkbox"/> Modo	<input type="checkbox"/> Tipo	
<input type="checkbox"/> Especie	<input type="checkbox"/> Tipo de especie	<input type="checkbox"/> Átomos
<input type="checkbox"/> Individual	<input type="checkbox"/> P3D <input type="checkbox"/> P2D	<input type="checkbox"/> Hoyos
<input checked="" type="checkbox"/> Grupo		<input checked="" type="checkbox"/> Dipolos
<input type="checkbox"/> Caja		<input type="checkbox"/> Conexiones
<input type="checkbox"/> Esfera		<input type="checkbox"/> Elementos de todos los tipos
<input checked="" type="checkbox"/> Lista		<input type="checkbox"/> Molécula
- El **SISTEMA** llama a **INDICADOR DE SELECCIÓN** y le proporciona el modo y el tipo:

Modo: Grupo – Lista Tipo: Dipolos

Tipo de especie: X
- El **INDICADOR DE SELECCIÓN** de acuerdo al tipo de selección marca los **DIPOLOS** indicados
- El **INDICADOR DE SELECCIÓN** obtiene las etiquetas de los **DIPOLOS** seleccionados
- El **INDICADOR DE SELECCIÓN** devuelve las etiquetas al **SISTEMA**

ESCENARIO B2.7: SELECCIONAR ÁTOMOS INDIVIDUALES

- El **USUARIO** indica al **SISTEMA** el modo de selección y el tipo de objeto a seleccionar:

<input type="checkbox"/> Modo	<input type="checkbox"/> Tipo	
<input type="checkbox"/> Especie	<input type="checkbox"/> Tipo de especie	<input checked="" type="checkbox"/> Átomos
<input checked="" type="checkbox"/> Individual	<input checked="" type="checkbox"/> P3D <input type="checkbox"/> P2D	<input type="checkbox"/> Hoyos
<input type="checkbox"/> Grupo		<input type="checkbox"/> Dipolos
<input type="checkbox"/> Caja		<input type="checkbox"/> Conexiones
<input type="checkbox"/> Esfera		<input type="checkbox"/> Elementos de todos los tipos
<input type="checkbox"/> Lista		<input type="checkbox"/> Molécula
- El **SISTEMA** llama a **INDICADOR DE SELECCIÓN** y le proporciona el modo y el tipo:

Modo: Individual Tipo: ÁTOMOS

Tipo de especie: X
- El **INDICADOR DE SELECCIÓN** de acuerdo al tipo de selección marca los **ÁTOMOS** indicados
El **INDICADOR DE SELECCIÓN** usa un **PUNTERO 3D o 2D**
- El **INDICADOR DE SELECCIÓN** obtiene las etiquetas de los **ÁTOMOS** seleccionados
- El **INDICADOR DE SELECCIÓN** devuelve las etiquetas al **SISTEMA**

9.1.2.1.2.3. CASO DE USO B3: CONEXIÓN DE ÁTOMOS

1. El **USUARIO** indica al **SISTEMA** que pinte las **CONEXIONES**

2. El **SISTEMA** solicita el radio de conexión, el tipo de conexión, el origen de la tabla de conectividad y las especies a conectar al **USUARIO**

	Tipo		Tabla de conectividad
<input type="checkbox"/>	Entre especies de átomos	<input type="checkbox"/>	Leer de un archivo
<input type="checkbox"/>	Entre todos los átomos	<input type="checkbox"/>	Calcular

El **USUARIO** introduce el radio de conexión

Radio de conexión

Si es necesario:

El **USUARIO** indica que especies de **ÁTOMOS** desea conectar entre sí de las disponibles en la **MOLÉCULA**

3. El **SISTEMA** obtiene los **ÁTOMOS** a conectar

El **SISTEMA** lee una tabla de conectividad de un **ARCHIVO DE CONFIGURACIÓN**

ó El **SISTEMA** calcula la tabla de conectividad

El **SISTEMA** valida las **CONEXIONES**

Si el tipo de conexión es entre especies de átomos:

El **SISTEMA** conecta sólo **ÁTOMOS** de una especie con **ÁTOMOS** de otra especie dentro del radio de conexión, no hay conexión entre **ÁTOMOS** de la misma especie

Si el tipo de conexión es entre todos los átomos:

El **SISTEMA** conecta a todos los **ÁTOMOS** de cualquier especie entre sí, dentro del radio de conexión

4. El **SISTEMA** conecta los **ÁTOMOS** (dibuja las conexiones)

ESCENARIO B3.1: CONEXIÓN DE ÁTOMOS ENTRE ESPECIES DESDE UN ARCHIVO

1. El **USUARIO** indica al **SISTEMA** que pinte las **CONEXIONES**

2. El **SISTEMA** solicita el radio de conexión, el tipo de conexión, el origen de la tabla de conectividad y las especies a conectar al **USUARIO**

	Tipo		Tabla de conectividad
<input checked="" type="checkbox"/>	Entre especies de átomos	<input checked="" type="checkbox"/>	Leer de un archivo
<input type="checkbox"/>	Entre todos los átomos	<input type="checkbox"/>	Calcular

El **USUARIO** introduce el radio de conexión

Radio de conexión

Si es necesario:

El **USUARIO** indica que especies de **ÁTOMOS** desea conectar entre sí de las disponibles en la **MOLÉCULA**

<input checked="" type="checkbox"/>	Oxígeno - Aluminio
<input checked="" type="checkbox"/>	Oxígeno - Silicio
<input type="checkbox"/>	Silicio - Aluminio

3. El **SISTEMA** obtiene los **ÁTOMOS** a conectar

El **SISTEMA** lee una tabla de conectividad de un **CONTENEDOR DE CONEXIONES**

El **SISTEMA** valida las **CONEXIONES**

El **SISTEMA** conecta sólo **ÁTOMOS** de una especie con **ÁTOMOS** de otra especie dentro del radio de conexión, no hay conexión entre **ÁTOMOS** de la misma especie

4. El **SISTEMA** conecta los **ÁTOMOS** (dibuja las conexiones)

ESCENARIO B3.2: CONEXIÓN DE ÁTOMOS ENTRE ESPECIES DESDE CÁLCULO

1. El **USUARIO** indica al **SISTEMA** que pinte las **CONEXIONES**
2. El **SISTEMA** solicita el radio de conexión, el tipo de conexión, el origen de la tabla de conectividad y las especies a conectar al **USUARIO**

Tipo	Tabla de conectividad
<input checked="" type="checkbox"/> Entre especies de átomos	<input type="checkbox"/> Leer de un archivo
<input type="checkbox"/> Entre todos los átomos	<input checked="" type="checkbox"/> Calcular

El **USUARIO** introduce el radio de conexión
Radio de conexión 3.2

Si es necesario:

El **USUARIO** indica que especies de **ÁTOMOS** desea conectar entre sí de las disponibles en la **MOLÉCULA**

<input type="checkbox"/> Oxígeno - Aluminio
<input type="checkbox"/> Oxígeno - Silicio
<input checked="" type="checkbox"/> Silicio - Aluminio

3. El **SISTEMA** obtiene los **ÁTOMOS** a conectar
El **SISTEMA** calcula la tabla de conectividad
El **SISTEMA** valida las **CONEXIONES**
El **SISTEMA** conecta sólo **ÁTOMOS** de una especie con **ÁTOMOS** de otra especie dentro del radio de conexión, no hay conexión entre **ÁTOMOS** de la misma especie
4. El **SISTEMA** conecta los **ÁTOMOS** (dibuja las conexiones)

ESCENARIO B3.3: CONEXIÓN ENTRE TODOS LOS ÁTOMOS DESDE UN ARCHIVO

1. El **USUARIO** indica al **SISTEMA** que pinte las **CONEXIONES**
2. El **SISTEMA** solicita el radio de conexión, el tipo de conexión, el origen de la tabla de conectividad y las especies a conectar al **USUARIO**

Tipo	Tabla de conectividad
<input type="checkbox"/> Entre especies de átomos	<input checked="" type="checkbox"/> Leer de un archivo
<input checked="" type="checkbox"/> Entre todos los átomos	<input type="checkbox"/> Calcular

El **USUARIO** introduce el radio de conexión
Radio de conexión 3.7

3. El **SISTEMA** obtiene los **ÁTOMOS** a conectar
El **SISTEMA** lee una tabla de conectividad de un **CONTENEDOR DE CONEXIONES**
El **SISTEMA** valida las **CONEXIONES**
El **SISTEMA** conecta a todos los **ÁTOMOS** de cualquier especie entre sí, dentro del radio de conexión
4. El **SISTEMA** conecta los **ÁTOMOS** (dibuja las conexiones)

9.1.2.1.2.4. CASO DE USO B4: TRANSFORMACIÓN ESPACIAL

1. El **USUARIO** indica al **SISTEMA** que tipo de transformación espacial va a realizar, el dispositivo a usar y el sentido (eje):

Tipo	Dispositivo	Eje / sentido
<input type="checkbox"/> Rotación	<input type="checkbox"/> Perillas	<input type="checkbox"/> X <input type="checkbox"/> Z
<input type="checkbox"/> Traslación	<input type="checkbox"/> Ratón	<input type="checkbox"/> - X <input type="checkbox"/> - Z
<input type="checkbox"/> Escalamiento	<input type="checkbox"/> Teclado	<input type="checkbox"/> Y
	<input type="checkbox"/> Interfaz	<input type="checkbox"/> - Y

2. El **SISTEMA** obtiene el tipo de transformación y el dispositivo y Eje:

Tipo
Dispositivo
Eje

3. El **USUARIO** realiza la **TRANSFORMACIÓN ESPACIAL** con el **DISPOSITIVO** sobre el eje
El **SISTEMA** obtiene las coordenadas de las nuevas posiciones y las envía a la **CELDA COMPUTACIONAL**

4. El **SISTEMA** actualiza la **ESCENA GRÁFICA**

La **CELDA COMPUTACIONAL** pinta la **MOLÉCULA** y/o la **CAJA** y/o el **SISTEMA DE CAVIDADES** y/o la **SUPERFICIE DE POTENCIAL** en la nueva posición.
Los **EJES** se pintan en la nueva posición
El **PUNTERO 3D** se pinta en la nueva posición

ESCENARIO B4.1: TRASLACIÓN CON PERILLAS SOBRE EL EJE X

1. El **USUARIO** indica al **SISTEMA** que tipo de transformación espacial va a realizar, el dispositivo a usar y el sentido (eje):

Tipo	Dispositivo	Eje / sentido	
<input type="checkbox"/> Rotación	<input checked="" type="checkbox"/> Perillas	<input checked="" type="checkbox"/> X	<input type="checkbox"/> Z
<input checked="" type="checkbox"/> Traslación	<input type="checkbox"/> Ratón	<input type="checkbox"/> - X	<input type="checkbox"/> - Z
<input type="checkbox"/> Escalamiento	<input type="checkbox"/> Teclado	<input type="checkbox"/> Y	
	<input type="checkbox"/> Interfaz	<input type="checkbox"/> - Y	

2. El **SISTEMA** obtiene el tipo de transformación, el dispositivo y eje:

Tipo Traslación
 Dispositivo Perillas
 Eje X

3. El **USUARIO** realiza la **TRASLACIÓN** con las **PERILLAS** sobre el eje X

El **SISTEMA** obtiene las coordenadas de las nuevas posiciones y las envía a la **CELDA COMPUTACIONAL**

4. El **SISTEMA** actualiza la **ESCENA GRÁFICA**

ESCENARIO B4.2: ROTACIÓN CON RATON SOBRE EL EJE Y

1. El **USUARIO** indica al **SISTEMA** que tipo de transformación espacial va a realizar, el dispositivo a usar y el sentido (eje):

Tipo	Dispositivo	Eje / sentido	
<input checked="" type="checkbox"/> Rotación	<input type="checkbox"/> Perillas	<input type="checkbox"/> X	<input type="checkbox"/> Z
<input type="checkbox"/> Traslación	<input checked="" type="checkbox"/> Ratón	<input type="checkbox"/> - X	<input type="checkbox"/> - Z
<input type="checkbox"/> Escalamiento	<input type="checkbox"/> Teclado	<input checked="" type="checkbox"/> Y	
	<input type="checkbox"/> Interfaz	<input type="checkbox"/> - Y	

2. El **SISTEMA** obtiene el tipo de transformación, el dispositivo y eje:

Tipo Rotación
 Dispositivo Ratón
 Eje Y

3. El **USUARIO** realiza la **ROTACIÓN** con el **RATÓN** sobre el eje Y

El **SISTEMA** obtiene las coordenadas de las nuevas posiciones y las envía a la **CELDA COMPUTACIONAL**

4. El **SISTEMA** actualiza la **ESCENA GRÁFICA**

9.1.2.1.2.5. CASO DE USO B5: ANIMACIONES

1. Si es necesario:

El **USUARIO** elige animar el **OBJETO DE ANIMACIÓN**

ó El **SISTEMA** determina el **OBJETO DE ANIMACIÓN** inicial:

Molécula Sistema de cavidades Superficie de potencial

2. El **USUARIO** elige el estado inicial del **OBJETO DE ANIMACIÓN**

ó El **SISTEMA** determina el estado inicial o actual del **OBJETO DE ANIMACIÓN**:

Congelado
 Movimiento en sentido normal (izquierda a derecha)
 Movimiento en sentido contrario (derecha a izquierda)

3. Si es necesario:

El **USUARIO** indica la acción a realizar al **SISTEMA**:

Congelar animación
 Sentido normal de animación
 Invertir sentido de animación

4. El **SISTEMA** indica el estado a la **ESCENA GRÁFICA**

5. Si es necesario:

La **ESCENA GRÁFICA** asigna o cambia el estado al **OBJETO DE ANIMACIÓN**

6. La **ESCENA GRÁFICA** dibuja el **OBJETO DE ANIMACIÓN** en el estado indicado

ESCENARIO B5.1: ANIMAR MOLÉCULA

1. El **SISTEMA** determina el **OBJETO DE ANIMACIÓN** inicial:
 Molécula Sistema de cavidades Superficie de potencial
2. El **SISTEMA** determina el estado inicial o actual de la **MOLÉCULA**:
 Congelado
 En movimiento en sentido normal (izquierda a derecha)
 En movimiento en sentido contrario (derecha a izquierda)
3. El **SISTEMA** indica el estado a la **ESCENA GRÁFICA**
4. La **ESCENA GRÁFICA** asigna el estado a la **MOLÉCULA**
5. La **ESCENA GRÁFICA** dibuja la **MOLÉCULA** en movimiento en sentido normal

ESCENARIO B5.2: CONGELAR MOLÉCULA

1. El **USUARIO** elige el **OBJETO DE ANIMACIÓN**:
 Molécula Sistema de cavidades Superficie de potencial
2. El **SISTEMA** determina el estado inicial o actual de la **MOLÉCULA**:
 Congelado
 En movimiento en sentido normal (izquierda a derecha)
 En movimiento en sentido contrario (derecha a izquierda)
3. El **USUARIO** indica la acción a realizar al **SISTEMA**:
 Congelar animación
 Sentido normal de animación
 Invertir sentido de animación
4. El **SISTEMA** indica el estado a la **ESCENA GRÁFICA**
5. La **ESCENA GRÁFICA** cambia el estado a la **MOLÉCULA**
6. La **ESCENA GRÁFICA** dibuja la **MOLÉCULA** congelada (en un cuadro x)

ESCENARIO B5.3: ANIMAR MOLÉCULA EN SENTIDO NORMAL

1. El **USUARIO** elige animar la **MOLÉCULA**
 Molécula Sistema de cavidades Superficie de potencial
2. El **SISTEMA** determina el estado inicial o actual de la **MOLÉCULA**:
 Congelado
 En movimiento en sentido normal (izquierda a derecha)
 En movimiento en sentido contrario (derecha a izquierda)
3. El **USUARIO** indica la acción a realizar al **SISTEMA**:
 Congelar animación
 Sentido normal de animación
 Invertir sentido de animación
4. El **SISTEMA** indica el estado a la **ESCENA GRÁFICA**
5. La **ESCENA GRÁFICA** cambia el estado a la **MOLÉCULA**
6. La **ESCENA GRÁFICA** dibuja la **MOLÉCULA** en movimiento en sentido normal

ESCENARIO B5.4: ANIMAR MOLÉCULA EN SENTIDO CONTRARIO

1. El **USUARIO** elige animar la **MOLÉCULA**
 Molécula Sistema de cavidades Superficie de potencial
2. El **SISTEMA** determina el estado inicial o actual de la **MOLÉCULA**:
 Congelado
 En movimiento en sentido normal (izquierda a derecha)
 En movimiento en sentido contrario (derecha a izquierda)
3. El **USUARIO** indica la acción a realizar al **SISTEMA**:
 Congelar animación
 Sentido normal de animación
 Invertir sentido de animación
4. El **SISTEMA** indica el estado a la **ESCENA GRÁFICA**
5. La **ESCENA GRÁFICA** cambia el estado a la **MOLÉCULA**
6. La **ESCENA GRÁFICA** dibuja la **MOLÉCULA** en movimiento en sentido contrario

ESCENARIO B5.5: ANIMAR MOLÉCULA Y SISTEMA DE CAVIDADES EN SENTIDO CONTRARIO

1. El **USUARIO** elige animar la **MOLÉCULA** y el **SISTEMA DE CAVIDADES**
 Molécula Sistema de cavidades Superficie de potencial
2. El **SISTEMA** determina el estado inicial o actual de la **MOLÉCULA** y EL **SISTEMA DE CAVIDADES**
 SC Congelado
 M En movimiento en sentido normal (izquierda a derecha)
 En movimiento en sentido contrario (derecha a izquierda)
3. El **USUARIO** indica la acción a realizar al **SISTEMA**:
 Congelar animación
 Sentido normal de animación
 Invertir sentido de animación
4. El **SISTEMA** indica los estados a la **ESCENA GRÁFICA**
5. La **ESCENA GRÁFICA** cambia el estado a la **MOLÉCULA** y al **SISTEMA DE CAVIDADES**
6. La **ESCENA GRÁFICA** dibuja la **MOLÉCULA** y el **SISTEMA DE CAVIDADES** en movimiento en sentido contrario

ESCENARIO B5.6: ANIMAR SUPERFICIE DE POTENCIAL

1. El **USUARIO** elige animar el **OBJETO DE ANIMACIÓN**
 Molécula Sistema de cavidades Superficie de potencial
2. El **SISTEMA** determina el estado inicial o actual de la **SUPERFICIE DE POTENCIAL**:
 Congelado
 Movimiento en sentido normal (izquierda a derecha)
 Movimiento en sentido contrario (derecha a izquierda)
3. El **USUARIO** indica la acción a realizar al **SISTEMA**:
 Congelar animación
 Sentido normal de animación
 Invertir sentido de animación
4. El **SISTEMA** indica el estado a la **ESCENA GRÁFICA**
5. La **ESCENA GRÁFICA** cambia el estado a la **SUPERFICIE DE POTENCIAL**
6. La **ESCENA GRÁFICA** dibuja la **SUPERFICIE DE POTENCIAL** en movimiento en sentido normal

9.1.2.1.2.6. CASO DE USO B6: MODIFICAR PROPIEDADES

1. El **USUARIO** indica al **SISTEMA** que propiedad o propiedades desea cambiar
 Color Material Slices Radio (excepto a hoyos)
 Calidad Transparencia Stacks
- ó El **USUARIO** indica al **SISTEMA** que deshaga los cambios efectuados
 ó El **USUARIO** indica al **SISTEMA** que rehaga los cambios efectuados
2. En caso de cambio de propiedades:
 Si no hay selección previa:
 El **SISTEMA** pregunta al **USUARIO** por el tipo de objeto u objetos a los que se les cambiara la(s) propiedad(es) indicada(s)
 El **USUARIO** indica al **SISTEMA** la(s) clase(s) de objeto(s) a modificar
 ÁTOMOS Hoyos Dipolos Conexiones Todos

 El **USUARIO** indica al **SISTEMA** los nuevos valores
 El **SISTEMA** indica a la **ESCENA GRÁFICA** los tipos de objeto, las propiedades a cambiar y los nuevos valores
 La **ESCENA GRÁFICA** cambia las propiedades a todos los elementos de las clases seleccionadas

 Si hay una selección previa:
 El **USUARIO** indica al **SISTEMA** los nuevos valores
 El **SISTEMA** indica a la **ESCENA GRÁFICA** los tipos de objetos, los elementos seleccionados, las propiedades a cambiar y los nuevos valores
 La **ESCENA GRÁFICA** cambia las propiedades sólo a los elementos seleccionados de las clases seleccionadas
3. En caso de deshacer cambios
 El **SISTEMA** deshace cambios hechos a la **ESCENA GRÁFICA**
4. En caso de rehacer cambios
 El **SISTEMA** rehace cambios hechos a la **ESCENA GRÁFICA**
5. El **SISTEMA** dibuja los cambios de la **ESCENA GRÁFICA**

ESCENARIO B6.1: MODIFICAR RADIO A ÁTOMOS SIN SELECCIÓN PREVIA

1. El **USUARIO** indica al **SISTEMA** que propiedad o propiedades desea cambiar
 Color Material Slices Radio (excepto a hoyos)
 Calidad Transparencia Stacks
2. El **SISTEMA** pregunta al **USUARIO** por el tipo de objeto u objetos a los que se les cambiara la(s) propiedad(es) indicada(s)
El **USUARIO** indica al **SISTEMA** la(s) clase(s) de objeto(s) a modificar
 ÁTOMOS Hoyos Dipolos Conexiones Todos

El **USUARIO** indica al **SISTEMA** el nuevo radio 3.2
El **SISTEMA** indica a la **ESCENA GRÁFICA** los tipos de objetos, las propiedades a cambiar y los nuevos valores
La **ESCENA GRÁFICA** cambia el radio a todos los **ÁTOMOS** en la **MOLECULA**
3. El **SISTEMA** dibuja los cambios de la **ESCENA GRÁFICA**

ESCENARIO B6.2: MODIFICAR COLOR A ÁTOMOS CON SELECCIÓN PREVIA

1. El **USUARIO** indica al **SISTEMA** que propiedad o propiedades desea cambiar
 Color Material Slices Radio (excepto a hoyos)
 Calidad Transparencia Stacks
2. El **USUARIO** indica al **SISTEMA** el nuevo color Rojo (255,0,0)
El **SISTEMA** indica a la **ESCENA GRÁFICA** los tipos de objetos, los elementos seleccionados, las propiedades a cambiar y los nuevos valores
La **ESCENA GRÁFICA** cambia el color sólo a los **ÁTOMOS** seleccionados
3. El **SISTEMA** dibuja los cambios de la **ESCENA GRÁFICA**

ESCENARIO B6.3: MODIFICAR TRANSPARENCIA A HOYOS SIN SELECCIÓN PREVIA

1. El **USUARIO** indica al **SISTEMA** que propiedad o propiedades desea cambiar
 Color Material Slices Radio (excepto a hoyos)
 Calidad Transparencia Stacks
2. El **SISTEMA** pregunta al **USUARIO** por el tipo de objeto u objetos a los que se les cambiara la(s) propiedad(es) indicada(s)
El **USUARIO** indica al **SISTEMA** la(s) clase(s) de objeto(s) a modificar
 ÁTOMOS Hoyos Dipolos Conexiones Todos

El **USUARIO** indica al **SISTEMA** el nuevo valor 0.2
El **SISTEMA** indica a la **ESCENA GRÁFICA** los tipos de objetos, las propiedades a cambiar y los nuevos valores
La **ESCENA GRÁFICA** cambia la transparencia a todos los **HOYOS** en la **MOLECULA**
3. El **SISTEMA** dibuja los cambios de la **ESCENA GRÁFICA**

ESCENARIO B6.4: MODIFICAR CALIDAD A CONEXIONES CON SELECCIÓN PREVIA

1. El **USUARIO** indica al **SISTEMA** que propiedad o propiedades desea cambiar
 Color Material Slices Radio (excepto a hoyos)
 Calidad Transparencia Stacks
2. El **USUARIO** indica al **SISTEMA** el nuevo valor 2
El **SISTEMA** indica a la **ESCENA GRÁFICA** los tipos de objetos, los elementos seleccionados, las propiedades a cambiar y los nuevos valores
La **ESCENA GRÁFICA** cambia la calidad sólo a las **CONEXIONES** seleccionadas
3. El **SISTEMA** dibuja los cambios de la **ESCENA GRÁFICA**

ESCENARIO B6.5: MODIFICAR SLICES Y STACKS A TODOS SIN SELECCIÓN PREVIA

1. El **USUARIO** indica al **SISTEMA** que propiedad o propiedades desea cambiar
 Color Material Slices Radio (excepto a hoyos)
 Calidad Transparencia Stacks
2. El **SISTEMA** pregunta al **USUARIO** por el tipo de objeto u objetos a los que se les cambiara la(s) propiedad(es) indicada(s)
 El **USUARIO** indica al **SISTEMA** la(s) clase(s) de objeto(s) a modificar
 ÁTOMOS Hoyos Dipolos Conexiones Todos
 El **USUARIO** indica al **SISTEMA** los nuevos valores
 El **SISTEMA** indica a la **ESCENA GRÁFICA** los tipos de objeto, las propiedades a cambiar y los nuevos valores
 La **ESCENA GRÁFICA** cambia los Slices y Stacks a todos los elementos de las clases seleccionadas
3. El **SISTEMA** dibuja los cambios de la **ESCENA GRÁFICA**

ESCENARIO B6.6: DESHACER CAMBIOS

1. El **USUARIO** indica al **SISTEMA** que deshaga los cambios efectuados
2. El **SISTEMA** deshace cambios hechos a la **ESCENA GRÁFICA**
3. El **SISTEMA** dibuja los cambios de la **ESCENA GRÁFICA**

9.1.2.1.2.7. CASO DE USO B7: EDITAR MOLÉCULA

1. El **USUARIO** indica al **SISTEMA** que acción desea efectuar
 - a) Si hay selección previa:

<input type="checkbox"/> Copiar	<input type="checkbox"/> Agrupar	<input type="checkbox"/> Unir
<input type="checkbox"/> Cortar	<input type="checkbox"/> Desagrupar	<input type="checkbox"/> Desunir
<input type="checkbox"/> Pegar		
 - b) Si no hay selección previa:

<input type="checkbox"/> Agregar			
----------------------------------	--	--	--

 El **USUARIO** indica al **SISTEMA** que objeto desea agregar:
 Átomo Dipolo Conexión Molécula
 Si es agregar átomo:
 El **SISTEMA** proporciona al **USUARIO** una **TABLA PERIÓDICA**
 El **USUARIO** elige el tipo de átomo que desea agregar
 Si es agregar molécula:
 El **SISTEMA** crea una **MOLÉCULA** con los datos de los **CONTENEDORES DE DATOS** disponibles indicados por el **USUARIO**
 ÁTOMOS Dipolos Conexiones
 - c) Si la selección es individual:
 Las opciones Agrupar y Desagrupar no están activadas
 - d) Si la acción es unir átomos:
 El **USUARIO** indica al **SISTEMA** si la unión es:
 Entre dos átomos
 Entre más de dos átomos
 Sólo entre enlace y átomo
 - e) Si la acción es unir molécula
 El **USUARIO** debe seleccionar la(s) molécula(s) que se van a unir
 - f) Si la acción es deshacer:
 EL **SISTEMA** deshace los cambios efectuados
 - g) Si la acción es rehacer:
 EL **SISTEMA** rehace los cambios efectuados
2. El **SISTEMA** obtiene los datos de los objetos seleccionados, los tipos de objetos, y la acción a realizar
 ó El **SISTEMA** crea el **OBJETO GRÁFICO** a agregar
3. El **SISTEMA** manda al **EDITOR MOLECULAR** los tipos de objetos, los datos de los elementos seleccionados o creados y la acción que debe realizarse
1. El **EDITOR MOLECULAR** aplica la acción que debe realizarse sobre la **MOLÉCULA**
5. Si es necesario:
 El **SISTEMA** dibuja los cambios de la **ESCENA GRÁFICA**

ESCENARIO B7.1: AGREGAR ÁTOMOS

1. El **USUARIO** indica al **SISTEMA** que objeto desea agregar:
 Átomo Dipolo Conexión Molécula
2. El **SISTEMA** proporciona al **USUARIO** una **TABLA PERIÓDICA**
El **USUARIO** elige el tipo de átomo que desea agregar
3. El **SISTEMA** crea el **ÁTOMO** a agregar
4. El **SISTEMA** manda al **EDITOR MOLECULAR** los tipos de objetos, los datos creados y la acción que debe realizarse
5. El **EDITOR MOLECULAR** agrega un **ÁTOMO** sobre la **MOLÉCULA**
6. El **SISTEMA** dibuja los cambios de la **ESCENA GRÁFICA**

ESCENARIO B7.2: AGREGAR MOLÉCULAS

1. El **USUARIO** indica al **SISTEMA** que objeto desea agregar:
 Átomo Dipolo Conexión Molécula
2. El **SISTEMA** crea una **MOLÉCULA** con los datos de los contenedores de datos disponibles indicados por el **USUARIO**
 ÁTOMOS **Dipolos** **Conexiones**
3. El **SISTEMA** crea la **MOLÉCULA** a agregar
4. El **SISTEMA** manda al **EDITOR MOLECULAR** los tipos de objetos, los datos creados y la acción que debe realizarse
5. El **EDITOR MOLECULAR** agrega una **MOLÉCULA** sobre la **CELDA COMPUTACIONAL**
6. El **SISTEMA** dibuja los cambios de la **ESCENA GRÁFICA**

ESCENARIO B7.3: COPIAR UN GRUPO DE ÁTOMOS

1. El **USUARIO** indica al **SISTEMA** que acción desea efectuar
 Copiar Agrupar Unir
 Cortar Desagrupar Desunir
 Pegar
2. El **SISTEMA** obtiene los datos de los objetos seleccionados, los tipos de objetos, y la acción a realizar
3. El **SISTEMA** manda al **EDITOR MOLECULAR** los tipos de objetos, los datos de los elementos seleccionados o creados y la acción que debe realizarse
4. El **EDITOR MOLECULAR** copia los **ÁTOMOS** de la **MOLÉCULA**

ESCENARIO B7.4: PEGAR UN GRUPO DE ÁTOMOS

1. El **USUARIO** indica al **SISTEMA** que acción desea efectuar
 Copiar Agrupar Unir
 Cortar Desagrupar Desunir
 Pegar
2. El **SISTEMA** obtiene los datos de los objetos seleccionados, los tipos de objetos, y la acción a realizar
3. El **SISTEMA** manda al **EDITOR MOLECULAR** los tipos de objetos, los datos de los elementos seleccionados o creados y la acción que debe realizarse
4. El **EDITOR MOLECULAR** pega los **ÁTOMOS** de la **MOLÉCULA**
5. El **SISTEMA** dibuja los cambios de la **ESCENA GRÁFICA**

ESCENARIO B7.5: AGRUPAR UN GRUPO DE ÁTOMOS, CONEXIONES Y DIPOLOS

1. El **USUARIO** indica al **SISTEMA** que acción desea efectuar
 Copiar Agrupar Unir
 Cortar Desagrupar Desunir
 Pegar
2. El **SISTEMA** obtiene los datos de los objetos seleccionados, los tipos de objetos, y la acción a realizar
3. El **SISTEMA** manda al **EDITOR MOLECULAR** los tipos de objetos, los datos de los elementos seleccionados o creados y la acción que debe realizarse
4. El **EDITOR MOLECULAR** agrupa los **ÁTOMOS, CONEXIONES Y DIPOLOS** de la **MOLÉCULA**
5. El **SISTEMA** dibuja los cambios de la **ESCENA GRÁFICA**

ESCENARIO B7.6: CORTAR CONEXIONES

1. El **USUARIO** indica al **SISTEMA** que acción desea efectuar
 Copiar Agrupar Unir
 Cortar Desagrupar Desunir
 Pegar
2. El **SISTEMA** obtiene los datos de los objetos seleccionados, los tipos de objetos, y la acción a realizar
3. El **SISTEMA** manda al **EDITOR MOLECULAR** los tipos de objetos, los datos de los elementos seleccionados o creados y la acción que debe realizarse
4. El **EDITOR MOLECULAR** corta las **CONEXIONES** de la **MOLÉCULA**
5. El **SISTEMA** dibuja los cambios de la **ESCENA GRÁFICA**

ESCENARIO B7.7: UNIR ÁTOMOS

1. El **USUARIO** indica al **SISTEMA** que acción desea efectuar
 Copiar Agrupar Unir
 Cortar Desagrupar Desunir
 Pegar
2. El **USUARIO** indica al **SISTEMA** si la unión es:
 Entre dos átomos
 Entre más de dos átomos
 Sólo entre enlace y átomo
3. El **SISTEMA** obtiene los datos de los objetos seleccionados, los tipos de objetos, y la acción a realizar
4. El **SISTEMA** manda al **EDITOR MOLECULAR** los tipos de objetos, los datos de los elementos seleccionados o creados y la acción que debe realizarse
5. El **EDITOR MOLECULAR** une los **ÁTOMOS** de la **MOLÉCULA**
6. El **SISTEMA** dibuja los cambios de la **ESCENA GRÁFICA**

ESCENARIO B7.8: UNIR MOLÉCULAS

1. El **USUARIO** indica al **SISTEMA** que acción desea efectuar
 Copiar Agrupar Unir
 Cortar Desagrupar Desunir
 Pegar
2. El **USUARIO** selecciona la(s) molécula(s) que se van a unir
3. El **SISTEMA** obtiene los datos de los objetos seleccionados, los tipos de objetos, y la acción a realizar
4. El **SISTEMA** manda al **EDITOR MOLECULAR** los tipos de objetos, los datos de los elementos seleccionados o creados y la acción que debe realizarse
5. El **EDITOR MOLECULAR** une las **MOLÉCULAS**
6. El **SISTEMA** dibuja los cambios de la **ESCENA GRÁFICA**

ESCENARIO B7.9: DESHACER CAMBIOS

1. El **SISTEMA** obtiene los datos de los objetos seleccionados, los tipos de objetos, y la acción a realizar
2. El **SISTEMA** manda al **EDITOR MOLECULAR** los tipos de objetos, los datos de los elementos seleccionados o creados y la acción que debe realizarse
3. El **EDITOR MOLECULAR** deshace los cambios
4. El **SISTEMA** dibuja los cambios de la **ESCENA GRÁFICA**

9.1.2.1.2.8. CASO DE USO B8: CREAR ESTRUCTURA CRISTALINA

1. El **USUARIO** indica al **SISTEMA** que multiplique la **CELDA UNITARIA**
2. El **SISTEMA** solicita los factores
3. El **USUARIO** introduce los factores
4. El **SISTEMA** verifica que la **CELDA UNITARIA** sea válida (que sea una base)
 - Si la **CELDA UNITARIA** no es una base
 - El **SISTEMA** se lo indica al **USUARIO**
 - Si la **CELDA UNITARIA** si es una base
 - El **SISTEMA** multiplica la **CELDA UNITARIA**
 - El **SISTEMA** dibuja la **ESTRUCTURA CRISTALINA**

CASO DE USO B8.1: MULTIPLICAR CELDA UNITARIA

1. El **USUARIO** indica al **SISTEMA** que multiplique la **CELDA UNITARIA**
2. El **SISTEMA** solicita los factores
3. El **USUARIO** introduce el factor1 3 factor2 3 factor3 3 ($3^3 = 27$ cubos)
4. El **SISTEMA** verifica que la **CELDA UNITARIA** sea válida (que sea una base)
 - El **SISTEMA** multiplica la **CELDA UNITARIA**
 - El **SISTEMA** dibuja la **ESTRUCTURA CRISTALINA**

9.1.2.1.3. Monitor de Procesos de Cálculo

9.1.2.1.3.1. CASO DE USO C1: MONITOREO Y ACTIVACIÓN DE PROCESOS

1. El **USUARIO** llama al **MONITOR DE PROCESOS DE CÁLCULO**
2. El **SISTEMA** activa el **MONITOR DE PROCESOS DE CÁLCULO**
3. El **MONITOR DE PROCESOS DE CÁLCULO** identifica los **PROCESOS RESIDENTES**:
 - Si hay **PROCESOS RESIDENTES**:
 - El **SISTEMA** muestra información sobre los **PROCESOS RESIDENTES** al **USUARIO**
 - El **SISTEMA** avisa sobre el estado de los **PROCESOS RESIDENTES** al **USUARIO**
 - Si el **PROCESO ACTIVO** concluyo:
 - El **SISTEMA** recibe los datos del **CÁLCULO ASOCIADO**
 - El **SISTEMA** avisa al **USUARIO** que el resultado esta listo para la visualización
 - El **USUARIO** cierra el **PROCESO TERMINADO** para que el **SISTEMA** libere espacio de memoria
 - Si el **USUARIO** suspende el **PROCESO ACTIVO**:
 - El **SISTEMA** avisa al **CÁLCULO ASOCIADO** para que aborte la operación
 - El **SISTEMA** termina el **PROCESO DE CÁLCULO** correspondiente
4. El **USUARIO** elige el tipo de **PROCESO DE CÁLCULO** que desea ejecutar

PROCESOS LOCALES MULTIPLICAR LA CELDA UNITARIA <u> </u> GENERAR RAYOS X <u> </u> GENERAR VECINOS <u> </u> GENERAR ANALISIS DE <u> </u> COORDINACION <u> </u> GENERAR IMAGEN <u> </u>	PROCESOS REMOTOS GENERAR HOYOS <u> </u> GENERAR POTENCIAL <u> </u> GENERAR DIPOLOS <u> </u> GENERAR DINÁMICA <u> </u>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------
5. El **SISTEMA** obtiene los datos de configuración necesarios de los **CONTENEDORES DE DATOS**
 - ó El **SISTEMA** solicita al **USUARIO** que elija los archivos que se necesitan:
 - En el caso del **PROCESO DE GENERAR DINÁMICA**:
 - El **USUARIO** puede generar la configuración de la dinámica
 - El **USUARIO** puede elegir los datos de inicialización de la dinámica
 - Si el **PROCESO ACTIVADO** es un **PROCESO REMOTO**:
 - El **SISTEMA** usa una configuración establecida inicial para la conexión en red
 - ó El **USUARIO** introduce los parámetros para la conexión en red

Dirección IP: <u> </u>	Login: <u> </u>	Password: <u> </u>
-------------------------------------------------	--------------------------------------------------	-----------------------------------------------------
6. El **SISTEMA** ejecuta el **CÁLCULO** elegido
7. El **SISTEMA** crea un **PROCESO ACTIVO**

ESCENARIO C1.1: CÁLCULO DE HOYOS TERMINADO

1. El USUARIO llama al MONITOR DE PROCESOS DE CÁLCULO
2. El SISTEMA activa el MONITOR DE PROCESOS DE CÁLCULO
3. El MONITOR DE PROCESOS DE CÁLCULO identifica los PROCESOS RESIDENTES:
 - El SISTEMA muestra información sobre los PROCESOS RESIDENTES al USUARIO
 - El SISTEMA avisa sobre el estado de los PROCESOS RESIDENTES al USUARIO
 - El SISTEMA recibe los datos del CÁLCULO DE HOYOS
 - El SISTEMA avisa al USUARIO que el resultado del CÁLCULO DE HOYOS esta listo para la visualización
 - El USUARIO cierra el PROCESO CALCULO DE HOYOS para que el SISTEMA libere espacio de memoria

ESCENARIO C1.2: CÁLCULO DE POTENCIALES SUSPENDIDO

1. El USUARIO llama al MONITOR DE PROCESOS DE CÁLCULO
2. El SISTEMA activa el MONITOR DE PROCESOS DE CÁLCULO
3. El MONITOR DE PROCESOS DE CÁLCULO identifica los PROCESOS RESIDENTES:
 - El SISTEMA muestra información sobre los PROCESOS RESIDENTES al USUARIO
 - El SISTEMA avisa sobre el estado de los PROCESOS RESIDENTES al USUARIO
 - El USUARIO suspende el PROCESO DE CALCULO DE POTENCIALES
 - El SISTEMA avisa al CÁLCULO DE POTENCIALES para que aborte la operación
 - El SISTEMA termina el PROCESO DE CÁLCULO DE POTENCIALES

ESCENARIO C1.3: ACTIVACIÓN DEL PROCESO DE CÁLCULO GENERAR IMAGEN

1. El USUARIO llama al MONITOR DE PROCESOS DE CÁLCULO
2. El SISTEMA activa el MONITOR DE PROCESOS DE CÁLCULO
3. El MONITOR DE PROCESOS DE CÁLCULO identifica los PROCESOS RESIDENTES:
 - El SISTEMA muestra información sobre los PROCESOS RESIDENTES al USUARIO
 - El SISTEMA avisa sobre el estado de los PROCESOS RESIDENTES al USUARIO
4. El USUARIO elige el tipo de PROCESO DE CÁLCULO que desea ejecutar

PROCESOS LOCALES	PROCESOS REMOTOS
MULTPLICAR LA CELDA UNITARIA _____	GENERAR HOYOS _____
GENERAR RAYOS X _____	GENERAR POTENCIAL _____
GENERAR VECINOS _____	GENERAR DIPOLOS _____
GENERAR ANALISIS DE COORDINACION _____	
GENERAR IMAGEN <input checked="" type="checkbox"/> _____	GENERAR DINÁMICA _____
5. El SISTEMA obtiene los datos de configuración necesarios del CONTENEDOR DE IMAGEN
6. El SISTEMA ejecuta el CÁLCULO DE IMAGEN
7. El SISTEMA crea un PROCESO ACTIVO

ESCENARIO C1.4: ACTIVACIÓN DEL PROCESO DE CÁLCULO GENERAR POTENCIAL

1. El USUARIO llama al MONITOR DE PROCESOS DE CÁLCULO
2. El SISTEMA activa el MONITOR DE PROCESOS DE CÁLCULO
3. El MONITOR DE PROCESOS DE CÁLCULO identifica los PROCESOS RESIDENTES:
 - El SISTEMA muestra información sobre los PROCESOS RESIDENTES al USUARIO
 - El SISTEMA avisa sobre el estado de los PROCESOS RESIDENTES al USUARIO
4. El USUARIO elige el tipo de PROCESO DE CÁLCULO que desea ejecutar

PROCESOS LOCALES	PROCESOS REMOTOS
MULTPLICAR LA CELDA UNITARIA _____	GENERAR HOYOS _____
GENERAR RAYOS X _____	GENERAR POTENCIAL <input checked="" type="checkbox"/> _____
GENERAR VECINOS _____	GENERAR DIPOLOS _____
GENERAR ANALISIS DE COORDINACION _____	
GENERAR IMAGEN _____	GENERAR DINÁMICA _____

5. El **SISTEMA** obtiene los datos de configuración necesarios del **CONTENEDOR DE PARTÍCULAS**
El **SISTEMA** usa una configuración establecida inicial para la conexión en red

Dirección IP: 132.148.165.40 Login: lja Password: XXXXXXXXXX

6. El **SISTEMA** ejecuta el **CÁLCULO DE POTENCIALES**
7. El **SISTEMA** crea un **PROCESO ACTIVO**

ESCENARIO C1.5: ACTIVACIÓN DEL PROCESO DE CÁLCULO GENERAR DIPOLOS

1. El **USUARIO** llama al **MONITOR DE PROCESOS DE CÁLCULO**
2. El **SISTEMA** activa el **MONITOR DE PROCESOS DE CÁLCULO**
3. El **MONITOR DE PROCESOS DE CÁLCULO** identifica los **PROCESOS RESIDENTES**:
El **SISTEMA** muestra información sobre los **PROCESOS RESIDENTES** al **USUARIO**
El **SISTEMA** avisa sobre el estado de los **PROCESOS RESIDENTES** al **USUARIO**

4. El **USUARIO** elige el tipo de **PROCESO DE CÁLCULO** que desea ejecutar

PROCESOS LOCALES	PROCESOS REMOTOS
MULTIPlicAR LA CELDA UNITARIA <u> </u>	GENERAR HOYOS <u> </u>
GENERAR RAYOS X <u> </u>	GENERAR POTENCIAL <u> </u>
GENERAR VECINOS <u> </u>	GENERAR DIPOLOS <u> ✓ </u>
GENERAR ANALISIS DE <u> </u>	
COORDINACION <u> </u>	
 GENERAR IMAGEN <u> </u>	 GENERAR DINÁMICA <u> </u>

5. El **SISTEMA** obtiene los datos de configuración necesarios del **CONTENEDOR DE PARTÍCULAS**
El **USUARIO** introduce los parámetros para la conexión en red
Dirección IP: 132.148.165.40 Login: becarios Password: XXXXXXXXXX

6. El **SISTEMA** ejecuta el **CÁLCULO DE DIPOLOS**
7. El **SISTEMA** crea un **PROCESO ACTIVO**

ESCENARIO C1.6: ACTIVACIÓN DEL PROCESO DE CÁLCULO GENERAR DINÁMICA

1. El **USUARIO** llama al **MONITOR DE PROCESOS DE CÁLCULO**
2. El **SISTEMA** activa el **MONITOR DE PROCESOS DE CÁLCULO**
3. El **MONITOR DE PROCESOS DE CÁLCULO** identifica los **PROCESOS RESIDENTES**:
El **SISTEMA** muestra información sobre los **PROCESOS RESIDENTES** al **USUARIO**
El **SISTEMA** avisa sobre el estado de los **PROCESOS RESIDENTES** al **USUARIO**

4. El **USUARIO** elige el tipo de **PROCESO DE CÁLCULO** que desea ejecutar

PROCESOS LOCALES	PROCESOS REMOTOS
MULTIPlicAR LA CELDA UNITARIA <u> </u>	GENERAR HOYOS <u> </u>
GENERAR RAYOS X <u> </u>	GENERAR POTENCIAL <u> </u>
GENERAR VECINOS <u> </u>	GENERAR DIPOLOS <u> </u>
GENERAR ANALISIS DE <u> </u>	
COORDINACION <u> </u>	
 GENERAR IMAGEN <u> </u>	 GENERAR DINÁMICA <u> ✓ </u>

5. El **SISTEMA** solicita al **USUARIO** que elija los archivos que se necesitan:
ARCHIVO DE CONFIGURACIÓN DE DINÁMICA
ARCHIVO DE INICIALIZACIÓN DE DINÁMICA
El **USUARIO** genera la configuración de la dinámica ✓
El **USUARIO** elige los datos de inicialización de la dinámica ✓

El **SISTEMA** usa una configuración establecida inicial para la conexión en red

Dirección IP: 132.148.165.40 Login: lja Password: XXXXXXXXXX

6. El **SISTEMA** ejecuta el **CÁLCULO GENERAR DINÁMICA**
7. El **SISTEMA** crea un **PROCESO ACTIVO**

9.1.2.1.4. Impresiones

9.1.2.1.4.1. CASO DE USO D1: IMPRESIÓN DE DOCUMENTOS

1. El **USUARIO** elige el tipo del **DOCUMENTO** a imprimir:

ARCHIVO DE CONFIGURACIÓN	_____	IMAGEN	_____	MANUAL	_____
PARTICULAS	_____	IMAGEN DE DINÁMICA	_____	DE USUARIO	_____
HOYOS	_____	IMAGEN DE GRÁFICA	_____		
DIPÓLOS	_____				
POTENCIALES	_____				
RAYOS X	_____				
CONEXIONES	_____				
INICIALIZACIÓN DE DINÁMICA	_____				
CONFIGURACIÓN DE DINÁMICA	_____				

2. El **USUARIO** establece la **CONFIGURACIÓN DE IMPRESIÓN**

3. El **SISTEMA** verifica el estado de la impresora

Si no hay problemas:

IMPRESIONES verifica el tipo del **DOCUMENTO**

IMPRESIONES solicita los datos al **CONTENEDOR DE DATOS** correspondiente

Si es un **ARCHIVO DE CONFIGURACIÓN** o el **MANUAL**:

IMPRESIONES imprime el **DOCUMENTO** en modo texto inicialmente

Si es una **IMAGEN**:

IMPRESIONES imprime la **IMAGEN** usando un formato de PostScript inicialmente

IMPRESIONES notifica al **SISTEMA** sobre el resultado

Si hay problemas:

IMPRESIONES notifica al **SISTEMA** el tipo de problema

4. El **SISTEMA** avisa al **USUARIO** que la impresión fue realizada o que hubo problemas

ESCENARIO D1.1: IMPRESIÓN DE IMAGEN DE DINÁMICA

1. El **USUARIO** elige el tipo del **DOCUMENTO** a imprimir:

ARCHIVO DE CONFIGURACIÓN	_____	IMAGEN	_____	MANUAL	_____
PARTICULAS	_____	IMAGEN DE DINÁMICA	<u> ✓ </u>	DE USUARIO	_____
HOYOS	_____	IMAGEN DE GRÁFICA	_____		
DIPÓLOS	_____				
POTENCIALES	_____				
RAYOS X	_____				
CONEXIONES	_____				
INICIALIZACIÓN DE DINÁMICA	_____				
CONFIGURACIÓN DE DINÁMICA	_____				

2. El **USUARIO** establece la **CONFIGURACIÓN DE IMPRESIÓN**

3. El **SISTEMA** verifica el estado de la impresora

Si no hay problemas:

IMPRESIONES verifica el tipo de **IMAGEN**

IMPRESIONES solicita los datos al **CONTENEDOR DE IMAGEN**

IMPRESIONES imprime la **IMAGEN** usando un formato de PostScript inicialmente

IMPRESIONES notifica al **SISTEMA** sobre el resultado

Si hay problemas:

IMPRESIONES notifica al **SISTEMA** el tipo de problema

4. El **SISTEMA** avisa al **USUARIO** que la impresión fue realizada o que hubo problemas

ESCENARIO D1.2: *IMPRESIÓN DE ARCHIVO DE PARTÍCULAS*

1. El **USUARIO** elige el tipo del **DOCUMENTO** a imprimir:

ARCHIVO DE CONFIGURACIÓN		IMAGEN		MANUAL
PARTÍCULAS	<input checked="" type="checkbox"/>	IMAGEN DE DINÁMICA	<input type="checkbox"/>	DE USUARIO
HOYOS	<input type="checkbox"/>	IMAGEN DE GRÁFICA	<input type="checkbox"/>	
DIPOLOS	<input type="checkbox"/>			
POTENCIALES	<input type="checkbox"/>			
RAYOS X	<input type="checkbox"/>			
CONEXIONES	<input type="checkbox"/>			
INICIALIZACIÓN DE DINÁMICA	<input type="checkbox"/>			
CONFIGURACIÓN DE DINÁMICA	<input type="checkbox"/>			
2. El **USUARIO** establece la **CONFIGURACIÓN DE IMPRESIÓN**
3. El **SISTEMA** verifica el estado de la impresora
 - Si no hay problemas:
 - IMPRESIONES** verifica el tipo del **DOCUMENTO**
 - IMPRESIONES** solicita los datos al **CONTENEDOR DE PARTÍCULAS**
 - IMPRESIONES** imprime el **ARCHIVO DE PARTÍCULAS** en modo texto inicialmente
 - IMPRESIONES** notifica al **SISTEMA** sobre el resultado
 - Si hay problemas:
 - IMPRESIONES** notifica al **SISTEMA** el tipo de problema
4. El **SISTEMA** avisa al **USUARIO** que la impresión fue realizada o que hubo problemas

9.1.2.2. Identificar eventos entre objetos (identificar sucesos = señales, entradas, decisiones, interrupciones, transiciones, acciones externas, condiciones de error)

9.1.2.2.1. ESCENARIO A1.1: OBTENCION DE DATOS DE PARTICULAS

SISTEMA	Obtener nombre de Solicitar nombre de archivo de particulas a	Archivo de particulas USUARIO
USUARIO	Seleccionar nombre de	Archivo de particulas
CONTENEDOR DE PARTICULAS	Verificar tipo de Leer Obtener datos de particulas Traducir datos de particulas Colocar datos de particulas en un vector de datos Devolver un vector de datos de particulas Devolver archivo de particulas	Archivo de particulas Archivo de particulas IMPRESIONES MOLECULA MONITOR DE PROCESOS MONITOR DE PROCESOS
SISTEMA	Devolver archivo de particulas a	USUARIO

9.1.2.2.2. ESCENARIO A2.1: RENOMBRAR ARCHIVO DE PARTICULAS

USUARIO	Elegir para renombrar	Archivo de particulas
SISTEMA	Solicitar nuevo nombre para el archivo de particulas a	USUARIO
USUARIO	Introducir nuevo nombre para	Archivo de particulas
CONTENEDOR DE PARTICULAS	Verificar nuevo nombre de Cambiar nombre anterior por nombre nuevo	Archivo de particulas
SISTEMA	Notificar a	USUARIO

9.1.2.2.3. SCENARIO A3.1: GUARDAR ARCHIVO DE PARTICULAS

USUARIO	Elegir para guardar	Archivo de particulas
CONTENEDOR DE PARTICULAS	Guardar los datos en	Archivo de particulas
SISTEMA	Notificar sobre el resultado	USUARIO

9.1.2.2.4. ESCENARIO A4.1: GUARDAR CON OTRO NOMBRE ARCHIVO DE PARTÍCULAS

USUARIO	Elegir configuración de partículas para guardar con otro nombre en	Archivo de partículas
SISTEMA	Solicitar nuevo nombre para el archivo de partículas a	USUARIO
USUARIO	Introducir nuevo nombre para	Archivo de partículas
SISTEMA	Validar nuevo nombre de Crear un CONTENEDOR DE PARTÍCULAS Copiar la configuración de partículas al nuevo CONTENEDOR DE PARTÍCULAS Pasarle el nombre al nuevo CONTENEDOR DE PARTÍCULAS	Archivo de partículas
CONTENEDOR DE PARTICULAS	Cambiar nombre a Guardar datos en Cerrar sin guardar cambios	Archivo de partículas Archivo de partículas Archivo de partículas
SISTEMA	Eliminar CONTENEDOR DE PARTÍCULAS de la memoria Indicar resultado de la operación a	USUARIO

9.1.2.2.5. ESCENARIO B1.1: INICIAR ESCENA GRÁFICA EJEMPLO 1

SISTEMA	Reconocer configuración disponible (Hay un Archivo de partículas) Obtener el tipo y datos de caja de (Caja cúbica) Obtener datos de configuración de PARTÍCULAS de Crear una ESCENA GRÁFICA Indicar que la caja es cúbica a Acomodar datos de configuración de PARTÍCULAS en Dibujar MOLÉCULA con apariencia inicial	CONTENEDOR DE PARTÍCULAS CONTENEDOR DE PARTÍCULAS CELDA COMPUTACIONAL MOLÉCULA
---------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------

9.1.2.2.6. ESCENARIO B1.2: INICIAR ESCENA GRÁFICA EJEMPLO 2

<p>SISTEMA</p>	<p>Reconocer configuración disponible (Hay un Archivo de partículas) (Hay un Archivo de hoyos) (Hay un Archivo de potenciales) (Hay un Archivo de dipolos)</p> <p>Obtener el tipo y datos de caja de (Caja cúbica)</p> <p>Obtener datos de configuración de PARTÍCULAS de Obtener datos de configuración de HOYOS de Obtener datos de configuración de POTENCIALES de Obtener datos de configuración de DIPOLOS de</p> <p>Crear una ESCENA GRÁFICA Indicar que la caja es cúbica a Acomodar datos de configuración de PARTÍCULAS en Acomodar datos de configuración de DIPOLOS en Acomodar datos de configuración de HOYOS en Acomodar datos de configuración de POTENCIALES en</p> <p>Dibujar MOLÉCULA con apariencia inicial</p>	<p>CONTENEDOR DE PARTÍCULAS</p> <p>CONTENEDOR DE PARTÍCULAS CONTENEDOR DE HOYOS CONTENEDOR DE POTENCIALES CONTENEDOR DE DIPOLOS</p> <p>CELDA COMPUTACIONAL MOLÉCULA</p> <p>SISTEMA DE CAVIDADES SUPERFICIE POTENCIAL</p> <p>MOLÉCULA</p>
-----------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

9.1.2.2.7. ESCENARIO B2.1: SELECCIONAR ÁTOMOS POR ESPECIE

<p>USUARIO</p>	<p>Indicar modo de selección y tipo de objeto a seleccionar (Especie Oxígeno, tipo átomos)</p>	<p>SISTEMA</p>
<p>SISTEMA</p>	<p>Proporcionar modo y tipo a</p>	<p>INDICADOR DE SELECCIÓN</p>
<p>INDICADOR DE SELECCIÓN</p>	<p>Marcar átomos indicados (Sólo átomos de una especie) (especie oxígeno) Obtener etiquetas de átomos seleccionados Devolver etiquetas a</p>	<p>ÁTOMOS</p> <p>ÁTOMOS SISTEMA</p>

9.1.2.2.8. ESCENARIO B2.3: SELECCIONAR ELEMENTOS DE TODOS LOS TIPOS POR GRUPO CON ESFERA

<p>USUARIO</p>	<p>Indicar modo de selección y tipo de objeto a seleccionar (Grupo-esfera, todos los tipos: átomos, dipolos, conexiones y hoyos)</p>	<p>SISTEMA</p>
<p>SISTEMA</p>	<p>Proporcionar modo y tipo a</p>	<p>INDICADOR DE SELECCIÓN</p>
<p>INDICADOR DE SELECCIÓN</p>	<p>Marcar elementos indicados (que sean seleccionados con una esfera) Obtener etiquetas de elementos seleccionados Devolver etiquetas a</p>	<p>ÁTOMOS, DIPOLOS, CONEXIONES Y HOYOS</p> <p>ÁTOMOS, DIPOLOS, CONEXIONES Y HOYOS SISTEMA</p>

9.1.2.2.9. ESCENARIO B3.1: CONEXIÓN DE ÁTOMOS ENTRE ESPECIES DESDE UN ARCHIVO

USUARIO	Indicar pintar CONEXIONES a	SISTEMA
SISTEMA	Solicitar radio de conexión, tipo de conexión, origen de la tabla de conectividad y las especies a conectar	USUARIO
USUARIO	Introducir radio de conexión, tipo de conexión, origen de la tabla de conectividad y las especies a conectar (radio 4.5, entre especies, leer tabla de archivo, Oxígeno-Aluminio, Oxígeno-Silicio)	SISTEMA
SISTEMA	Obtener ÁTOMOS a conectar de Leer tabla de conectividad de un Validar CONEXIONES (sólo conectar átomos de una especie con átomos de otra especie diferente dentro del radio de conexión, no hay conexión entre átomos de la misma especie) Conectar ÁTOMOS (dibujar CONEXIONES)	INDICADOR DE SELECCIÓN CONTENEDOR DE CONEXIONES

9.1.2.2.10. ESCENARIO B4.2: ROTACIÓN CON RATON SOBRE EL EJE Y

USUARIO	Indicar tipo de transformación, dispositivo y eje a (Rotación, Ratón, Eje Y)	SISTEMA
SISTEMA	Obtener tipo de transformación, dispositivo y eje	USUARIO
USUARIO	Realizar rotación con ratón sobre el eje Y	ESCENA GRAFICA
SISTEMA	Solicitar coordenadas de las posiciones de objetos gráficos a Enviar coordenadas a Actualizar ESCENA GRÁFICA	INDICADOR DE SELECCIÓN ESCENA GRAFICA

9.1.2.2.11. ESCENARIO B5.1: ANIMAR MOLÉCULA

SISTEMA	Determinar objeto de animación Determinar estado inicial o actual de MOLÉCULA (En movimiento en sentido normal (izquierda a derecha)) Indicar estado para MOLÉCULA a	ESCENA GRAFICA
ESCENA GRAFICA	Asignar estado de animación a Dibujar la MOLÉCULA en movimiento en sentido normal	MOLÉCULA

9.1.2.2.12. ESCENARIO B5.5: ANIMAR MOLÉCULA Y SISTEMA DE CAVIDADES EN SENTIDO CONTRARIO

USUARIO	Elegir Objeto de animación (Molécula y Sistema de Cavidades)	
SISTEMA	Determinar estado inicial o actual de MOLÉCULA (En movimiento en sentido normal (izquierda a derecha)) Determinar estado inicial o actual de SISTEMA DE CAVIDADES (Congelado)	
USUARIO	Indicar acción a realizar a (Invertir sentido de animación (derecha a izquierda))	SISTEMA
SISTEMA	Indicar estados a	ESCENA GRAFICA
ESCENA GRAFICA	Cambiar estado de animación a Cambiar estado de animación a Dibujar la MOLÉCULA en movimiento en sentido contrario (derecha a izquierda) Dibujar la SISTEMA DE CAVIDADES en movimiento en sentido contrario (derecha a izquierda)	MOLÉCULA SISTEMA DE CAVIDADES

9.1.2.2.13. ESCENARIO B6.2: MODIFICAR COLOR A ÁTOMOS CON SELECCIÓN PREVIA

USUARIO	Indicar propiedades a cambiar a (Color)	SISTEMA
SISTEMA	Preguntar por tipo de objetos y coordenadas a	INDICADOR DE SELECCIÓN
USUARIO	Indicar nuevo color a (Rojo)	SISTEMA
SISTEMA	Indicar tipos de objetos, elementos seleccionados, propiedades a cambiar y nuevos valores a	ESCENA GRAFICA
ESCENA GRAFICA	Cambiar color sólo a los átomos seleccionados	
SISTEMA	Dibujar los cambios de la ESCENA GRAFICA	

9.1.2.2.14. ESCENARIO B7.1: AGREGAR ÁTOMOS

USUARIO	Indicar tipo de objeto a agregar a (Átomo)	SISTEMA
SISTEMA	Proporcionar una <i>Tabla Periódica</i> a	USUARIO
USUARIO	Elegir el tipo de átomo a agregar	
SISTEMA	Crear átomo del tipo indicado Mandar tipo de objeto, objeto creado y la acción a realizar a	EDITOR MOLECULAR

EDITOR MOLECULAR	Agregar un ÁTOMO sobre la MOLÉCULA	
SISTEMA	Dibujar los cambios de la ESCENA GRAFICA	

9.1.2.2.15. ESCENARIO C1.3: ACTIVACIÓN DEL PROCESO DE CÁLCULO GENERAR IMAGEN

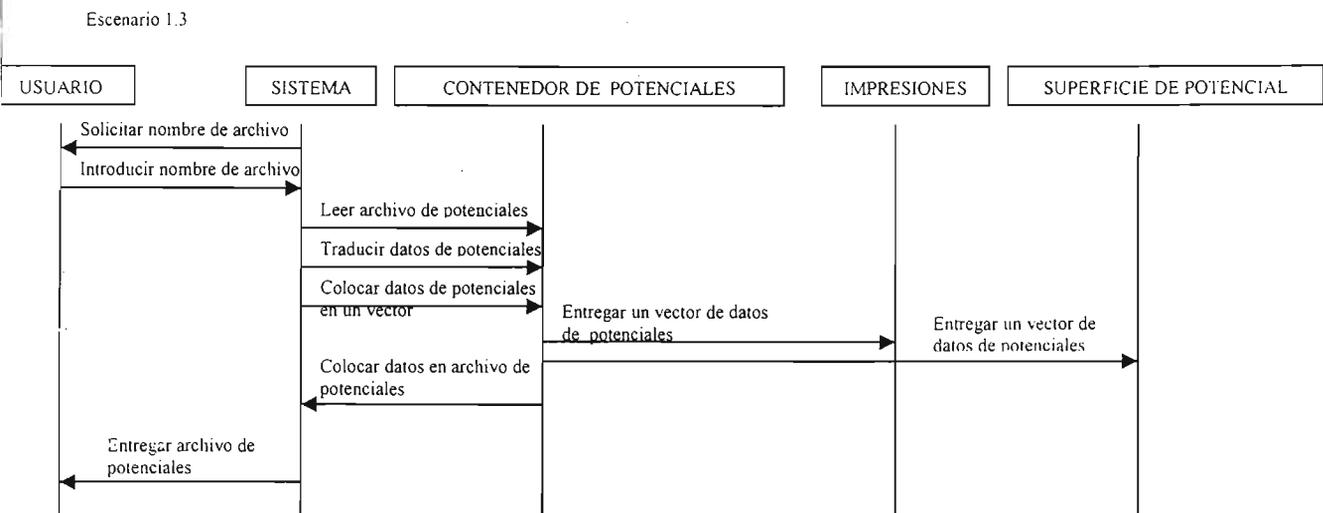
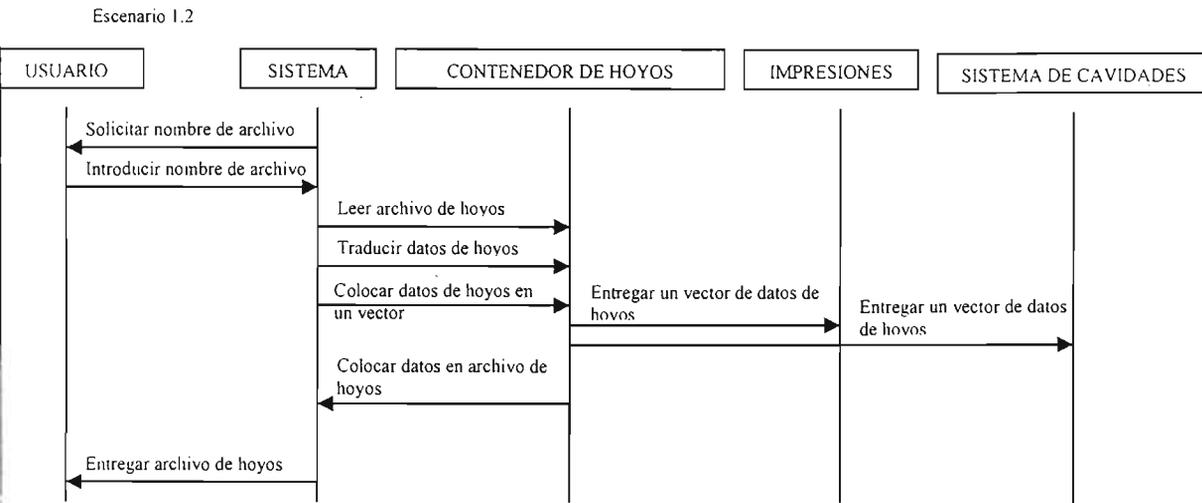
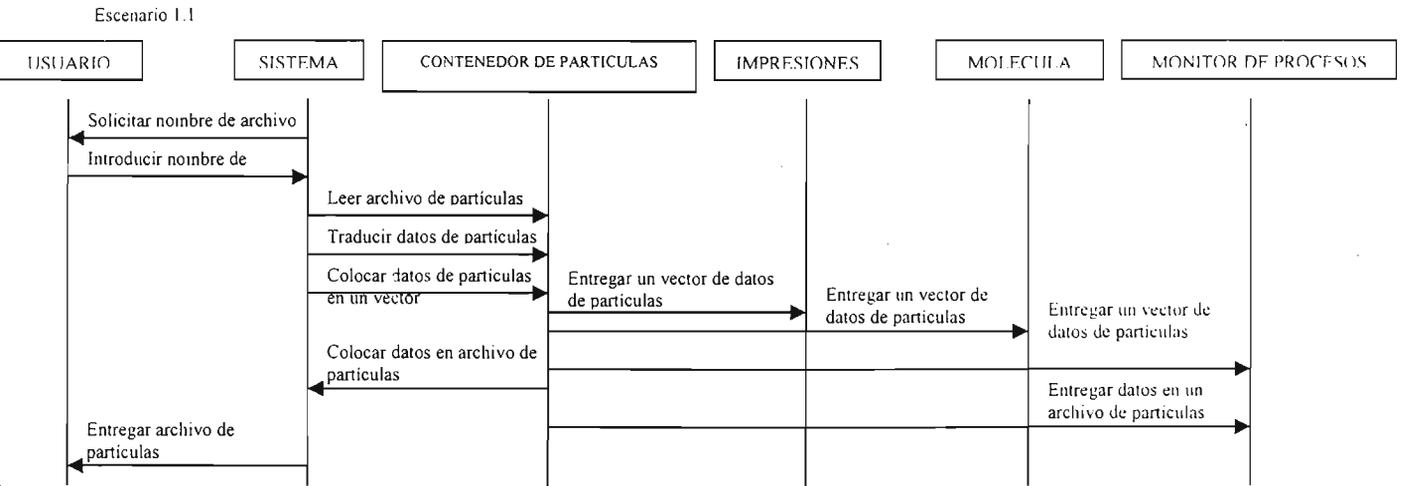
USUARIO	Llamar a	MONITOR DE PROCESOS
SISTEMA	Activar	MONITOR DE PROCESOS
MONITOR DE PROCESOS	Identificar <i>procesos residentes</i>	
SISTEMA	Mostrar información sobre <i>procesos residentes</i> a Avisar sobre el estado de los <i>procesos residentes</i> a	USUARIO USUARIO
USUARIO	Elegir tipo de PROCESO DE CÁLCULO a ejecutar (Generar imagen)	
SISTEMA	Ejecutar CÁLCULO DE IMAGEN Crear un <i>proceso activo</i>	

9.1.2.2.16. ESCENARIO D1.1: IMPRESIÓN DE IMAGEN DE DINÁMICA

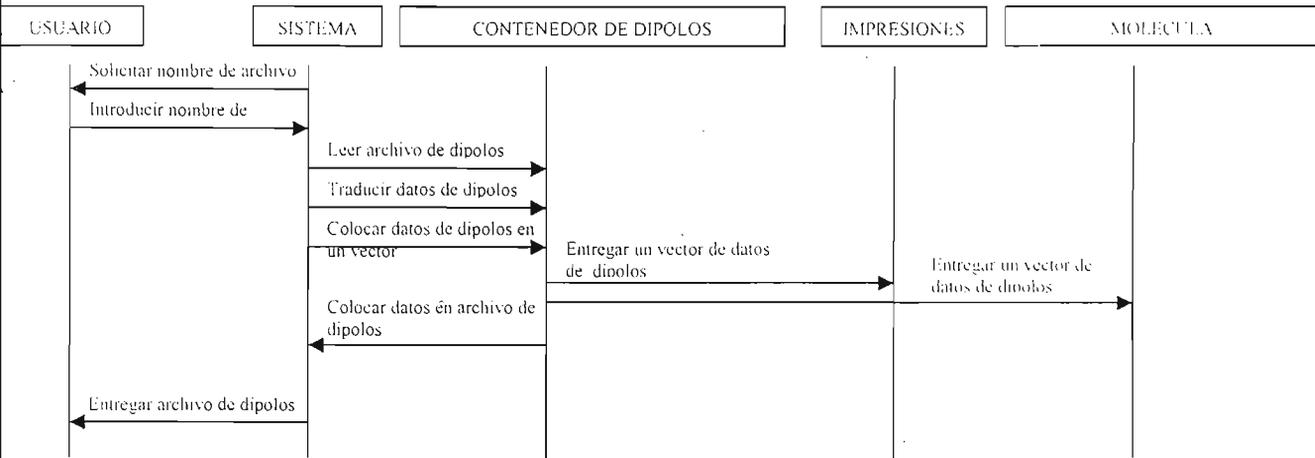
USUARIO	Elegir documento a imprimir (Imagen de dinámica) Establecer configuración de impresión	
SISTEMA	Verificar estado de	<i>Impresora</i>
IMPRESIONES	Sin problemas: Verificar tipo de Solicitar datos a Imprimir usando un formato de PostScript inicialmente Notificar sobre el resultado a Con problemas: Notificar el tipo de problema a	<i>Imagen</i> CONTENEDOR DE IMAGEN <i>Imagen</i> SISTEMA SISTEMA
SISTEMA	Avisar que la impresión fue hecha o que hubo problemas	USUARIO

9.1.2.3. Diagrama de eventos para cada escenario

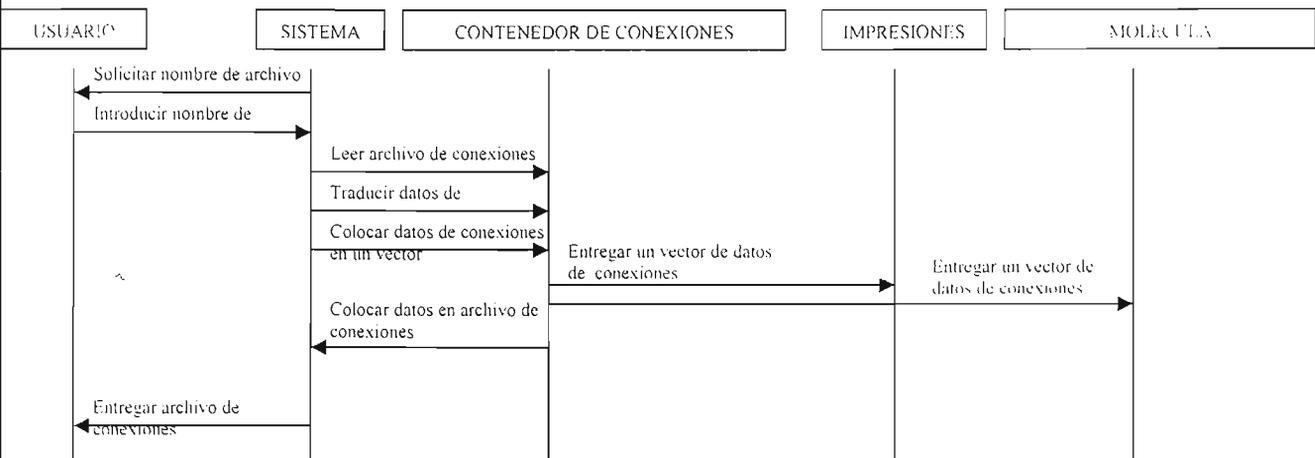
9.1.2.3.1. Manejo de archivos



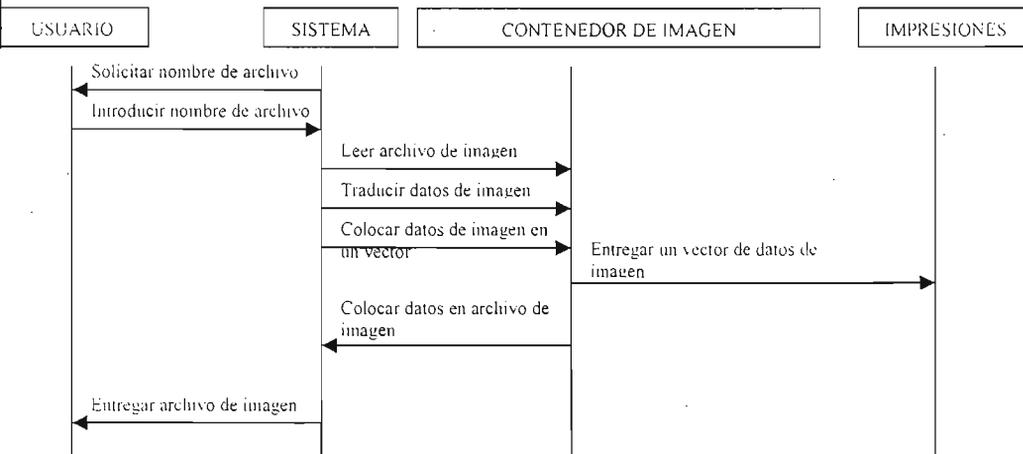
Escenario 1.4



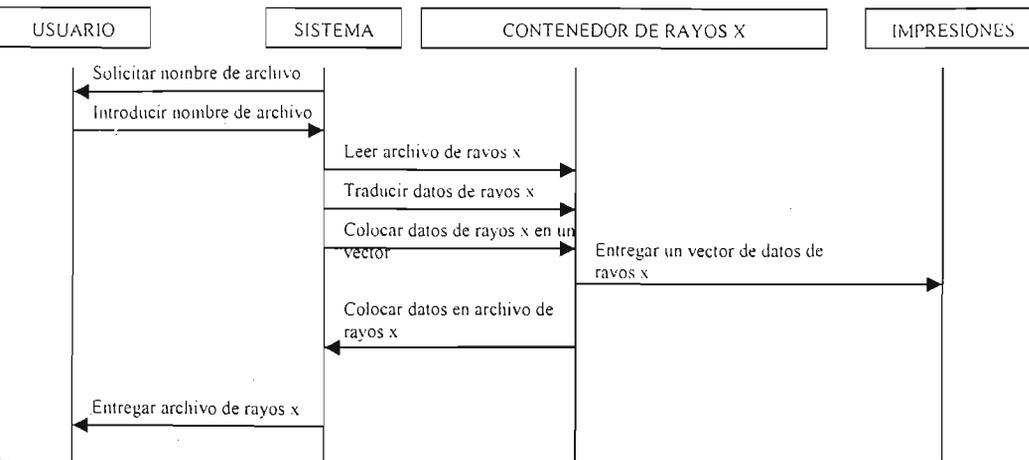
Escenario 1.5



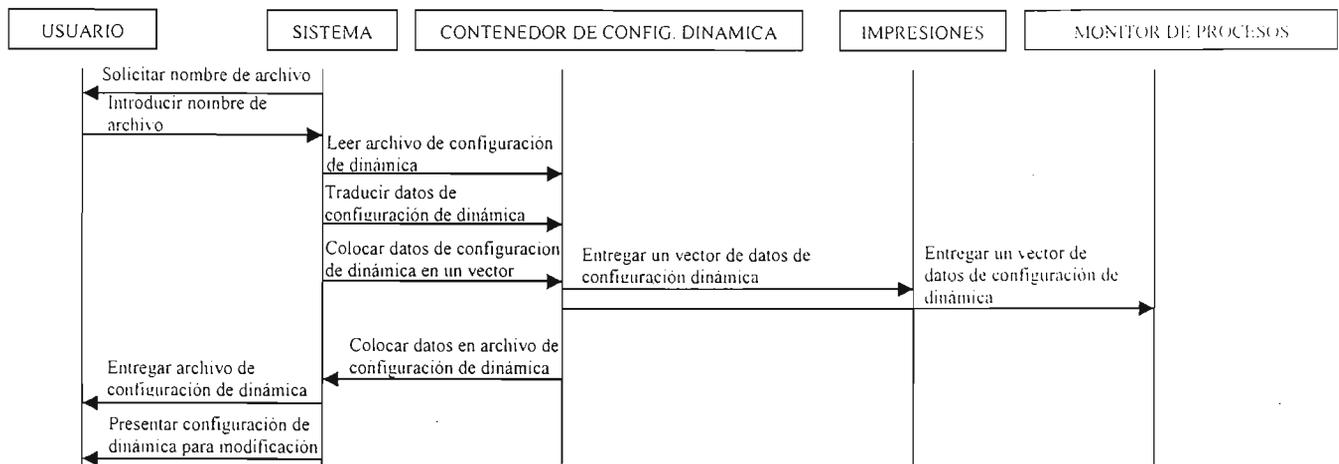
Escenario 1.6



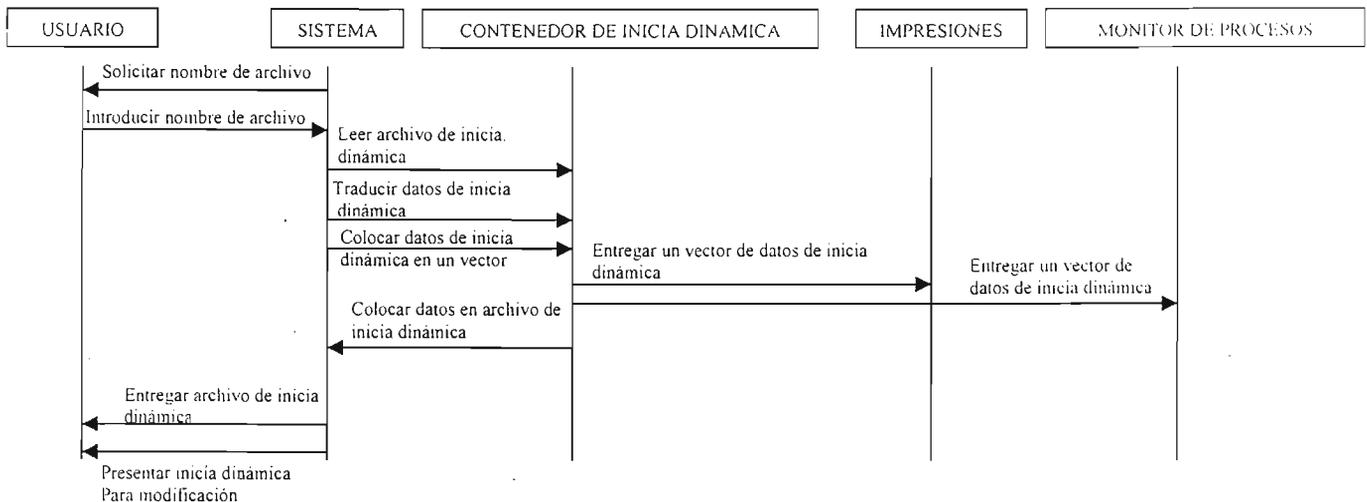
Escenario 1.7



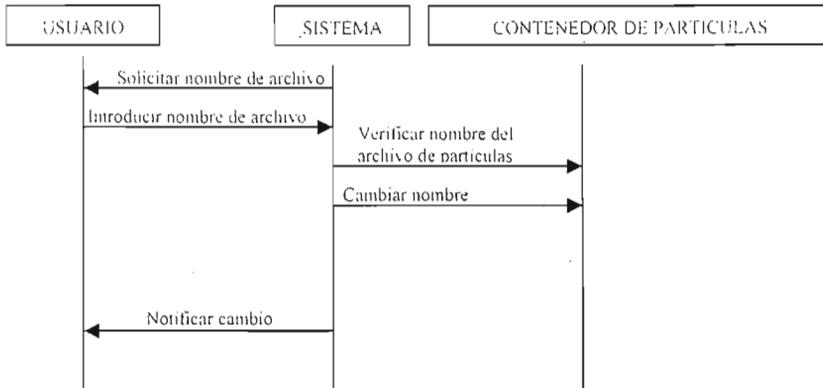
Escenario 1.8



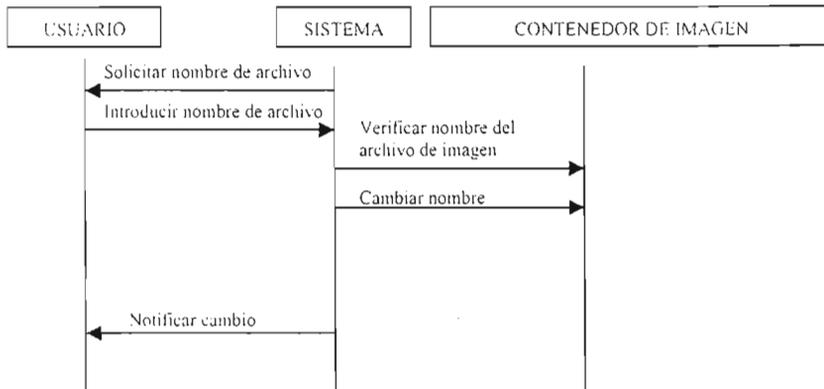
Escenario 1.9



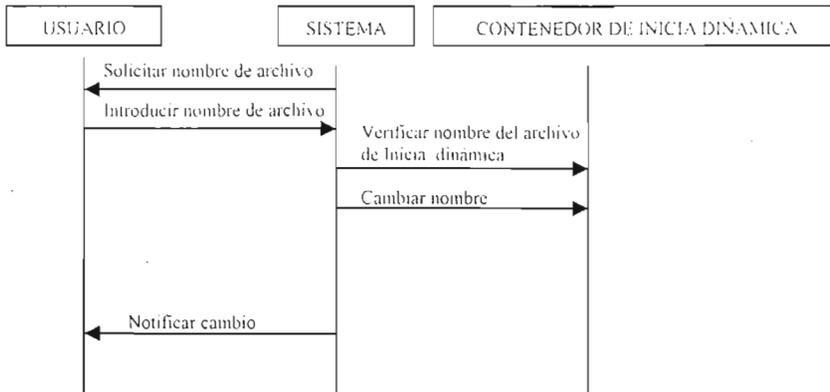
Escenario 2.1



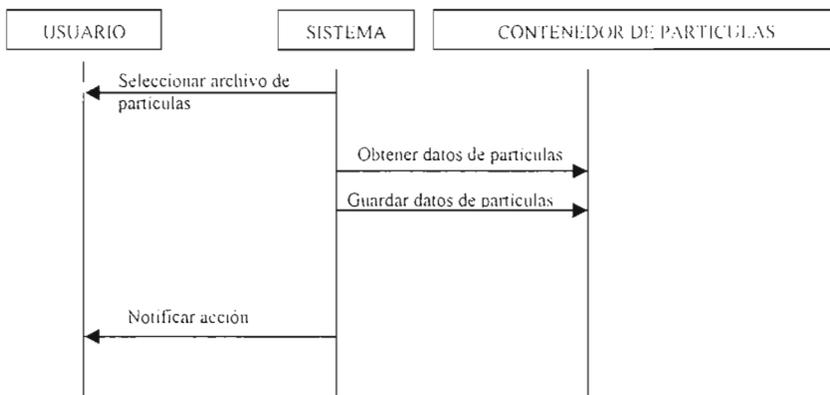
Escenario 2.2



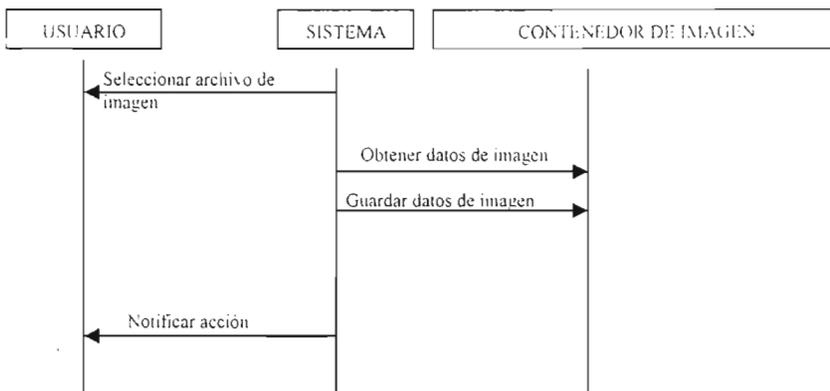
Escenario 2.3



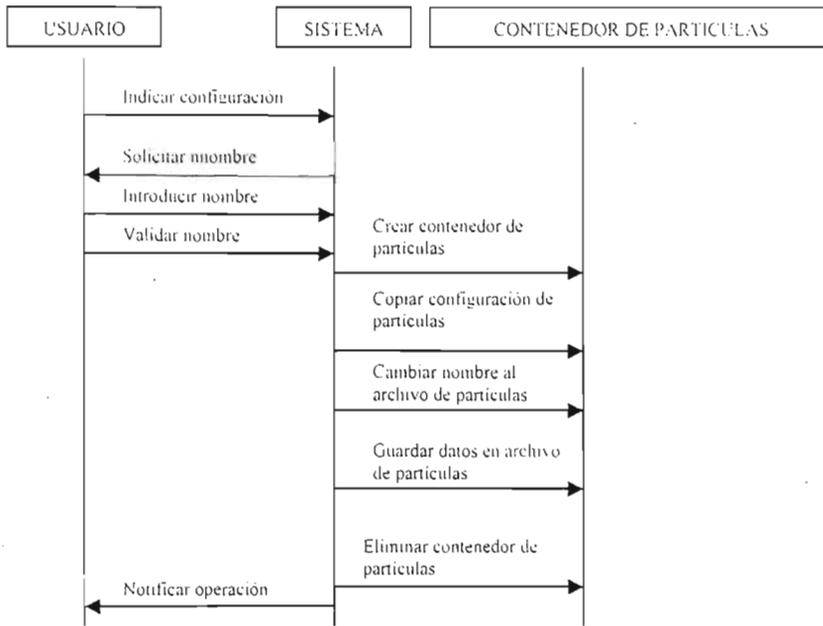
Escenario 3.1



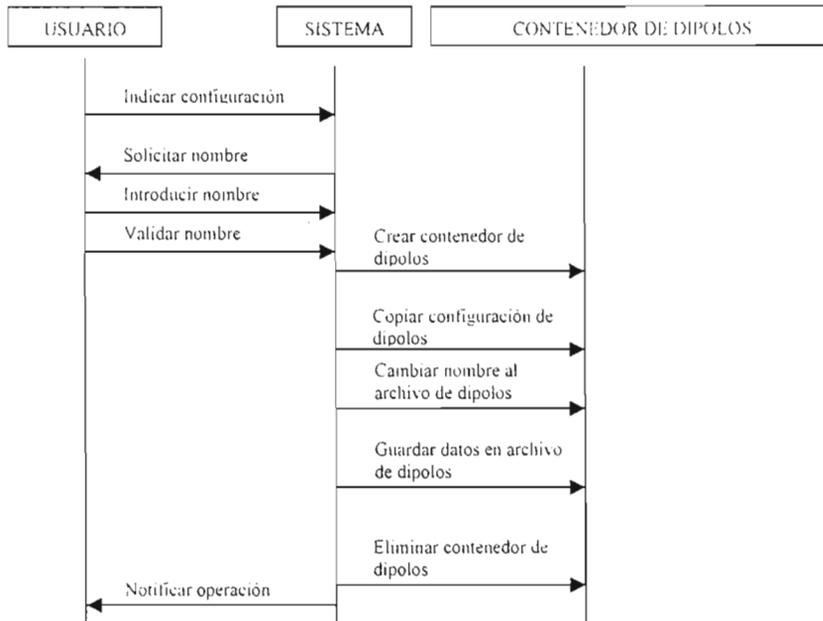
Escenario 3.2

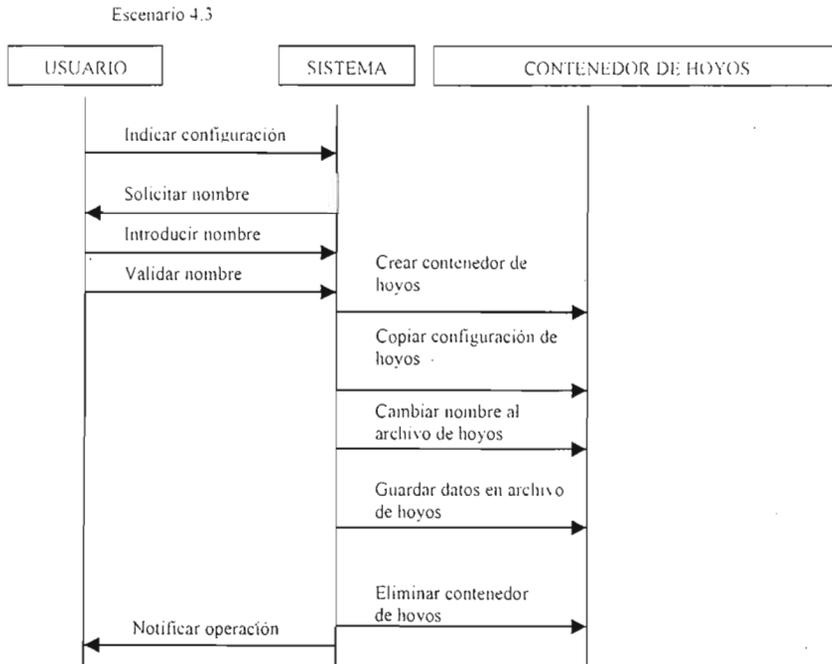


Escenario 4.1

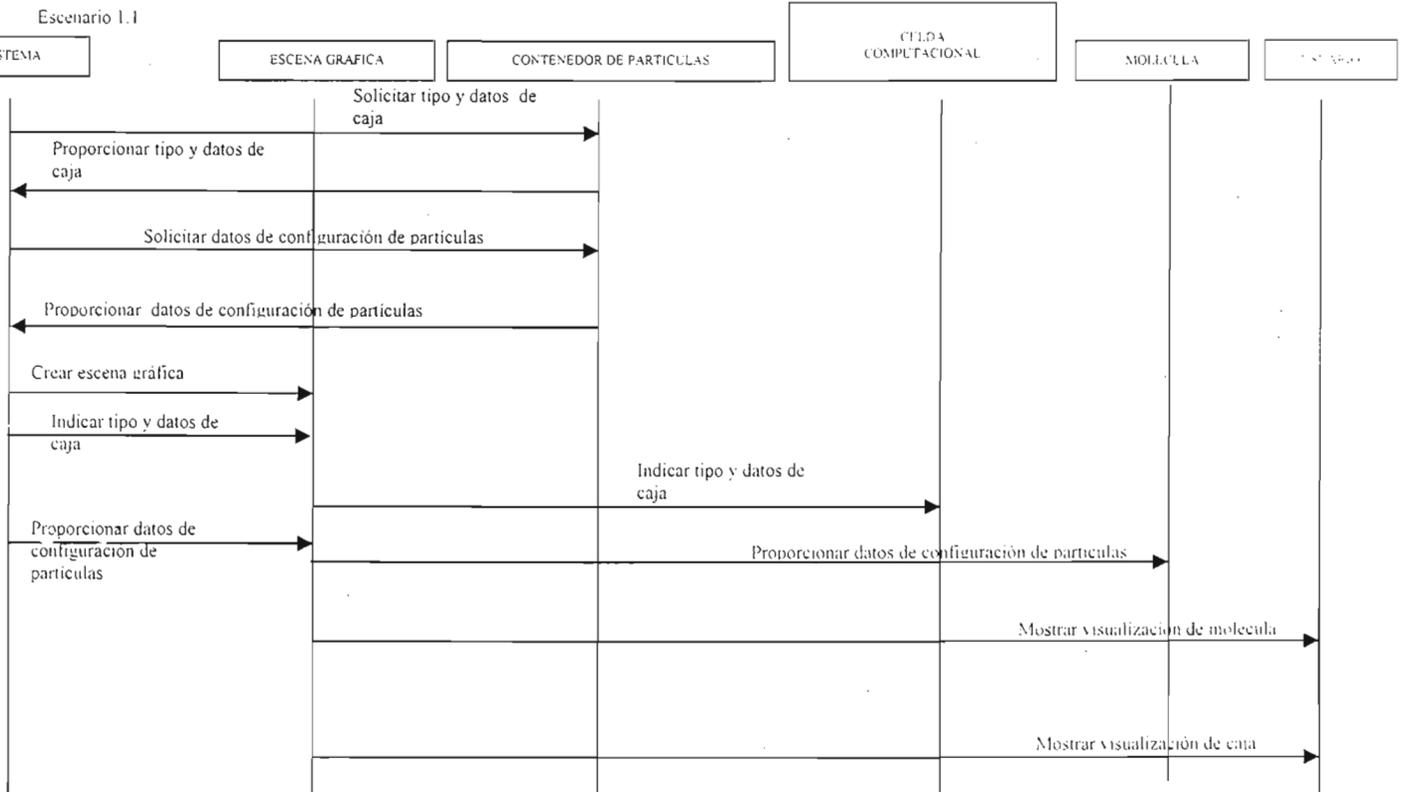


Escenario 4.2

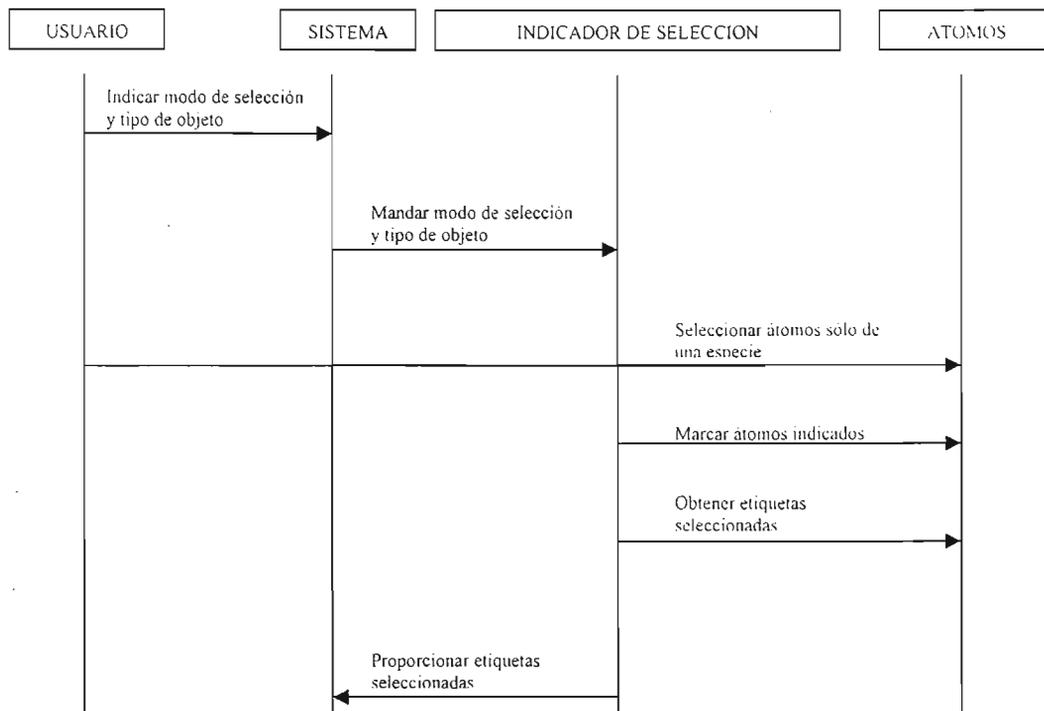




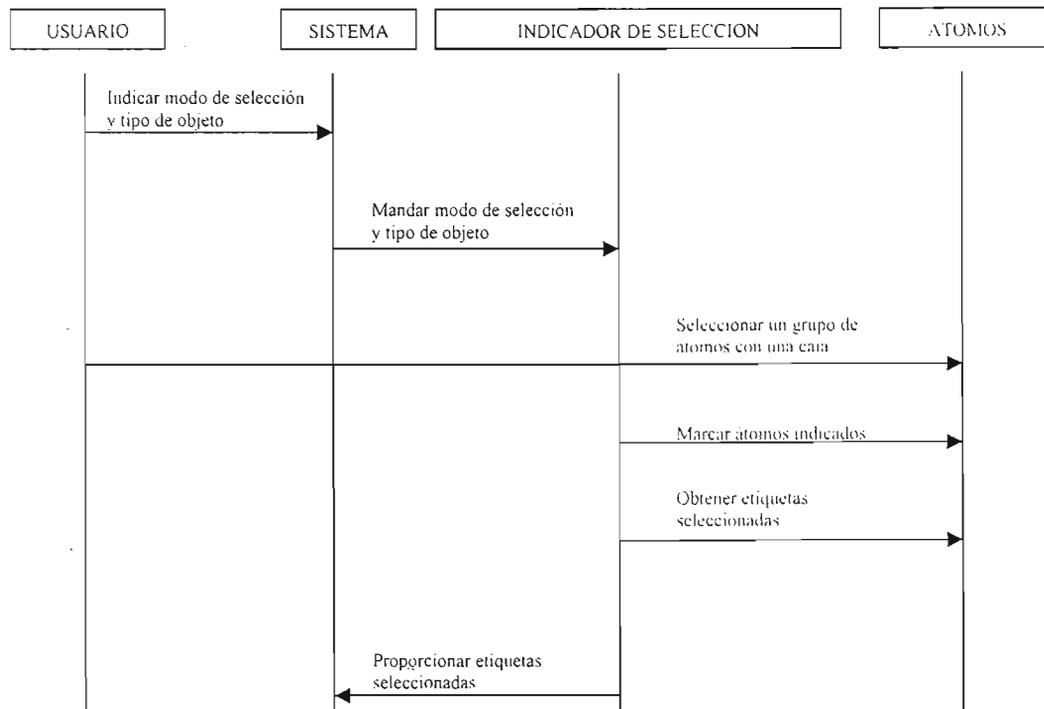
9.1.2.3.2. Visualizar y editar molécula



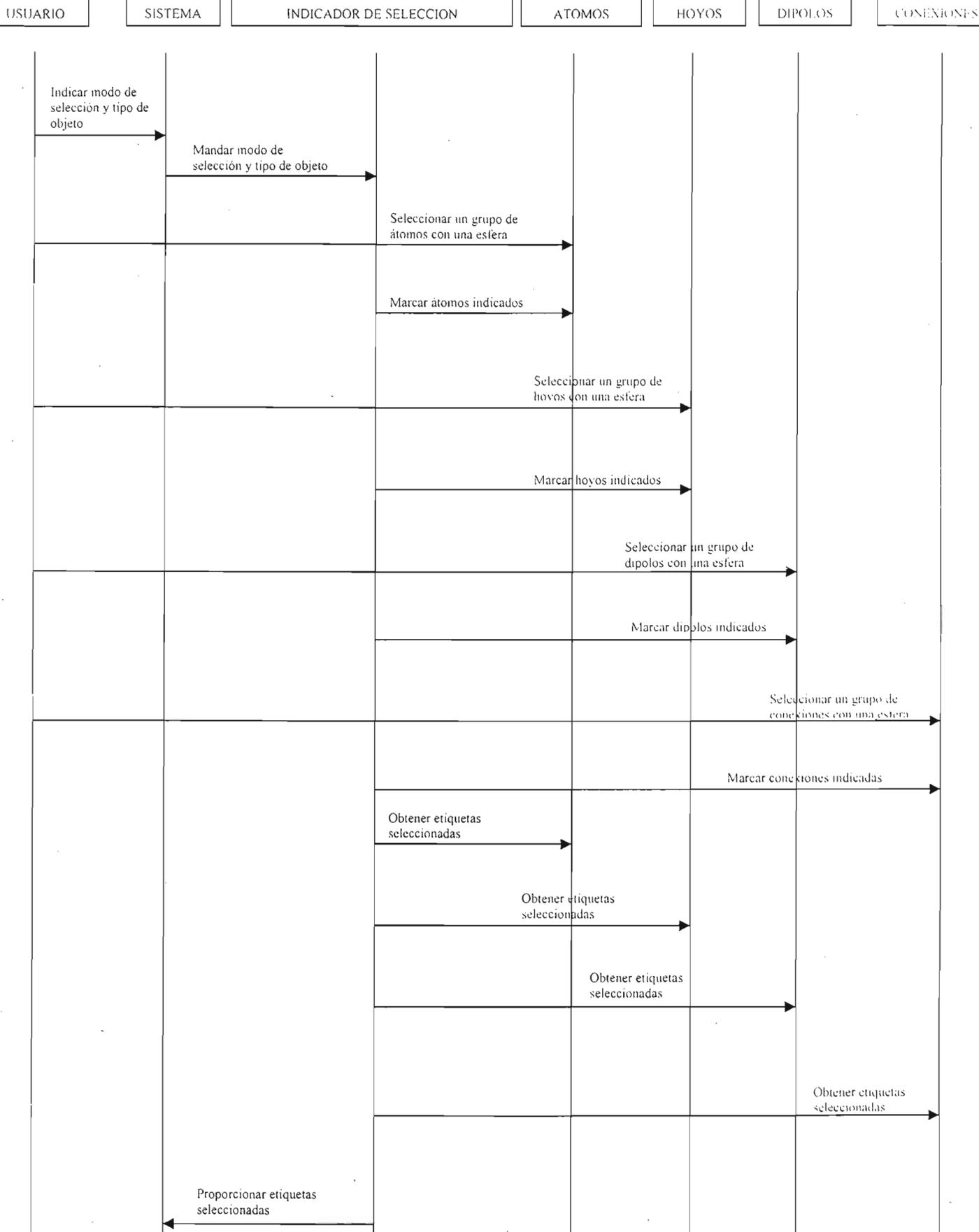
Escenario 2.1



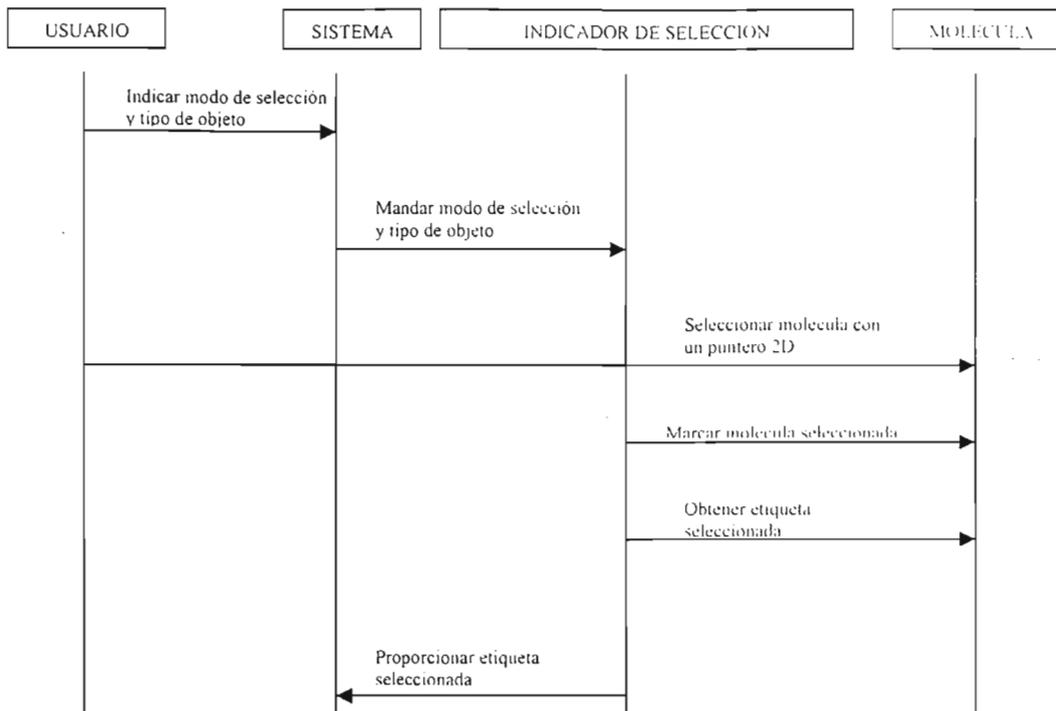
Escenario 2.2



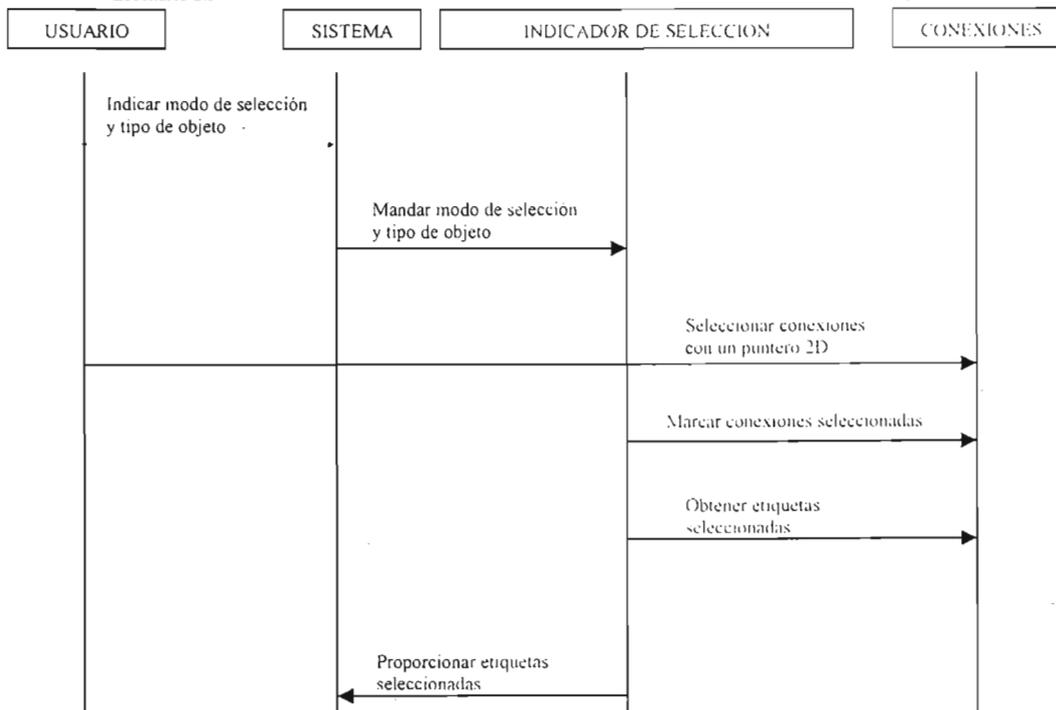
Escenario 2.3



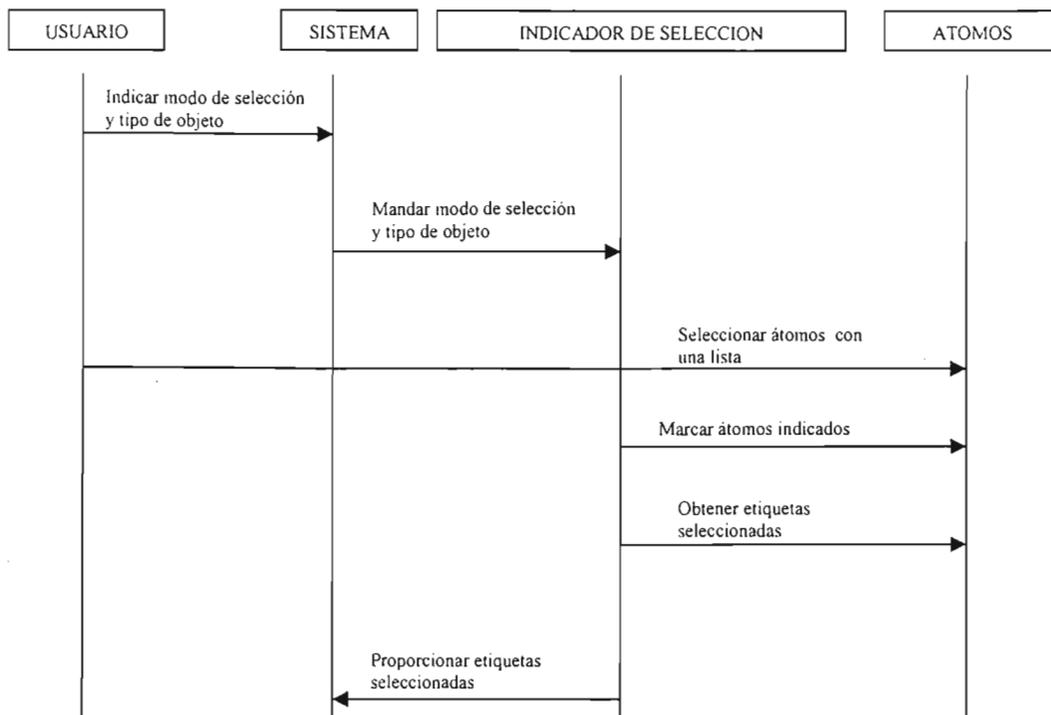
Escenario 2.4



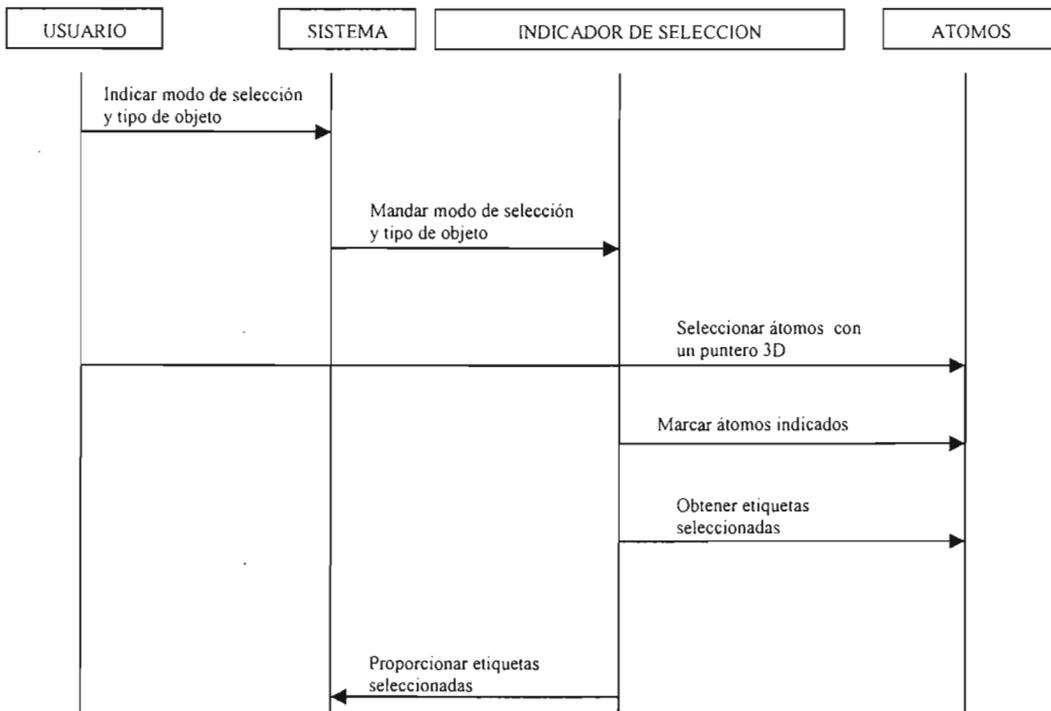
Escenario 2.5



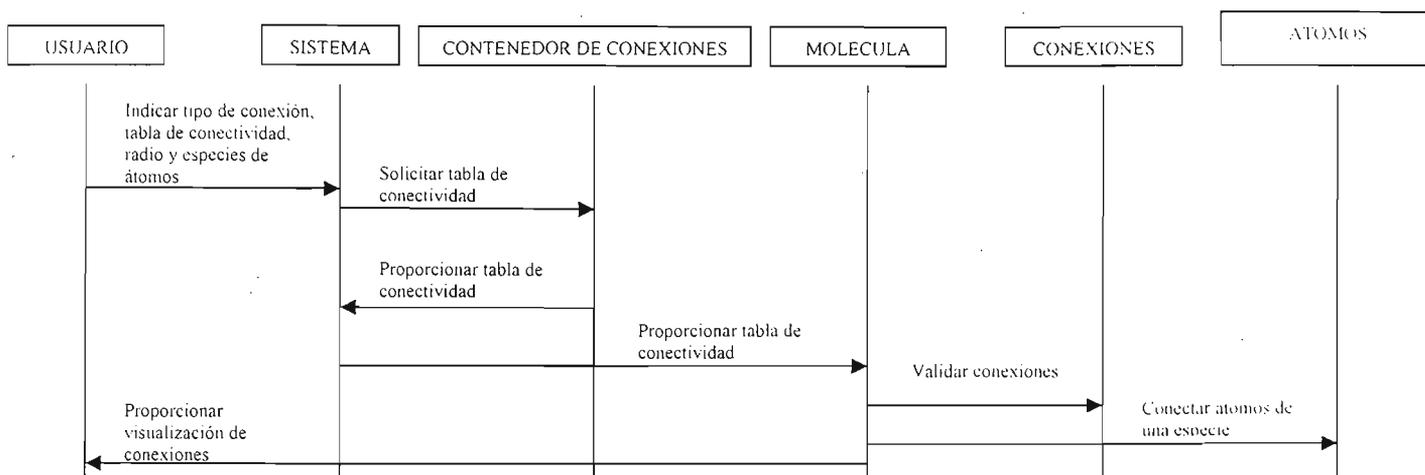
Escenario 2.6



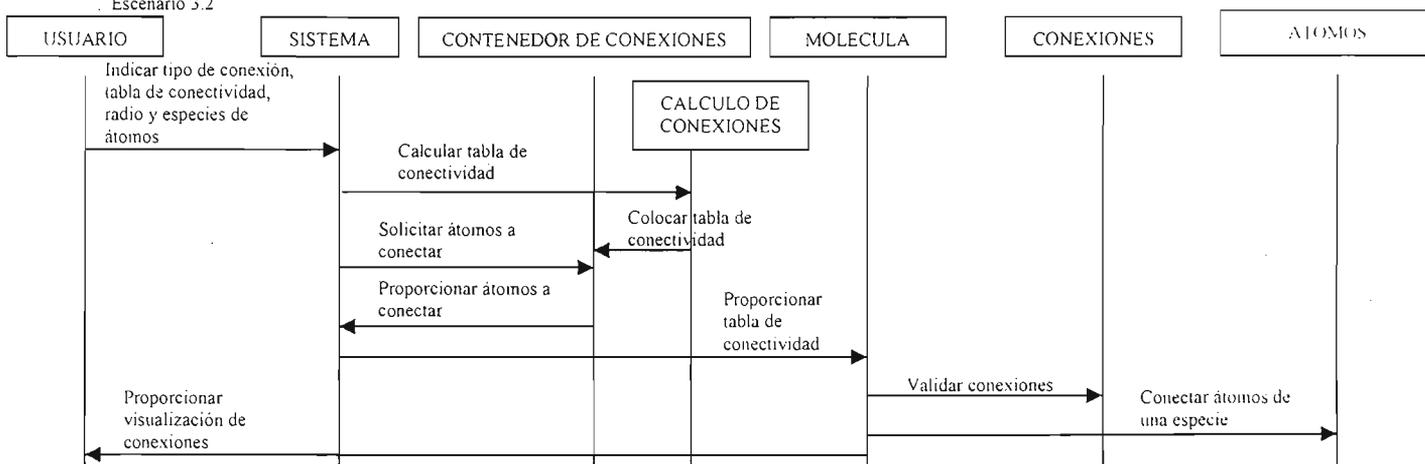
Escenario 2.7



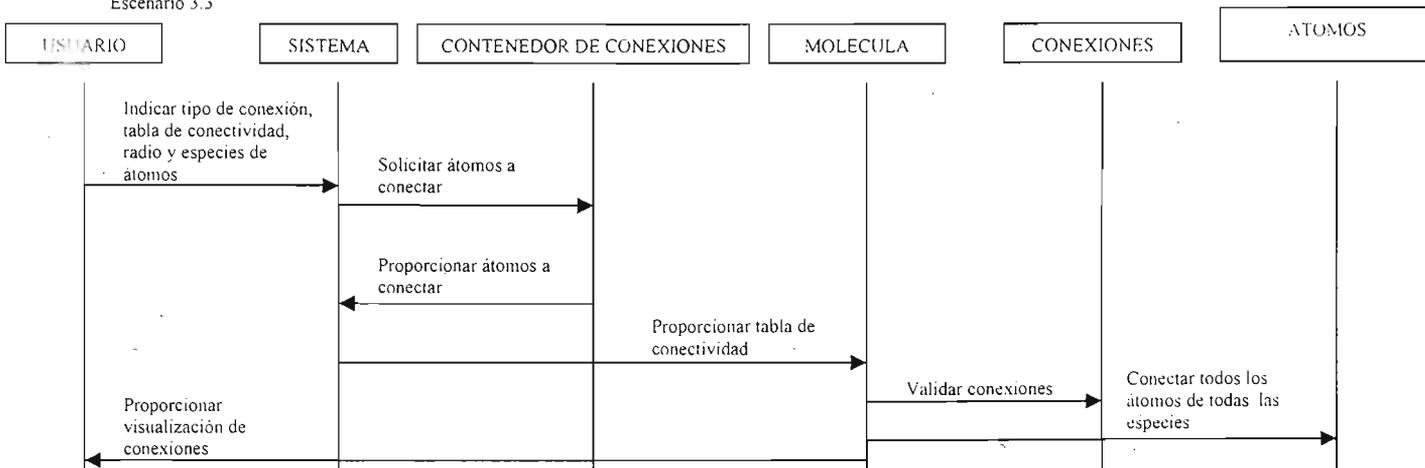
Escenario 3.1



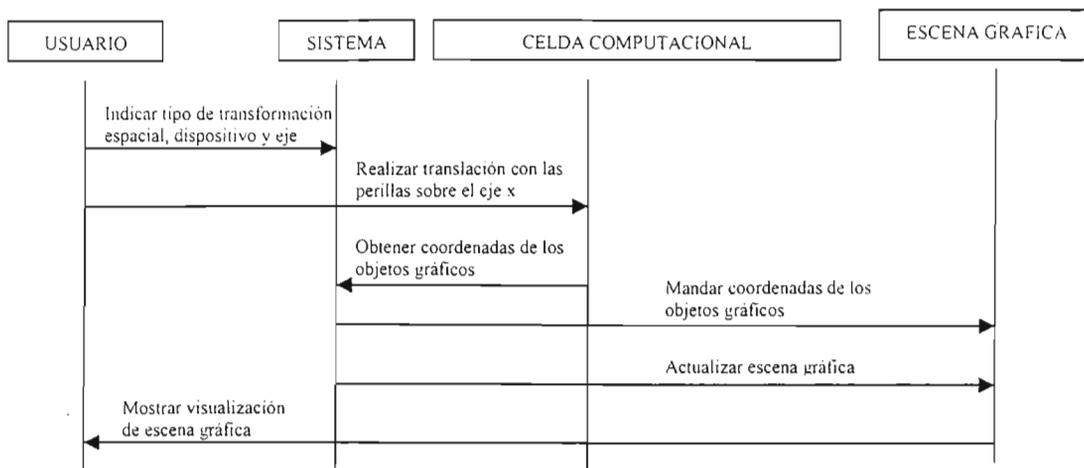
Escenario 3.2



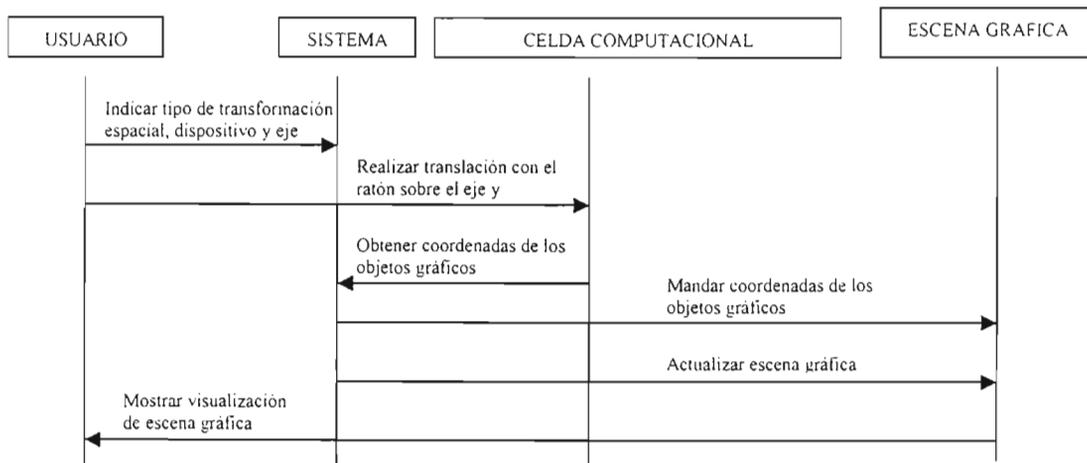
Escenario 3.3



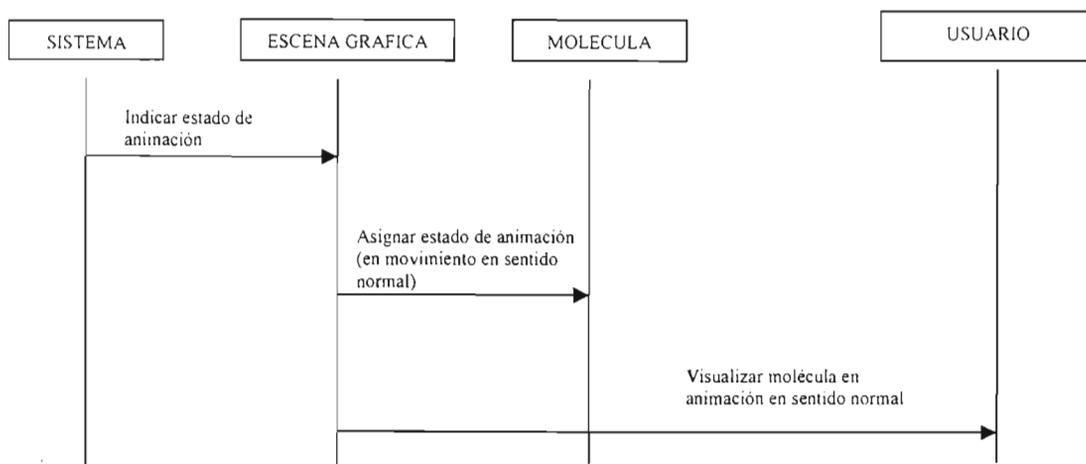
Escenario 4.1



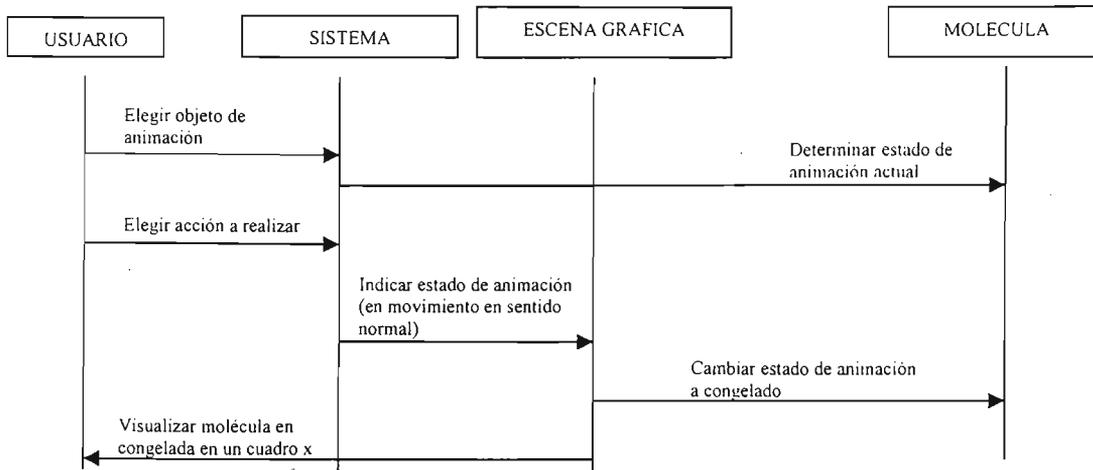
Escenario 4.2



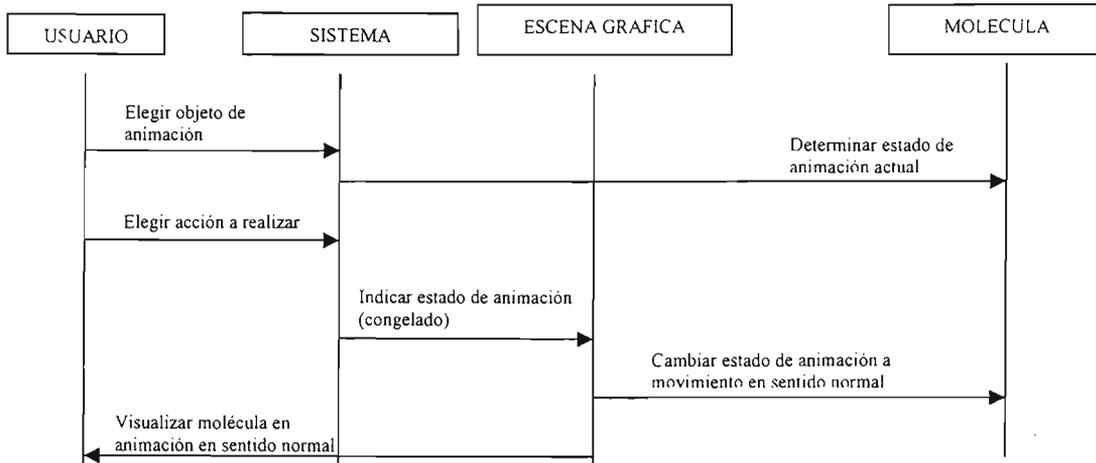
Escenario 5.1



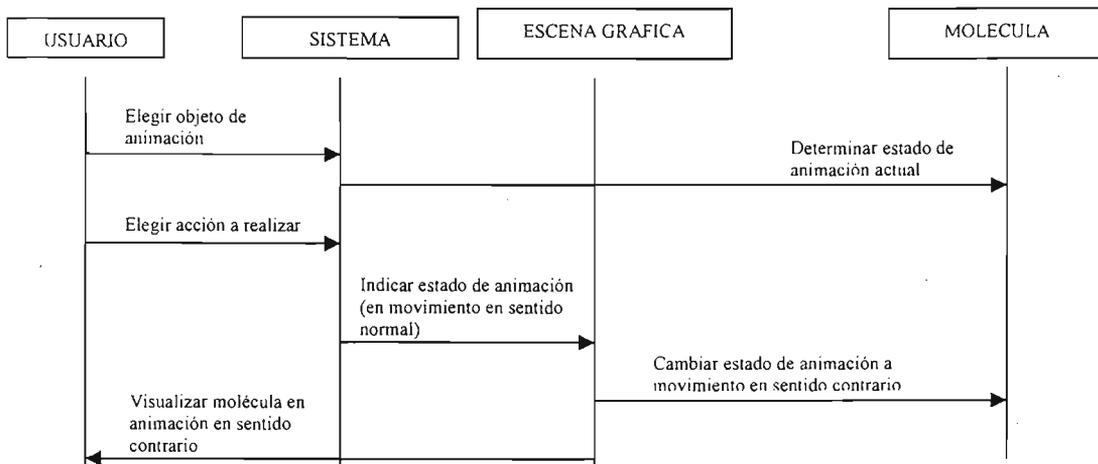
Escenario 5.2



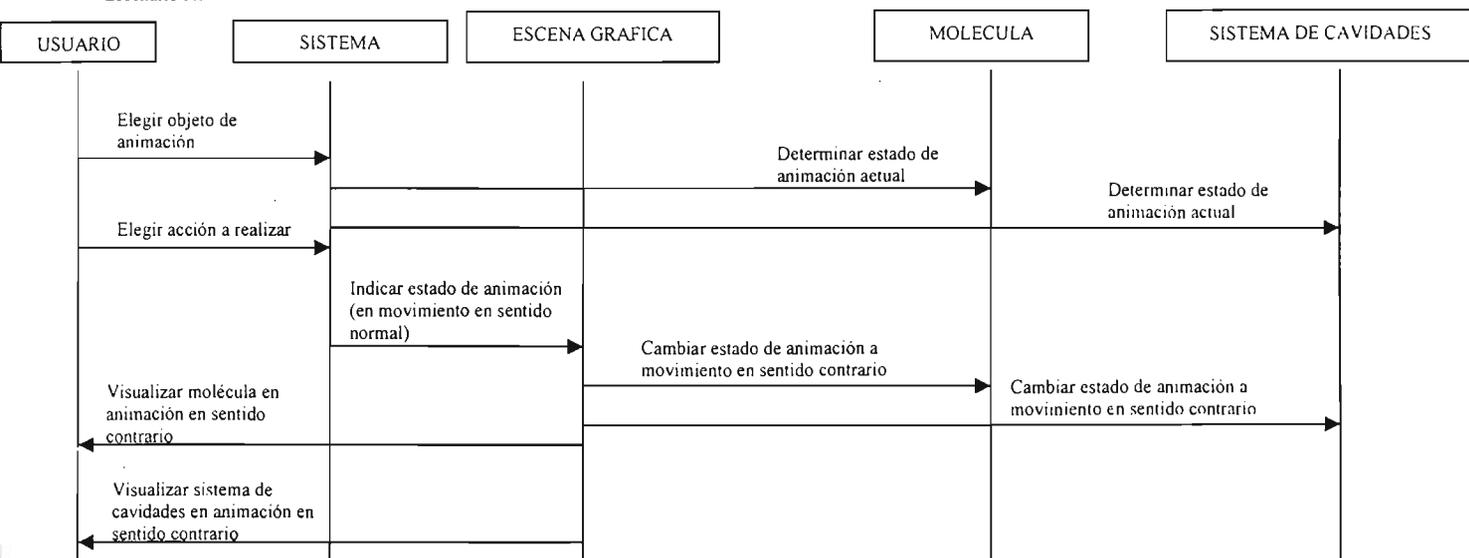
Escenario 5.3



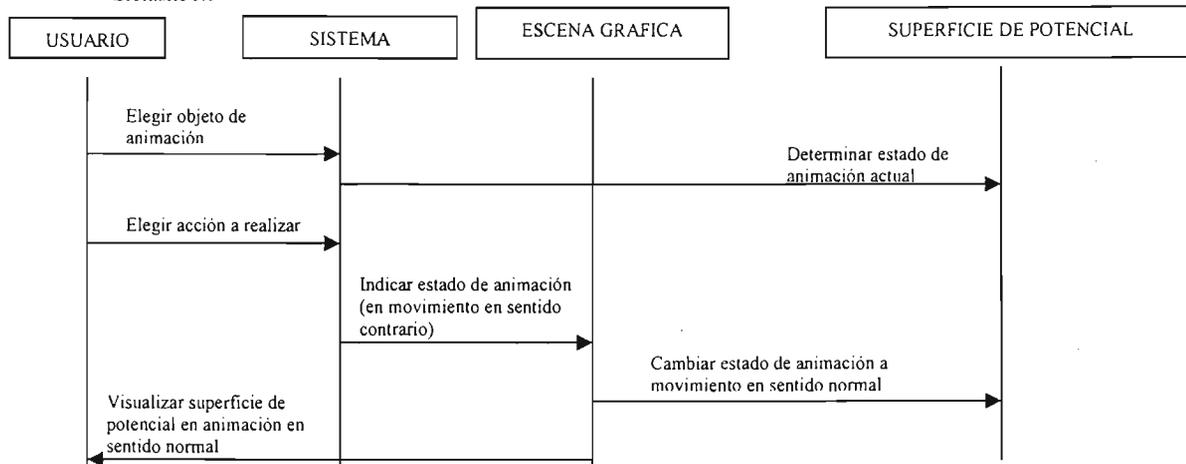
Escenario 5.4



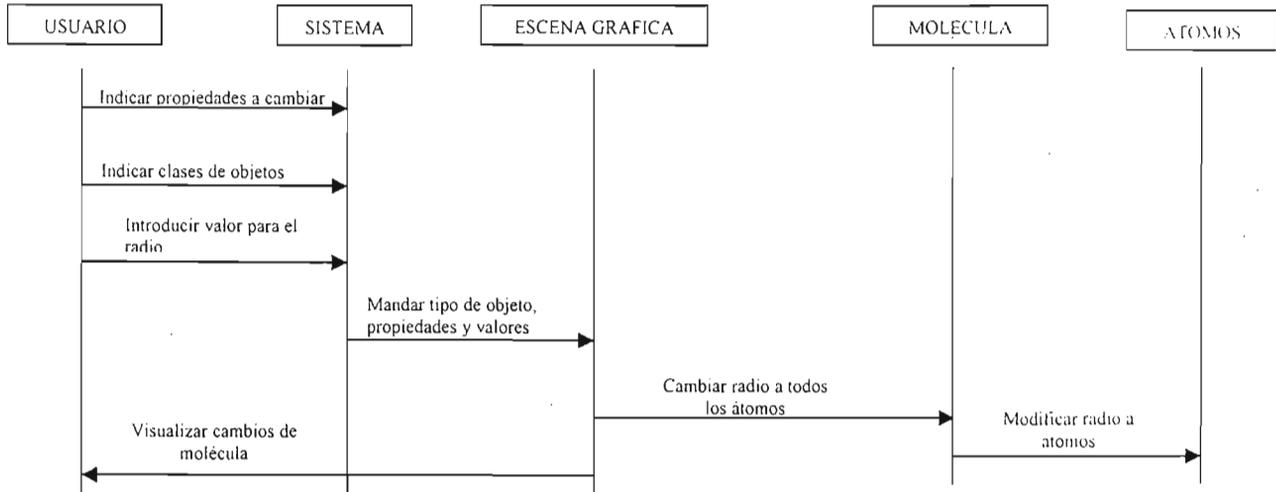
Escenario 5.5



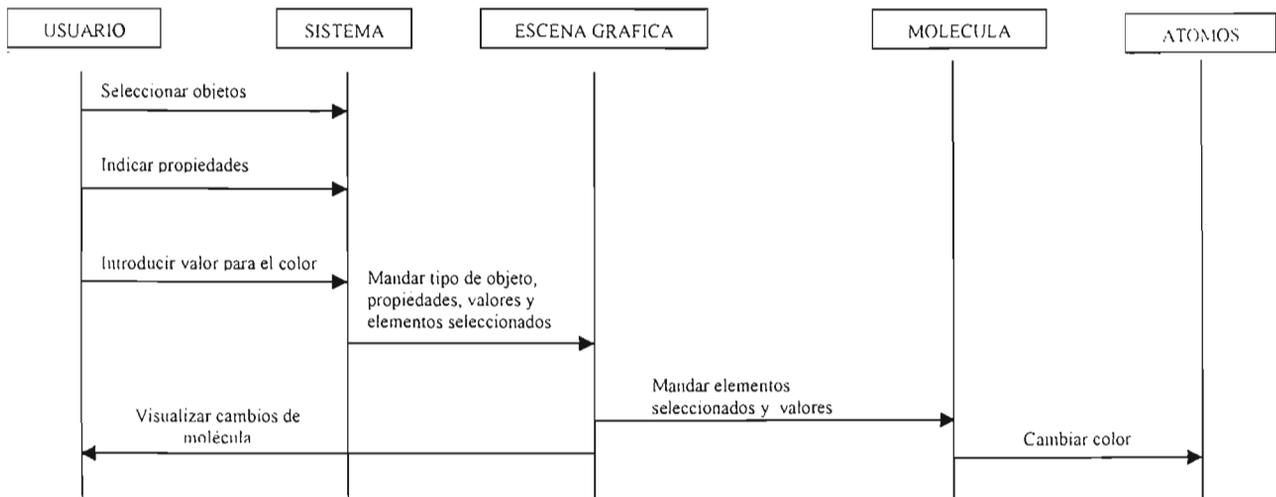
Escenario 5.6



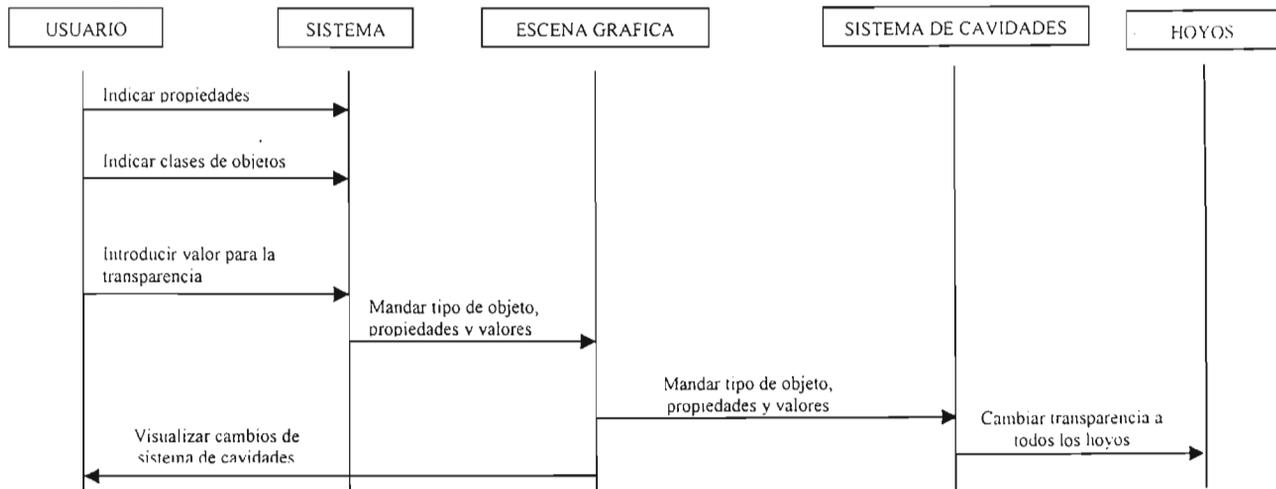
Escenario 6.1



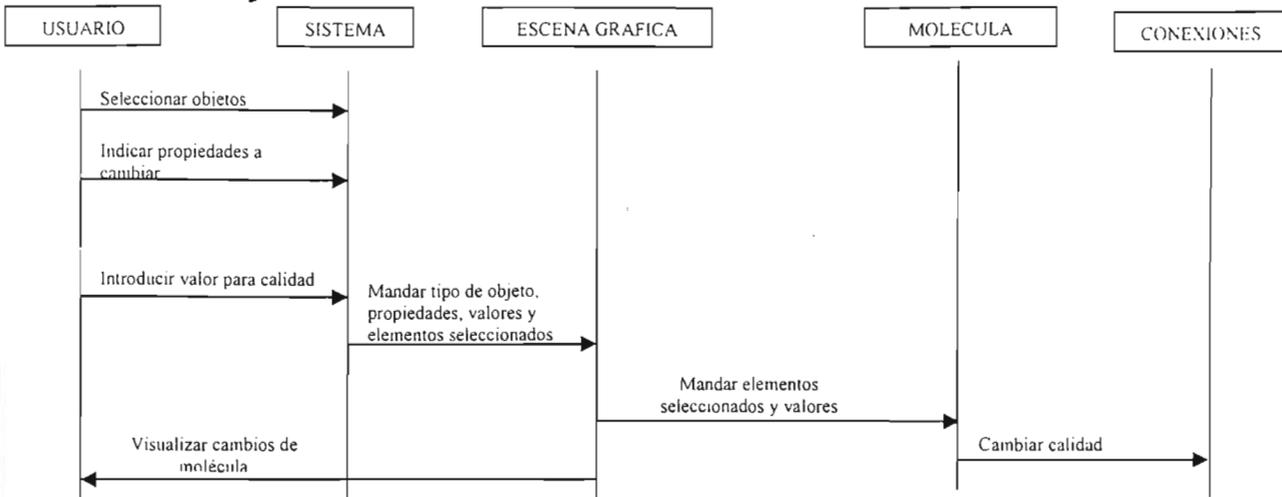
Escenario 6.2



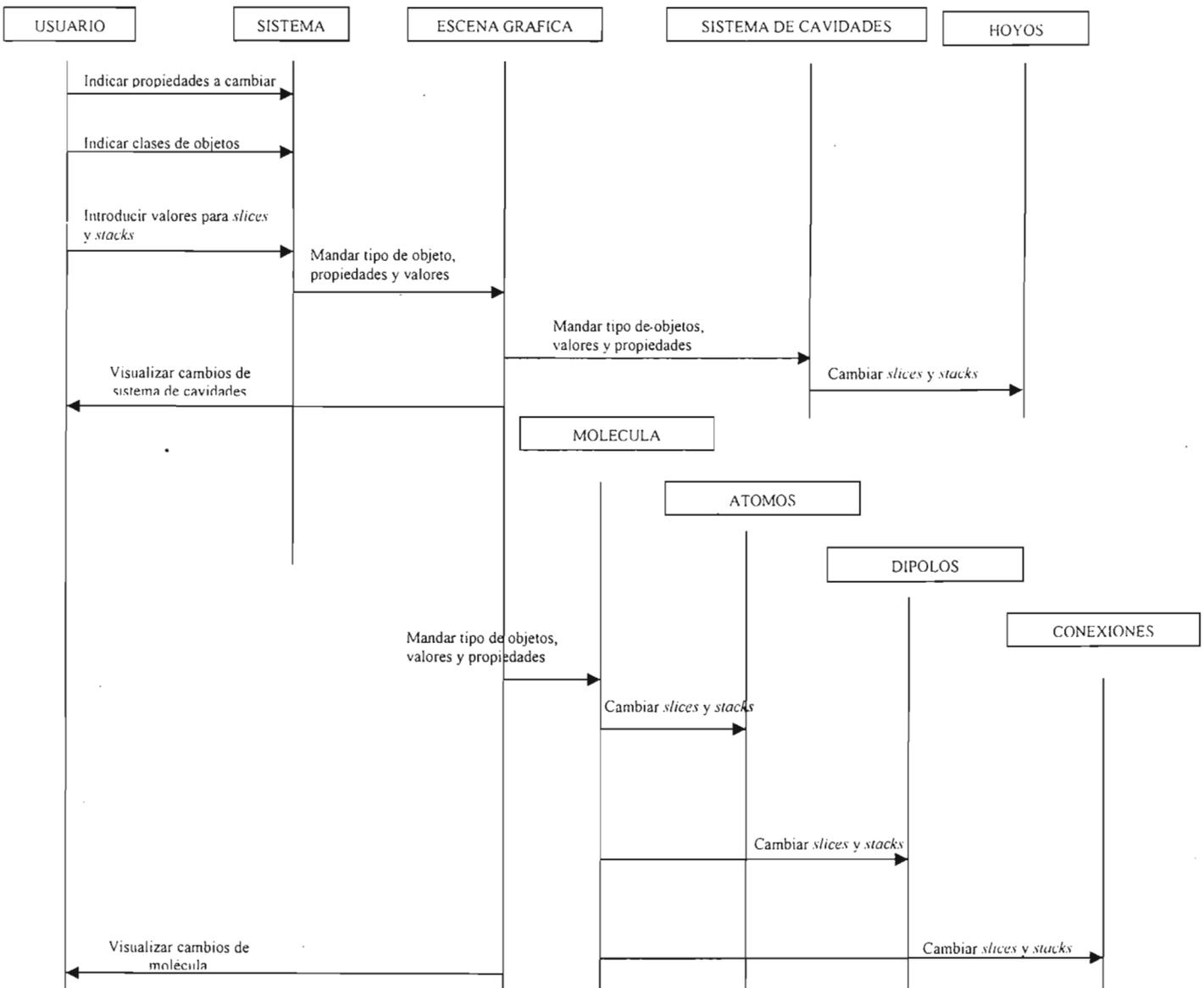
Escenario 6.3



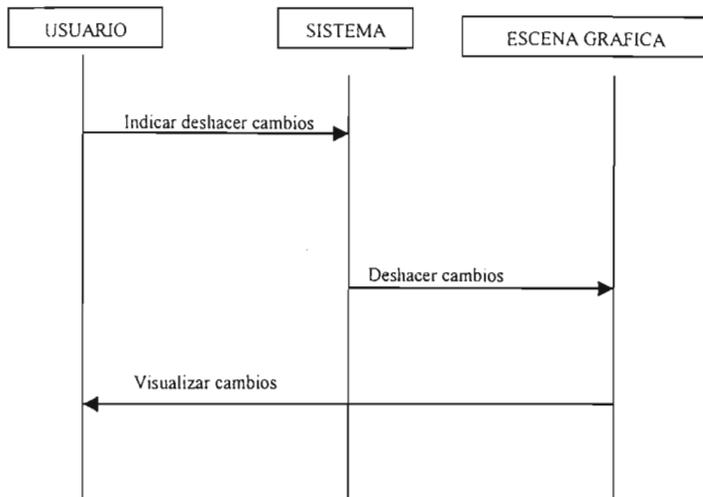
Escenario 6.4



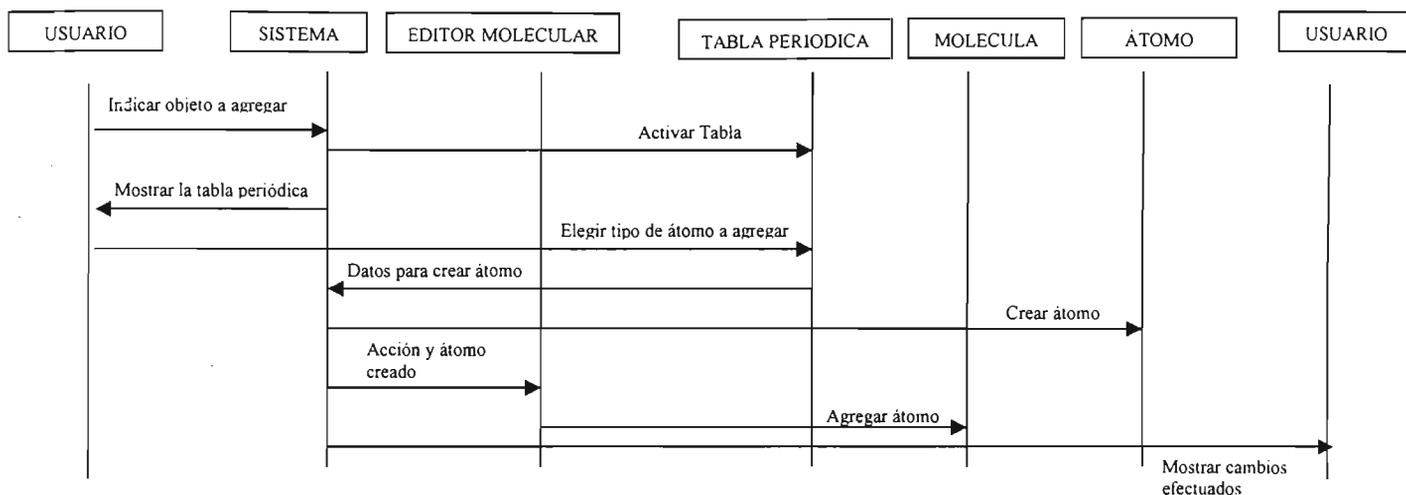
Escenario 6.5



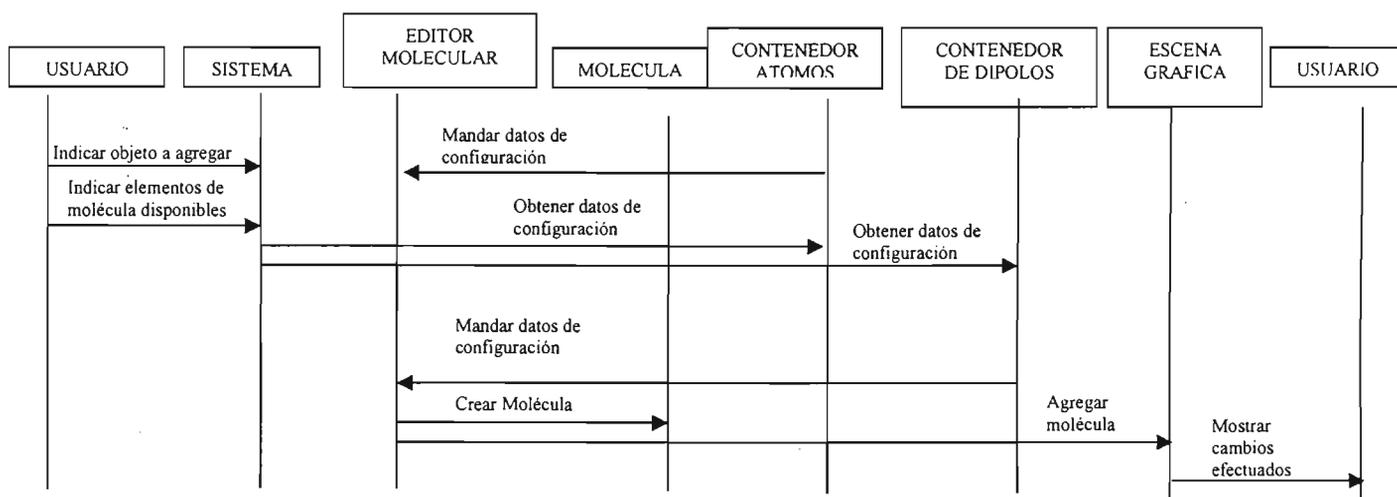
Escenario 6.6



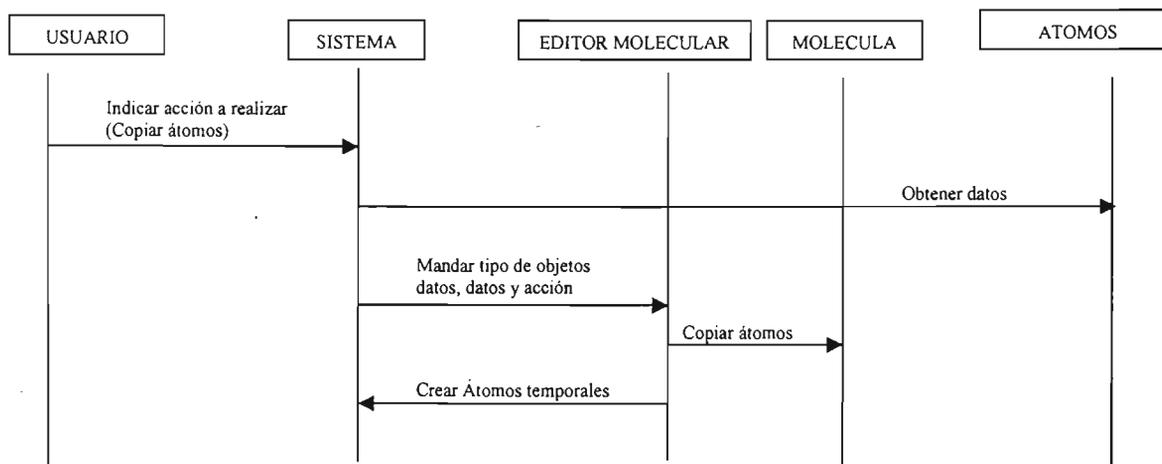
Escenario 7.1



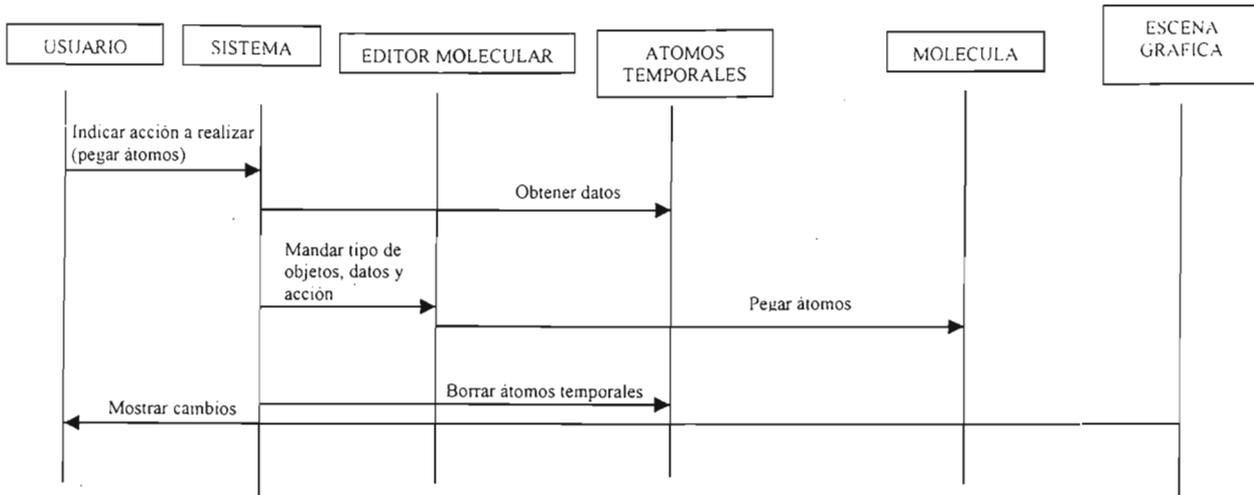
Escenario 7.2



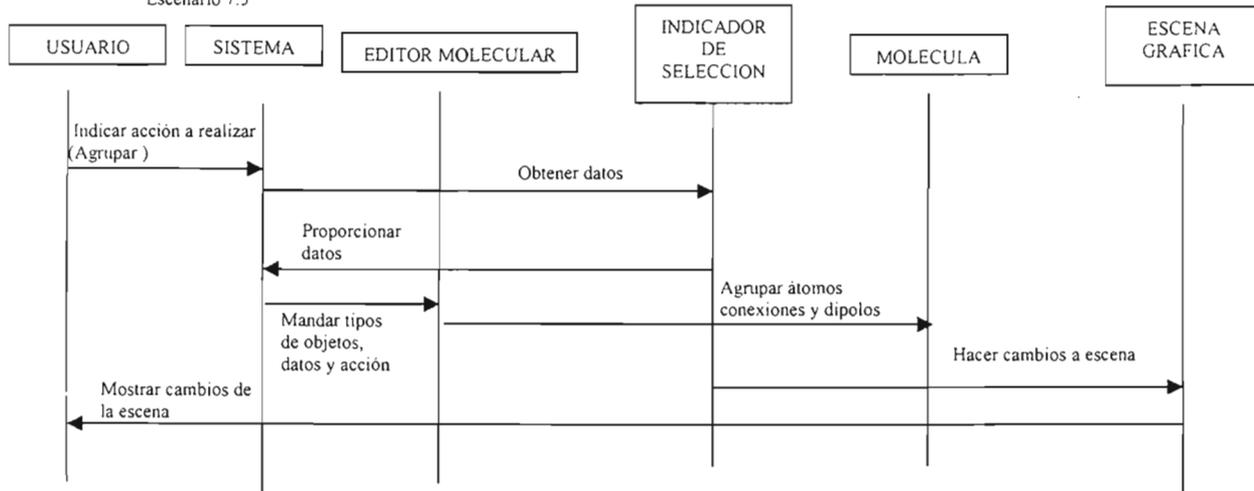
Escenario 7.3



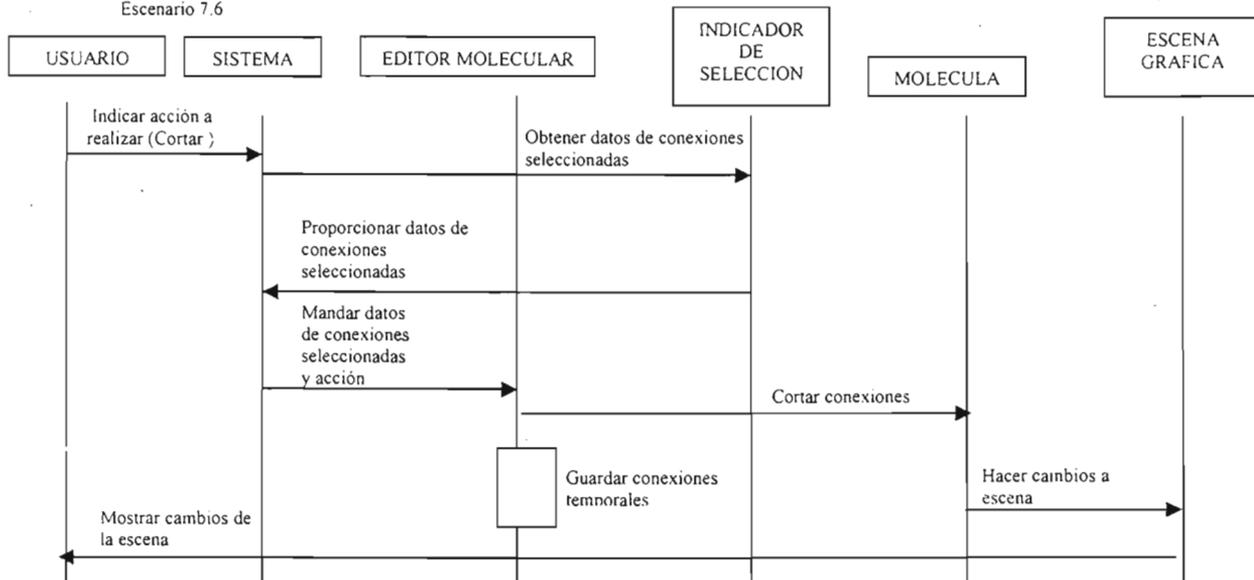
Escenario 7.4



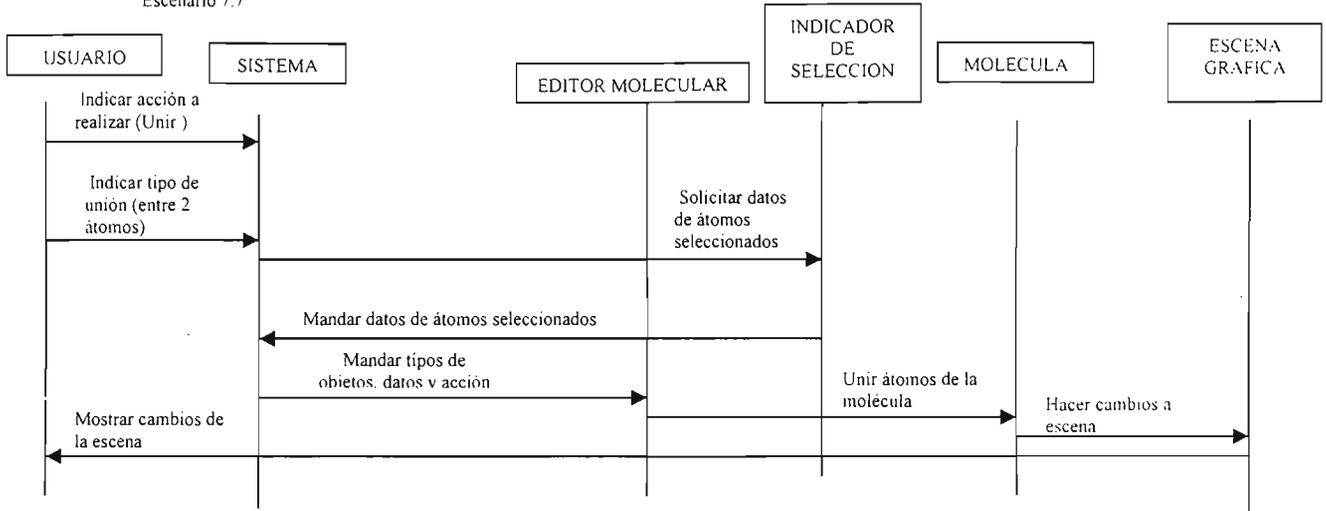
Escenario 7.5



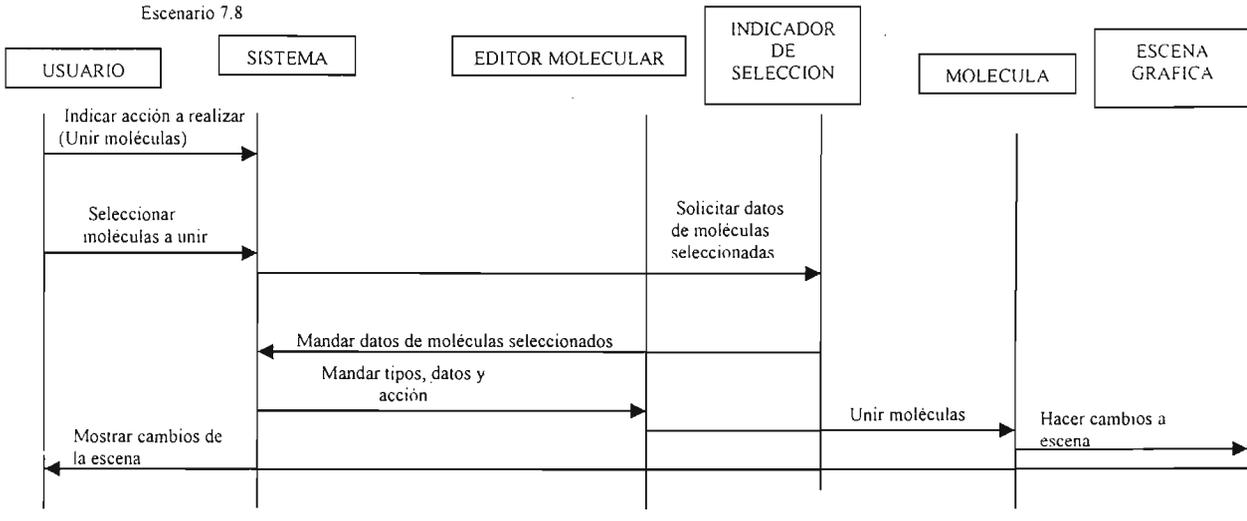
Escenario 7.6



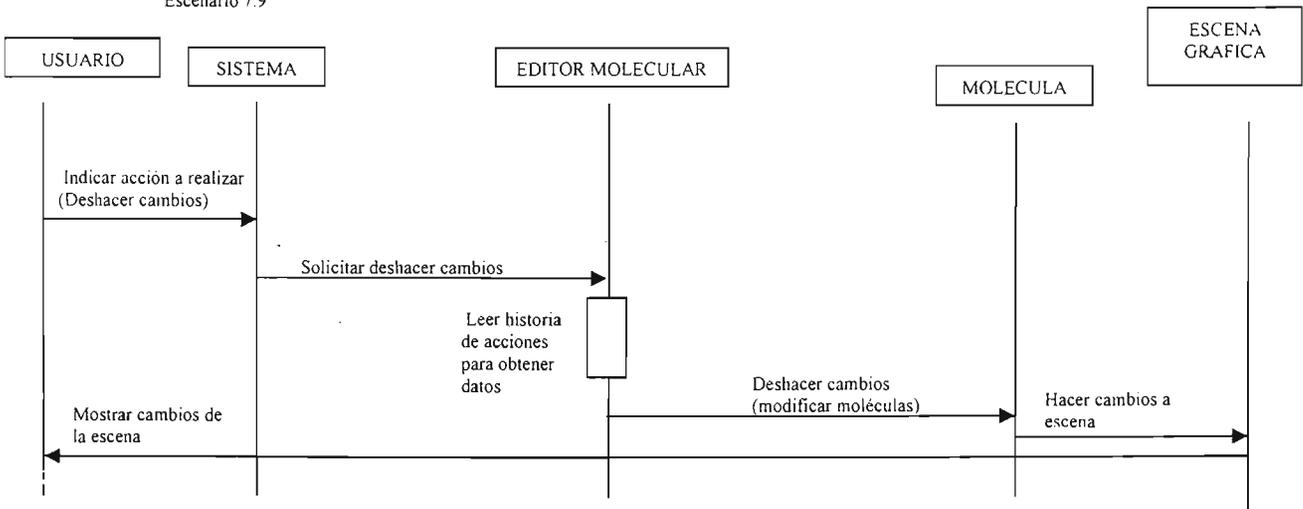
Escenario 7.7



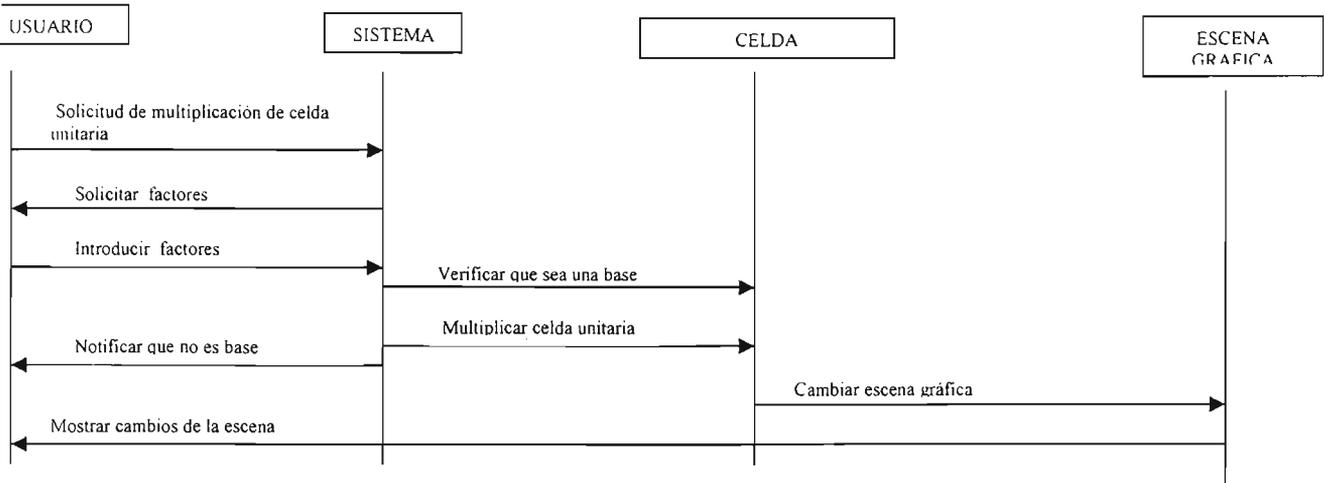
Escenario 7.8



Escenario 7.9

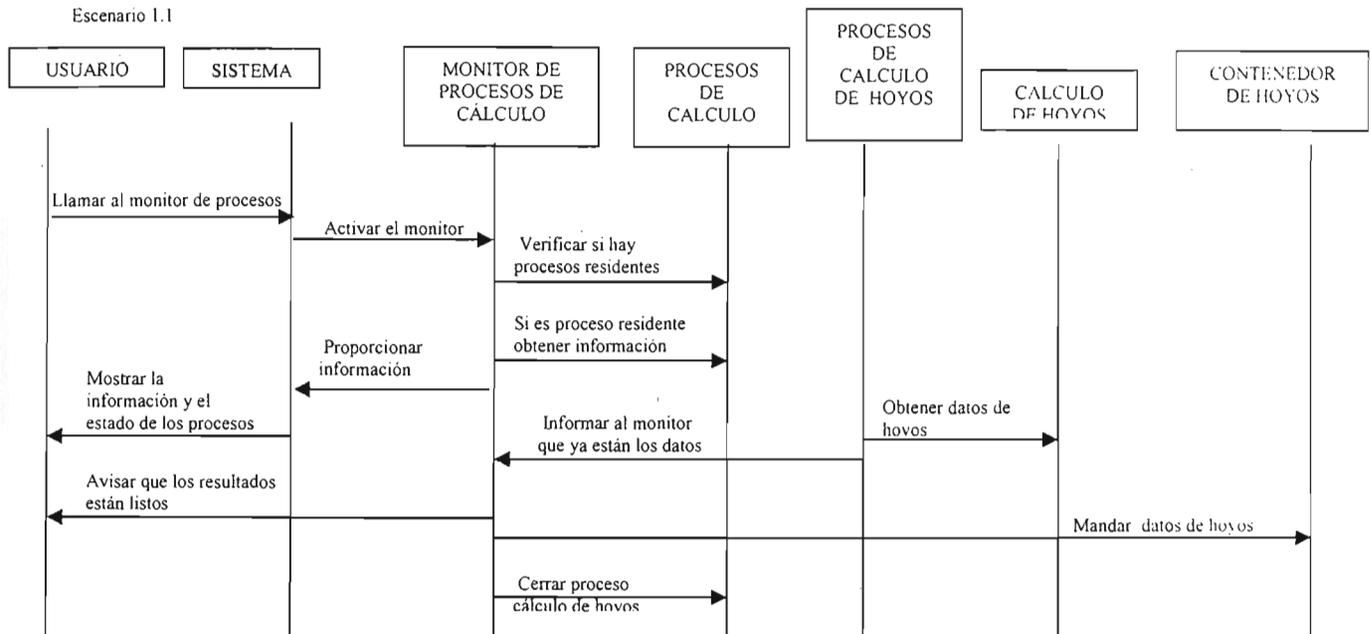


Escenario 8.1

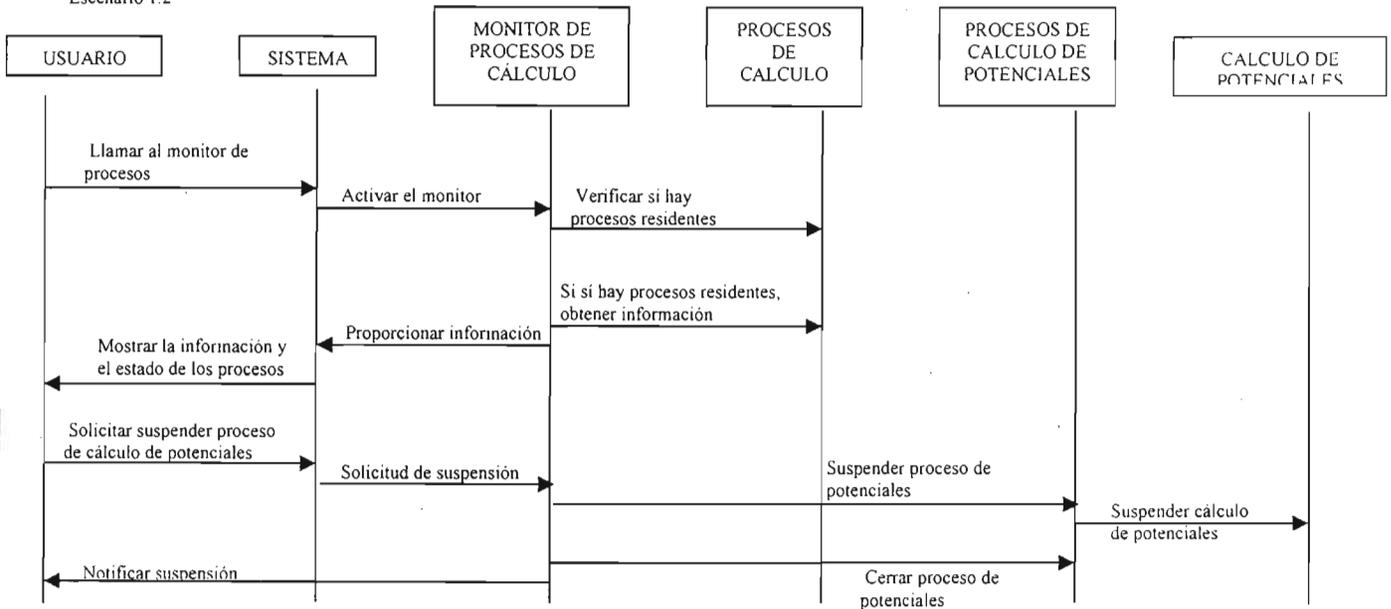


9.1.2.3.3. Monitor de procesos de cálculo

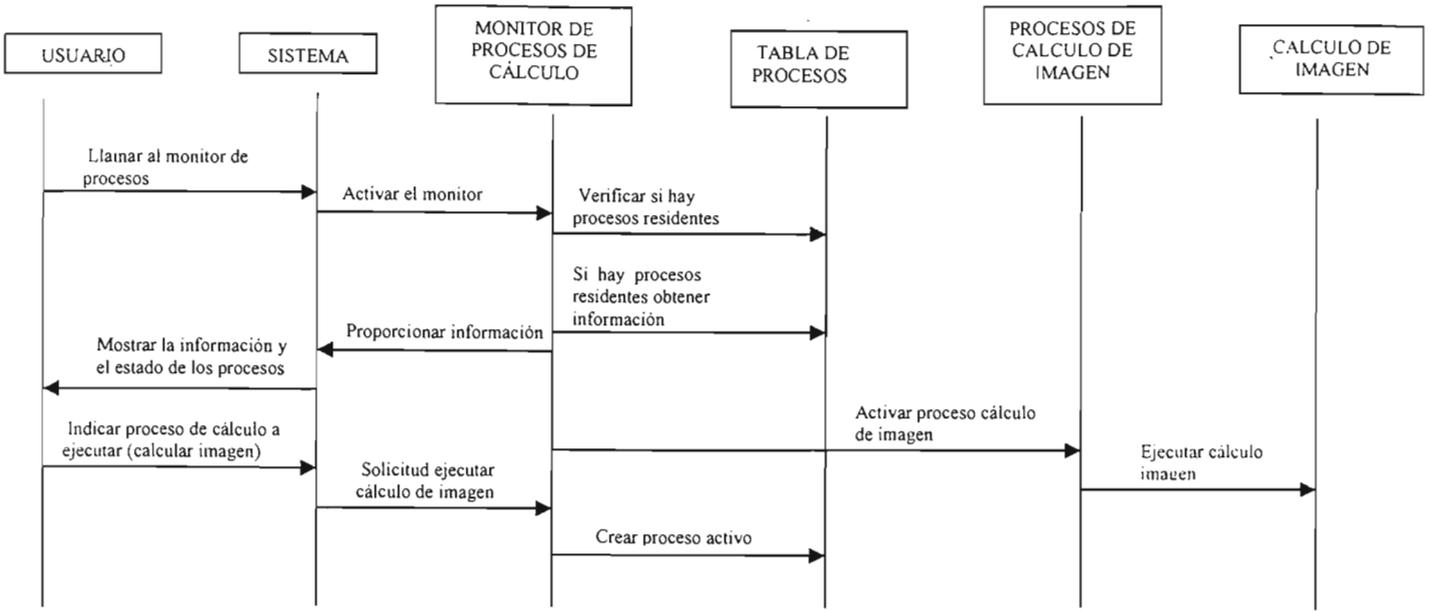
Escenario 1.1



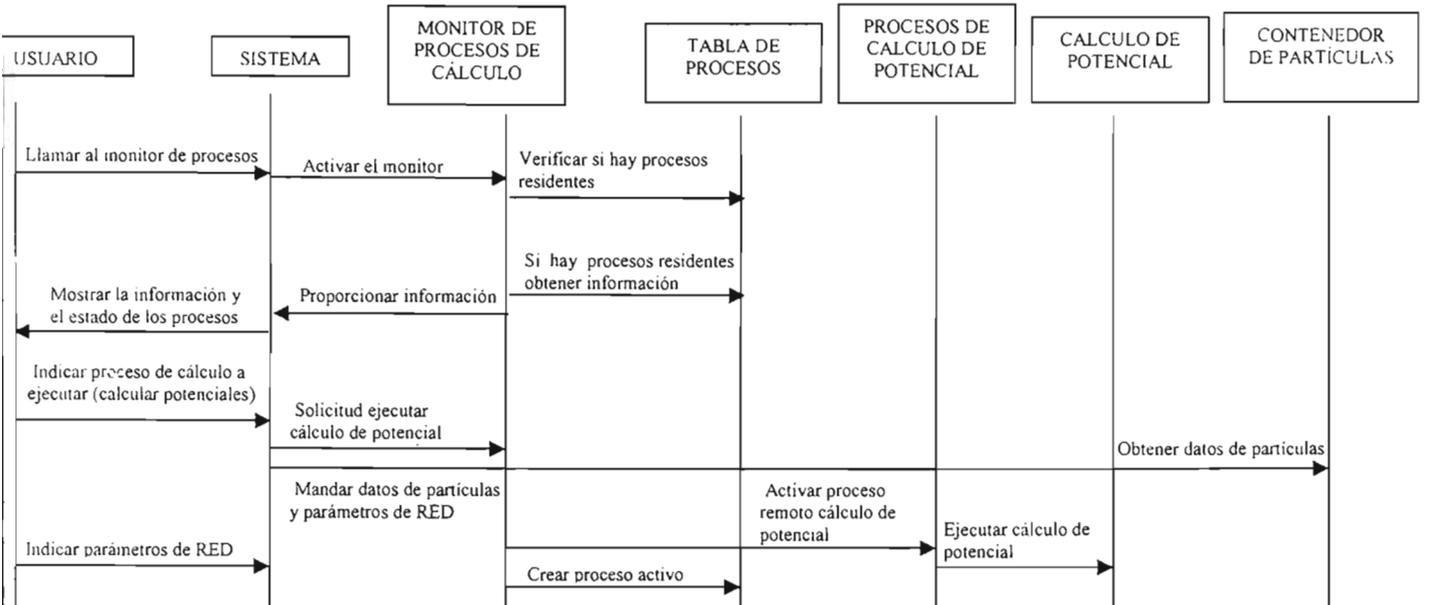
Escenario 1.2



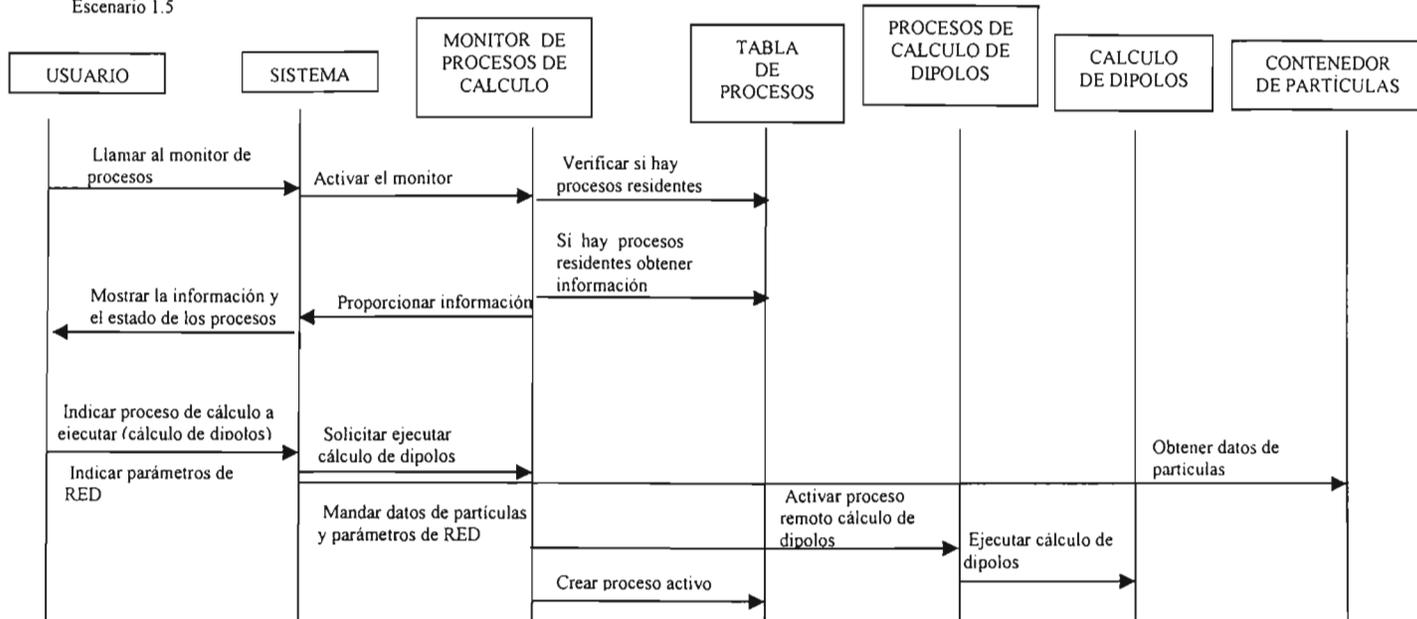
Escenario 1.3



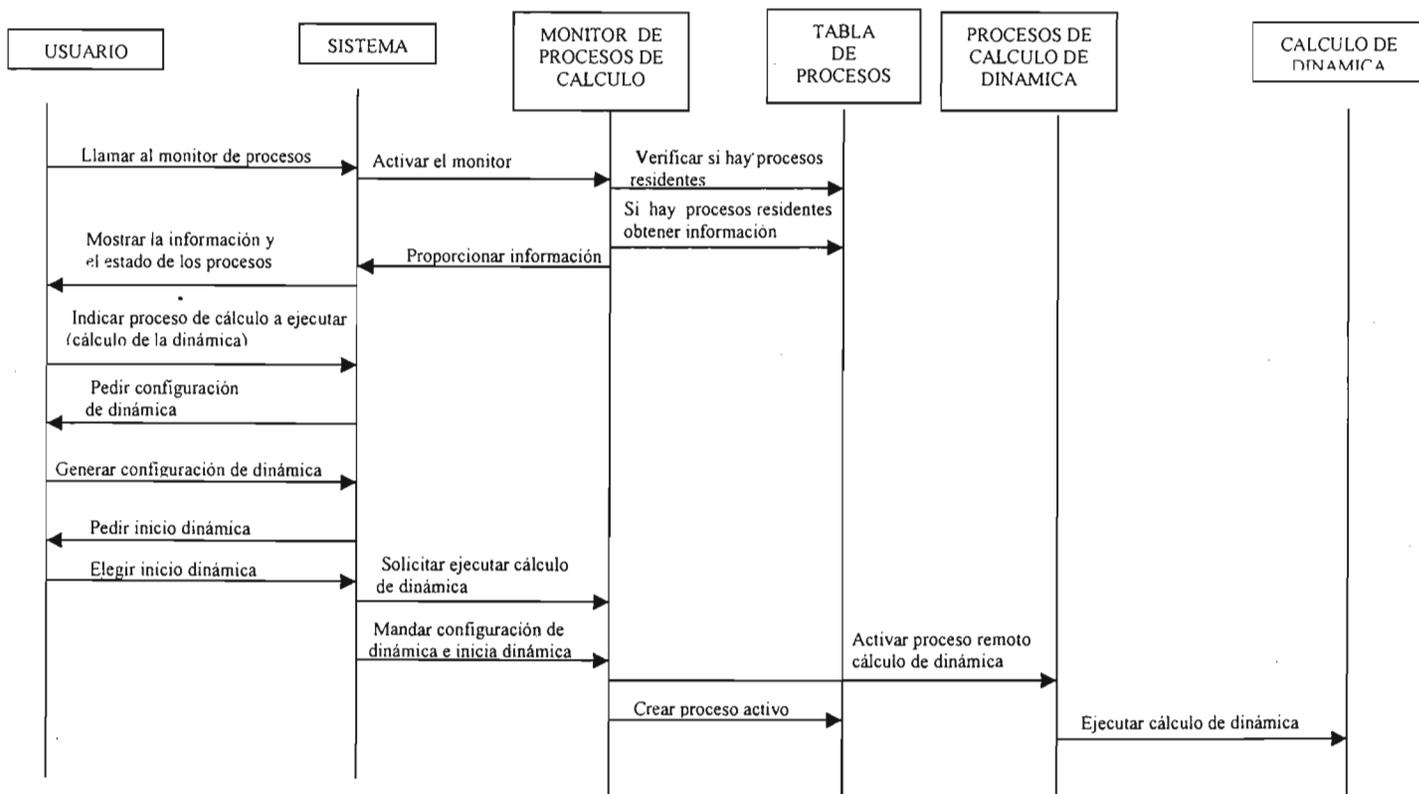
Escenario 1.4



Escenario 1.5

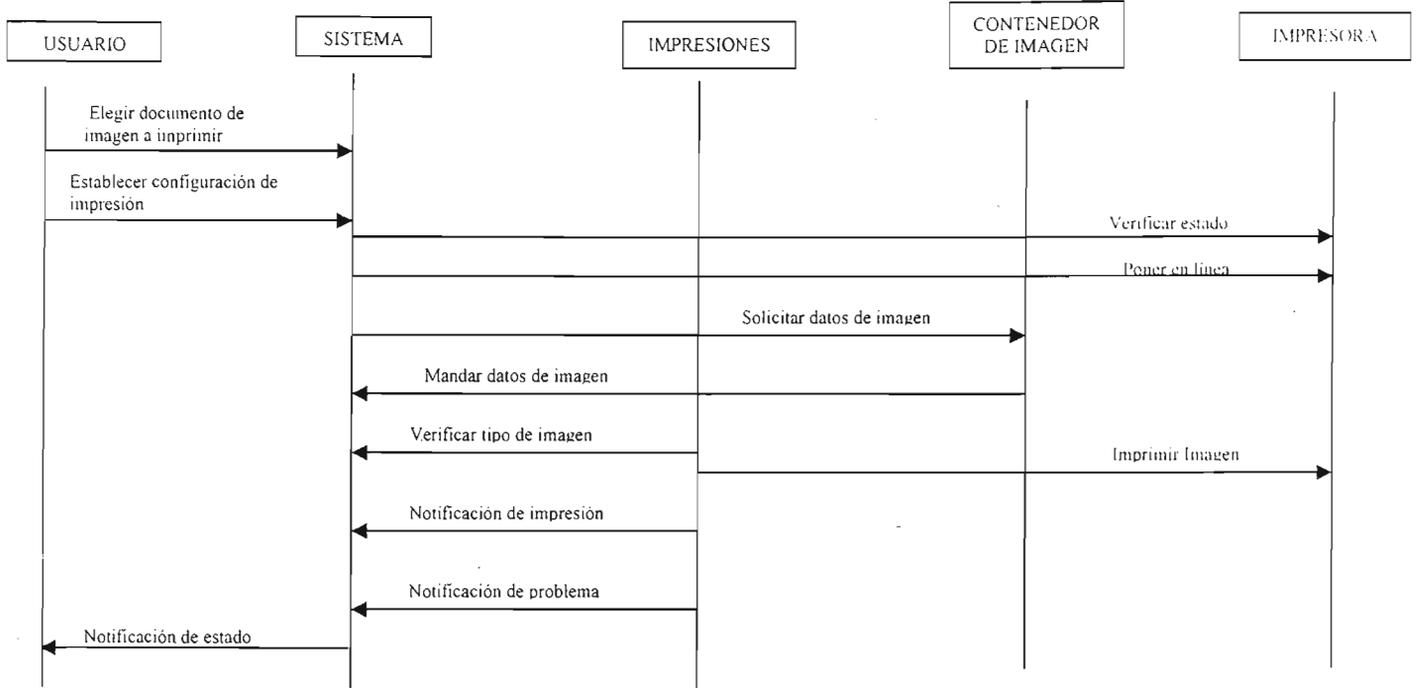


Escenario 1.6

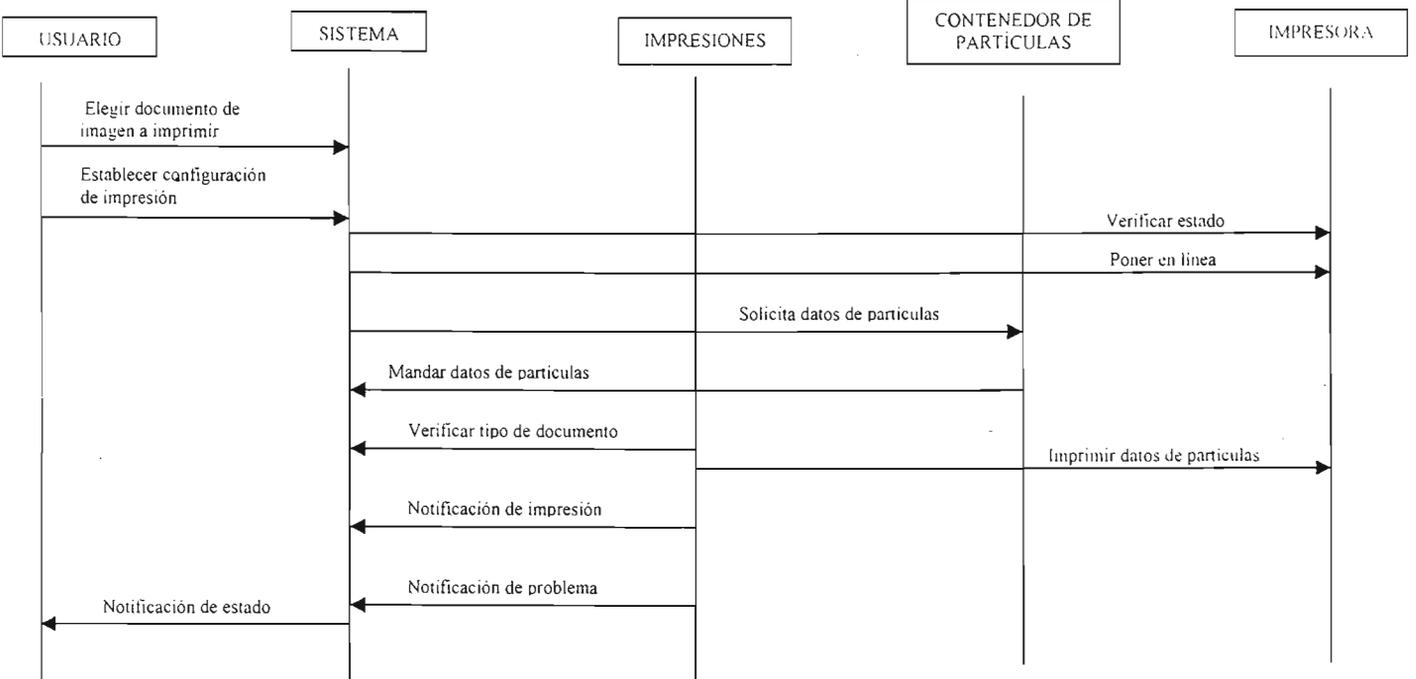


9.1.2.3.4. Impresiones

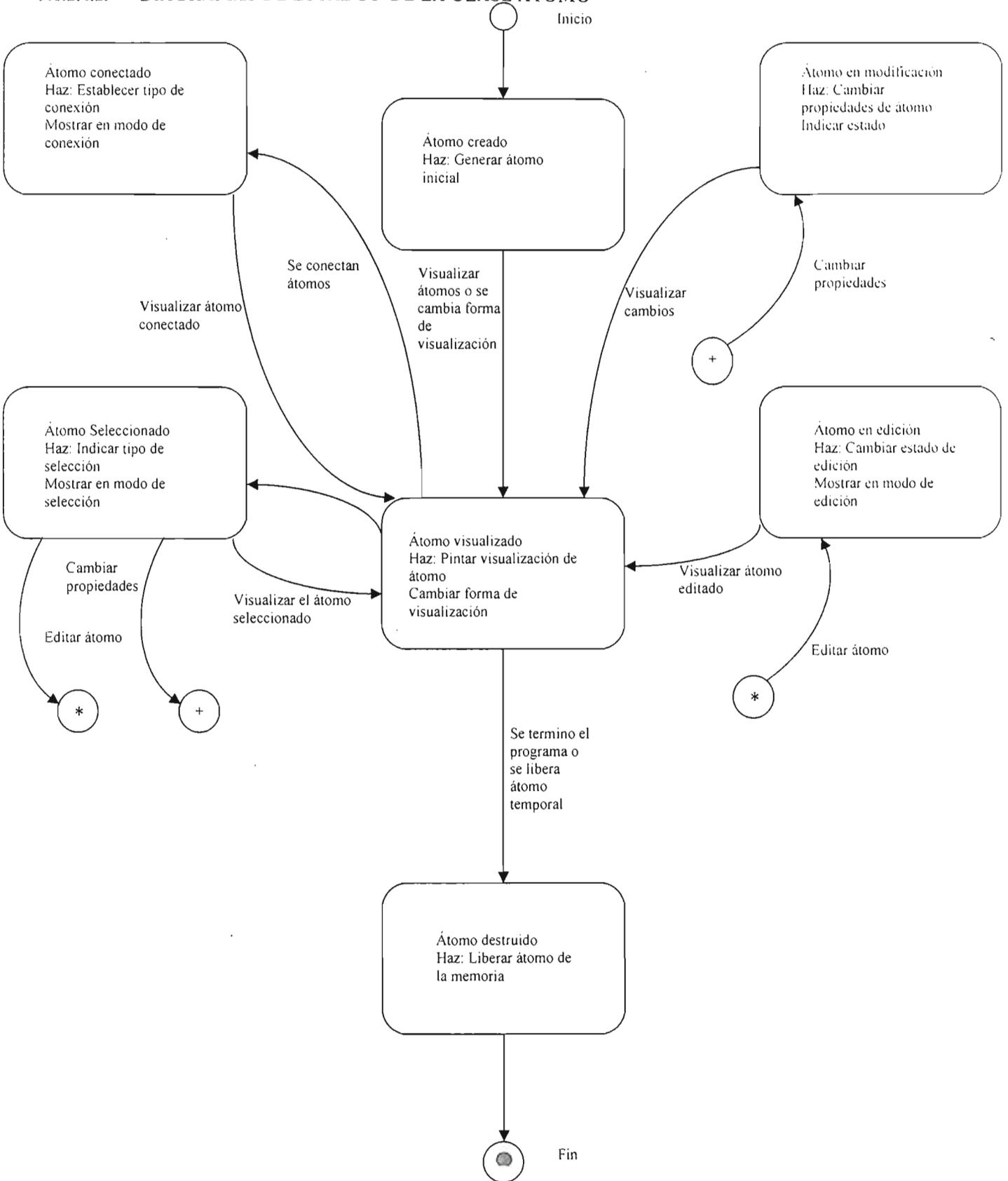
Escenario 1.1

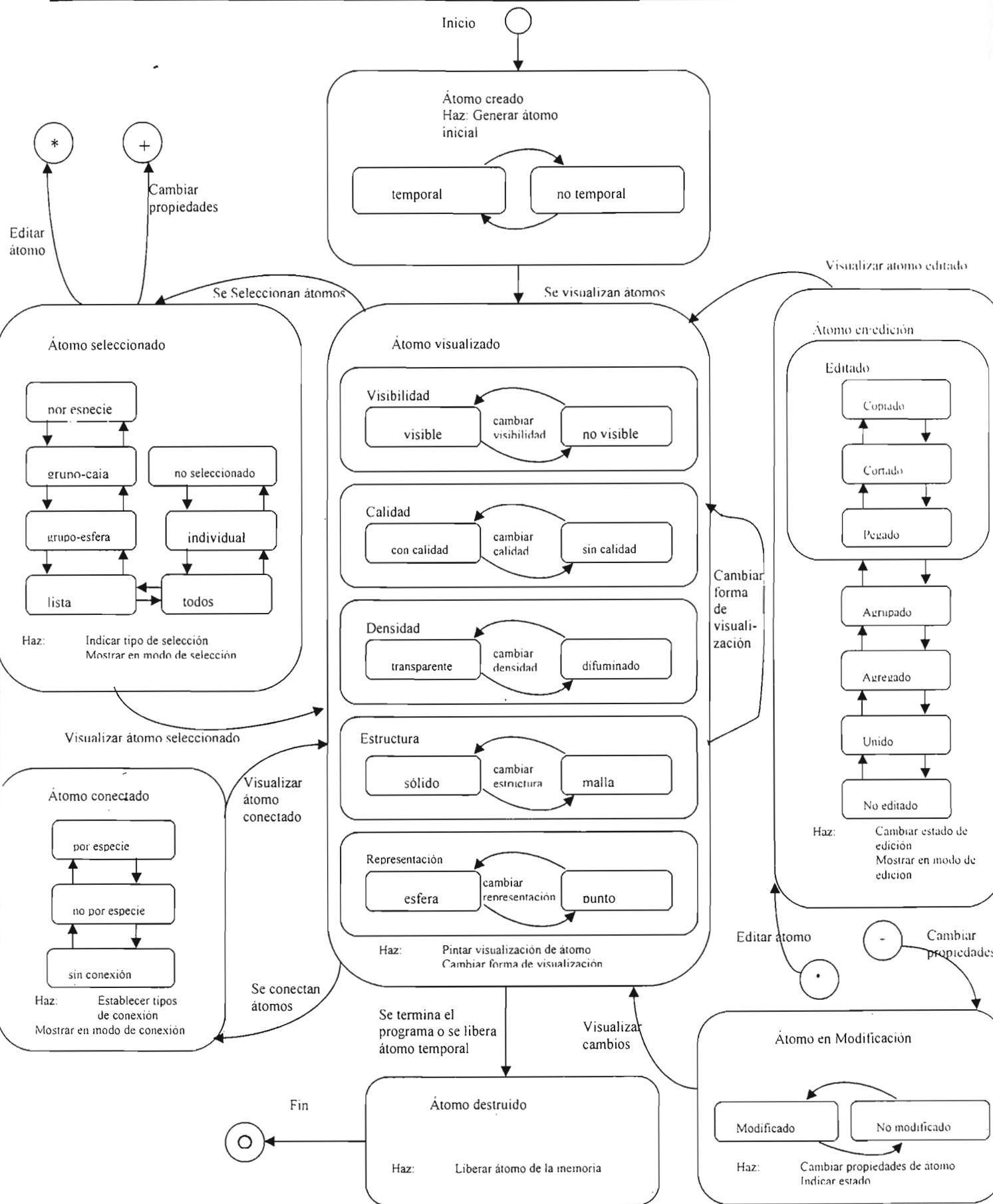


Escenario 1.2

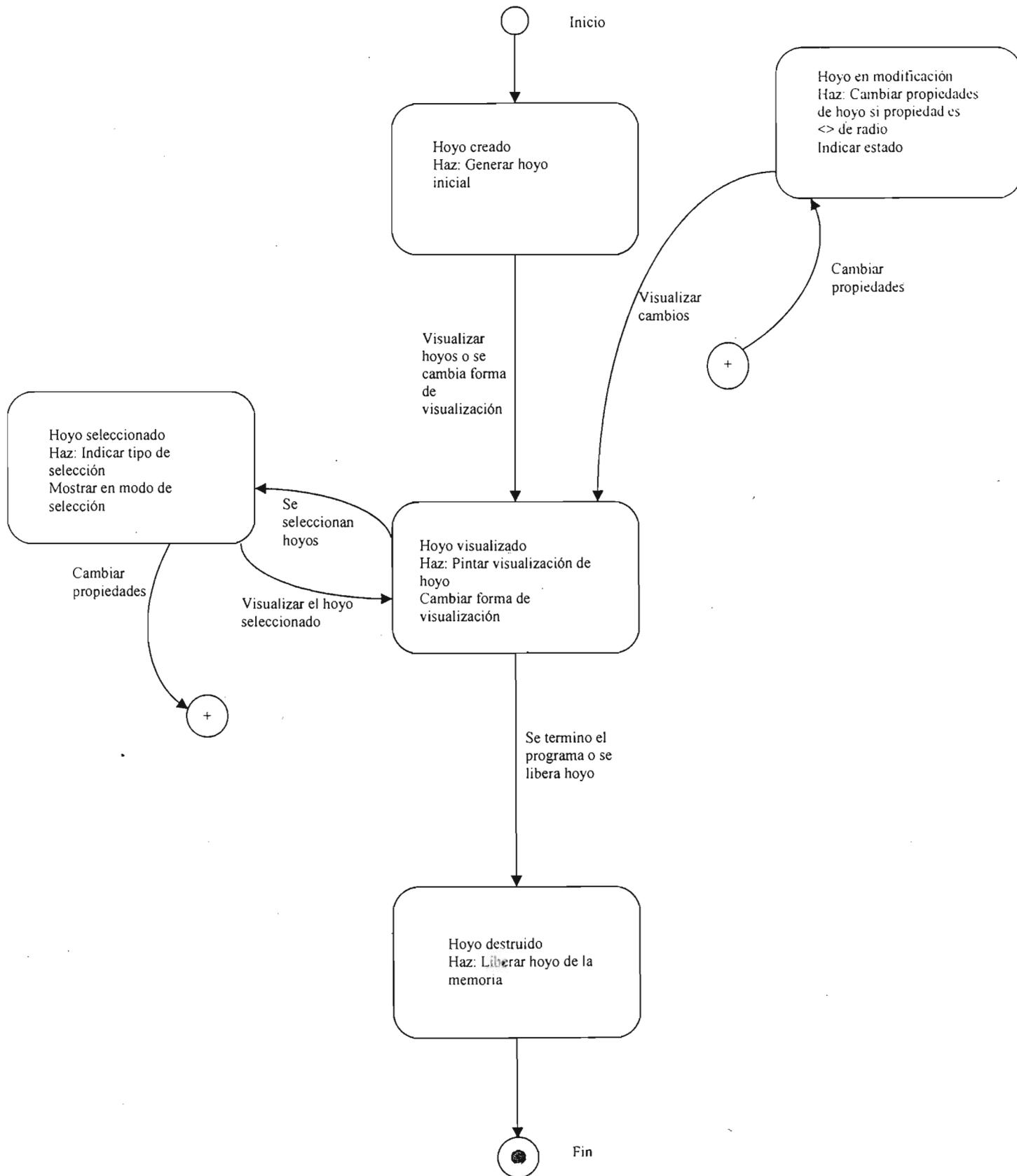


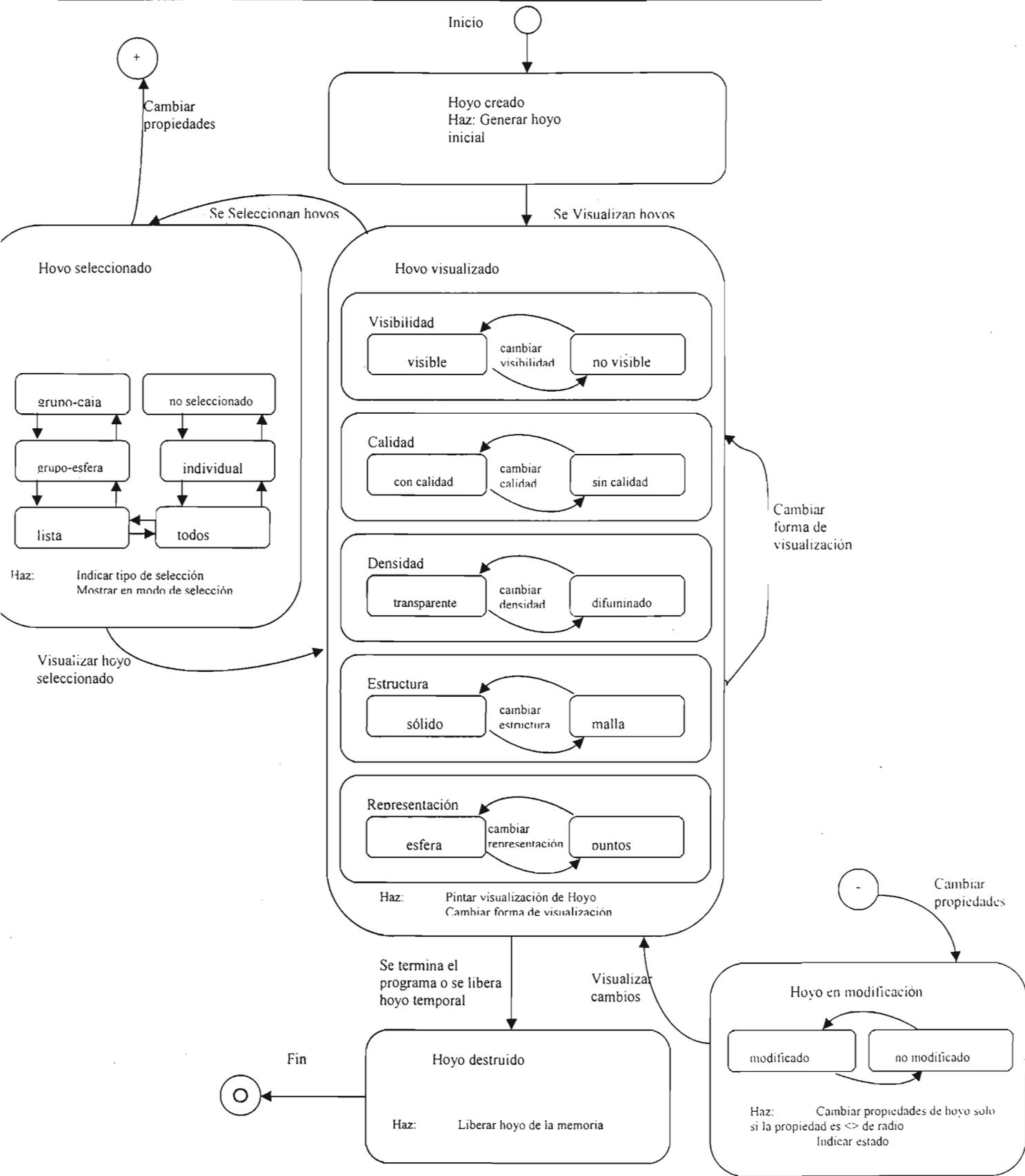
9.1.2.4.2. DIAGRAMAS DE ESTADOS DE LA CLASE ÁTOMO



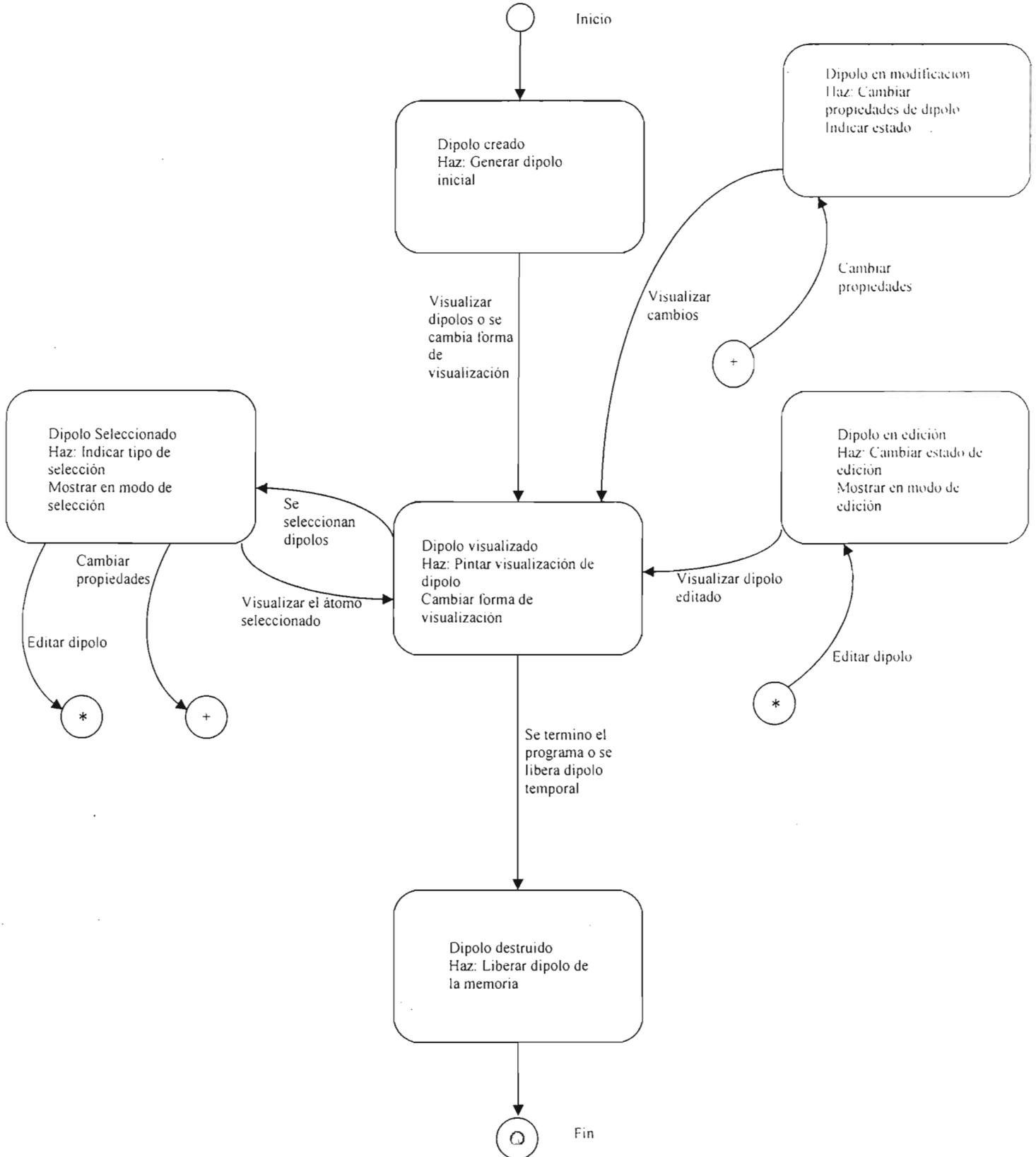


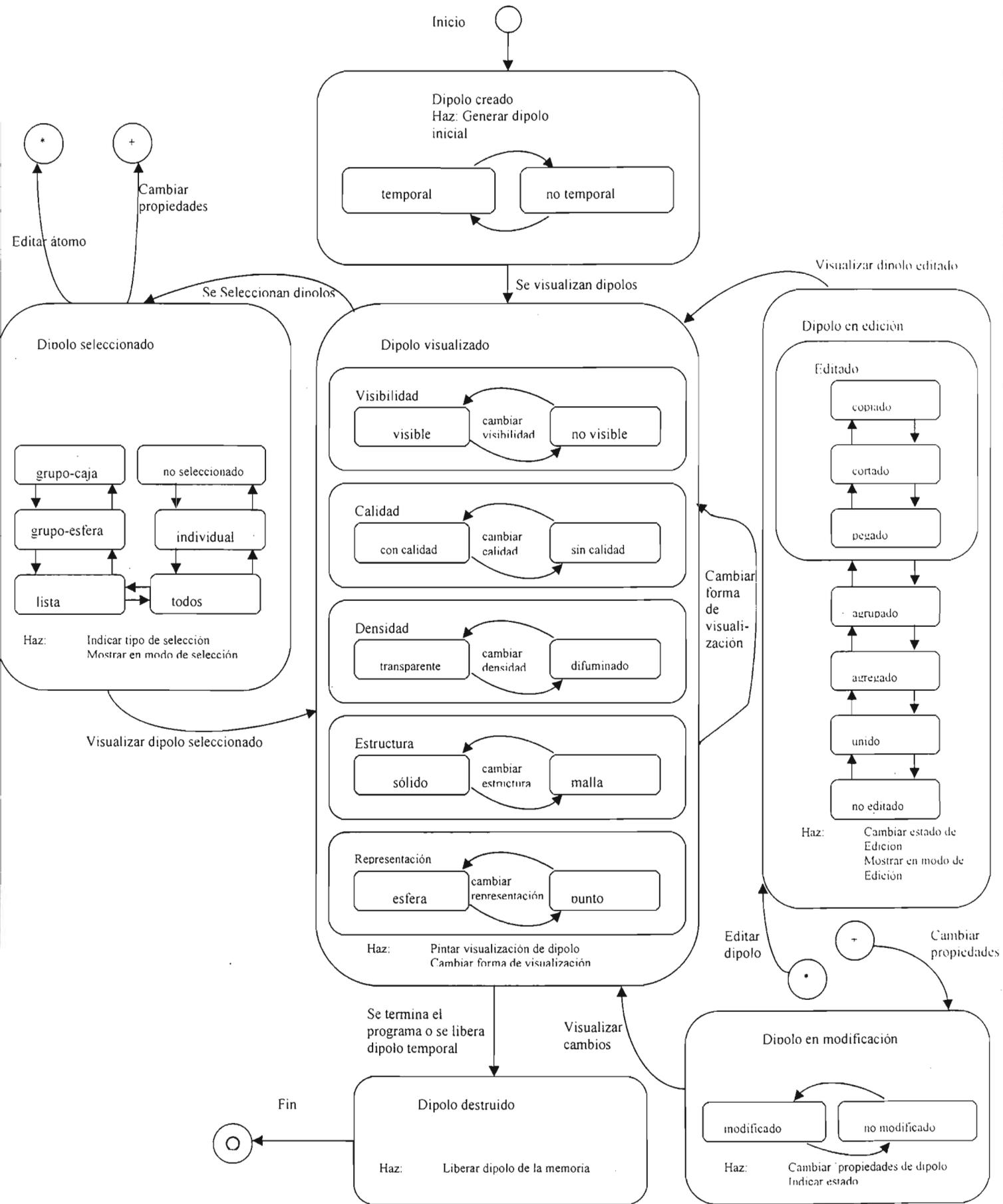
9.1.2.4.3. DIÁGRAMAS DE ESTADOS DE LA CLASE HOYO



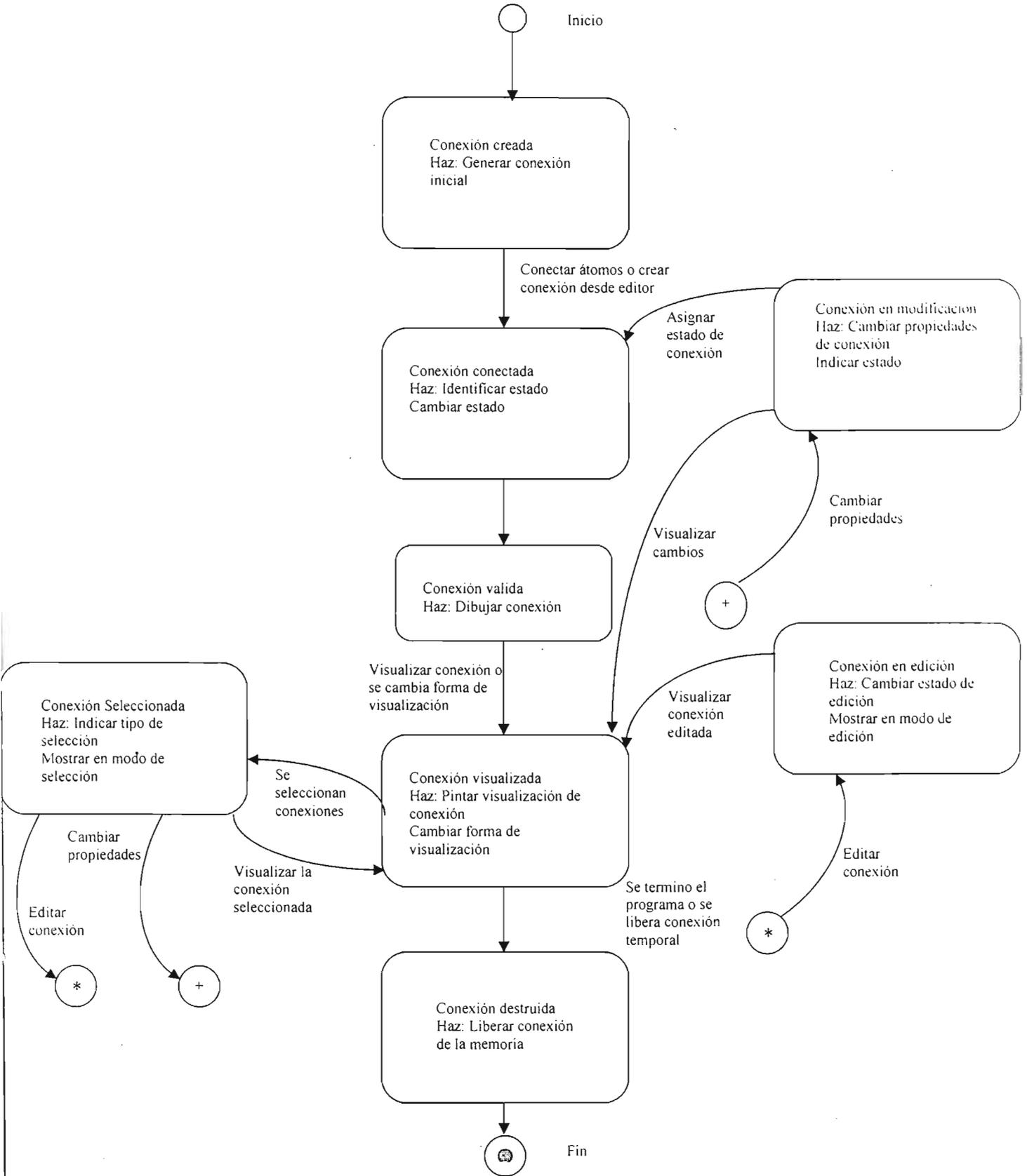


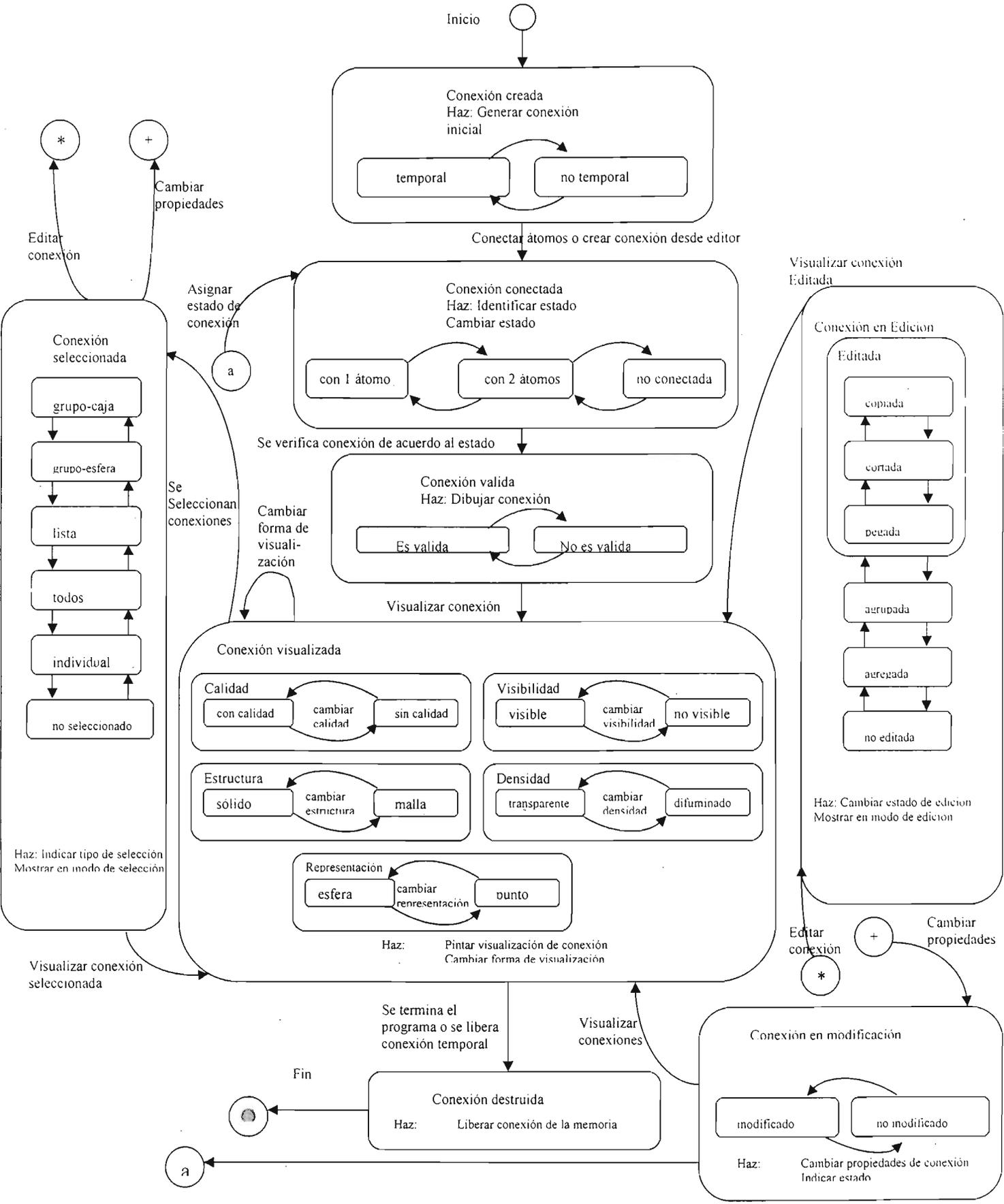
9.1.2.4.4 DIAGRAMAS DE ESTADOS DE LA CLASE DIPOLO





9.1.2.4.5. DIÁGRAMAS DE ESTADOS DE LA CLASE CONEXION





9.1.2.4.6 DIAGRAMAS DE ESTADOS DE LA CLASE MOLÉCULA

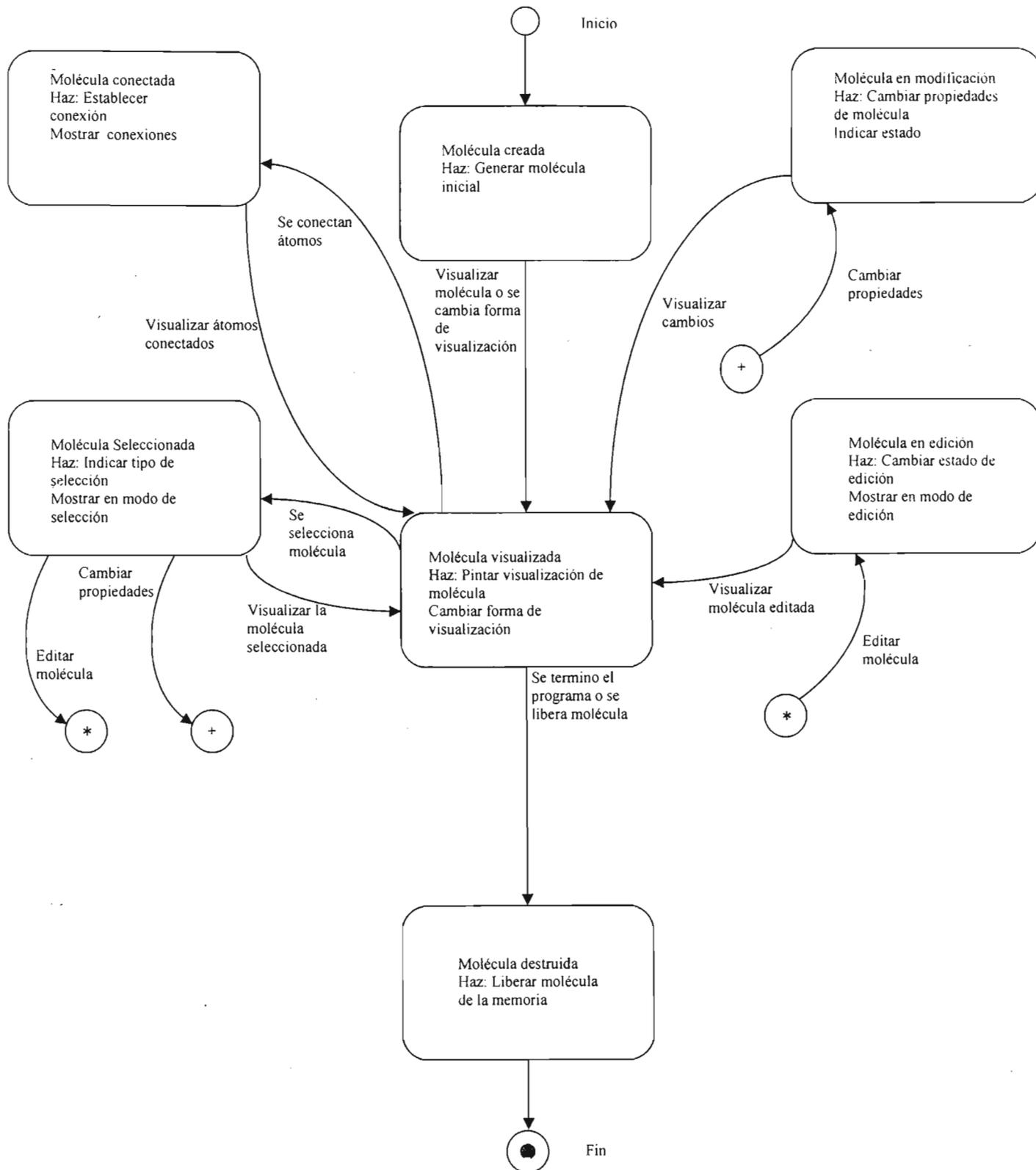
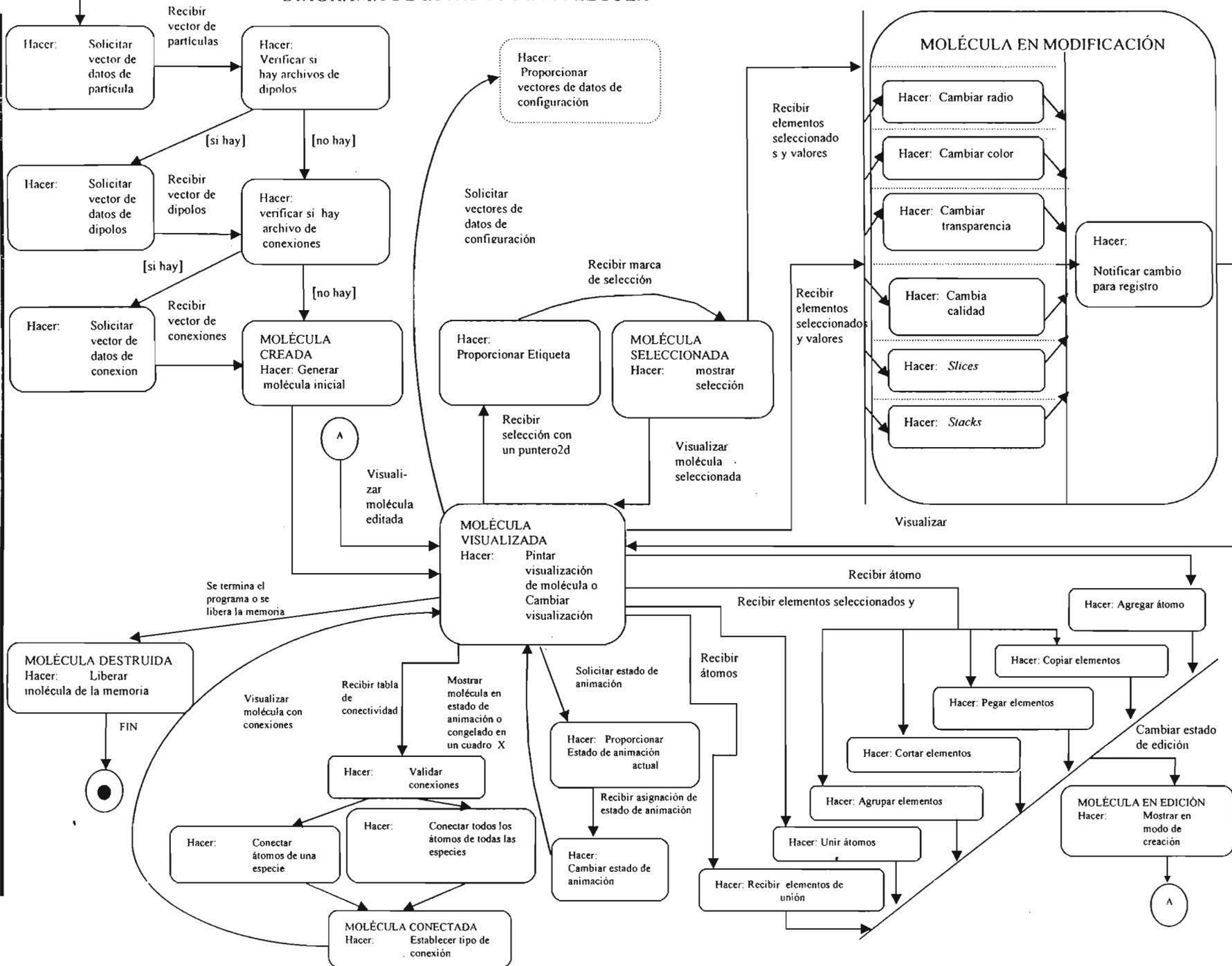


DIAGRAMA DE ESTADOS DE MOLÉCULA



9.1.2.4.7 DIAGRAMAS DE ESTADOS DE LA CLASE SISTEMA DE CAVIDADES

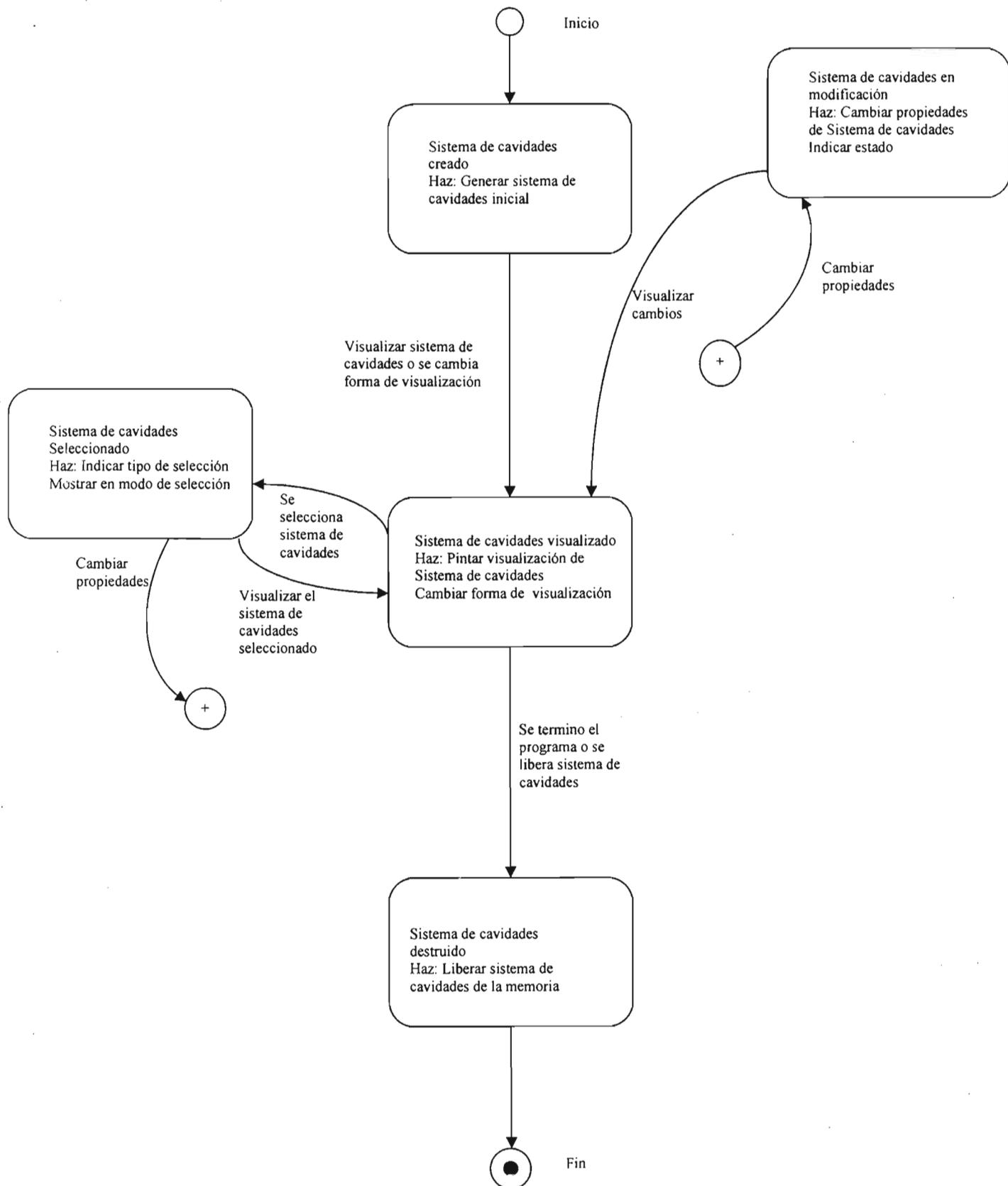
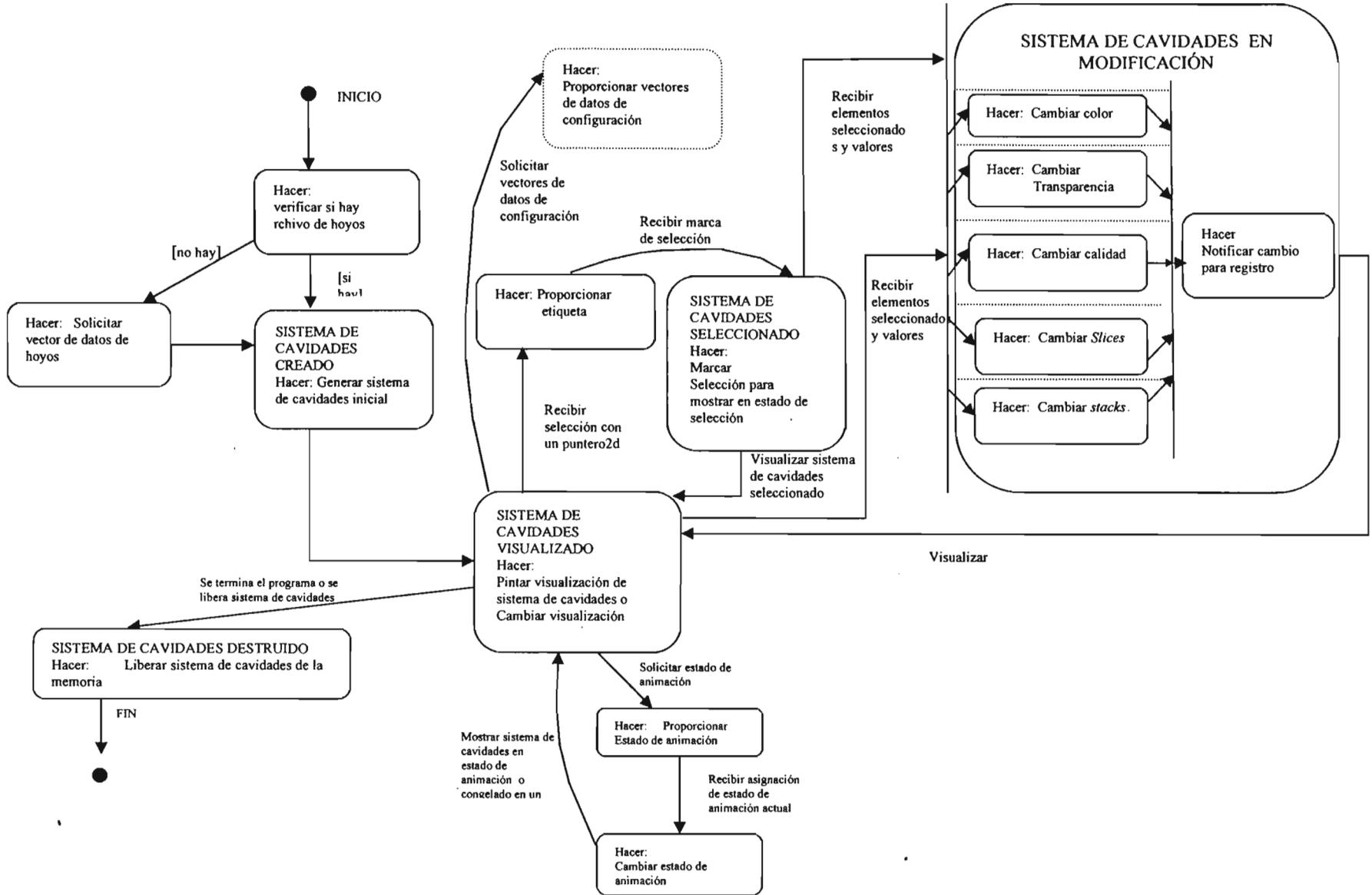


DIAGRAMA DE ESTADOS DE SISTEMA DE CAVIDADES



9.1.2.4.8 DIAGRAMAS DE ESTADOS DE LA CLASE SUPERFICIE DE POTENCIAL

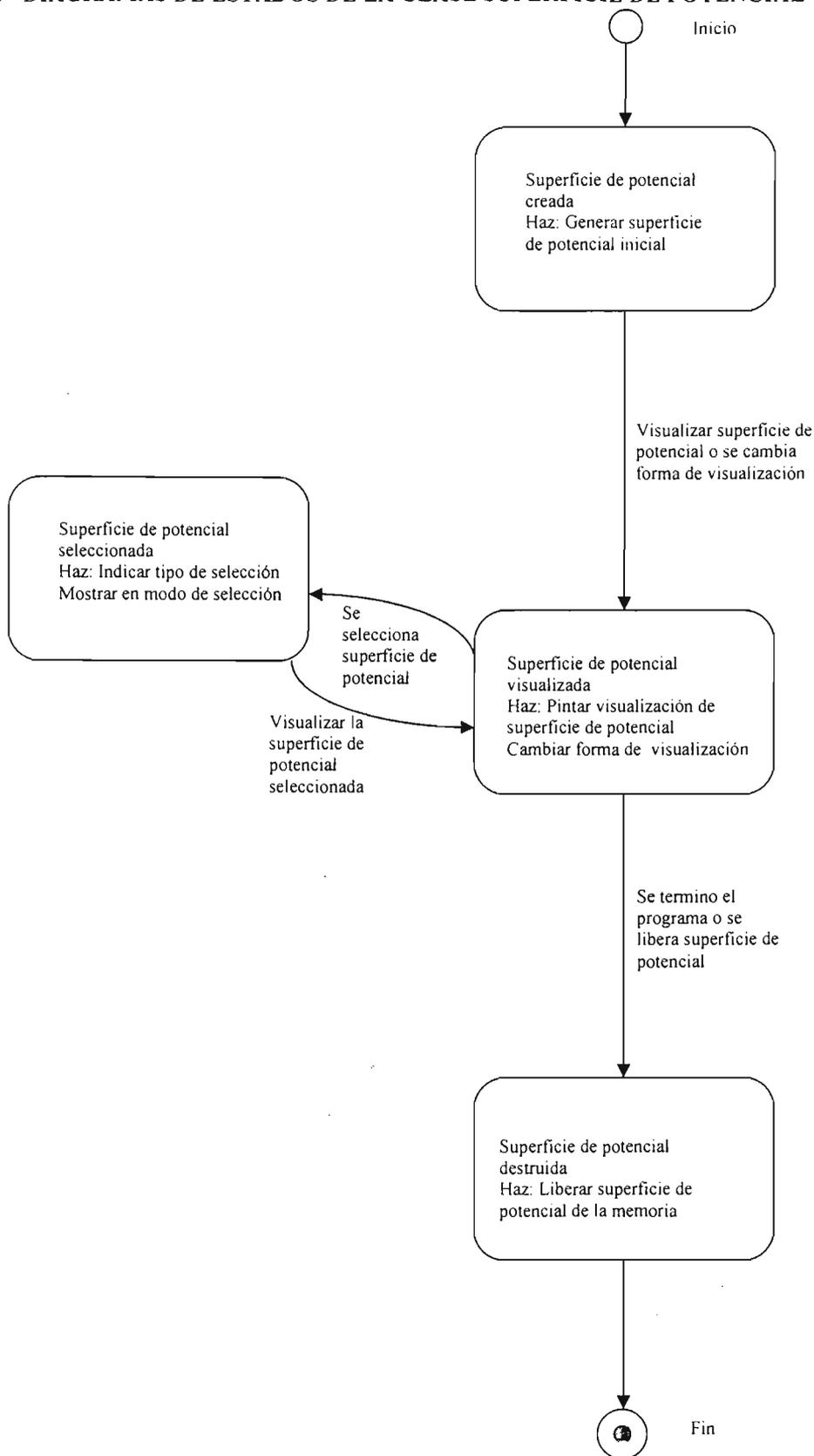
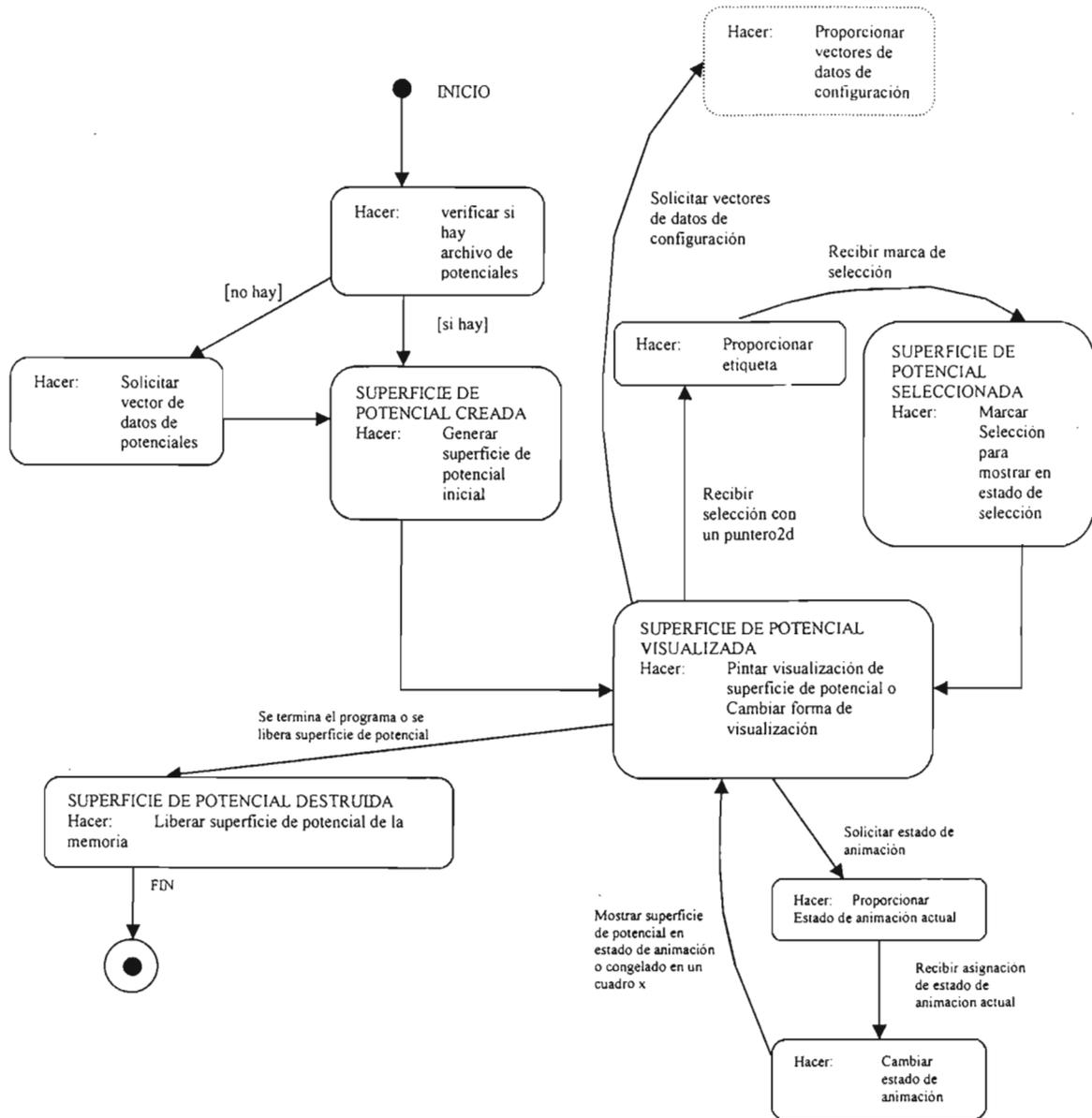
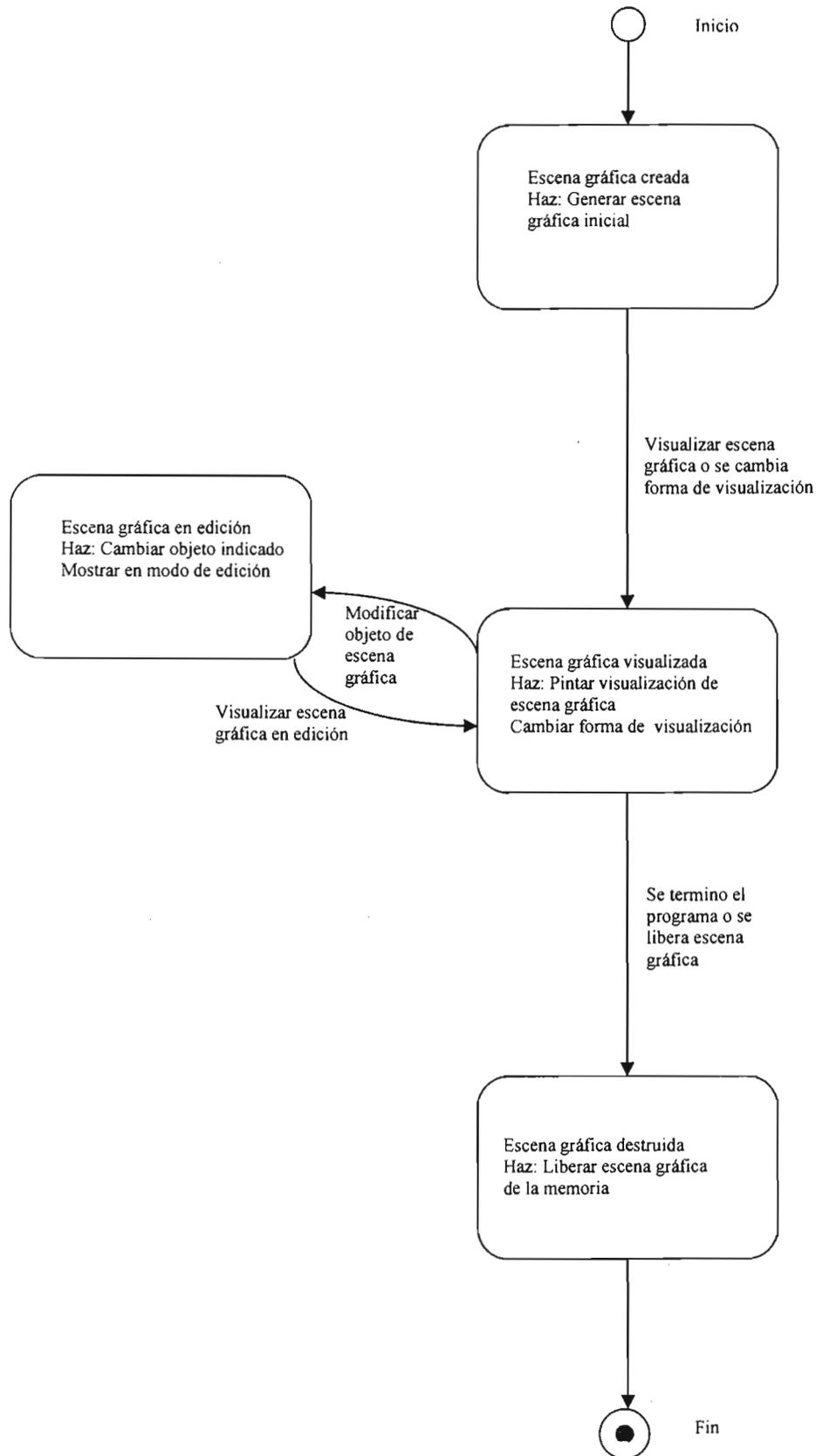
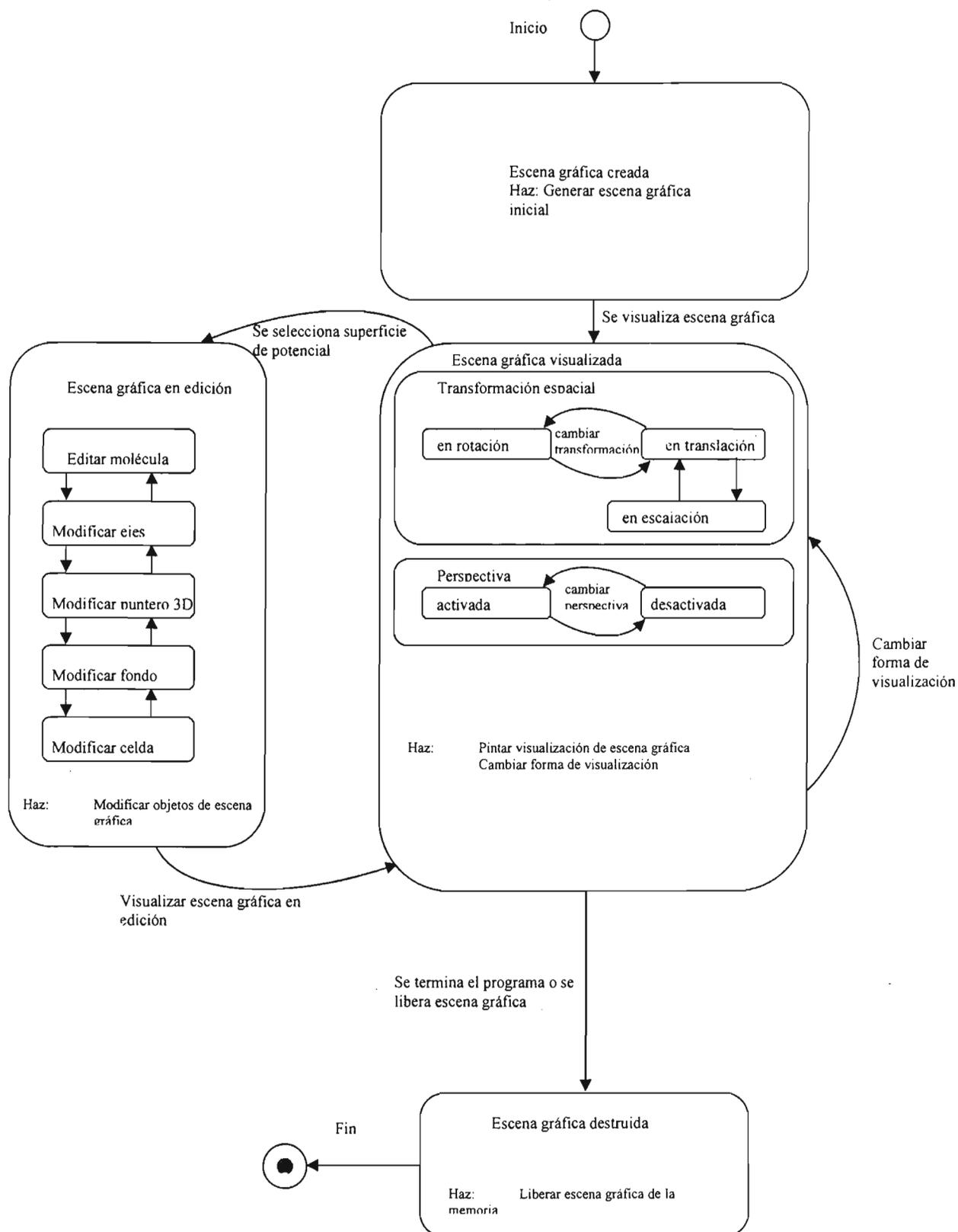


DIAGRAMA DE ESTADOS DE SUPERFICIE DE POTENCIAL

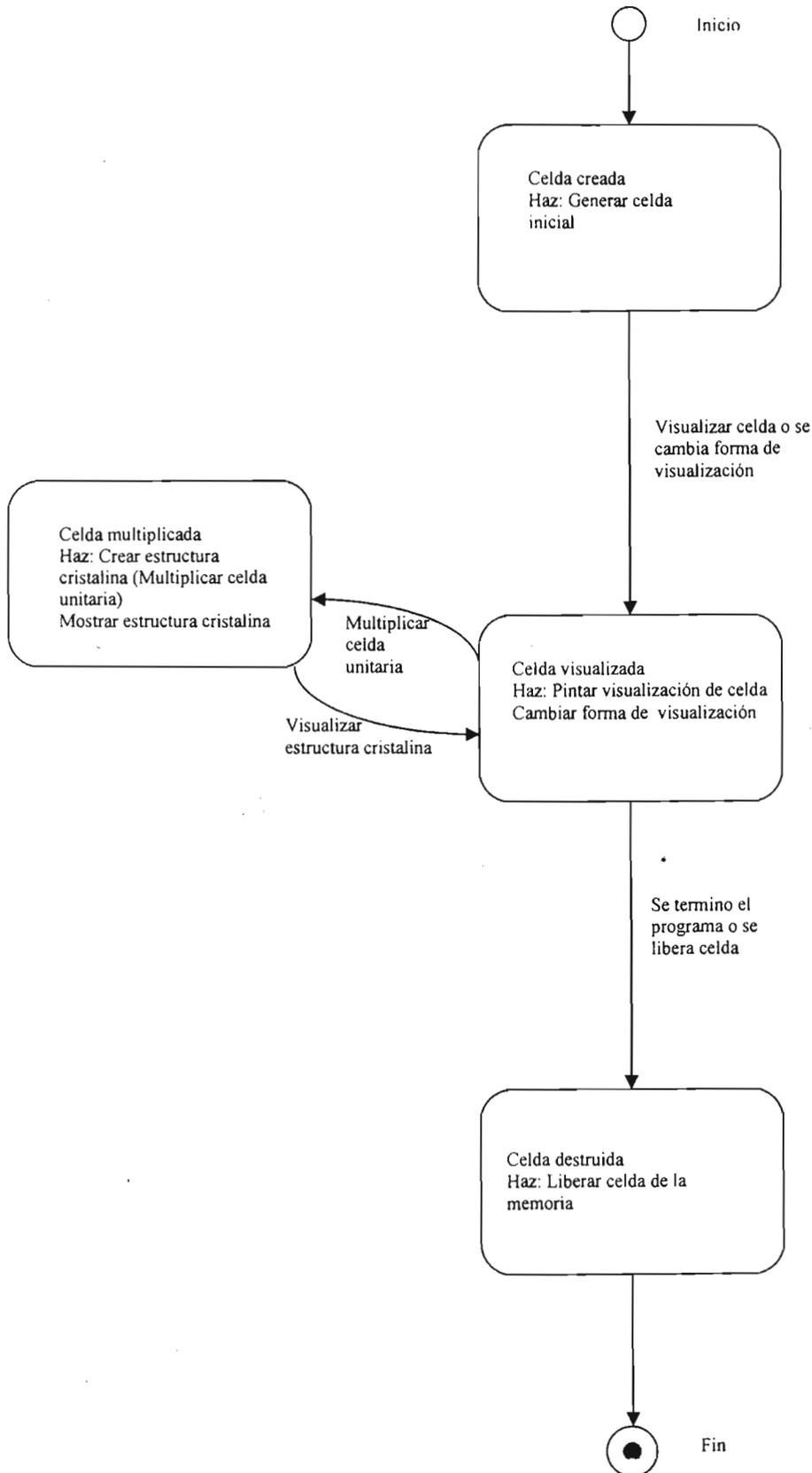


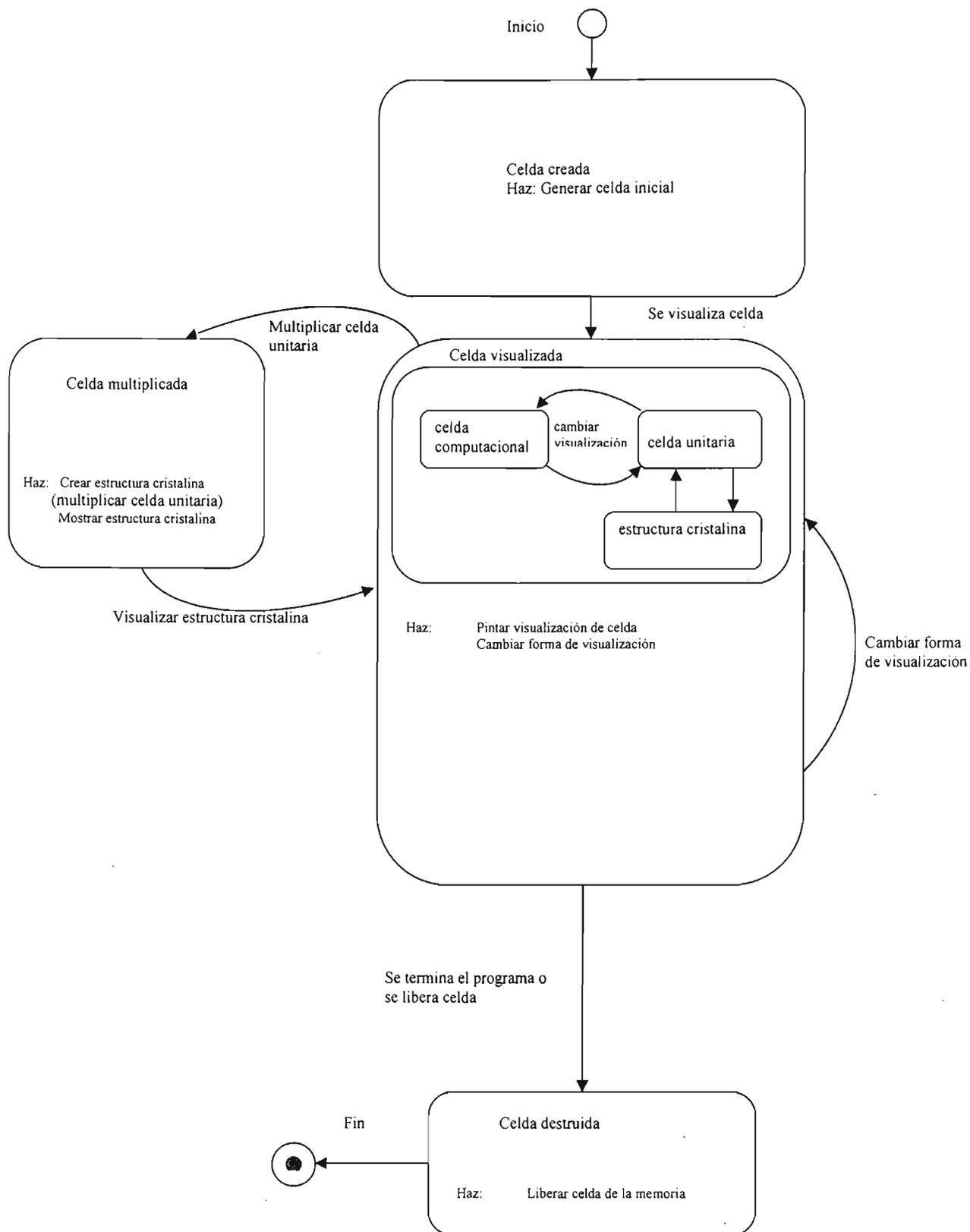
9.1.2.4.9 DIAGRAMAS DE ESTADOS DE LA CLASE ESCENA GRAFICA



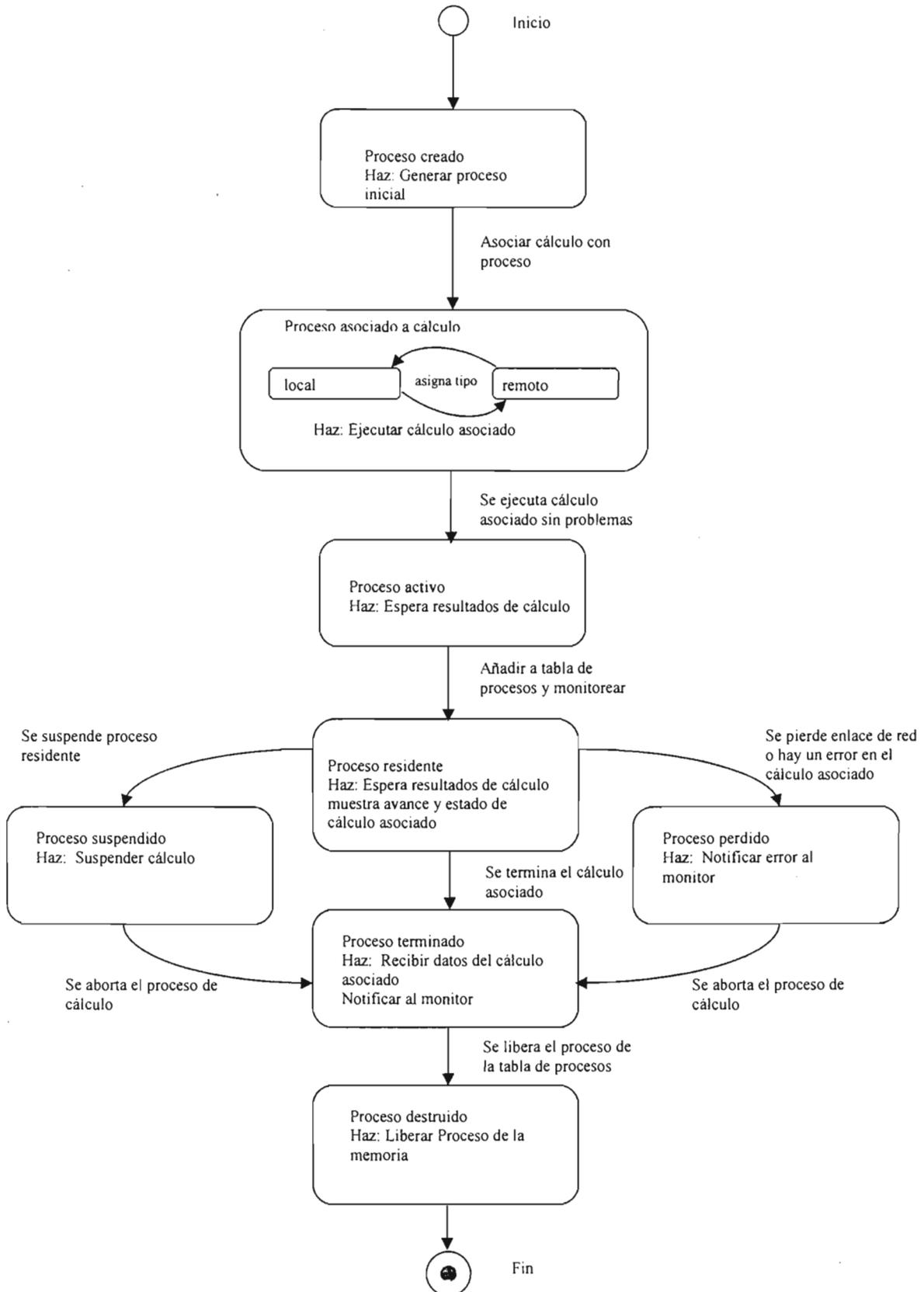


9.1.2.4.10 DIAGRAMAS DE ESTADOS DE LA CLASE CELDA

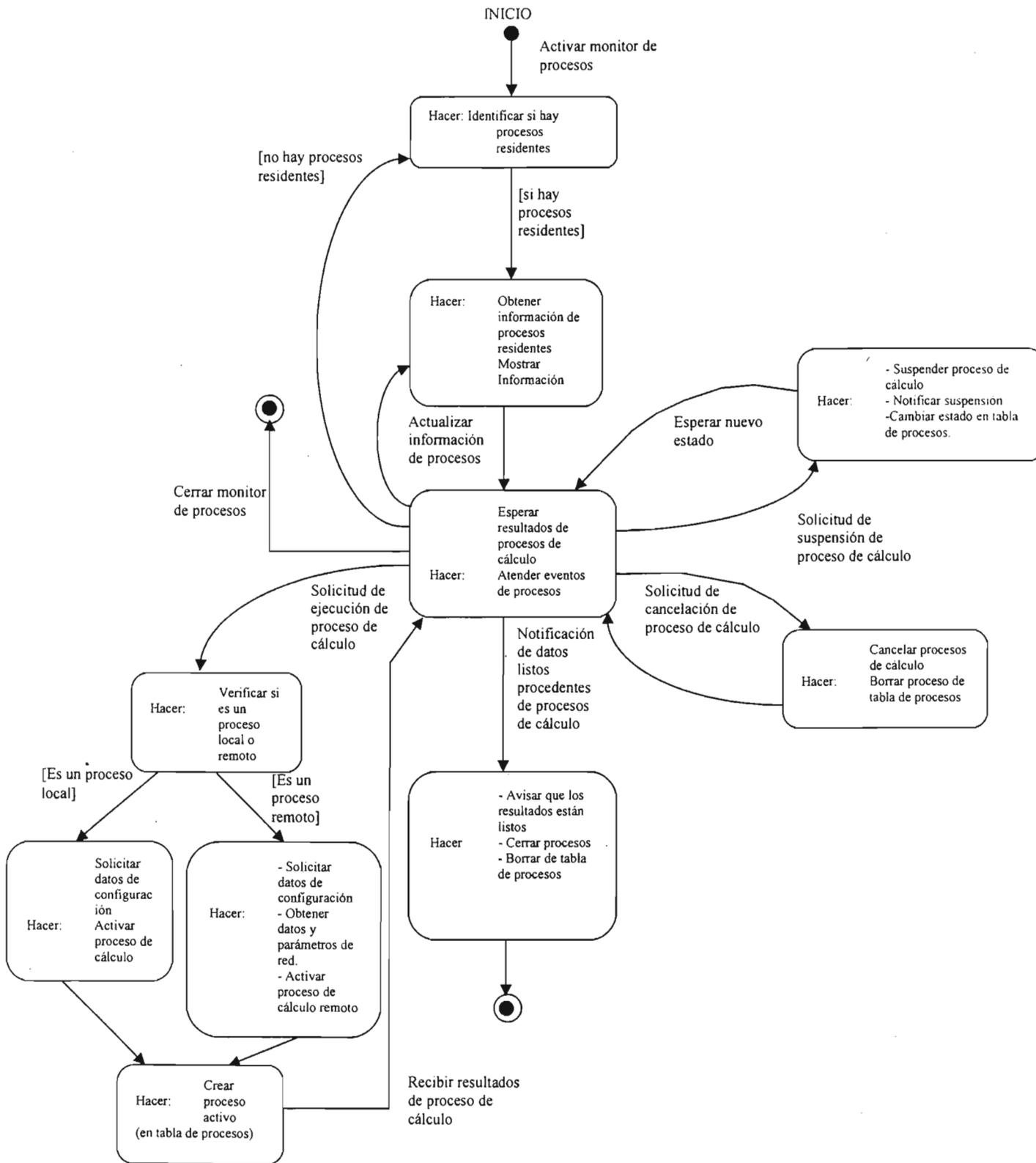




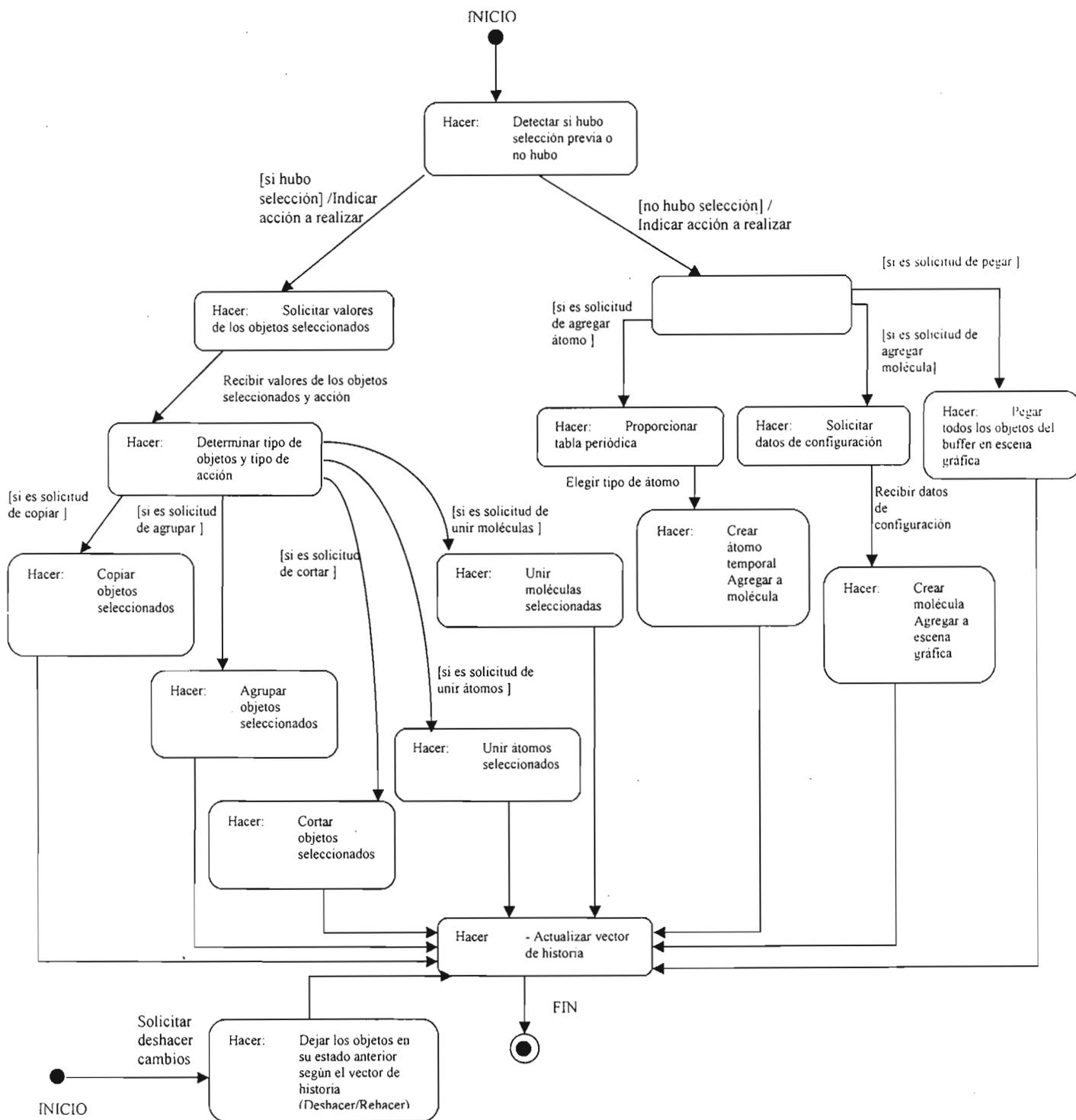
9.1.2.4.11 DIAGRAMA DE ESTADOS DE LA CLASE PROCESO



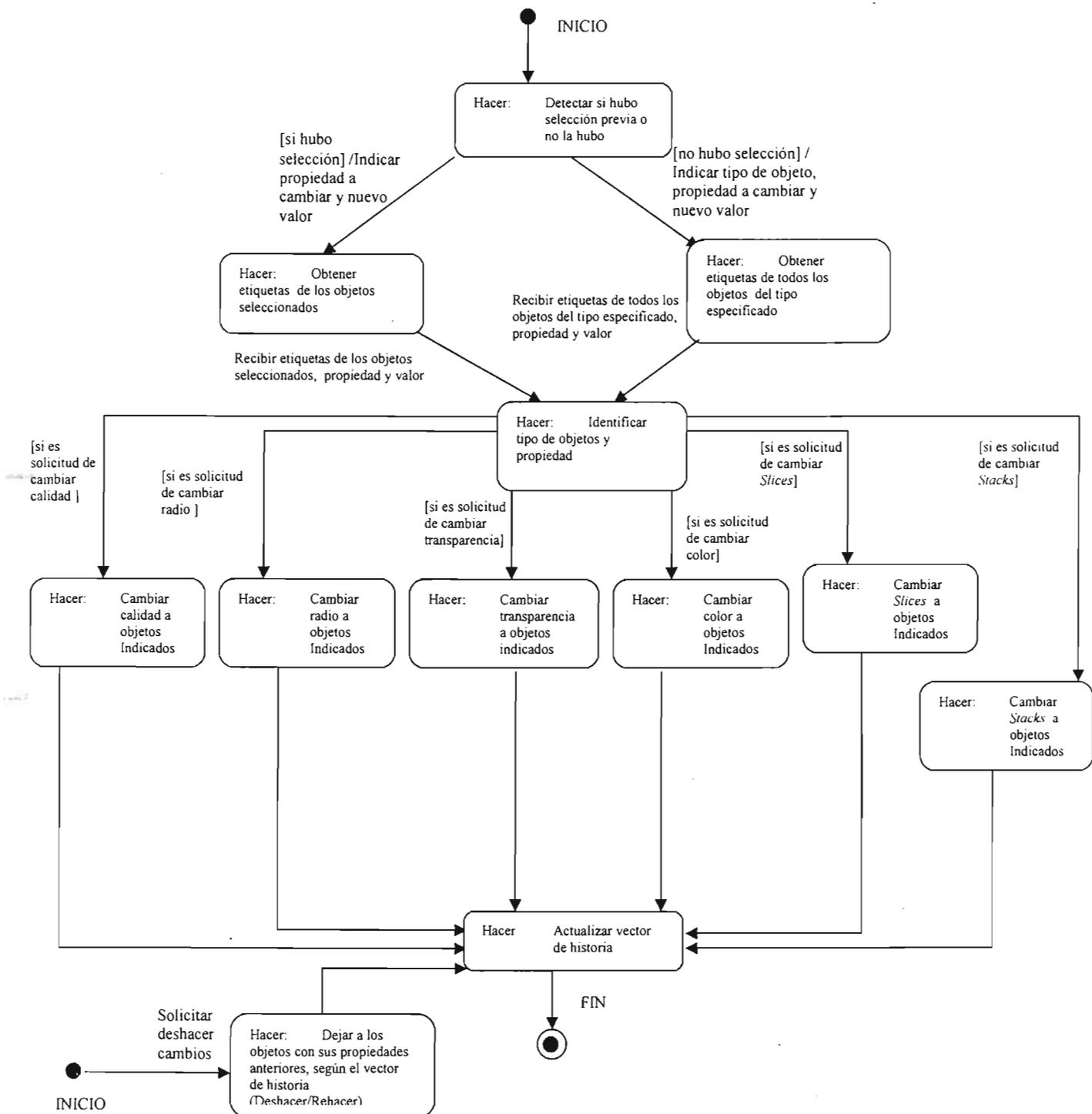
9.1.2.4.12 - DIAGRAMA DE ESTADOS DE LA CLASE MONITOR DE PROCESOS



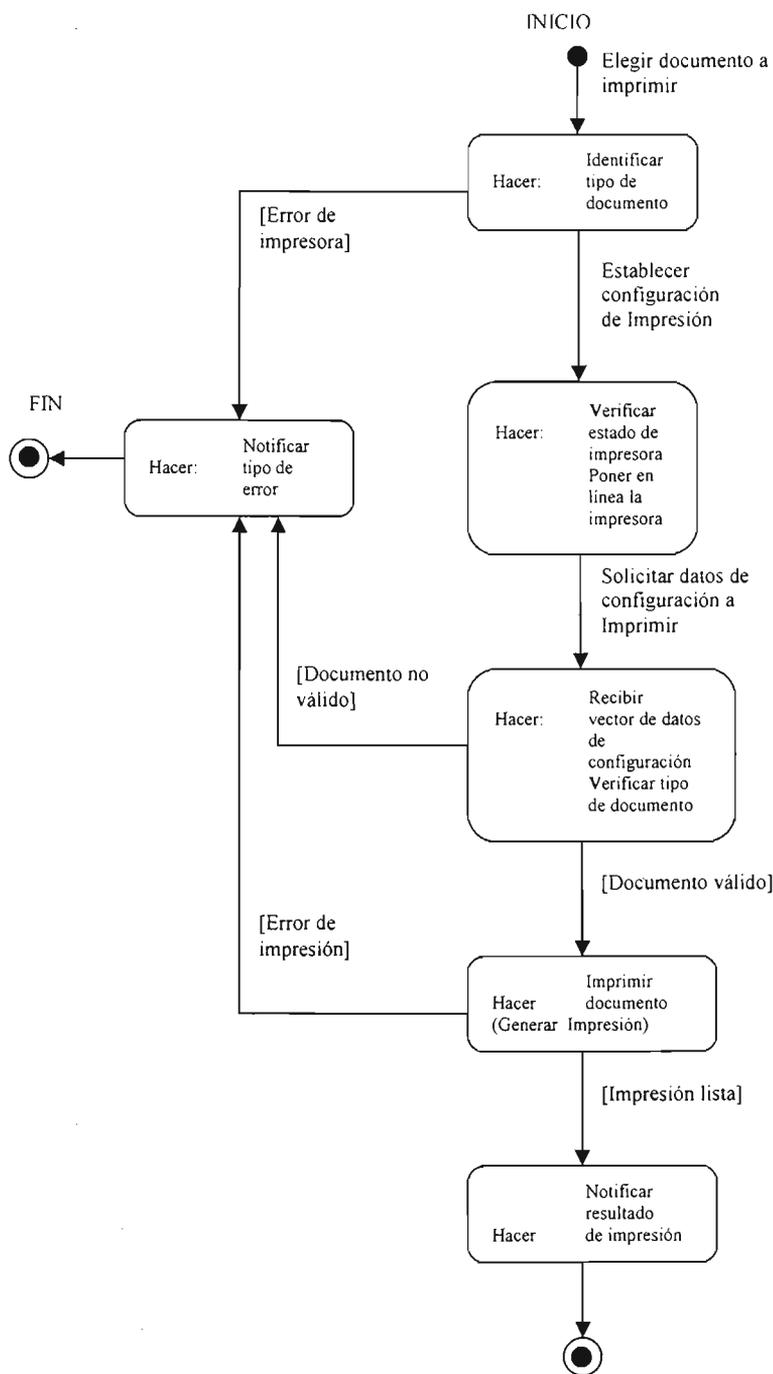
9.1.2.4.13 DIAGRAMA DE ESTADOS DE LA CLASE EDITOR MOLECULAR



9.1.2.4.14 DIAGRAMA DE ESTADOS DE LA CLASE MODIFICADOR DE PROPIEDADES



9.1.2.4.15 DIAGRAMA DE ESTADOS DE LA CLASE IMPRESIONES



9.1.3 Modelo funcional

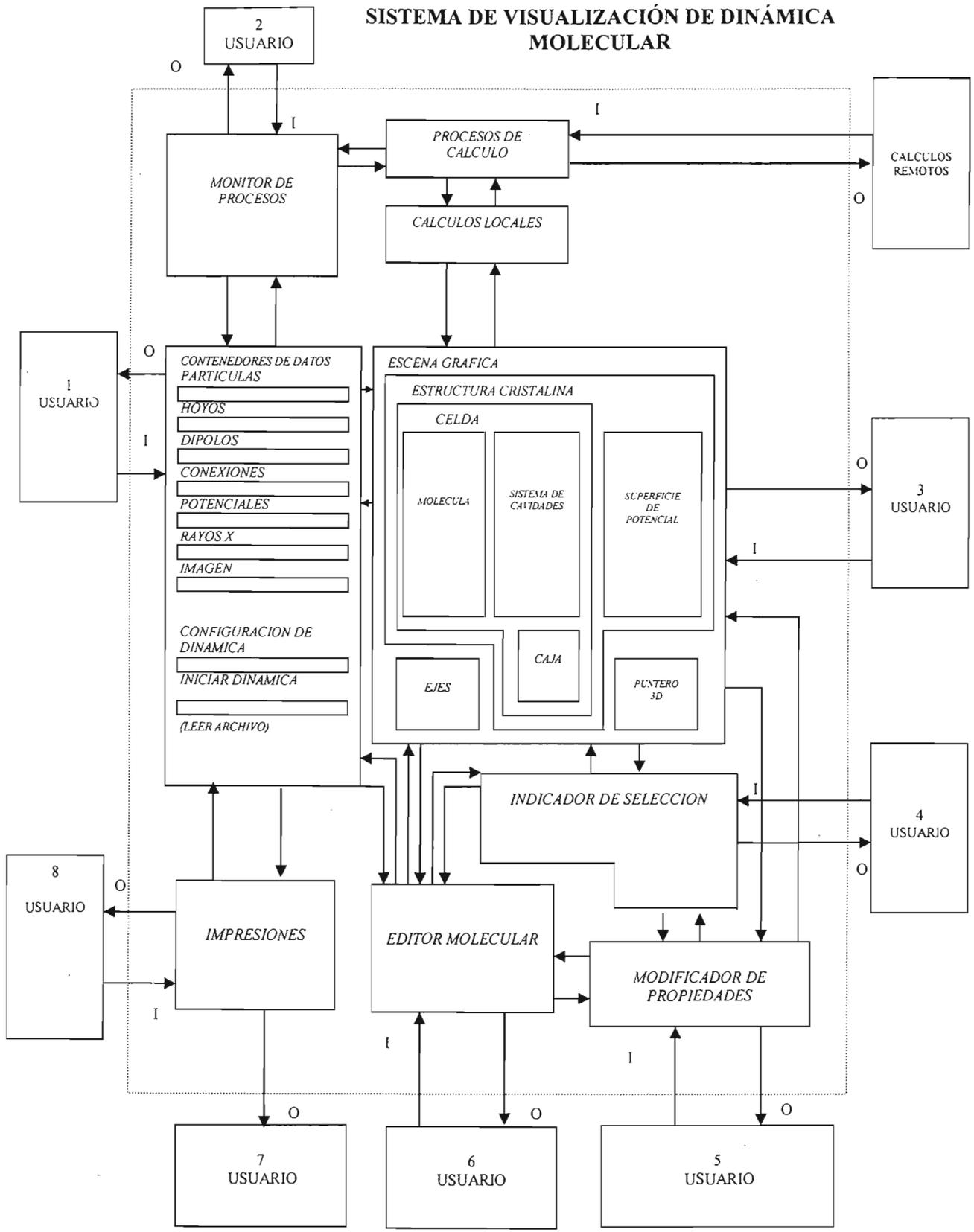
9.1.3.1 Identificar valores de entrada / salida

ENTRADAS	SALIDAS
1i) <ul style="list-style-type: none"> • Introducir nombre de archivo • Seleccionar archivo • Indicar configuración • Generar configuración de dinámica • Generar inicia dinámica 	1o) <ul style="list-style-type: none"> • Solicitar nombre de archivo • Entregar archivo de particulas • Entregar archivo de hoyos • Entregar archivo de potencial • Entregar archivo de dipolos • Entregar archivo de conexiones • Entregar archivo de imagen • Entregar archivo de rayos X • Notificar resultados de operaciones • Entregar archivo de configuración de dinámica • Entregar archivo de inicia dinámica • Presentar inicia dinámica para modificación
2i) <ul style="list-style-type: none"> • Llamar al monitor de procesos de cálculo • Solicitar suspender proceso de cálculo • Indicar proceso de cálculo a ejecutar (calcular imagen, dinámica, potenciales, dipolos, hoyos) • Indicar parámetros de RED 	2o) <ul style="list-style-type: none"> • Mostrar la Información y el estado de los procesos
3i) <ul style="list-style-type: none"> • Seleccionar átomos sólo de una especie • Seleccionar un grupo de átomos con una caja • Seleccionar un grupo de átomos con una esfera • Seleccionar un grupo de dipolos con una esfera • Seleccionar un grupo de hoyos con una esfera • Seleccionar una molécula con un puntero 2D • Seleccionar conexiones con un puntero 2D • Seleccionar átomos con un puntero 3D • Indicar tipo y radio de conexión, tabla de conectividad y especies de átomos • Indicar tipo de transformación espacial, dispositivo y eje • Realizar transformación con las perillas sobre el eje x • Elegir objeto de animación • Elegir acción a realizar • Solicitar multiplicación de celda unitaria • Introducir factores 	3o) <ul style="list-style-type: none"> • Mostrar cambios efectuados • Notificar si la celda es una base o no lo es
4i) <ul style="list-style-type: none"> • Indicar modo de selección y tipo de objeto a seleccionar 	
5i) <ul style="list-style-type: none"> • Indicar propiedades a cambiar • Indicar clases de objetos • Introducir valores • Seleccionar objetos • Indicar deshacer cambios 	
6i) <ul style="list-style-type: none"> • Indicar objeto a agregar • Elegir tipo de átomo a agregar • Indicar objetos de molécula disponibles • Indicar acción a realizar (copiar, pegar, agrupar, cortar, unir, deshacer cambios) • Indicar tipo de unión (entre dos átomos, unir moléculas) • Seleccionar molécula a unir 	
	7o) <ul style="list-style-type: none"> • Generar impresión de documento
8i) <ul style="list-style-type: none"> • Establecer configuración de impresora • Elegir documento a imprimir 	8o) <ul style="list-style-type: none"> • Notificar estado de impresión

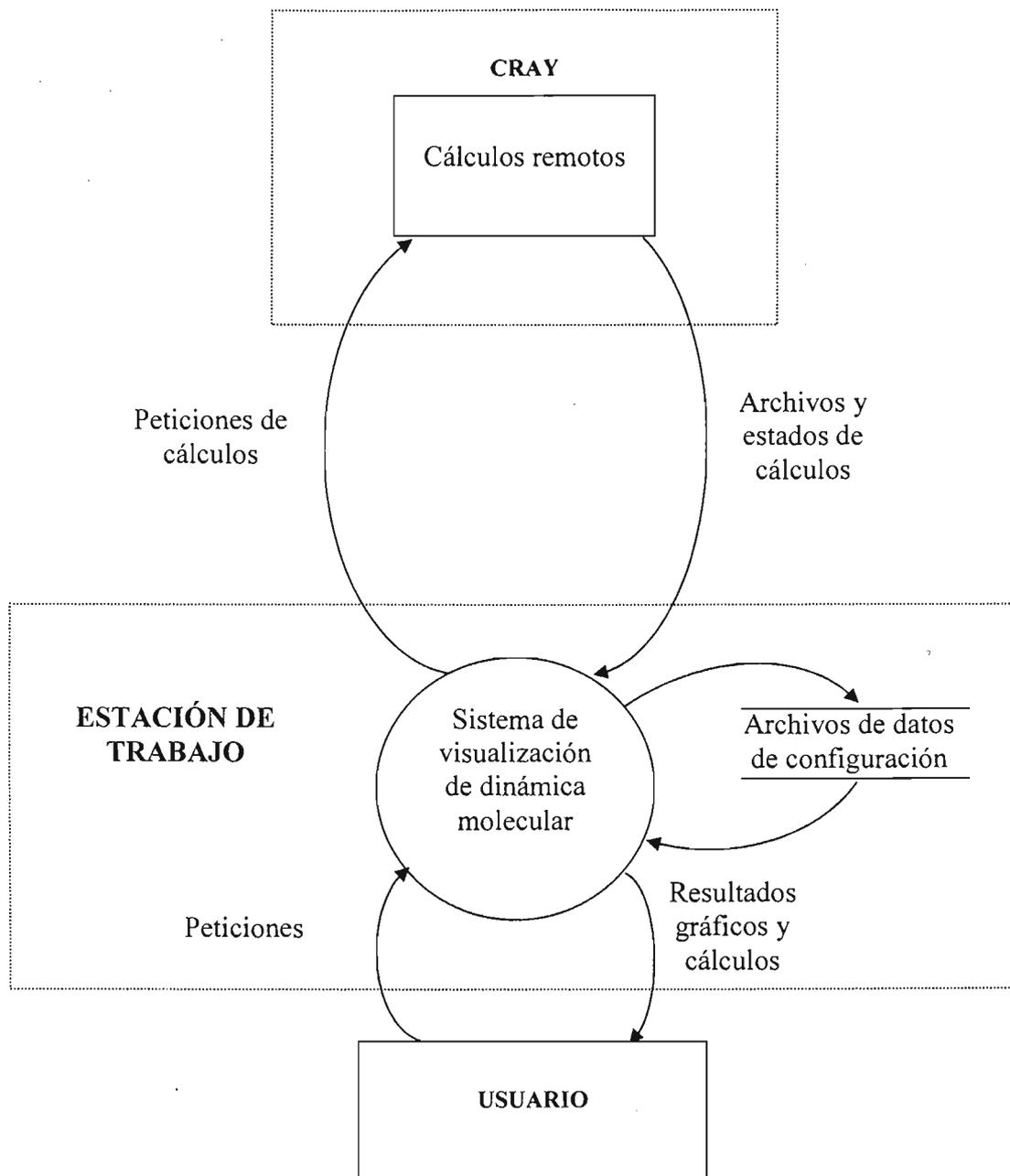
9.1.3.2 Depurar entradas y salidas

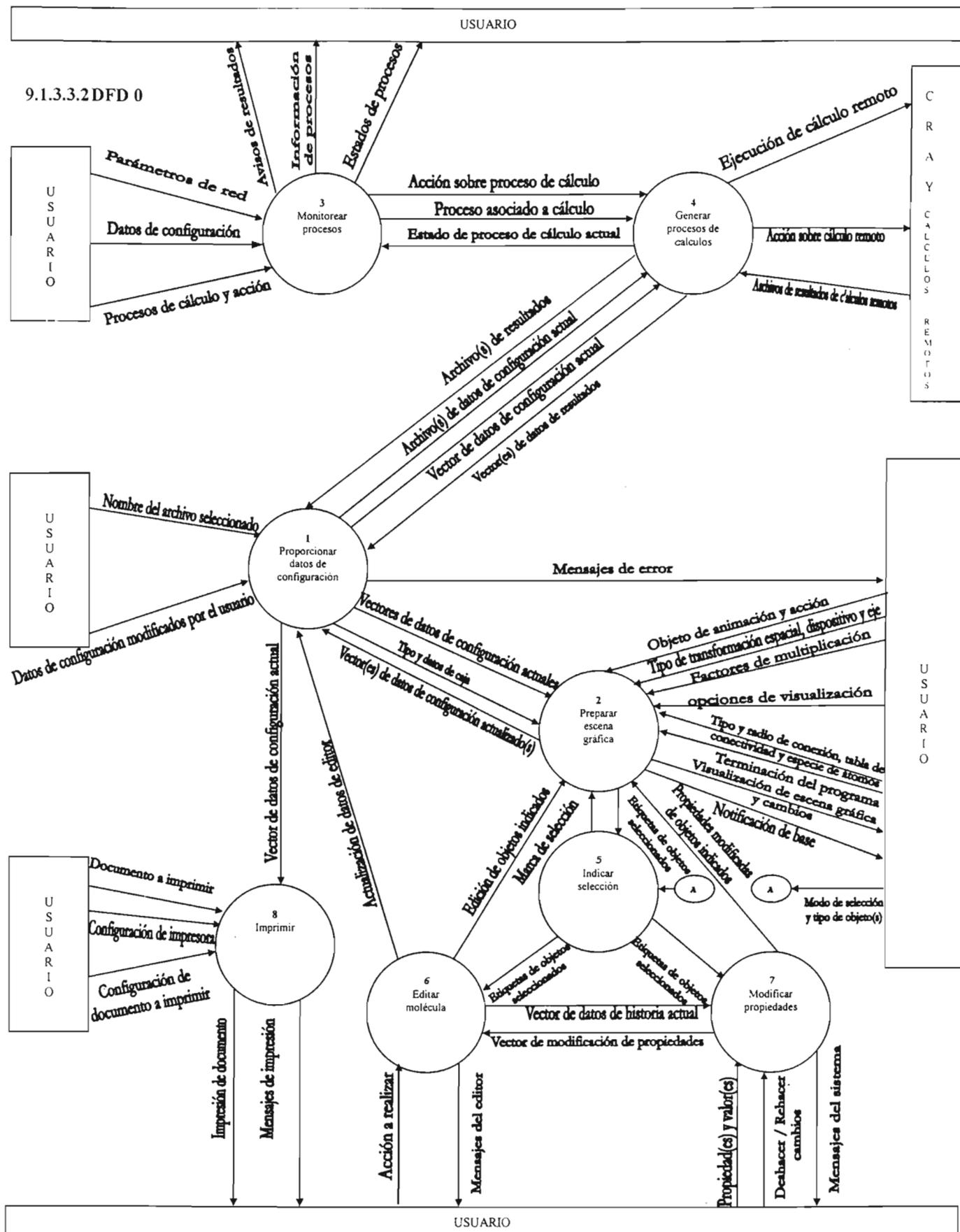
1) Proporcionar datos de configuración	
Entradas en el nivel 0	Salidas en el nivel 0
<ol style="list-style-type: none"> 1) Nombre del archivo seleccionado 2) Datos de configuración modificados por el usuario 3) Actualización de datos del editor. 4) Vectores de datos de configuración actualizados 5) Vector(es) de datos de resultados 6) Archivos de resultados 	<ol style="list-style-type: none"> 1) Vector de datos de configuración actual (va hacia imprimir) 2) Tipo y datos de caja (va hacia preparar escena gráfica) 3) Vectores de datos de configuración actuales (va hacia preparar escena gráfica) 4) Mensajes de error (va hacia usuario) 5) Vector de datos de configuración actual (va hacia generar procesos de cálculo) 6) Archivos de datos de configuración actual (va hacia procesos de cálculo)
2) Preparar Escena Gráfica	
Entradas en el nivel 0	Salidas en el nivel 0
<ol style="list-style-type: none"> 1. Vectores de datos de configuración actuales 2. Tipo y datos de caja 3. Edición de objetos indicados 4. Marca de selección 5. Propiedades modificadas de objetos indicados 6. Terminación del programa 7. Tipo y radio de conexión, tabla de conectividad y especie de átomos 8. Factores de multiplicación 9. Tipo de transformación espacial, dispositivo y eje 10. Objeto de animación y acción 11. Opciones de visualización 	<ol style="list-style-type: none"> 1. Vectores de datos de configuración actualizados (va hacia proporcionar datos de configuración) 2. Etiquetas de objetos seleccionados (va hacia indicar selección) 3. Notificación de base (va hacia usuario) 4. Visualización de escena gráfica y cambios (va hacia usuario)
3) Monitorear procesos	
Entradas en el nivel 0	Salidas en el nivel 0
<ol style="list-style-type: none"> 1. Parámetros de red 2. Procesos de cálculo y acción 3. Estado de proceso de cálculo actual 4. Datos de configuración 	<ol style="list-style-type: none"> 1. Avisos de resultados (va hacia usuario) 2. Información de procesos (va hacia usuario) 3. Estados de procesos (va hacia usuario) 4. Acción sobre proceso de cálculo (va hacia generar procesos de cálculo) 5. Proceso asociado a cálculo (va hacia generar procesos de cálculo)
4) Procesos de Cálculos	
Entradas en el nivel 0	Salidas en el nivel 0
<ol style="list-style-type: none"> 1. Acción sobre proceso de cálculo 2. Proceso asociado a cálculo 3. Archivos de resultados de cálculos remotos 4. Vector de datos de configuración actual 5. Archivo(s) de datos de configuración actual 	<ol style="list-style-type: none"> 1. Estado de proceso de cálculo actual (va hacia monitorear procesos) 2. Acción sobre cálculo remoto (va hacia cálculos remotos) 3. Vector(es) de datos de resultados (va hacia proporcionar datos de configuración) 4. Archivo(s) de resultados (va hacia proporcionar datos de configuración) 5. Ejecución de cálculo remoto (va hacia cálculos remotos)
5) Indicar Selección	
Entradas en el nivel 0	Salidas en el nivel 0
<ol style="list-style-type: none"> 1. Etiquetas de objetos seleccionados 2. Modo de selección y tipo de objeto(s) 	<ol style="list-style-type: none"> 1. Etiquetas de objetos seleccionados (va hacia editor molecular) 2. Etiquetas de objetos seleccionados (va hacia modificar propiedades) 3. Marca de selección (va hacia preparar escena gráfica)
6) Editar Molécula	
Entradas en el nivel 0	Salidas en el nivel 0
<ol style="list-style-type: none"> 1. Acción a realizar 2. Vector de modificación de propiedades 3. Etiquetas de objetos seleccionados 	<ol style="list-style-type: none"> 1. Actualización de datos del editor (va hacia proporcionar datos de configuración) 2. Mensajes del editor (va hacia usuario) 3. Vector de datos de historia actual (va hacia modificar propiedades) 4. Edición de objetos indicados (va hacia preparar escena gráfica)
7) Modificar Propiedades	
Entradas en el nivel 0	Salidas en el nivel 0
<ol style="list-style-type: none"> 1. Vector de datos de historia actual 2. Propiedad(es) y valor(es) 3. Deshacer / Rehacer cambios 4. Etiquetas de objetos seleccionados 	<ol style="list-style-type: none"> 1. Vector de modificación de propiedades (va hacia editor molecular) 2. Mensajes del sistema (va hacia usuario) 3. Propiedades modificadas de objetos indicados (va hacia preparar escena gráfica)
8) Imprimir	
Entradas en el nivel 0	Salidas en el nivel 0
<ol style="list-style-type: none"> 1. Documento a imprimir 2. Configuración de impresora 3. Configuración de documento a imprimir 4. Vector de datos de configuración actual 	<ol style="list-style-type: none"> 1. Impresión de documento (va hacia usuario) 2. Mensajes de impresión (va hacia usuario)

9.1.3.3 Construir diagramas de flujo de datos.

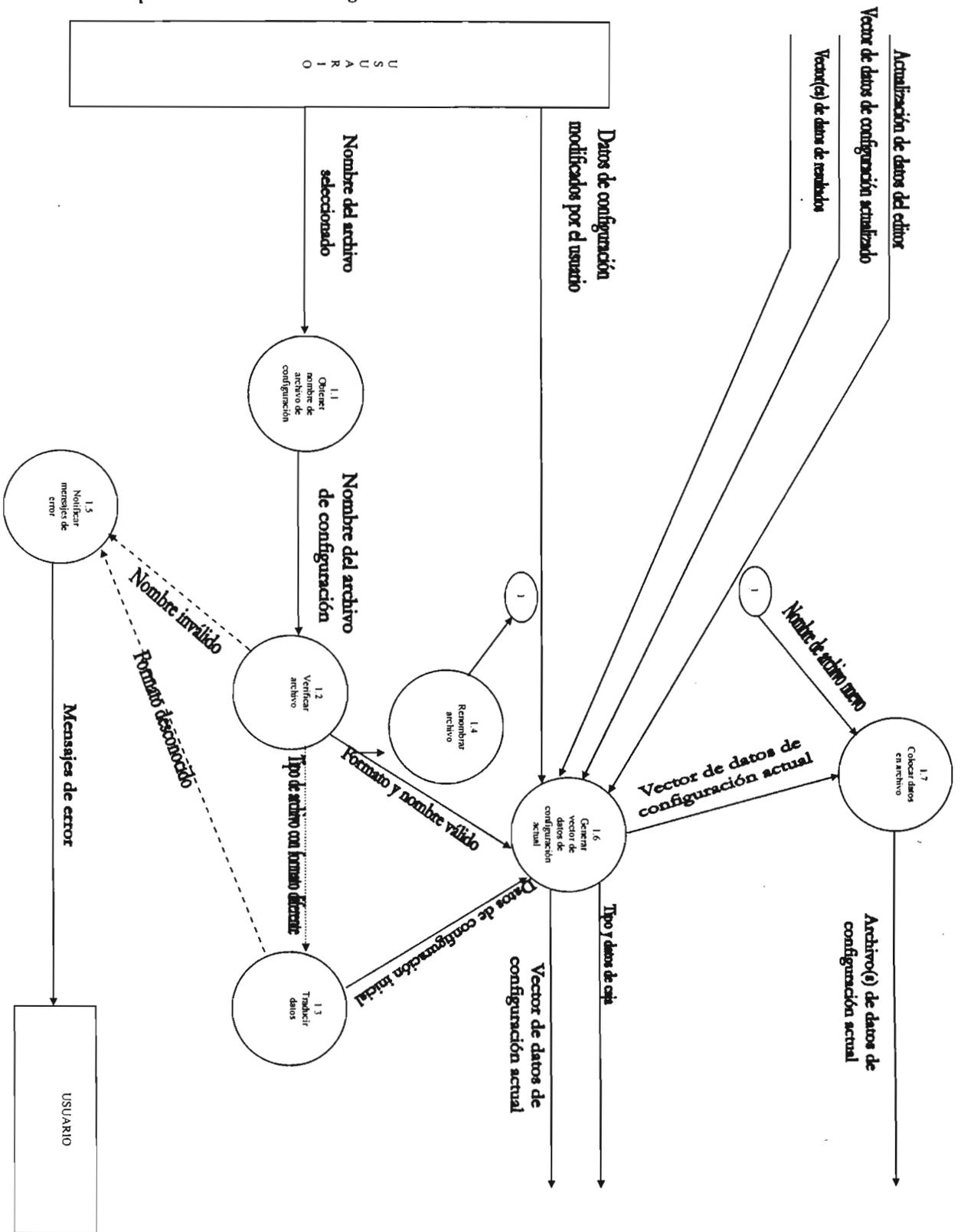


9.1.3.3.1 Diagrama de contexto

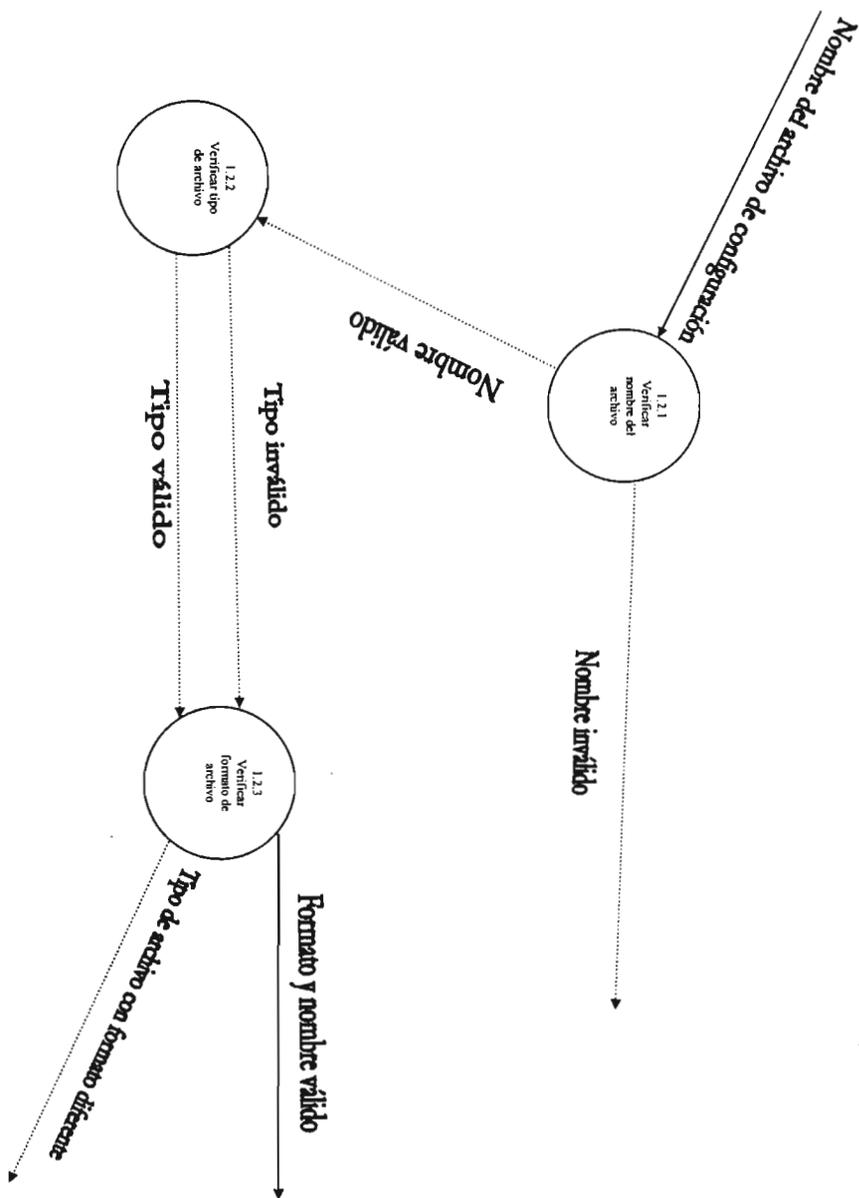




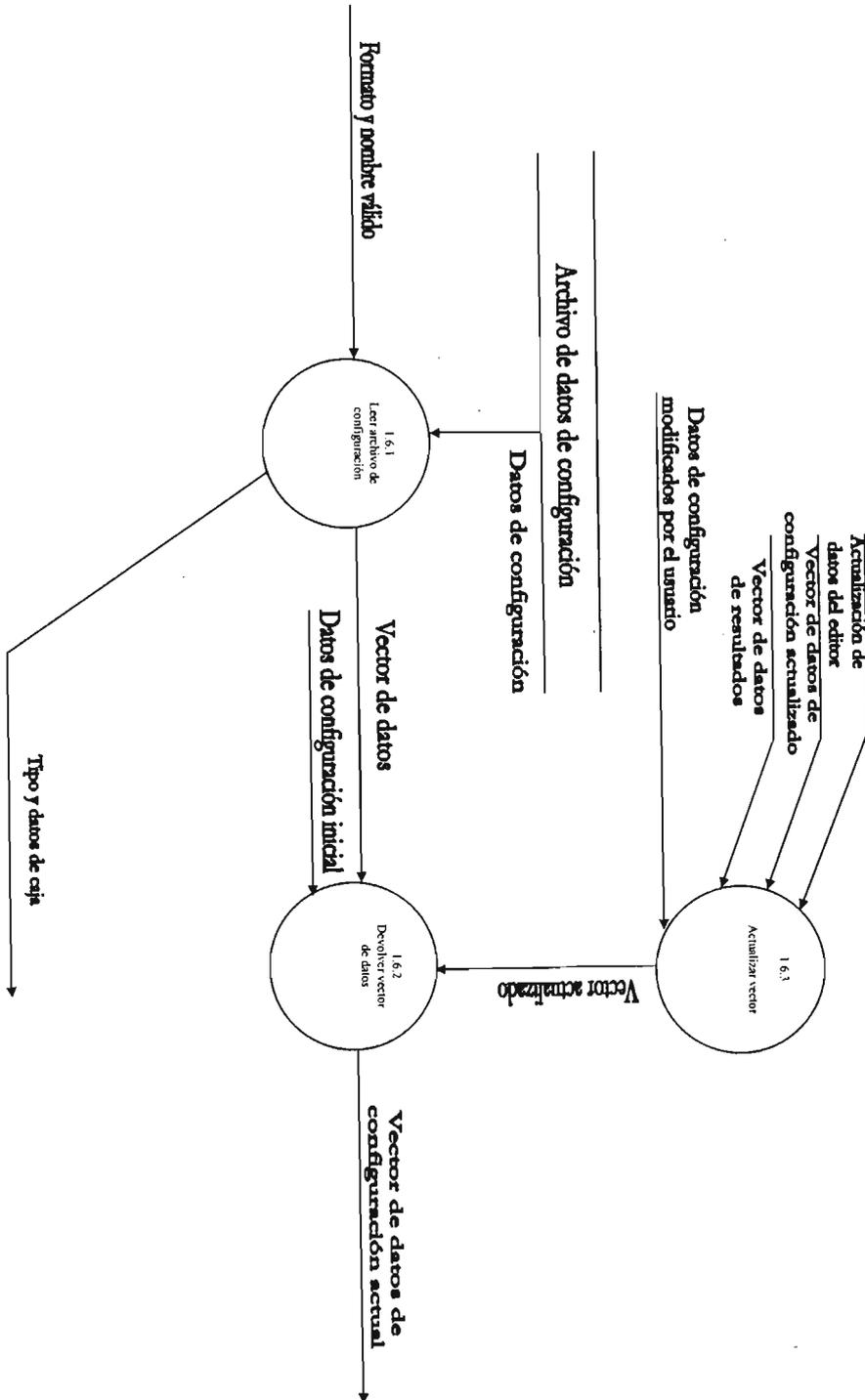
9.1.3.3.3 DFD 1 : Proporcionar datos de configuración



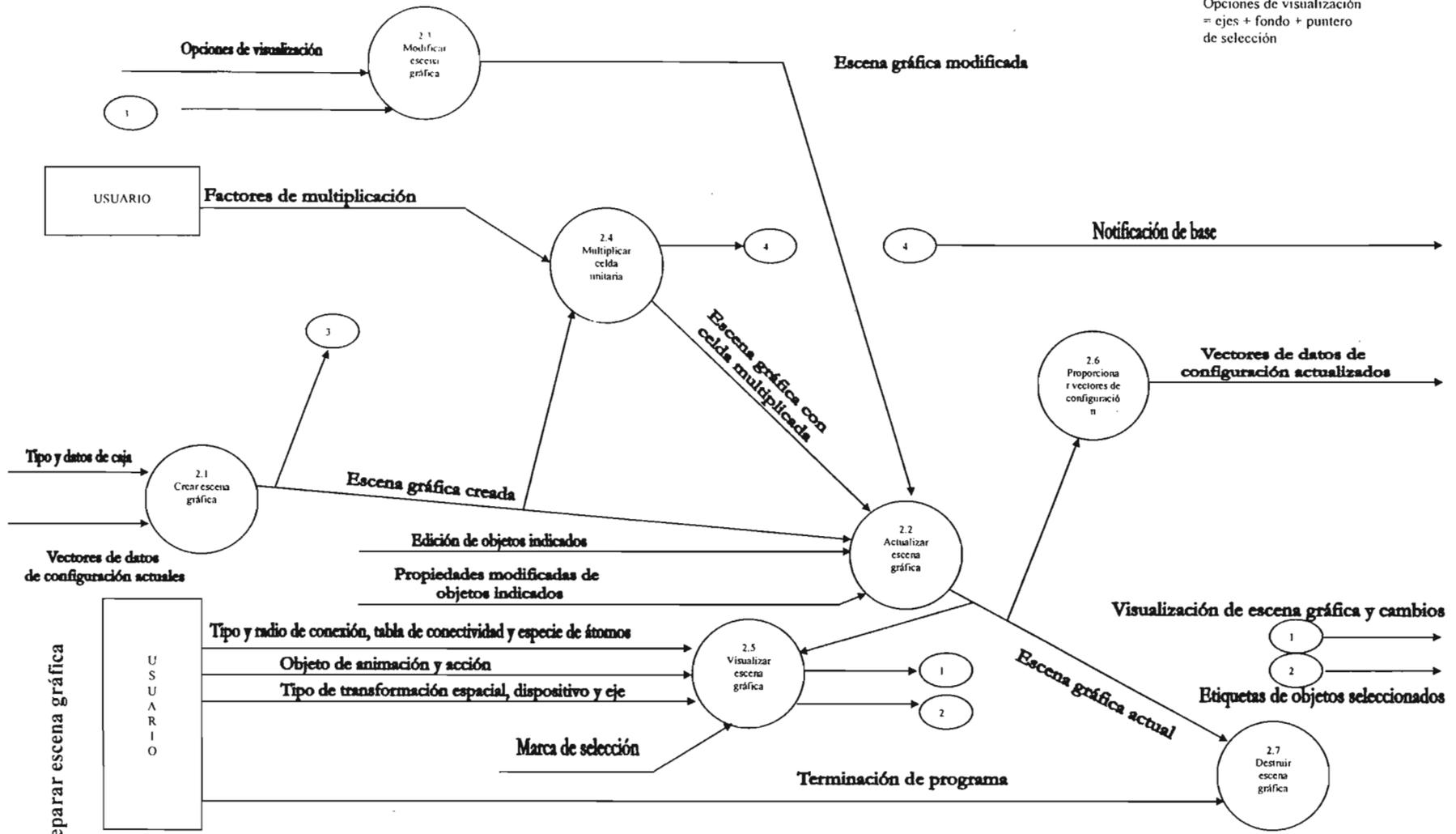
9.1.3.3.1 DFD 1.2 : Verificar archivo



9.1.3.3.3.2 DFD 1.6 : Generar vector de datos configuración actual

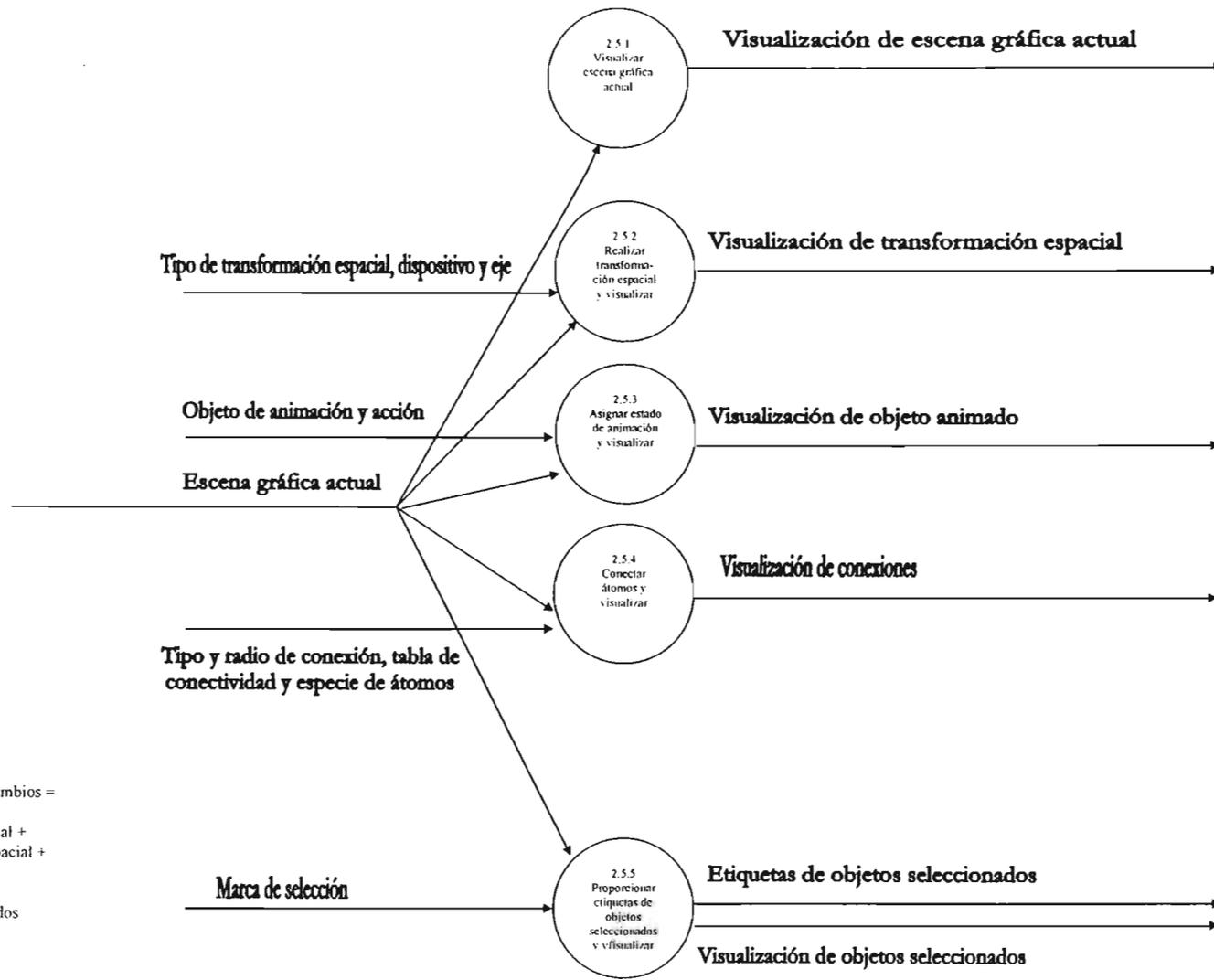


9.1.3.3.4 DFD 2 : Preparar escena gráfica

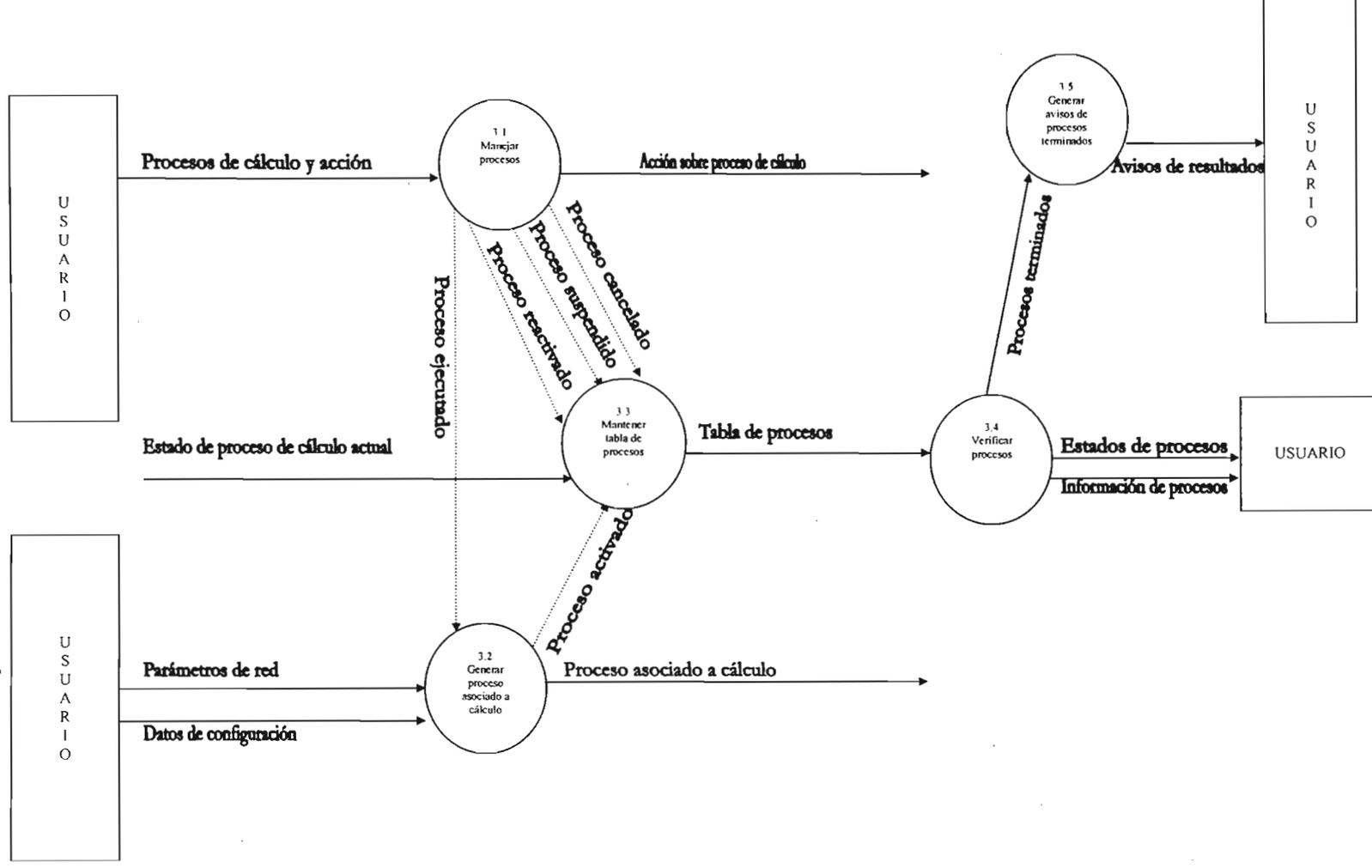


9.1.3.3.4.1 DFD 2.5 : Visualizar escena gráfica

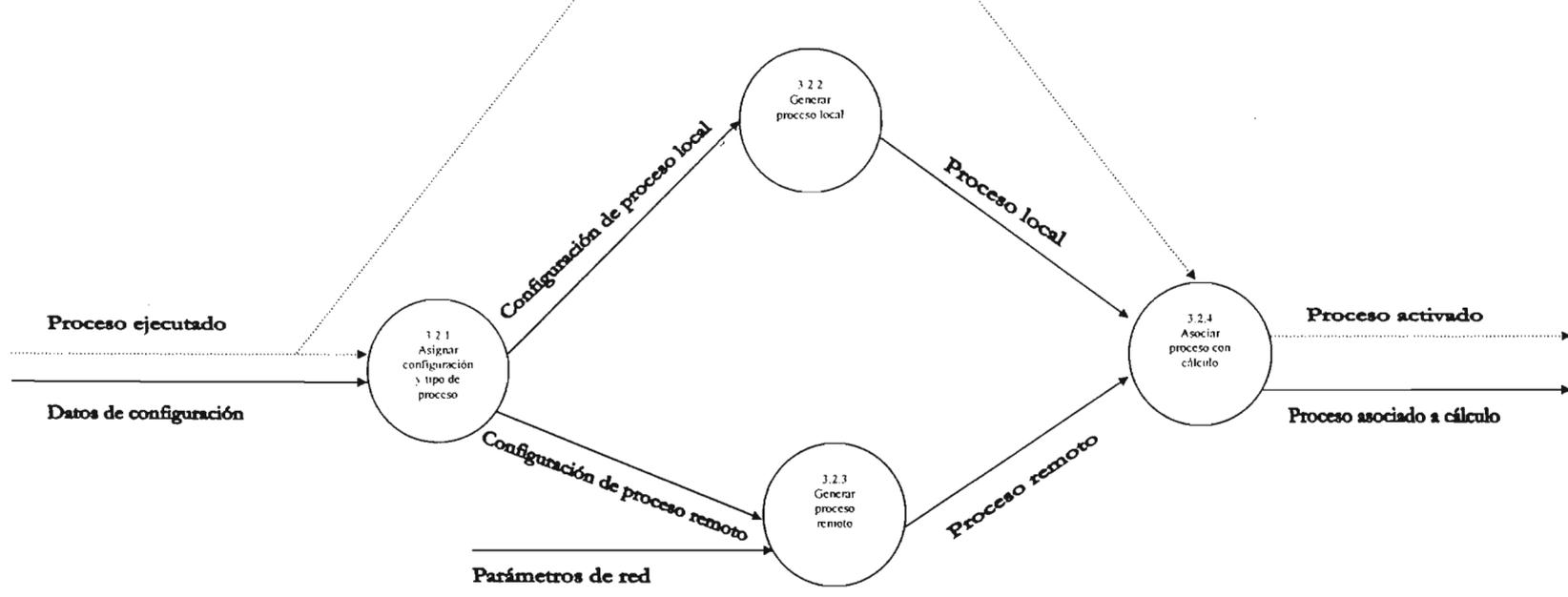
Visualización de escena gráfica y cambios =
Visualización de escena gráfica actual +
Visualización de transformación espacial +
Visualización de objeto animado +
Visualización de conexiones +
Visualización de objetos seleccionados



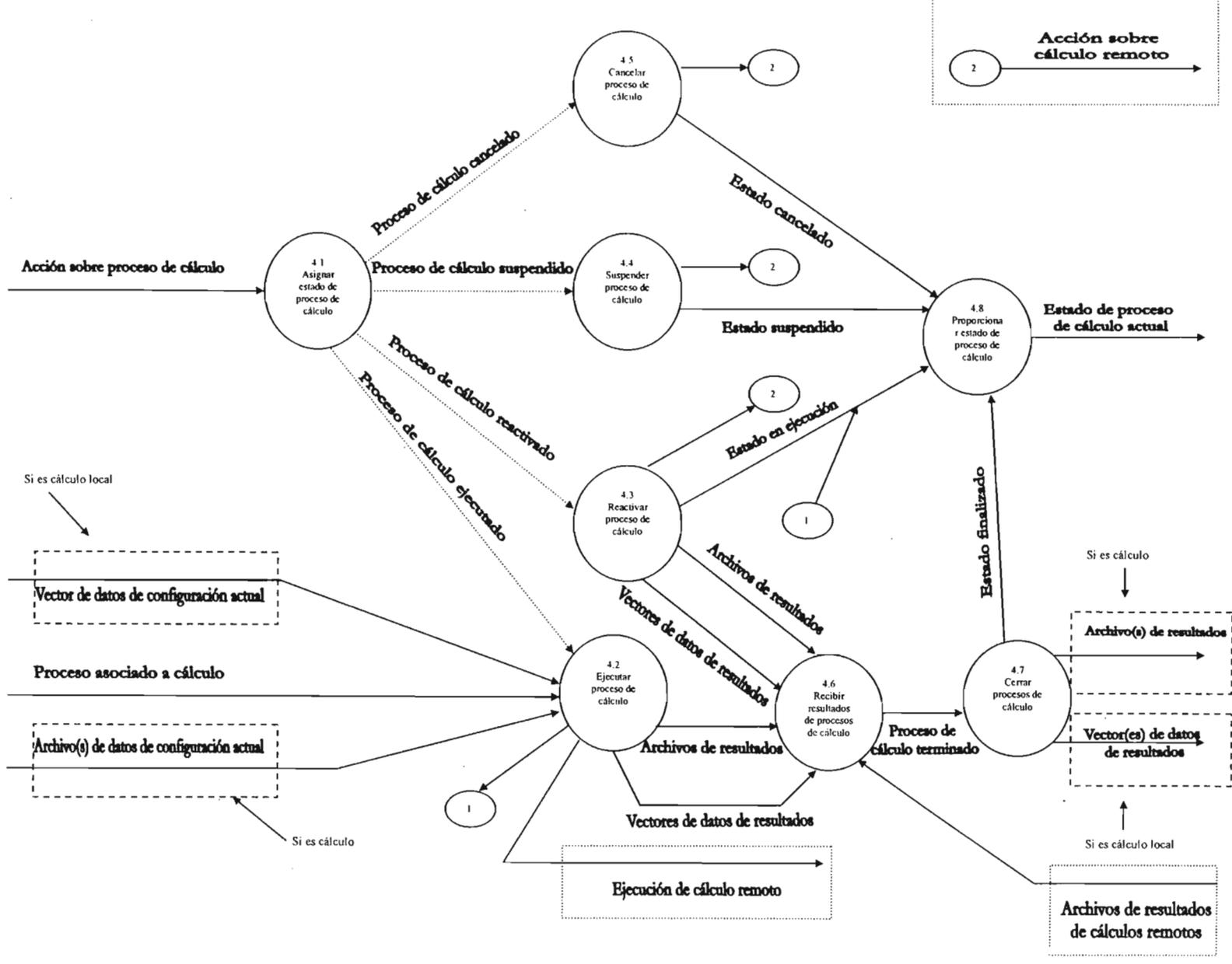
9.1.3.3.5 DFD 3 : Monitorear procesos



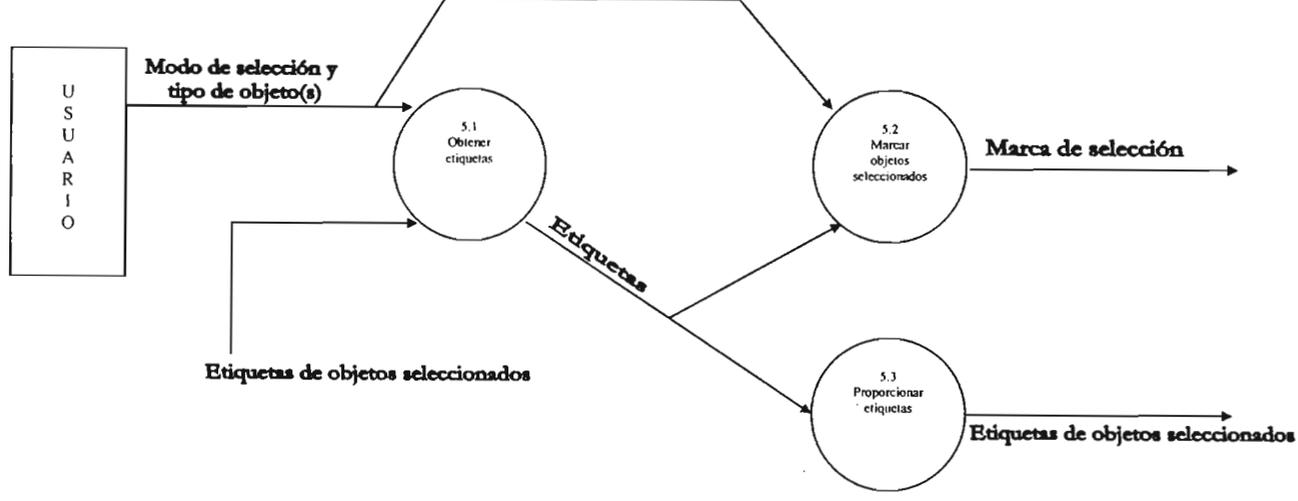
9.1.3.3.5.1 DFD 3.2 : Generar proceso asociado a cálculo



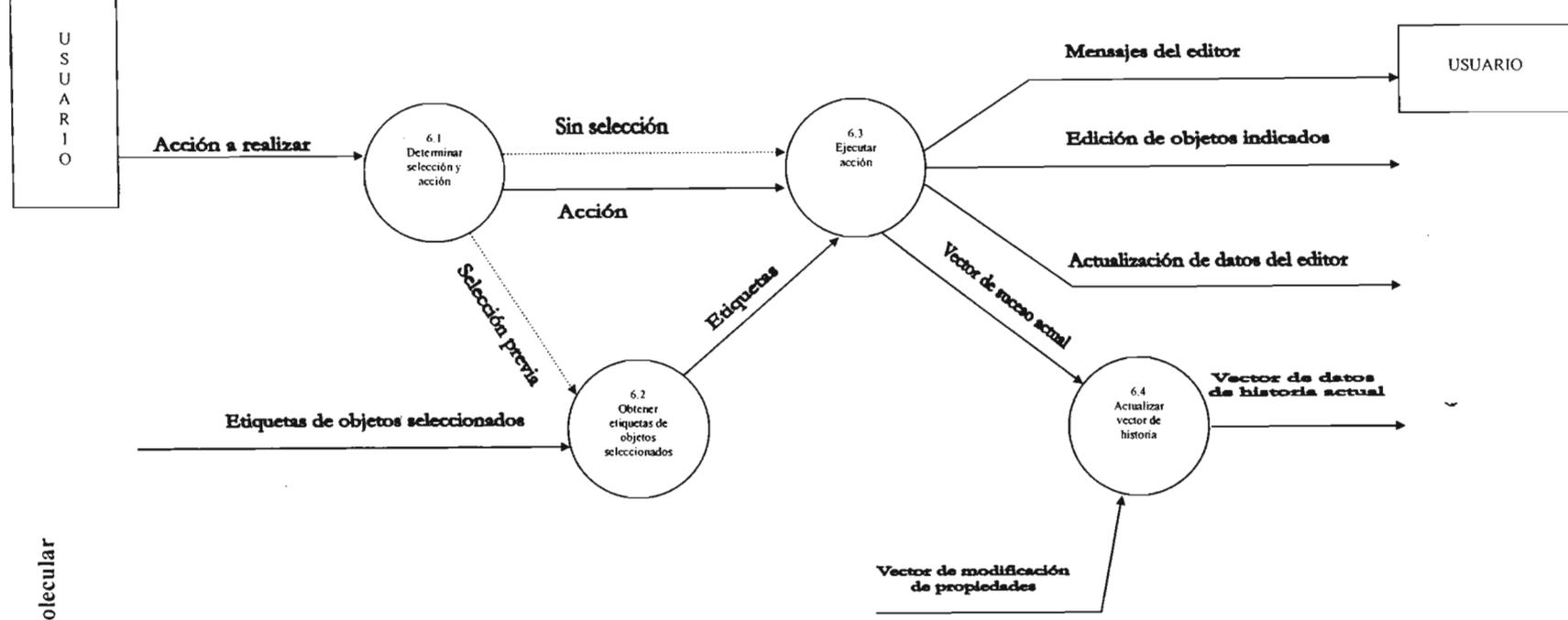
9.1.3.3.6 DFD 4 : Generar procesos de cálculo



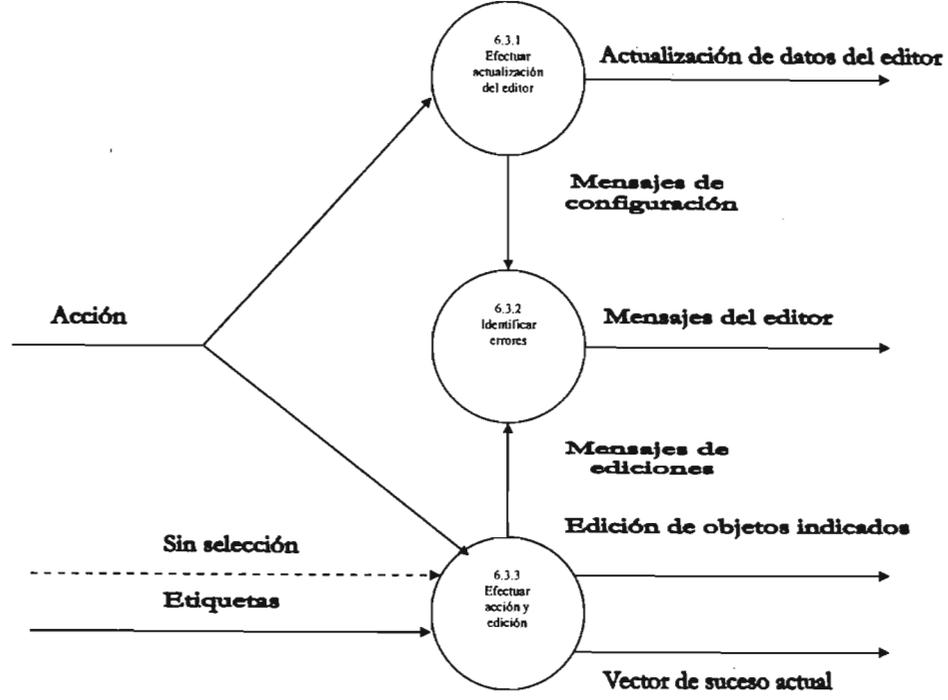
9.1.3.3.7 DFD 5 : Indicar selección



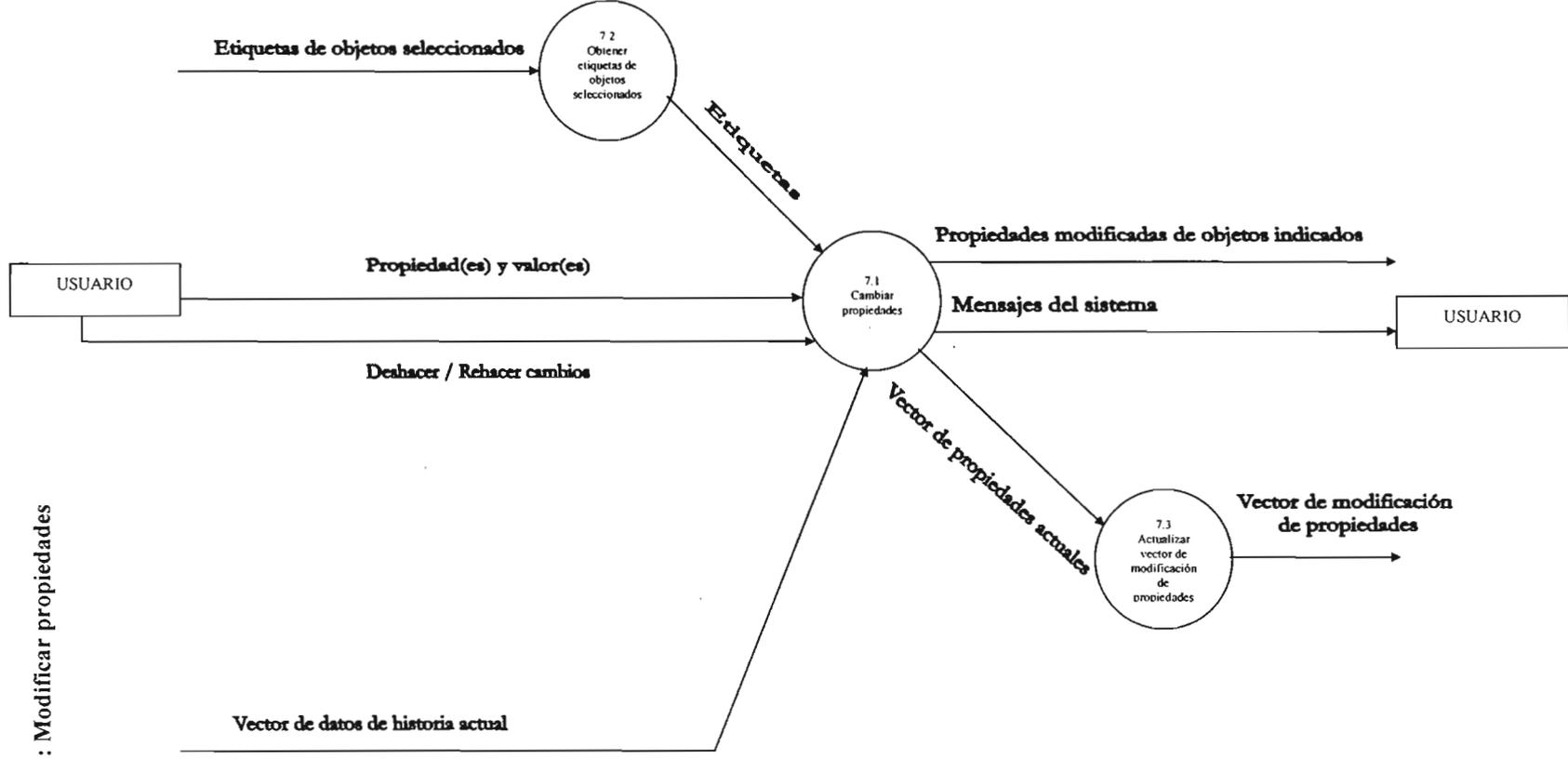
9.1.3.3.8 DFD 6 : Editor molecular



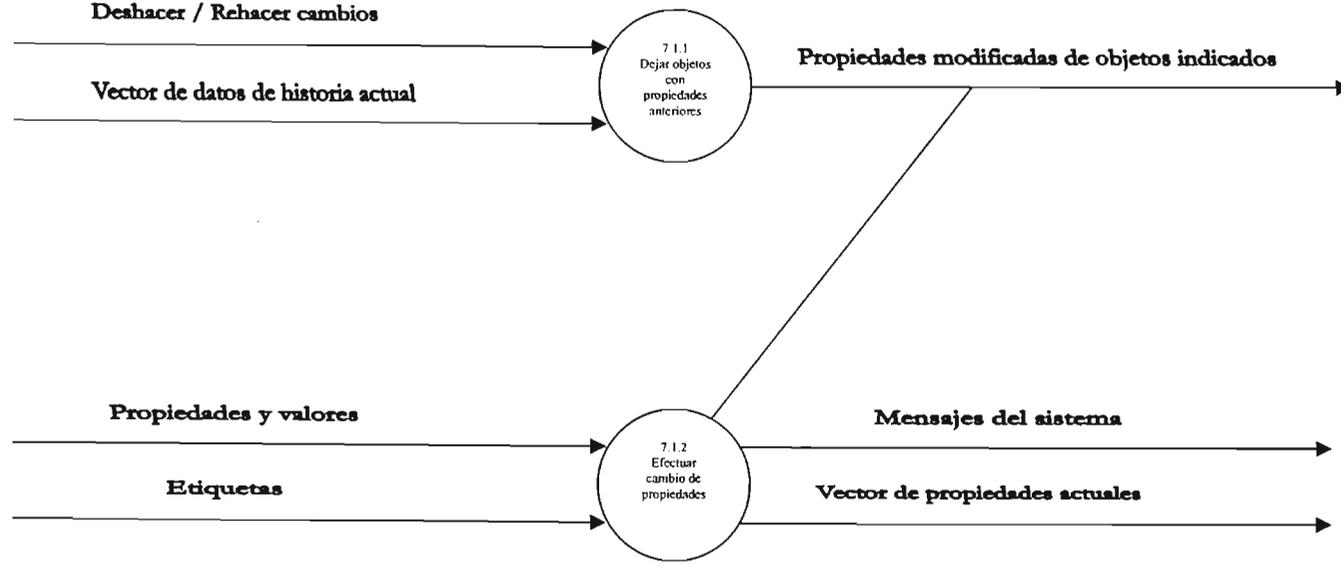
9.1.3.3.8.1 DFD 6.3 : Ejecutar acción



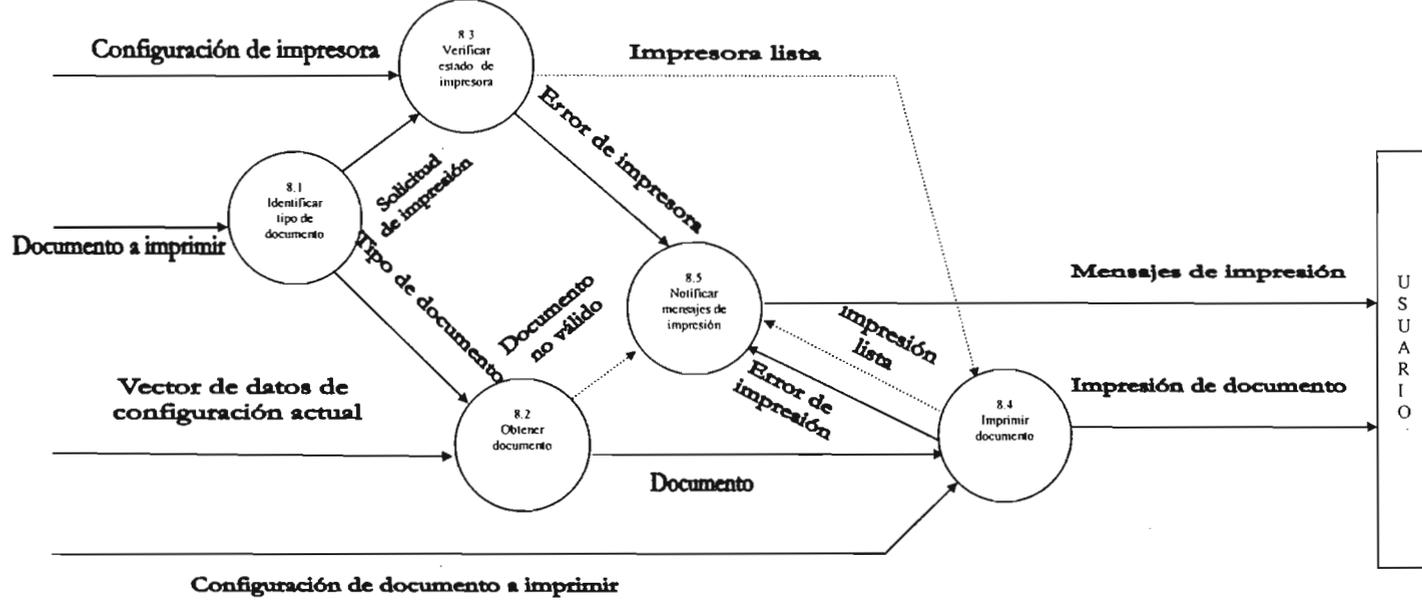
9.1.3.3.9 DFD 7 : Modificar propiedades



9.1.3.3.9.1 DFD 7.1 : Cambiar propiedades



9.1.3.3.10 DFD 8 : Imprimir



9.1.3.4 Describir los procesos y funciones

9.1.3.4.1 DFD 1 Proporcionar datos de configuración

1.1 Función : Obtener nombre de archivo de configuración

- Entrada: Recibe el nombre del archivo de seleccionado.
- Salida: Devuelve el nombre del archivo de configuración.
- Transformación: No transforma datos, ya que solo pide el nombre del archivo al usuario, o puede obtener el nombre de los archivos de datos especificados (hoyos, dipolos, etc) en base al nombre del archivo de partículas que fue especificado por el usuario.

1.2 Verificar archivo

1.2.1 Función: Verificar nombre del archivo.

- Entrada: Recibe el nombre del archivo.
- Salida: Devuelve un valor, entero corto, que indica si el nombre del archivo es un nombre válido o no lo es.
- Transformación: Realiza una evaluación, dentro de la cual chequea si el nombre es válido, verifica que la cadena del nombre tenga caracteres válidos y cumpla el formato del SO.

1.2.2 Función: Verificar tipo del archivo.

- Entrada: Recibe la señal que indica que el nombre del archivo fue válido.
- Salida: Devuelve un valor que indica si el archivo es del tipo adecuado, o si fue de un tipo inválido.
- Transformación: Realiza una verificación del tipo, revisando la extensión del nombre del archivo.

1.2.3 Función: Verificar formato del archivo.

- Entrada: Recibe la señal de que el archivo es de un tipo válido.
- Salida: Devuelve un valor que indica si el archivo es del formato adecuado, o entrega una señal que dice que el archivo es de algún otro formato distinto.
- Transformación: Verifica el formato del archivo de datos, chequeando las primeras líneas para asegurarse de que los datos correspondan al archivo de datos especificado.

1.3 Función : Traducir datos

- Entrada: Recibe la señal de que el archivo de datos es de un tipo diferente (datos leídos de archivos de configuración).
- Salida: Entrega los datos de la configuración inicial en formato de gráfica (SVDM), o un mensaje que indica que el archivo fue de un formato desconocido.
- Transformación: Esta función realiza la traducción de los datos de otros formatos al formato de gráfica (SVDM).

1.4 Función : Renombrar archivo

- Entrada: Recibe una señal de formato válido y el nombre válido.
- Salida: Devuelve el nombre nuevo para el archivo.
- Transformación: Salva el archivo con el nuevo nombre, y cierra el archivo original sin los cambios realizados. Realiza el renombramiento del archivo.

1.5 Función : Notificar mensajes de error

- Entrada: Recibe la notificación de que fue nombre inválido o la notificación de que el formato del archivo es desconocido.
- Salida: Entrega un mensaje al usuario notificando que ocurrió un error.
- Transformación: Únicamente realiza una visualización de mensajes para el usuario.

1.6 Generar vector de datos de configuración actual

1.6.1 Función : Leer archivo de configuración

- Entrada: Recibe el nombre válido del archivo de datos de configuración, y el archivo mismo y los datos de la configuración inicial.
- Salida: Devuelve el vector de datos de configuración y los datos de la caja, como son: tipo, ángulos, etc.
- Transformación: Únicamente recibe el nombre del archivo de datos de configuración, lee y almacena los datos en un vector.

1.6.2 Función : Devolver vector de datos

- Entrada: Recibe los datos de configuración inicial, el vector de datos y el vector actualizado.
- Salida: Devuelve los datos de configuración actual.
- Transformación: Solo es una interfaz que devuelve los datos del vector actual.

1.6.3 Función : Actualizar vector

- Entrada: Recibe los datos actualizados del Editor Molecular, el vector de datos de configuración actualizado, el vector de datos de resultados, y los datos de configuración que fueron modificados por el usuario.
- Salida: Devuelve el vector de datos actualizado.
- Transformación: Esta función realiza una actualización de los datos del vector que esta en memoria.

1.7 Colocar datos en archivo (Guardar vector de datos, Guardar archivo de datos)

- Entrada: Recibe el vector de datos de configuración actual, el nuevo nombre del archivo.
- Salida: Devuelve el archivo de datos de configuración actual.
- Transformación: Realiza la escritura en disco de los datos del archivo o lo entrega al proceso que se lo solicite.

9.1.3.4.2 DFD 2 Preparar escena gráfica

2.1 Función : Crear escena gráfica.

Esta función genera una instancia de la clase escena gráfica con los archivos de configuración disponibles.

- Entrada: Vector de potenciales, número de celdas de la estructura cristalina, vector de hoyos. tipo de caja. datos de caja (si la caja es cúbica : ángulo A , ángulo B, ángulo C, lado a, lado b, lado c), número de moléculas en la celda, vector de partículas por molécula, vector de dipolos por molécula.
 - Salida: Escena gráfica creada.
 - Transformación : La escena gráfica se genera con tres ejes, y una estructura cristalina, (teniendo esta solo una celda al momento de crearla con al menos una molécula ,al crear la celda se genera igualmente la caja con los valores indicados), dependiendo de las opciones indicadas por el usuario, si existe un vector de hoyos se genera el sistema de cavidades de la celda, si no se pudo crear el sistema de cavidades devolver un mensaje de error. si hay un vector de potenciales se crea la superficie de potencial de la estructura cristalina, si no se pudo crear el sistema de potenciales devolver un mensaje de error.
- Si no se pudo crear la caja, o la molécula o la celda, devolver la señal de error correspondiente, en caso contrario señalar que la escena gráfica se creo exitosamente.

2.2 Función : Actualizar escena gráfica

Esta función actualiza el objeto escena grafica de acuerdo a las opciones de visualización indicadas por el usuario, las marcas del editor molecular y del modificador de propiedades, y visualiza la escena gráfica actual.

- Entrada: Escena gráfica creada o escena gráfica modificada o escena gráfica con celda multiplicada, opciones de visualización, edición de objetos (marca), propiedades modificadas de objetos indicados (señal que indica que se efectuaron cambios sobre las propiedades)
- Salida: Escena gráfica actual, visualización de escena gráfica actual
- Transformación: Dibuja la escena gráfica actual con las opciones de visualización de menor calidad para optimización a menos que se indique lo contrario en las opciones de visualización, o con los objetos editados resaltados si la marca edición de objetos es igual a 1. Si hubo algún error en la visualización indicar la señal de error correspondiente en caso contrario señalar que se visualizo exitosamente.

2.3 Función : Modificar escena gráfica

Esta función modifica los ejes, el fondo y el puntero de selección de la escena gráfica.

- Entrada: Opciones de visualización, escena gráfica creada.
- Salida: Escena gráfica modificada.
- Transformación: Cambiar los objetos ejes, fondo y puntero de selección de acuerdo a los parámetros dados por el usuario en las opciones de visualización

2.4 Multiplicar celda unitaria

Esta función genera una estructura cristalina a partir de la celda unitaria y los factores de multiplicación.

- Entrada: Factores de multiplicación, escena grafica creada (o actual).
- Salida: Notificación de base, escena gráfica con celda multiplicada.
- Transformación: Verifica si la celda unitaria es una base,
 - Si lo es :
Multiplica la celda unitaria usando el factor de multiplicación indicado por el usuario.
Genera una notificación de base indicando que si fue una base.
 - Si no lo es :
Genera una notificación de base indicando que no fue una base.

2.5 Visualizar escena gráfica.

2.5.1 Función : Visualizar escena gráfica actual.

- Entrada: Escena gráfica actual.
- Salida: Visualización de escena gráfica actual.
- Transformación: Visualiza los objetos de la escena gráfica con las opciones establecidas por el usuario a través del editor molecular, el modificador de propiedades o el indicador de propiedades o con las opciones establecidas inicialmente.

2.5.2 Función : Realizar transformación espacial y visualizar.
Esta función efectúa la transformación espacial indicada por el usuario sobre el eje seleccionado usando el dispositivo elegido

- Entrada: Escena gráfica actual, tipo de transformación espacial, dispositivo (usado), eje (sobre el que se realizara la transformación espacial).
- Salida: Visualización de transformación espacial.
- Transformación: Aplicar transformación espacial indicada a la escena gráfica, dibujar después la escena gráfica.

2.5.3 Función : Asignar estado de animación y visualizar.
Esta función asigna el estado de animación a un objeto animado de la escena gráfica y lo visualiza.

- Entrada: Escena gráfica actual, objeto animado, acción.
- Salida: Visualización de objeto animado.
- Transformación: Asignar estado de animación al objeto animado de la escena grafica, dibujar la escena grafica en animación, si hubo algún error al efectuar la operación indicarlo, de lo contrario señalar que la operación se realizo exitosamente.

2.5.4 Función : Conectar átomos y visualizar.
Esta función visualiza las conexiones entre los átomos de la(s) molécula(s) de la escena gráfica

- Entrada: Escena gráfica actual, tipo y radio de conexión, tabla de conectividad y especie de átomos.
- Salida: Visualización de conexiones.
- Transformación: Dibujar conexiones de la molécula en la escena gráfica (conectar átomos).

2.5.5 Función : Proporcionar etiquetas de objetos seleccionados y visualizar.
Esta función entrega las etiquetas de los objetos que componen la molécula, el sistema de cavidades o superficie de potencial y visualiza los objetos seleccionados restándolos de los demás si la marca e selección es igual a 1.

- Entrada: Marca de selección, escena gráfica actual.
- Salida: Etiquetas de objetos seleccionados, visualización de objetos seleccionados.
- Transformación: Buscar los objetos seleccionados por la marca de selección, entregar etiquetas y visualizar los objetos seleccionados resaltándolos.

2.6 Función : Proporcionar vectores de configuración.
Es una función que proporciona los datos de configuración de los objetos que componen la escena gráfica.

- Entrada: Escena gráfica actual.
- Salida: Vectores de datos de configuración actuales.
- Transformación: Obtiene los vectores de datos solicitados de los objetos en la escena gráfica del tipo (molécula, sistema de cavidades y potenciales).

2.7 Proceso : Destruir escena gráfica.
Este proceso llama al destructor de la clase para liberar la memoria ocupada por este objeto.

- Entrada: Escena grafica actual, terminación del programa.
- Salida: Ninguna
- Transformación: Llamar al destructor de escena gráfica.

9.1.3.4.3 DFD 3 Monitorear procesos.

3.1 Función : Manejar Procesos

- Entrada: Recibe los procesos de cálculo y la acción a realizar.
- Salida: Entrega ya sea el proceso cancelado, el proceso suspendido, el proceso reactivado o el proceso ejecutado, y además entrega la acción sobre el proceso de cálculo.
- Transformación: se realiza el manejo de los procesos y se entregan éstos a los objetos solicitantes, además de entregar sus estados actuales cuando se requiera.

3.2 Generar proceso asociado a cálculo

3.2.1 Función : Asignar configuración y tipo de proceso.

- Entrada: Recibe el proceso ejecutado y los datos de configuración.
- Salida: Entrega la configuración del proceso local o la configuración del proceso remoto.
- Transformación: Asigna la configuración y el tipo de proceso solicitado para que sea posible generar el proceso.

3.2.2 Función : Generar proceso local.

- Entrada: Recibe la configuración de proceso local, necesaria para generar el proceso local.
- Salida: Entrega el proceso local generado.
- Transformación: Realiza la creación del nuevo proceso local.

3.2.3 Función : Generar proceso remoto.

- Entrada: Recibe la configuración del proceso remoto y los parámetros de red.
- Salida: Entrega el proceso remoto generado.
- Transformación: Realiza la creación del nuevo proceso remoto.

3.2.4 Función : Asociar proceso con cálculo.

- Entrada: Recibe el proceso local, y el proceso remoto.
- Salida: Entrega el proceso activado, y el proceso asociado a cálculo.
- Transformación: Recibe el proceso, ya sea local o remoto.

3.3 Función : Mantener tabla de procesos

- Entrada: Recibe la señal de proceso cancelado, proceso suspendido, o proceso reactivado, el estado de procesos de cálculos actuales, y el proceso actualizado.
- Salida: Entrega la tabla de procesos actualizada.
- Transformación: Realiza el mantenimiento y la actualización de la tabla de procesos.

3.4 Verificar procesos

- Entrada: Recibe la tabla de procesos.
- Salida: Entrega los identificadores de los procesos terminados, los estados de procesos, y la información de los procesos.
- Transformación: Realiza un chequeo de los procesos actuales.

3.5 Función : Generar avisos de procesos terminados

- Entrada: Recibe los identificadores de los procesos terminados.
- Salida: Entrega los avisos de resultados.
- Transformación: solo genera los avisos de los procesos que han finalizado.

9.1.3.4.4 DFD 4 Generar procesos de cálculos.

4.1 Función : Asignar estado de proceso de cálculo

- Entrada: Recibe la acción a realizar sobre el proceso de cálculo.
- Salida: Entrega una señal que indica si el proceso fue:
 - cancelado,
 - suspendido,
 - reactivado,
 - ejecutado.
- Transformación: Solamente realiza una asignación de estado al proceso de cálculo especificado.

4.2 Función : Ejecutar proceso de cálculo.

- Entrada: Recibe la señal que indica si el proceso de cálculo fue ejecutado, así como los vectores de datos de configuración actual, el proceso asociado al cálculo y los archivos de datos de configuración actual que sean requeridos por el proceso.
- Salida: Entrega la señal que indica que el proceso se está ejecutando actualmente, proporciona los archivos y vectores de datos de resultados, además de entregar los datos necesarios para la ejecución del cálculo remoto.
- Transformación: Envía a ejecutar el proceso especificado.

4.3 Función : Reactivar proceso de cálculo.

- Entrada: Recibe la señal que indica que el proceso será reactivado.
- Salida: Entrega el estado del proceso, el cual indica que se encuentra en ejecución, así como los archivos y los vectores de datos de resultados, y la acción sobre cálculo remoto.
- Transformación: Cambia el estado del proceso reactivando su ejecución.

4.4 Función : Suspender proceso de cálculo.

- Entrada: Recibe la señal que indica que el proceso deberá ser suspendido y el identificador del proceso.
- Salida: Entrega una señal que indica que el estado del proceso es suspendido y la acción a realizar sobre el cálculo remoto.
- Transformación: Ejecuta una suspensión sobre el proceso actual.

4.5 Función : Cancelar proceso de cálculo

- Entrada: Recibe la señal que indica la cancelación del proceso y el identificador del proceso.
- Salida: Entrega el estado del proceso que indica que éste ha sido cancelado y la acción a realizar sobre el proceso remoto.
- Transformación: Cancela el proceso indicado.

4.6 Función : Recibir resultados de procesos de cálculo

- Entrada: Recibe los archivos y los vectores de datos de resultados, así como también los datos de resultados generados por cálculos remotos.
- Salida: Entrega el proceso de cálculo determinado.
- Transformación: Recibe los resultados y entrega el nombre proceso de cálculo.

4.7 Función : Cerrar proceso de cálculo

- Entrada: Recibe el proceso de cálculo terminado, es decir, el proceso que será cerrado.
- Salida: Entrega alguno de los siguientes:

- Archivo de resultados, o
- Vectores de datos de resultados.
- Transformación: Solo realiza el cierre del proceso que le fue indicado.

4.8 Función : Proporcionar estado de proceso de cálculo

- Entrada: Recibe la indicación del estado del proceso: cancelado, suspendido, y en ejecución.
- Salida: Entrega el estado de proceso de cálculo actual.
- Transformación: Solo proporciona el estado del proceso.

9.1.3.4.5 DFD 5 Indicar selección.

5.1 Obtener etiquetas

- Entrada: Recibe el modo de selección y tipo de objetos, además de la indicación de selección y las etiquetas de los objetos seleccionados.
- Salida: Entrega las etiquetas.
- Transformación: Únicamente proporciona las etiquetas de los objetos.

5.2 Marcar objetos seleccionados

- Entrada: Recibe el modo de selección, así como el tipo de objetos a seleccionar.
- Salida: Entrega el valor de la marca de selección.
- Transformación: Entrega el indicador de que se ha seleccionado el objeto o no se ha seleccionado.

5.3 Proporcionar etiquetas

- Entrada: Recibe las etiquetas.
- Salida: Entrega las etiquetas de los objetos seleccionados.
- Transformación: Entrega las etiquetas de los objetos que fueron seleccionados.

9.1.3.4.6 DFD 6 Editar molécula

6.1 Determinar selección y acción

- Entrada: Recibe el identificador que indica la acción a realizar.
- Salida: Entrega la acción y uno de los siguientes indicadores:
 - Selección previa
 - Sin selección.
- Transformación: Identifica si la acción va a ser realizada cuando hubo una selección previa o no, y se envía la acción siempre.

6.2 Obtener etiquetas de objetos seleccionados

- Entrada: Recibe la señal de que hubo una selección previa y las etiquetas de los objetos seleccionados.
- Salida: Entrega las etiquetas de los objetos seleccionados.
- Transformación: Obtiene las etiquetas de los objetos seleccionados para que se pueda efectuar la edición sobre ellos.

6.3 Ejecutar acción

6.3.1 Efectuar actualización del editor

- Entrada: Recibe acción a realizar.
- Salida: Entrega la actualización de datos del editor, y los mensajes de configuración.
- Transformación: Actualiza el editor con los datos de configuración.

6.3.2 Identificar errores

- Entrada: Recibe los mensajes de configuración y los mensajes de ediciones.
- Salida: Entrega los mensajes provenientes del editor.
- Transformación: Hace la distinción entre los mensajes recibidos de configuración y edición, e identifica los errores para reportarlos.

6.3.3 Efectuar acción y edición

- Entrada: recibe la acción a realizar, la señal de que no se realizó una selección previa y las etiquetas de una selección realizada por un patrón determinado.
- Salida: entrega la edición de objetos indicados, y el vector de suceso actual.
- Transformación: realiza la acción y modifica la edición.

6.4 Actualizar vector de historia

- Entrada: recibe el vector del suceso actual y el vector de modificación de propiedades.
- Salida: entrega el vector de datos de historia actual.
- Transformación: actualiza el vector de historia con los datos de la última acción.

9.1.3.4.7 DFD 7 Modificar propiedades

7.1 * Cambiar propiedades

7.1.1 Dejar objetos con propiedades anteriores

- Entrada: Recibe la señal de deshacer o rehacer cambios, y el vector de historia actual.
- Salida: Entrega una indicación a la escena gráfica para que actualice la visualización de los objetos que cambiaron sus propiedades.
- Transformación: modifica el vector de historia y lo deja con los datos anteriores.

7.1.2 Efectuar cambio de propiedades

- Entrada: Recibe las propiedades y valores, así como las etiquetas.
- Salida: Entrega los mensajes del sistema, el vector de propiedades actual, así como una indicación de propiedades modificadas para la actualización de la visualización.
- Transformación: modifica las propiedades especificadas.

7.2 Obtener etiquetas de objetos seleccionados

- Entrada: recibe las etiquetas de los objetos seleccionados.
- Salida: devuelve la indicación de selección, y las etiquetas de los objetos.
- Transformación: obtiene las etiquetas de los objetos seleccionados y las entrega a cambiar propiedades.

7.3 Actualizar vector de modificación de propiedades

- Entrada: recibe vector de propiedades actuales.
- Salida: entrega vector de modificación de propiedades.
- Transformación: cambia el contenido del vector de modificación de propiedades por los nuevos datos.

9.1.3.4.8 DFD 8 Imprimir

8.1 Identificar tipo de documento

- Entrada: Recibe el documento a imprimir.
- Salida: Entrega la solicitud de impresión, e informa de que tipo de documento se trata.
- Transformación: Identifica el tipo de documento que se va a imprimir.

8.2 Obtener documento

- Entrada: recibe el tipo de documento y el vector de datos de configuración actual.
- Salida: entrega la señal que informa que el documento no fue válido, o el documento mismo.
- Transformación: obtiene el documento y lo entrega para imprimir.

8.3 Verificar estado de impresora

- Entrada: recibe la configuración de la impresora, y la indicación de solicitud de impresión.
- Salida: entrega la señal de impresora lista o el error de impresión.
- Transformación: chequea el estado de la impresora.

8.4 Imprimir documento

- Entrada: Recibe la configuración del documento a imprimir, la señal de que la impresora está lista y el documento.
- Salida: Entrega la impresión del documento, la señal de impresión lista o el error de impresión.
- Transformación: Realiza la impresión del documento.

8.5 Notificar Mensajes de Impresión

- Entrada: Recibe los mensajes de error de la impresora, la señal de documento no válido, la señal de que la impresión está lista o el mensaje de error de impresión.
- Salida: Entrega los mensajes de impresión, dependiendo de las señales recibidas por los otros procesos de impresión.
- Transformación: Únicamente canaliza los mensajes de impresión al usuario.

9.1.3.5 Identificar las restricciones

9.1.3.5.1 DFD 1 Proporcionar datos de configuración

- 1.1 Restricciones: Se puede obtener el nombre explícitamente solicitándolo al usuario o el sistema puede obtenerlo automáticamente, basándose en el nombre de un archivo de partículas seleccionado previamente por el usuario, en este caso el sistema buscará las referencias de los archivos restantes correspondientes a las demás opciones del sistema con respecto al archivo seleccionado por el usuario (hoyos, potenciales, etc). El archivo generalmente será seleccionado a través de la interfaz.
- 1.2.1 Restricciones: El nombre no debe tener espacios en blanco, y debe estar formado por los caracteres válidos por el SO, en este caso UNIX.
- 1.2.2 Restricciones: El nombre del archivo deberá tener las siguientes extensiones:
.dat: archivo de partículas.
.hoyos: archivo de hoyos.
.dipolos: archivo de dipolos.
.pot: archivo de potenciales.
.rayos_x: archivo de rayos X.
.conexiones: archivo de conexiones.
- 1.2.3 Restricciones: Las restricciones de los formatos se dan de acuerdo a la definición de los formatos de los archivos, si la función devuelve la señal de que el archivo es de un formato diferente, entonces éste será validado por la operación de traducir datos.
- 1.4 Restricciones: El nuevo nombre del archivo debe ser un nombre válido y debe cumplir con las características de validación de caracteres en base al SO. La función recibe el nuevo nombre válido del archivo, guarda el archivo anterior sin guardar los cambios y entrega el nuevo nombre de archivo, para que posteriormente se le agreguen los nuevos datos.
- 1.5 Restricciones: No hay restricciones, ya que solo recibe un estado de error y entrega un mensaje.
- 1.6.1 Restricciones: El nombre del archivo debe ser válido según las restricciones del SO y el formato del archivo debe ser correcto.
- 1.6.2 Restricciones: Esta función devuelve los datos, solo cuando algo cambia en la escena gráfica y el usuario decide guardar o la información es requerida por algún otro proceso (preparar la escena gráfica).
- 1.6.3 Restricciones: El vector de datos cambia si se modifica algo en la escena gráfica o se solicitan datos al editor molecular.
- 1.7 Restricciones: Solo se realiza la escritura a disco cuando el usuario guarda el archivo, en otros caso se entrega a los procesos correspondientes.

9.1.3.5.2 DFD 2 Preparar escena gráfica

- 2.1 Restricciones : El vector de potenciales es opcional, el número de celdas de la estructura cristalina inicialmente es igual a uno, el vector de hoyos es opcional, el tipo de caja a usar por el momento es cúbica, los valores para los ángulos y lados de la caja deben ser positivos, para crear una escena gráfica la celda debe tener al menos una molécula (es necesario tener al menos un vector de partículas), el vector de dipolos es opcional, la perspectiva esta activada inicialmente.
- 2.2 Restricciones : Ninguna
- 2.3 Restricciones : Ninguna
- 2.4 Restricciones : Que los factores de multiplicación generen una base para poder multiplicar la celda unitaria.
- 2.5.1 Restricciones : La visualización se hace siempre usando los datos de configuración actuales que tiene cada objeto grafico.
- 2.5.2 Restricciones : Se debe poder reestablecer el origen de la escena gráfica.
- 2.5.3 Restricciones : El objeto escena gráfica que será animado debe tener mas de un cuadro de animación.
- 2.5.4 Restricciones : Las conexiones se dibujaran haciendo uso de la tabla de conectividad dentro del radio de conexión indicado, si la conexión no proviene del editor molecular.
- 2.5.5 Restricciones : Ninguna.
- 2.6 Restricciones : Ninguna.
- 2.7 Restricciones : los datos de configuración deben haber sido guardados previamente por el usuario o debe haberse confirmado la acción sin guardar los datos.

9.1.3.5.3 DFD 3 Monitorear procesos.

- 3.1 Restricciones: Solamente proporciona los procesos y sus estados actuales para el resto de las funciones del monitor que lo requieren, y con esto hacer posible actualizar las tablas de procesos y avisar si ya finalizaron los procesos que se estén ejecutando.
- 3.2.1 Restricciones: Recibe la configuración requerida para el proceso local a generar y lo crea.
- 3.2.2 Restricciones: Los datos de configuración son requeridos para la generación del nuevo proceso.
- 3.2.3 Restricciones: Recibe la configuración requerida para el proceso remoto a generar y lo crea.
- 3.2.4 Restricciones: Solo recibe uno de los procesos:
 - Proceso local.
 - Proceso remoto.Pero solo uno a la vez.
- 3.3 Restricciones: Recibe una señal que indica únicamente alguno de los tres estados:
 - Proceso cancelado.
 - Proceso suspendido.
 - Proceso reactivado.Para actualizar el estado de dicho proceso, en la tabla de procesos.
- 3.4 Restricciones: Recibe la tabla de procesos y dependiendo del estado de los mismo, entrega al usuario un mensaje que indica:
 - Que el proceso termino, y ésta información va a la función de entrega de mensajes de procesos terminados.
 - O el estado actual de los procesos y la información de éstos.
- 3.5 Restricciones: Solo recibe los identificadores de cada proceso que termina, y genera el mensaje de aviso al usuario de que los resultados (archivos) han llegado

9.1.3.5.4 DFD 4 Generar procesos de cálculos.

- 4.1 Restricciones: la función devuelve un valor indicando únicamente solo uno de los cuatro estados anteriores (cancelado, suspendido, reactivado o en ejecución)
- 4.2 Restricciones: Se recibe la señal que indica si el proceso fue ejecutado, y se recibe lo siguiente según sea el caso :
 - Vectores de datos de configuración actual, solo en caso de ser un cálculo local
 - Archivo de datos de configuración actual, solo en caso de ser un cálculo remoto, en este caso, entrega los datos necesarios para la ejecución del mismo.
- 4.3 Restricciones: solo se reactivan los procesos previamente cancelados y se entregan los datos de resultados.
- 4.4 Restricciones: solo se suspenderán los procesos pero no se cancelaran.
- 4.5 Restricciones: se cancelan únicamente los procesos indicados.
- 4.6 Restricciones: recibe los archivos y vectores de resultados que vienen ya sea de:
 - Reactivar proceso de cálculo.
 - O de ejecutar proceso de cálculo.pero no de ambos, y entrega solo los procesos de cálculo que han finalizado.
- 4.7 Restricciones: Entrega lo siguiente según sea el caso:
 - Los archivos de resultados, solo en caso de tratarse de un cálculo remoto.
 - Los vectores de datos de resultados, solo en caso de tratarse de cálculos locales.
- 4.8 Restricciones: Recibe únicamente uno de los siguientes estados de proceso:
 - Cancelado.
 - Suspendido.
 - En ejecución.
 - FinalizadoY entrega el estado del proceso actual.

9.1.3.5.5 DFD 5 Indicar selección.

- 5.1 Restricciones: Solo entrega las etiquetas de los objetos que han sido seleccionados.
- 5.2 Restricciones: Solo indica que objetos están seleccionados y cuales no lo están.
- 5.3 Restricciones: Entrega únicamente las etiquetas de los objetos que fueron previamente seleccionados.

9.1.3.5.6 DFD 6 Editar molécula

- 6.1 Restricciones: Se entrega una señal ya sea:
 - Selección previa
 - Sin selección.
 En cambio, la acción siempre se entrega como salida.
- 6.2 Restricciones: Entrega solo las etiquetas de los objetos a los cuales se les va aplicar la acción y la señal de que se realizó una selección.
- 6.3.1 Restricciones: Recibe los datos de configuración únicamente si éstos son requeridos por alguna acción a realizar en el editor.
- 6.3.2 Restricciones: Los mensajes son generalmente de errores o reportan acciones faltantes.
- 6.3.3 Restricciones: Recibe la acción y la señal que indica que NO hubo selección, además de las etiquetas de los objetos que pueden ser de todos o únicamente los de una sola especie.
- 6.4 Restricciones: Se actualiza el vector de historia después de realizar cada acción.

9.1.3.5.7 DFD 7 Modificar propiedades

- 7.1.1 Restricciones:
 - Si recibe la señal de deshacer cambios, entrega las propiedades anteriores, y coloca la acción en un vector temporal.
 - Si recibe la señal de rehacer cambios, entrega las propiedades y acciones del vector temporal, para colocarlos en la primer localidad del vector de historia.
- 7.1.2 Restricciones: Cambia las propiedades y entrega los mensajes respectivos del sistema.
- 7.2 Restricciones: Al recibir las etiquetas de los objetos seleccionados, las entrega al proceso que cambia propiedades, e indica que hubo selección
- 7.3 Restricciones: Únicamente actualiza los datos del vector de modificación de propiedades.

9.1.3.5.8 DFD 8 Imprimir

- 8.1 Restricciones: Solo realiza una verificación para identificar el tipo de documento.
- 8.2 Restricciones: Obtiene el documento a partir de los datos formateados, el vector de datos de configuración actual, y el tipo de documento, y si no fue un documento válido, únicamente lo informa para los mensajes de error, en caso contrario entrega el documento.
- 8.3 Restricciones: Verifica si la impresora esta lista y lo informa para proceder a imprimir el documento, si por el contrario hay algún error, lo entrega para la notificación de mensajes.
- 8.4 Restricciones:
 - Recibe los datos necesarios para realizar la impresión, la lleva a cabo y notifica lo sucedido:
 - si la impresión se realizó
 - o si por el contrario hubo algún error.
 Estas señales las envía para que se notifique el mensaje adecuado.
- 8.5 Restricciones:
 - Puede recibir alguna de las siguientes señales:
 - error de impresora,
 - documento no válido,
 - error de impresión.
 - impresión lista.

9.1.3.6 Actualizar el diccionario de datos (diagramas de flujo)

Diccionario de datos de los diagramas de flujo

1. Acción :

Es la operación que se va a efectuar sobre los objetos de la *molécula* y puede ser alguna de las siguientes : Copiar, agrupar, cortar, unir átomos, unir moléculas, pegar, agregar átomo, agregar molécula, deshacer/ rehacer cambios.

2. Acción a realizar :

Es la indicación que proporciona el usuario para efectuar la acción de edición sobre los objetos de una o más *moléculas*.

3. Acción sobre cálculo remoto :

Es la indicación que envía un *proceso de cálculo* a un *cálculo remoto* y puede ser alguna de las siguientes: Cancelar, suspender, reactivar.

4. Acción sobre proceso de cálculo :

Es la indicación que el *monitor de procesos* envía a un *proceso de cálculo* ya sea remoto o local, puede ser alguna de las siguientes: Cancelar, suspender, reactivar o ejecutar.

5. Actualización de datos del editor :

Indica al *contenedor de datos* correspondiente que objetos de la *molécula* fueron editados para actualizar el vector de datos original.

6. Archivo de datos de configuración :

Es el archivo físico de donde un objeto del tipo *contenedor de datos* obtiene los datos de configuración para la visualización o cálculo de la dinámica molecular, incluyendo el manual de usuario.

7. Archivo(s) de datos de configuración actual :

Son los archivos físicos necesarios para la ejecución de un cálculo remoto, que contienen los datos mas actuales de la configuración en uso.

8. Archivos de resultados :

Son los nombres de los archivos de resultados de los *procesos de cálculos*.

9. Archivos de resultados de cálculos remotos :

Son los archivos físicos devueltos por los *cálculos remotos* con los resultados obtenidos.

10. Avisos de resultados :

Son los mensajes que el *monitor de procesos* envía al usuario para indicarle que algún *proceso de cálculo* ha finalizado exitosamente.

11. Configuración de documento a imprimir :

Son los parámetros necesarios y disponibles para ajustar el documento a la impresión requerida por el usuario.

12. Configuración de impresora :

Indica los datos necesarios del dispositivo de impresión al que el sistema enviara las impresiones de los documentos generados.

13. Configuración de proceso local :

Son los datos de configuración (de partículas, hoyos, etc.) necesarios para efectuar el cálculo que esta asociado al *proceso local*.

14. Configuración de proceso remoto :

Son los archivos de datos de configuración necesarios para efectuar el cálculo asociado al *proceso remoto*, generalmente el archivo de partículas o el archivo de configuración de la dinámica y el archivo de inicialización de la dinámica.

15. Datos de configuración :

Es la indicación que hace el usuario de los archivos o datos de configuración para generar un *proceso de cálculo*.

16. Datos de configuración modificados por el usuario:

Se refiere a los datos leídos originalmente de los archivos de configuración alterados por el usuario para visualizarlos nuevamente.

17. Deshacer / Rehacer cambios :

Indicación para regresar a un cambio previo en las propiedades modificadas (deshacer una acción realizada), o hacer el efecto contrario (rehacer una acción realizada).

18. Documento :

Es el documento identificado por tipo listo para imprimirse.

19. Documento a imprimir:

Es el nombre del documento que el usuario desea imprimir.

20. Documento no válido :

Es una señal de error generada para indicar que el documento especificado no pudo obtenerse de algún *contenedores de datos*.

21. Edición de objetos indicados :

Se refiere a la marca de edición que el *editor molecular* manda a los objetos Editados para que se muestren los cambios en la visualización de la *escena gráfica* actual.

22. Ejecución de cálculo remoto :

Es la petición que hace un *proceso remoto* a un *cálculo remoto*. Generalmente los cálculos remotos serán programas que se ejecutaran en una máquina lejana con una gran capacidad de procesamiento, en este caso una CRAY. Esta petición lleva incluidos los parámetros y datos necesarios para la ejecución del cálculo remoto específico.

23. Error de impresión :

Notifica el tipo de error que ocurre cuando un documento tuvo un problema durante la impresión.

24. Error de impresora :

Notifica el tipo de error que ocurre cuando el sistema no puede identificar el dispositivo de impresión.

25. Escena gráfica actual :

Es el objeto *escena gráfica* actualizado con las opciones de visualización indicadas por el usuario, por modificaciones del *editor molecular*, por el *modificador de propiedades*, o por la multiplicación de la *celda*.

26. Escena gráfica con celda multiplicada :

Es el objeto resultante de multiplicar la *celda* unitaria de la *escena gráfica* con los factores de multiplicación indicados por el usuario.

27. Escena gráfica creada :

Es el objeto *escena gráfica* con el estado inicial antes de sufrir alguna modificación.

28. Escena gráfica modificada :

Es el objeto *escena gráfica* con algún cambio en sus propiedades o en las de sus objetos componentes (*ejes*, fondo, *puntero de selección*, etc).

29. Estado cancelado :

Indica la cancelación de un *proceso de cálculo* específico.

30. Estado de proceso de cálculo actual :

Indica el estado de un *proceso de cálculo* específico para reportarlo al *monitor de procesos*.

31. Estado en ejecución :

Indica la ejecución de un *proceso de cálculo* específico.

32. Estado finalizado :

Indica la terminación de un *proceso de cálculo* específico.

33. Estado suspendido :

Indica la suspensión de un *proceso de cálculo* específico.

34. Estados de procesos :

Es la relación de los estados de todos los procesos que están siendo supervisados por el *monitor de procesos*.

35. Etiquetas :

Son los identificadores de los objetos seleccionados que se obtuvieron por el *indicador de selección* después de una operación selectiva (método de selección).

36. Etiquetas de objetos seleccionados :

Son los identificadores de los objetos seleccionados solicitados por el *editor molecular* o el *modificador de propiedades*.

37. Factores de multiplicación :

Parámetros indicados por el usuario para multiplicar la *celda* unitaria.

38. Formato desconocido :

Es una señal de error generada para indicar que el archivo de configuración no pudo ser traducido por el sistema.

39. Formato y nombre válido :

Indicación de que el archivo de configuración puede ser usado para la visualización de la Dinámica Molecular y el nombre del archivo puede ser leído para obtener los datos sin problemas.

40. Impresión de documento :

Es la hoja u hojas impresa(s) con la información generada o contenida en el Sistema de Visualización de Dinámica Molecular

41. Impresión lista :

Es una señal generada para indicar que el documento especificado terminó de imprimirse satisfactoriamente.

42. Impresora lista :

Es una señal generada para indicar que la impresora está disponible y puede ser utilizada.

43. Información de procesos :

Es una lista de datos con información sobre los procesos que están siendo supervisados por el *monitor de procesos*, tales como la hora y día de ejecución, la duración del proceso, el tiempo transcurrido, etc.

44. Marca de selección :

Es un valor o un conjunto de valores que indican a los objetos de la escena gráfica que fueron seleccionados para que se visualicen en ese estado y proporcionen sus identificadores al indicador de selección.

45. Mensajes de configuración :

Son avisos de errores que se generan al efectuar alguna actualización a la configuración del *editor molecular*.

46. Mensajes de ediciones :

Son avisos de errores que se generan al intentar efectuar una acción de edición.

47. Mensajes de error :

Son los avisos generados por los *contenedores de datos* y que se producen al fallar la identificación y lectura de los archivos de configuración.

48. Mensajes de impresión :

Son avisos generados por los objetos de impresión para informar al usuario el estado de las impresiones.

49. Mensajes del Editor :

Son mensajes de error generados por el *editor molecular*.

50. Mensajes del sistema :

Son mensajes generados por el *modificador de propiedades*.

51. Modo de selección y tipo de objeto(s) :

Se refiere a la forma en que se va a indicar la selección de los objetos, por especie, individual, o por grupo, y el tipo de objeto que se desea seleccionar, átomos, hoyos, dipolos, conexiones, etc.

52. Nombre de archivo nuevo :

Es el nombre del archivo de configuración renombrado por el usuario que fue asociado al mismo *contenedor de datos* al que estaba asociado el archivo sustituido. Con este nombre se colocarán los datos actuales de la configuración que le corresponda.

53. Nombre del archivo de configuración :

Es el nombre del archivo de datos de configuración que va a ser verificado por el sistema antes de su utilización.

54. Nombre del archivo seleccionado :

Es el nombre del archivo de datos de configuración indicado explícitamente por el usuario.

55. Nombre inválido :

Es una señal de error generada por el *contenedor de datos* para indicar que el nombre no fue reconocido como un nombre válido.

56. Nombre válido :

Es una señal generada por el *contenedor de datos* para indicar que el nombre si fue reconocido como un nombre válido.

57. Notificación de base :

Es un mensaje para notificarle al usuario si la *celda* es unitaria o no, si es una base se considera una celda unitaria y se puede multiplicar de lo contrario la multiplicación no se realiza.

58. Objeto de animación y acción :

Es el tipo de objeto que se va a animar (*molécula, sistema de cavidades, etc*) y el estado de animación en que se desea que éste se encuentre.

59. Opciones de visualización :

Son indicaciones del usuario para visualizar los objetos de la *escena gráfica*, principalmente de los *ejes*, el Fondo o el *puntero de selección*.

60. Parámetros de Red :

Se refiere a los parámetros necesarios para establecer la conexión con la máquina en donde se ejecutaran los cálculos remotos como dirección IP, usuario y contraseña de la cuenta.

61. Proceso activado :

Es una señal que indica que un *proceso* fue ejecutado para actualizar la tabla de procesos.

62. Proceso asociado a cálculo :

Es el tipo de proceso y de cálculo que se generará para su ejecución inmediata.

63. Proceso cancelado :

Es una señal que indica que un *proceso* fue cancelado para actualizar la tabla de procesos.

64. Proceso de cálculo cancelado :

Es una señal que le indica al *proceso* que debe abortar la operación de cálculo.

65. Proceso de calculo ejecutado :

Es una señal que le indica al *proceso* que debe ejecutar la operación de cálculo.

66. Proceso de cálculo reactivado :

Es una señal que le indica al *proceso* que continúe la operación de cálculo suspendida.

67. Proceso de cálculo suspendido :

Es una señal que le indica al *proceso* que debe suspender momentáneamente la operación de cálculo.

68. Proceso de cálculo terminado :

Es el objeto *proceso* que finalizó su operación de cálculo asociada y que es requerido para devolver los resultados obtenidos.

69. Proceso ejecutado :

Es una señal que indica que se requiere generar un *proceso de cálculo*.

70. Proceso local :

Es un objeto del tipo *proceso* generado para ejecutar un cálculo local.

71. Proceso reactivado :

Es una señal que indica que un *proceso suspendido* puede continuar su ejecución.

72. Proceso remoto :

Es un objeto del tipo *proceso*, generado para ejecutar un cálculo remoto.

73. Proceso suspendido :

Es una señal que indica que un *proceso* fue suspendido para actualizar la tabla de procesos.

74. Procesos de cálculo y acción :

Son los datos ingresados por el usuario para indicar al *monitor de procesos* el tipo de proceso y la acción que efectuara sobre él.

75. Procesos terminados :

Es la lista de procesos concluidos que se han obtenido de la tabla de procesos para generar los avisos de resultados al usuario.

76. Propiedad(es) y valor(es) :

Son las propiedades de los objetos a modificar y sus respectivos valores indicados por el usuario al *modificador de propiedades*.

77. Propiedades modificadas de objetos indicados :

Es una indicación que se hace a la *escena gráfica* para que actualice la visualización de los objetos a los cuales se les cambio alguna propiedad.

78. Selección previa :

Es una señal que determina el *editor molecular* y que indica que la acción se va a realizar sobre un conjunto específico de objetos de la *molécula*.

79. Sin selección :

Es una señal que determina el *editor molecular* y que indica que la acción se va a realizar no requiere de una selección específica de objetos de la *molécula*

80. Solicitud de impresión :

Se refiere al aviso que hace el sistema internamente para comprobar el dispositivo de impresión y preparar el documento.

81. Tabla de procesos :

Es una relación de todos los *procesos* con sus respectivos estados actuales y su información relevante que el *monitor de procesos* genera constantemente.

82. Terminación de programa :

Es la indicación que hace el usuario para finalizar el uso del sistema de visualización y que permite indicarle al sistema que libere la *escena grafica* de la memoria y finalice la ejecución.

83. Tipo de archivo con formato diferente :

Es una señal que se genera al revisar el formato de los archivos de configuración que utiliza el sistema, y que sirve para indicar que se debe intentar traducir el formato para poderlo visualizar.

84. Tipo de documento :

Indica el formato del documento a imprimir, si es de texto (archivos de configuración), o una imagen generada de la *escena gráfica*.

85. Tipo de transformación espacial, dispositivo y eje :

Son los parámetros seleccionados por el usuario para efectuar la transformación espacial, que puede ser rotación, traslación, escalamiento y el dispositivo puede ser alguno de los disponibles como perillas, ratón, teclado o la interfaz gráfica, sobre el eje x, y o z.

86. Tipo inválido :

Es una señal que indica que la extensión del archivo de configuración no es del tipo esperado por el sistema.

87. Tipo válido :

Es una señal que indica que la extensión del archivo de configuración es del tipo esperado por el sistema.

88. Tipo y datos de caja :

Son los parámetros de configuración de la *caja* que sirven para delimitar el espacio de la *celda*, y que provienen del *contenedor de datos* de configuración de partículas.

89. Tipo y radio de conexión, tabla de conectividad y especie de átomos :

Son los parámetros necesarios para la conexión de los *átomos* de la *molécula* o *moléculas* contenidas en la *escena gráfica* introducidos o seleccionados por el usuario.

90. Vector actualizado :

Es el arreglo que contiene los datos más actuales de configuración de un *contenedor de datos*.

91. Vector de datos :

Es el arreglo que contiene los datos que se obtuvieron del archivo de configuración leído inicialmente.

92. Vector de datos de configuración actual :

Es el arreglo que contiene los datos más actuales proporcionado por el *contenedor de datos* a los objetos que lo solicitan.

93. Vector de datos de configuración actualizado :

Es el arreglo que contiene los datos de configuración de un determinado conjunto de objetos contenidos en la *escena gráfica* en el momento de la visualización actual.

94. Vector de datos de historia actual :

Es el arreglo que lleva la relación de los objetos modificados tanto por el *editor molecular* como por el *modificador de propiedades* a través de un determinado número de acciones, y que permite deshacer la alteraciones realizadas a la *escena gráfica* o viceversa.

95. Vector (es) de datos de resultados :

Arreglo(s) que contiene(n) los datos de configuración obtenidos por los *procesos de cálculos*

96. Vector de modificación de propiedades :

Es el arreglo que contiene la relación de objetos y propiedades modificadas por el *modificador de propiedades* para actualizar el vector de historia.

97. Vector de propiedades actuales :

Es el arreglo que contiene la relación de propiedades actuales de un objeto modificado.

98. Vector de suceso actual :

Es el arreglo que contiene la acción realizada por el *editor molecular* sobre los objetos de

la *molécula* y los identificadores de los objetos en un determinado momento, para generar el vector de historia.

99. Vectores de datos de configuración actuales :

Más de un vector de datos de configuración actual.

100. Vectores de datos de configuración actualizados :

Más de un vector de datos de configuración actualizado.

101. Visualización de conexiones :

Es la representación gráfica de la conexión entre varios átomos de una *molécula*.

102. Visualización de escena gráfica actual :

Es la representación gráfica de la *escena gráfica* en forma estática.

103. Visualización de objeto animado :

Es la representación gráfica de un objeto animado (*molécula, sistema de cavidades o superficie de potencial*)

104. Visualización de objetos seleccionados :

Es la representación gráfica que muestra a los objetos seleccionados resaltados de los que no están seleccionados.

105. Visualización de transformación espacial :

Es la visualización de la *escena gráfica* cuando sufre una transformación espacial.

106. Visualización de escena gráfica y cambios = Visualización de escena gráfica actual + Visualización de transformación espacial + Visualización de objeto animado + Visualización de conexiones + Visualización de objetos seleccionados.

9.1.3.7 Comparar los métodos

- ▲ Métodos del modelo de objetos
- * Métodos del modelo dinámico
- ⊕ Métodos del modelo funcional

<p>Objeto gráfico</p> <ul style="list-style-type: none"> ▲ Ajustar configuración al tipo de caja <p>Objeto estático</p> <ul style="list-style-type: none"> ▲ Cambiar grosor de línea ▲ Cambiar tipo de línea <p>Caja</p> <ul style="list-style-type: none"> ▲ Pinta caja ▲ Seleccionar tipo de caja ▲ Acceder a propiedades ▲ Asignar propiedades <p>Eje</p> <ul style="list-style-type: none"> ▲ Pinta eje ▲ Acceder a propiedades ▲ Asignar propiedades <p>Plano potencial</p> <ul style="list-style-type: none"> ▲ Pinta campos escalares ▲ Acceder a propiedades ▲ Asignar propiedades <p>Objeto dinámico</p> <ul style="list-style-type: none"> ▲ Marcar por conjunto ▲ Marcar individual ▲ Indicar tipo de selección ▲ Cambiar forma de visualización ▲ Indicar estado de modificación <p>Esfera</p> <ul style="list-style-type: none"> ▲ Pinta esfera * Mostrar en modo de selección * Mostrar en modo de edición * Cambiar visibilidad * Cambiar calidad * Cambiar densidad * Cambiar estructura * Cambiar representación * Cambiar estado de edición <p>Átomo</p> <ul style="list-style-type: none"> ▲ Generar átomo inicial ▲ Pintar visualización de átomo * Mostrar en modo de conexión ▲ Acceder a propiedades * Indicar estado * Indicar tipo de selección ▲ Asignar propiedades * Cambiar propiedades * Establecer tipo de conexión ▲ Liberar átomo de memoria <p>Hoyo</p> <ul style="list-style-type: none"> ▲ Generar hoyo inicial ▲ Pintar visualización de hoyo ▲ Acceder a propiedades * Indicar estado * Indicar tipo de selección ▲ Asignar propiedades * Cambiar propiedades <math>\diamond</math> radio ▲ Liberar hoyo de memoria <p>Dipolo</p> <ul style="list-style-type: none"> ▲ Generar dipolo inicial ▲ Pintar visualización de dipolo ▲ Acceder a propiedades * Indicar estado * Indicar tipo de selección ▲ Asignar propiedades * Cambiar propiedades ▲ Liberar dipolo de memoria <p>Celda</p> <ul style="list-style-type: none"> ▲ Generar celda inicial ▲ Pintar celda ▲ Liberar celda de la memoria 	<p>Conexión</p> <ul style="list-style-type: none"> ▲ Generar conexión inicial ▲ Pintar visualización de conexión * Mostrar en modo de selección * Mostrar en modo de edición ▲ Acceder a propiedades * Indicar estado * Indicar tipo de selección ▲ Asignar propiedades * Cambiar visibilidad * Cambiar calidad * Cambiar densidad * Cambiar estructura * Cambiar representación * Cambiar propiedades ▲ Liberar conexión de memoria <p>Objeto animado</p> <ul style="list-style-type: none"> ▲ Detener animación en cualquier cuadro ▲ Invertir sentido de la animación ▲ Encender y apagar ▲ Recibir estado de animación ▲ Cambiar estado de animación ▲ Proporcionar estado de animación actual ▲ Indicar tipo de selección <p>Molécula</p> <ul style="list-style-type: none"> ▲ Generar molécula inicial ▲ Pintar visualización de molécula * Mostrar conexiones * Mostrar en modo de selección * Mostrar en modo de edición ▲ Fraccionar molécula ▲ Mover molécula ▲ Conectar átomos <ul style="list-style-type: none"> ▲ Lee tabla de conectividad ▲ Valida conexiones * Conectar todos los átomos de todas las especies * Conectar átomos de una especie ▲ Establecer tipo de conexión ▲ Modificar lista de átomos <ul style="list-style-type: none"> * Agregar elementos * Copiar elementos * Pegar elementos * Cortar elementos * Agrupar elementos ▲ Modificar lista de dipolos <ul style="list-style-type: none"> * Agregar elementos * Copiar elementos * Pegar elementos * Cortar elementos * Agrupar elementos ▲ Modificar lista de conexiones <ul style="list-style-type: none"> * Agregar elementos * Copiar elementos * Pegar elementos * Cortar elementos * Agrupar elementos ▲ Unir molécula ▲ Unir átomos ▲ Recibir vector de partículas ▲ Recibir vector de dipolos ▲ Recibir vector de conexiones ▲ Devolver vector de partículas ▲ Devolver vector de dipolos ▲ Devolver vector de conexiones ▲ Acceder a propiedades <ul style="list-style-type: none"> * Indicar estado * Indicar tipo de selección * Proporcionar estado de animación actual ▲ Cambiar propiedades <ul style="list-style-type: none"> * Cambiar visibilidad * Cambiar calidad * Cambiar densidad * Cambiar estructura * Cambiar representación * Cambiar estado de edición * Cambiar propiedades * Notificar cambio para registro ▲ Liberar molécula de memoria 	<p>Sistema de cavidades</p> <ul style="list-style-type: none"> ▲ Recibir vector de hoyos ▲ Generar sistema de cavidades inicial ▲ Pintar visualización de cavidades * Mostrar en modo de selección ▲ Acceder a propiedades * Indicar estado * Indicar tipo de selección * Proporcionar estado de animación actual ▲ Asignar propiedades <ul style="list-style-type: none"> * Cambiar visibilidad * Cambiar calidad * Cambiar densidad * Cambiar estructura * Cambiar representación * Cambiar estado de animación * Cambiar propiedades ▲ Devolver vector de hoyos * Notificar cambio para registro ▲ Liberar sistema de cavidades de memoria <p>Superficie potencial</p> <ul style="list-style-type: none"> ▲ Recibir vector de potenciales ▲ Generar superficie potencial inicial ▲ Pintar visualización de superficie potencia * Mostrar en modo de selección ▲ Acceder a propiedades <ul style="list-style-type: none"> * Indicar tipo de selección * Proporcionar estado de animación actual ▲ Asignar propiedades <ul style="list-style-type: none"> * Cambiar visibilidad * Cambiar calidad * Cambiar densidad * Cambiar estructura * Cambiar representación * Cambiar estado de animación * Cambiar propiedades ▲ Devolver vector de potenciales ▲ Liberar superficie potencial de memoria <p>Indicador de selección</p> <ul style="list-style-type: none"> ▲ Marcar objetos seleccionados ▲ Identificar objetos seleccionados ▲ Obtener lista de etiquetas de objetos seleccionados ▲ Regresar lista de etiquetas de objetos seleccionados <p>Estructura cristalina</p> <ul style="list-style-type: none"> ▲ Multiplicar celda unitaria ▲ Crear estructura cristalina ▲ Pintar estructura cristalina ▲ Cambiar factores de multiplicación ▲ Liberar estructura de la memoria <p>Escena gráfica</p> <ul style="list-style-type: none"> ▲ Generar escena gráfica inicial (crear escena gráfica) ⊕ Actualizar escena gráfica ▲ Modificar objetos de escena gráfica (modificar escena gráfica) <ul style="list-style-type: none"> * Modificar molécula * Modificar ejes * Modificar puntero 3D * Modificar fondo * Modificar celda ⊕ Multiplicar celda unitaria ▲ Deshacer cambios hechos en la escena ▲ Rehacer cambios ▲ Cambiar perspectiva ▲ Asignar estado de animación ⊕ Proporcionar vectores de configuración ⊕ Proporcionar etiquetas de objetos seleccionados ▲ Cambiar (realizar) transformación espacial <ul style="list-style-type: none"> * Rotación * Traslación * Escalamiento * Pintar visualización de escena gráfica (visualizar escena gráfica) ▲ Visualizar ejes ▲ Visualizar estructura cristalina ▲ Visualizar puntero 3D ▲ Liberar escena gráfica de la memoria (destruir escena gráfica)
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>Proceso cálculo</p> <ul style="list-style-type: none"> ▲ Generar vector de datos <p>Cálculo de magnitudes Y ángulos</p> <ul style="list-style-type: none"> ▲ Generar distancias ▲ Generar ángulos ▲ Mostrar resultados <p>Cálculo imagen</p> <ul style="list-style-type: none"> ▲ Generar Imagen ▲ Devolver vector de imagen <p>Cálculo de dinámica</p> <ul style="list-style-type: none"> ▲ Generar dinámica ▲ Devolver archivos de datos <p>Cálculo generado por partículas</p> <ul style="list-style-type: none"> ▲ Obtener vector de partículas ▲ Devolver vector de datos <p>Cálculo local</p> <ul style="list-style-type: none"> ▲ Generar rayos x ▲ Generar vecinos ▲ Generar análisis de coordinación ▲ Generar archivo de datos <p>Cálculo remoto</p> <ul style="list-style-type: none"> ▲ Generar dipolos ▲ Generar hoyos ▲ Generar campo escalar ▲ Generar corte <p>Contenedor de datos</p> <ul style="list-style-type: none"> ▲ Verificar tipo de archivo de datos ⚡ Obtener nombre de archivo de datos ▲ Verificar nombre de archivo de datos ▲ Verificar formato de archivo de datos ▲ Leer archivo de datos ▲ Actualizar vector ▲ Traducir datos ▲ Renombrar archivo de datos ▲ Guardar archivo de datos ▲ Cerrar archivo de datos ▲ Generar vector de datos de configuración actual ▲ Devolver vector de datos ▲ Devolver archivo de datos ▲ Notificar mensajes de error <p>Contenedor de hoyos</p> <ul style="list-style-type: none"> ▲ Verificar tipo de archivo de hoyos ▲ Verificar nombre de archivo de hoyos ▲ Verificar formato de archivo de hoyos ▲ Leer archivo de hoyos ▲ Actualizar vector ▲ Guardar archivo de hoyos ▲ Generar vector de configuración actual de hoyos ▲ Devolver vector de hoyos ▲ Devolver archivo de hoyos ▲ Obtener nombre de archivo ★ Cerrar archivo anterior sin guardar cambios ★ Crear archivo con nuevo nombre ★ Mostrar datos para modificación ⚡ Notificar mensajes de error 	<p>Contenedor de partículas</p> <ul style="list-style-type: none"> ▲ Verificar tipo de archivo de partículas ▲ Verificar nombre de archivo de partículas ▲ Verificar formato de archivo de partículas ▲ Leer archivo de partículas ▲ Actualizar vector ▲ Traducir datos de partículas ▲ Guardar archivo de partículas ▲ Generar vector de configuración actual de partículas ▲ Devolver vector de partículas ▲ Devolver archivo de partículas ▲ Proporcionar tipo y datos de caja ★ Obtener nombre de archivo ★ Cerrar archivo anterior sin guardar cambios ★ Crear archivo con nuevo nombre ★ Mostrar datos para modificación ⚡ Notificar mensajes de error <p>Contenedor de imagen</p> <ul style="list-style-type: none"> ▲ Actualizar vector ▲ Generar vector de configuración actual de imagen ▲ Guardar archivo de imagen ▲ Devolver vector de imagen ▲ Devolver archivo de imagen ★ Obtener nombre de archivo ★ Cerrar archivo anterior sin guardar cambios ★ Crear archivo con nuevo nombre ⚡ Notificar mensajes de error <p>Contenedor de potencial</p> <ul style="list-style-type: none"> ▲ Verificar tipo de archivo de potencial ▲ Verificar nombre de archivo de potencial ▲ Verificar formato de archivo de potencial ▲ Leer archivo de potencial ▲ Actualizar vector ▲ Guardar archivo de potencial ▲ Generar vector de configuración actual de potencial ▲ Devolver vector de potencial ▲ Devolver archivo de potencial ★ Obtener nombre de archivo ★ Cerrar archivo anterior sin guardar cambios ★ Crear archivo con nuevo nombre ★ Mostrar datos para modificación ⚡ Notificar mensajes de error <p>Contenedor de dipolos</p> <ul style="list-style-type: none"> ▲ Verificar tipo de archivo de dipolos ▲ Verificar nombre de archivo de dipolos ▲ Verificar formato de archivo de dipolos ▲ Leer archivo de dipolos ▲ Actualizar vector ▲ Guardar archivo de dipolos ▲ Generar vector de configuración actual de dipolos ▲ Devolver vector de dipolos ▲ Devolver archivo de dipolos ★ Obtener nombre de archivo ★ Cerrar archivo anterior sin guardar cambios ★ Crear archivo con nuevo nombre ★ Mostrar datos para modificación ⚡ Notificar mensajes de error <p>Contenedor de conexiones</p> <ul style="list-style-type: none"> ▲ Verificar tipo de archivo de conexiones ▲ Verificar nombre de archivo de conexiones ▲ Verificar formato de archivo de conexiones ▲ Leer archivo de conexiones ▲ Actualizar vector ▲ Guardar archivo de conexiones ▲ Generar vector de configuración actual de conexiones ▲ Devolver vector de conexiones ▲ Devolver archivo de conexiones ★ Obtener nombre de archivo ★ Cerrar archivo anterior sin guardar cambios ★ Crear archivo con nuevo nombre ★ Mostrar datos para modificación ⚡ Notificar mensajes de error 	<p>Contenedor de rayos x</p> <ul style="list-style-type: none"> ▲ Verificar tipo de archivo de rayos x ▲ Verificar nombre de archivo de rayos x ▲ Verificar formato de archivo de rayos x ▲ Leer archivo de rayos x ▲ Actualizar vector ▲ Guardar archivo de rayos x ▲ Generar vector de configuración actual de rayos x ▲ Devolver vector de rayos x ★ Obtener nombre de archivo ★ Cerrar archivo anterior sin guardar cambios ★ Crear archivo con nuevo nombre ★ Mostrar datos para modificación ⚡ Notificar mensajes de error <p>Contenedor de configuración de dinámica t</p> <ul style="list-style-type: none"> ▲ Verificar tipo de archivo de configuración de dinámica t ▲ Verificar nombre de archivo de configuración de dinámica t ▲ Verificar formato de archivo de configuración de dinámica t ▲ Leer archivo de configuración de dinámica t ▲ Actualizar vector ▲ Guardar archivo de configuración de dinámica t ▲ Generar vector actual de configuración de dinámica t ▲ Devolver vector de configuración de dinámica t ▲ Devolver archivo de configuración de dinámica T ★ Obtener nombre de archivo ★ Cerrar archivo anterior sin guardar cambios ★ Crear archivo con nuevo nombre ★ Mostrar datos para modificación ⚡ Notificar mensajes de error <p>Contenedor de configuración de dinámica t-dt</p> <ul style="list-style-type: none"> ▲ Verificar tipo de archivo de configuración de dinámica t-dt ▲ Verificar nombre de archivo de configuración de dinámica t-dt ▲ Verificar formato de archivo de configuración de dinámica t-dt ▲ Leer archivo de configuración de dinámica t-dt ▲ Actualizar Vector t-dt ▲ Guardar archivo de configuración de dinámica t-dt ▲ Generar vector actual de configuración de dinámica t-dt ▲ Devolver vector de configuración de dinámica t-dt ▲ Devolver archivo de configuración de dinámica t-dt ★ Obtener nombre de archivo ★ Cerrar archivo anterior sin guardar cambios ★ Crear archivo con nuevo nombre ★ Mostrar datos para modificación ⚡ Notificar mensajes de error <p>Contenedor de inicia dinámica</p> <ul style="list-style-type: none"> ▲ Verificar tipo de archivo de inicia dinámica. ▲ Verificar nombre de archivo de inicia dinámica ▲ Verificar formato de archivo de inicia dinámica ▲ Leer archivo de inicia dinámica ▲ Actualizar vector ▲ Guardar archivo de inicia dinámica ▲ Generar vector de configuración actual de Inicia Dinámica. ▲ Devolver vector de inicia dinámica ▲ Devolver Archivo de inicia dinámica ★ Obtener nombre de archivo ★ Cerrar archivo anterior sin guardar cambios ★ Crear archivo con nuevo nombre ★ Mostrar datos para modificación ⚡ Notificar mensajes de error
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>Monitor de procesos</p> <ul style="list-style-type: none"> ▲ Generar proceso ⚙ Manejar procesos ▲ Asociar cálculo con proceso <ul style="list-style-type: none"> ★ Verificar tipo de proceso ★ Solicitar datos de configuración ▲ Asigna tipo de proceso ▲ Identificar procesos residentes <ul style="list-style-type: none"> ★ Obtener información de procesos residentes ★ Mostrar información de procesos residentes ▲ Activar proceso <ul style="list-style-type: none"> ▲ Añadir proceso a tabla de procesos ★ Cambiar estado en tabla de procesos ⚙ (Mantener tabla de procesos) ▲ Obtener estado del proceso ▲ Desplegar estado del proceso ⚙ Reactivar proceso ▲ Suspender proceso ▲ Avisos del proceso <ul style="list-style-type: none"> ★ Notificar error al monitor ★ Notificar suspensión ⚙ Generar avisos de procesos terminados ★ Notificación de resultados ★ Cancelar proceso ▲ Cerrar proceso <p>Proceso</p> <ul style="list-style-type: none"> ★ Generar proceso inicial ▲ Generar estado del proceso ▲ Obtener identificadores del proceso ▲ Ejecutar cálculo asociado <ul style="list-style-type: none"> ★ Esperar resultados de cálculo ★ Recibir datos de cálculo asociado ★ Suspender cálculo ▲ Notificar mensajes a monitor ▲ Mostrar avance y estado de cálculo asociado ▲ Liberar proceso de memoria <p>Proceso local</p> <ul style="list-style-type: none"> ▲ Ejecutar cálculo local ▲ Enviar vector de datos de entrada ▲ Esperar resultados de cálculo <ul style="list-style-type: none"> ▲ Recibir vector de datos de salida ▲ Suspender cálculo ▲ Abortar cálculo <p>Proceso remoto</p> <ul style="list-style-type: none"> ▲ Ejecutar cálculo remoto ▲ Enviar archivos de entrada ▲ Esperar resultados de cálculo <ul style="list-style-type: none"> ▲ Recibir archivos de salida ▲ Suspender cálculo ▲ Abortar cálculo 	<p>Impresiones</p> <ul style="list-style-type: none"> ★ Identificar tipo de documento ★ Verificar estado de impresora ▲ Poner en línea la Impresora ▲ Recibir el vector de datos (T) ▲ Notificar resultado de impresión ▲ Notificar tipo de error ⚙ Notificar mensajes de impresión <p>Impresión de archivo de configuración</p> <ul style="list-style-type: none"> ▲ Recibir vector de configuración ▲ Verificar tipo de datos ▲ Imprimir archivo de configuración <p>Impresión de imagen</p> <ul style="list-style-type: none"> ▲ Recibir vector de imagen ▲ Identificar formato (verificar tipo de imagen) ▲ Imprimir imagen 	<p>Puntero 3D</p> <ul style="list-style-type: none"> ▲ Obtener coordenadas ▲ Cambiar tipo de puntero ▲ Cambiar color al puntero <p>Editor molecular</p> <ul style="list-style-type: none"> ⚙ Identificar selección y acción ⚙ Obtener etiquetas de objetos seleccionados ⚙ Ejecutar acción <ul style="list-style-type: none"> ▲ Copiar objetos en escena ▲ Pegar objetos en escena ▲ Cortar objetos de escena ▲ Unir objetos de escena ▲ Deshacer cambios ▲ Rehacer cambios ▲ Agrupar objetos de escena ▲ Mostrar tabla periódica ▲ Asignar propiedades para mostrar elemento sobre una ventana ▲ Generar átomo temporal ▲ Agregar átomo sobre una molécula ▲ Generar conexión temporal ▲ Agregar conexión a escena ▲ Generar dipolo temporal ▲ Agregar dipolo a escena ▲ Agregar molécula sobre celda ⚙ Identificar errores ⚙ Efectuar actualización del editor ⚙ Actualizar vector de historia <p>Modificador de propiedades</p> <ul style="list-style-type: none"> ▲ Identificar propiedad y objeto <ul style="list-style-type: none"> ★ Detectar selección previa ★ Obtener etiquetas de objetos seleccionados ▲ Cambiar propiedad <ul style="list-style-type: none"> ★ Cambiar calidad a objetos indicados ★ Cambiar radio a objetos indicados ★ Cambiar transparencia a objetos indicados ★ Cambiar color a objetos indicados ★ Cambiar estructura a objetos indicados ▲ Deshacer cambios ▲ Rehacer cambios ⚙ Actualizar vector de modificación de propiedades ★ Actualizar vector de historia ▲ Notificar a editor
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

9.2 Diseño orientado a objetos

9.2.1 Diseño de sistema

9.2.1.1 División del sistema en subsistemas

Se pueden identificar los siguientes subsistemas dentro del sistema de visualización de dinámica molecular, de acuerdo a los servicios que realizan:

- ** **Subsistema de interacción con el usuario:** (E/S de opciones) Este subsistema se encargará de realizar las entradas y salidas por parte de los usuarios, es decir, atenderá las peticiones del usuario y recibirá los datos de configuración de opciones que se especifiquen (formalmente esta constituido por la interfaz gráfica del sistema de visualización de dinámica molecular).
- ** **Subsistema de control de datos:** En este subsistema se realizará el manejo de la entrada de los datos requeridos para el funcionamiento del sistema, y la salida de los resultados provenientes de los subsistemas relacionados. Dentro de este contexto se maneja la lectura y escritura de archivos, así como el control interno de los datos solicitados por otros subsistemas.
- ** **Subsistema de realización de cálculos:** En este subsistema se realizarán todos los cálculos requeridos por los otros subsistemas, así como la entrega de los resultados obtenidos.
- ** **Subsistema de realización de gráficas:** Este subsistema se encargará de la función principal del sistema de visualización, que consiste en realizar la visualización gráfica de las moléculas, basándose en el sistema de bibliotecas gráficas de OpenGL.
- ** **Subsistema de control y monitoreo de procesos:** Este subsistema se encargará de crear y monitorear los procesos generados, intercambiando mensajes del estado actual de los mismos, y desplegando información relacionada con ellos. Cada proceso está asociado a un cálculo requerido.
- ** **Subsistema de realización de impresiones:** Este subsistema se encargará de realizar las impresiones.

9.2.1.1.1 Relación entre subsistemas

La relación entre los subsistemas es:

- ** Entre *interacción con el usuario* y *control de datos* hay una relación de tipo cliente-servidor, ya que *interacción con el usuario* pide a *control de datos* el archivo que se especificó (o archivos) y este le devuelve resultados, a través de los otros subsistemas.



** Así mismo la relación entre *interacción con el usuario* y el subsistema de *control y monitoreo de procesos* es de tipo cliente-servidor, ya que el subsistema de *interacción con el usuario* le pide a *control y monitoreo de procesos* la creación de un proceso de cálculo determinado y la información sobre el estado actual del mismo.



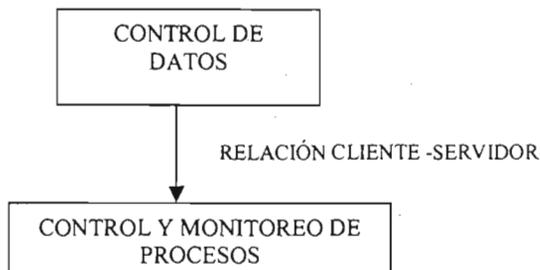
** La relación entre el subsistema de *interacción con el usuario* y *realización de gráficas* es de tipo cliente - servidor, ya que desde el subsistema de *interacción con el usuario* se solicita la visualización de los resultados generados por los cambios en tiempo real, al subsistema de *realización de gráficas*.



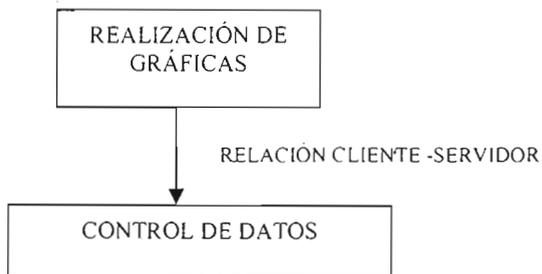
** La relación entre el subsistema de *interacción con el usuario* y el subsistema de *realización de impresiones* es cliente - servidor, debido a que se le solicita al subsistema de impresión las impresiones requeridas.



** La relación entre el subsistema de *control de datos* y el subsistema de *control y monitoreo de procesos* es de tipo cliente - servidor debido a que es el subsistema de *control de datos* es el que solicita la información generada por los procesos de cálculos.



- ** La relación entre el subsistema de *control de datos* y el subsistema de *realización de gráficas* es de tipo cliente - servidor, debido a que es el subsistema de *control de datos* el que proporciona o almacena información de acuerdo a las peticiones del subsistema de *realización de gráficas*.



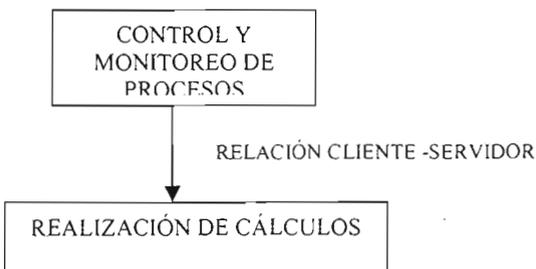
- ** La relación entre el subsistema de *control de datos* y el subsistema de *realización de Impresiones* es de tipo cliente - servidor, ya que el subsistema de *control de datos* le proporciona la información que le solicita el subsistema de *realización de impresiones* para llevar a cabo su trabajo.



- ** La relación entre el subsistema de *realización de cálculos* y el de *realización de gráficas* es del tipo cliente - servidor, ya que el subsistema de *realización de gráficas* solicita los datos correspondientes a los cálculos simples.



- ** La relación entre el subsistema de *control y monitoreo de procesos* y el subsistema de *realización de cálculos* es igualmente de tipo cliente - servidor, ya que el subsistema de *control y monitoreo* (cliente), solicita los datos de los cálculos, así como el estado actual de los procesos correspondientes.



9.2.1.1.2 Descomposición del sistema

- Diagrama de descomposición en capas y particiones:



El diagrama anterior indica que la parte alta del sistema es la de la interacción con el usuario (interfaz), enseguida van las bibliotecas necesarias para el funcionamiento de la interfaz, la siguiente capa que esta dividida en tres particiones, se encarga de interpretar la información recibida de las capas inferiores, proporcionando los resultados finales al usuario. a continuación se encuentra la capa que se encarga de generar y obtener la información para el funcionamiento adecuado del sistema. Finalmente se tienen las capas correspondientes a los recursos del equipo sobre el cual va a funcionar la aplicación, como son las bibliotecas gráficas, el sistema operativo, y el hardware.

El sistema se trata de una **arquitectura abierta**, ya que hay dependencia entre los diferentes subsistemas.

NOTA: Los subsistemas con * están ya implementados dentro de la configuración del sistema en el cual se realizara el SVDM.

9.2.1.1.3 Topología del sistema

El sistema de SVDM es un híbrido, dentro de las dos maneras en las que están definidas sus computaciones o cálculos. Lo anterior es porque en principio los cálculos y procesamientos realizados por el sistema son secuenciales, es decir, el sistema lee el archivo y lo traduce a coordenadas de modo que se obtenga la gráfica correspondiente y a partir de ahí espera para responder a las peticiones del usuario, siendo hasta este momento secuencial.

Posteriormente, cuando el sistema reciba una petición por parte del usuario, y ésta implica la realización de uno o mas procesos, ya sea locales o remotos, se comportará de manera centralizada, como un "subsistema" maestro, controlando y monitoreando todos los otros procesos que se necesario ejecutar, e interactuando con todas las tareas que sean parte del SVDM, y que por lo tanto le estén enviando y solicitando información.

9.2.1.2 Identificar concurrencia

9.2.1.2.1 Concurrencia de objetos

- ** Partículas, conexiones, dipolos, así como caja y ejes, pueden estar activos al mismo tiempo, o pueden ser independientes. De igual manera, pueden estar activos al mismo tiempo en cualquier combinación entre ellos.
- ** Cada eje va a estar activo siempre junto con los otros dos.
- ** El puntero 3D y la escena gráfica estarán activos al mismo tiempo.
- ** Escena gráfica es concurrente con la interfaz gráfica.
- ** Los objetos y clases de contenedores serán concurrentes o independientes según las solicitudes del usuario.
- ** El sistema será concurrente con el monitor de procesos.

9.2.1.2.2 Objetos secuenciales

- ** La realización de una impresión estará dada en función de si existen los datos necesarios, esto quiere decir que existe una dependencia entre realizar una impresión y la generación u obtención de los datos necesarios para esta. Entonces los objetos de manejo de archivos y realización de impresiones, tienen actividades secuenciales.
- ** Los objetos de control de datos y realización de gráficas son secuenciales.
- ** Control de datos y monitoreo de procesos son secuenciales.

9.2.1.2.3 Independencia de objetos

- ** Los objetos de control de datos y realización de cálculos son independientes.
- ** Los objetos de realización de gráficas y monitoreo de procesos son independientes.
- ** Los objetos de realización de impresiones y de realización de gráficas son independientes.

El orden de los objetos es el siguiente:

Objetos de contenedores

- ** Contenedor de hoyos
- ** Contenedor de imagen
- ** Contenedor de dipolos
- ** Contenedor de partículas
- ** Contenedor de potencial
- ** Contenedor de conexiones
- ** Contenedor de configuración de dinámica
- ** Contenedor de rayos x
- ** Contenedor de inicia dinámica

Objetos gráficos

- ** Objeto gráfico
- ** Objeto estático
- ** Caja
- ** Eje
- ** Plano potencial
- ** Objeto dinámico
- ** Esfera
- ** Átomo
- ** Hoyo
- ** Dipolo
- ** Conexión

- ** Objeto animado (estos objetos dependen de los objetos gráficos, es por eso que el orden de aparición va a estar dado por la creación de las partículas, conexiones, dipolos, etc., dentro de la escena)
- ** Sistema de cavidades
- ** Molécula
- ** Superficie potencial

- ** Interfaz gráfica (clases gráficas)
- ** Escena gráfica
- ** Celda
- ** Indicador de selección
- ** Modificador de propiedades
- ** Estructura cristalina

Nota: Los objetos anteriores dependen de los objetos de clases gráficas, como partículas, conexiones, etc., pero se especificó un orden de esta forma para dar secuencia en cuanto a la definición y propósito de los diferentes objetos.

Objetos de editor molecular

- ** Editor molecular
- ** Puntero 3D

Objetos de monitor de procesos

- ** Monitor de procesos
- ** Proceso
- ** Proceso local
- ** Proceso remoto

Objetos de cálculos

- ** Proceso cálculo
- ** Cálculo imagen
- ** Cálculo de dinámica
- ** Cálculo generado por partículas
- ** Cálculo de magnitudes y ángulos
- ** Cálculo local
- ** Cálculo remoto

Objetos de impresiones

- ** Impresiones
- ** Impresión de archivo de configuración.
- ** Impresión de imagen.

9.2.1.2.4 Hilos de control

Serán necesario implementar algo similar a hilos de control en los siguientes eventos:

- ** Cuando se realice algún proceso requerido por alguna función del sistema, sea éste un proceso local o uno remoto.
- ** Cuando se envíe un proceso a realizarse en la computadora remota (CRAY), es decir, cuando se realice alguno de los cálculos mayores, como pueden ser, cálculo de dinámica, cálculo de potenciales, etc.).
- ** Cuando un proceso se este ejecutando localmente, ya que es necesario tener un monitoreo constante del mismo para poder visualizar los resultados al finalizar dicho proceso.
- ** Cuando se lleve a cabo la escritura o lectura de un archivo, por ejemplo en caso de salvar configuración, o leer archivo de molécula, salvar archivo de hoyos, salvar cambios en archivo, etc.

Es similar debido a que sólo será necesario verificar los procesos creados por necesidades específicas del sistema, como son los cálculos mayores solicitados, que son enviados a la CRAY.

9.2.1.3 Asignar subsistemas a procesadores y tareas

Asignar a procesadores no será necesario, ya que solo se dispondrá de uno, y por lo tanto todos los procesos locales serán realizados en una estación de trabajo.

Por otro lado los procesos remotos serán efectuados en una supercomputadora, debido a la gran capacidad que esta posee para realizar grandes cálculos, de esta forma, y una vez obtenidos los resultados, serán enviados a la estación de trabajo, para mostrar la visualización correspondiente.

9.2.1.3.1 Estimar los requisitos de recursos

El sistema de SVDM, utilizará sólo el procesador de la estación de trabajo local para la realización de la visualización, ya que solo leerá un archivo de datos generado previamente.

Para llevar a cabo los cálculos de la generación de archivos de configuración, que constituyen los cálculos mas pesados, se requerirá de enviar datos a la supercomputadora CRAY, la cual los recibirá y entregará los resultados luego de haber efectuado los cálculos necesarios.

Para la comunicación entre ambas partes, el sistema y la supercomputadora, sólo se requerirá de una conexión vía software, es decir, se aprovecharán los recursos de red ya disponibles, trabajando sobre plataforma UNIX, la cual nos permitirá la acción de entablar comunicación a través de "sockets", que son interfaces de comunicación que funcionan por software.

9.2.1.3.2 Balancear entre *hardware* y *software*

Como ya se mencionó, los recursos serán los siguientes:

Recursos de *hardware*

Equipo (*hardware*):

- ** Supercomputadora CRAY,
- ** Estación de trabajo Indigo 2 de SGI ó Estación de trabajo (PC) con Linux Red Hat versión 6.2.

La supercomputadora llevará a cabo los cálculos que requieran de mayores recursos (CPU, memoria, capacidad de almacenamiento temporal, etc.). Lo anterior será porque únicamente se cuenta con los recursos ya mencionados de equipo, y uno de los objetivos del sistema es obtener el mayor beneficio de los mismos.

Los procesos de los cálculos mas pesados, como son: Cálculo de dinámica, cálculo de hoyos, cálculo de potenciales, etc., se llevarán a cabo en la supercomputadora CRAY.

Relación de *software*

Programas y sistemas operativos (software)

Se trabajará sobre ambiente UNIX, es por eso que se usaran los siguientes recursos de Software:

- ** Sistema operativo UNIX, IRIX y LINUX específicamente.
- ** Lenguajes C y C++ para SO UNIX.
- ** Bibliotecas gráficas OpenGL o MESA, bibliotecas para el desarrollo de interfaces GTK (*Gimp ToolKit*) y VDK (*Visual Development Kit*) Builder.

9.2.1.3.3 Asignar las tareas a los procesadores

Dentro de este contexto tomaremos en cuenta que se utilizará sólo el procesador de los 2 equipos, es decir, el de la indigo2 o en su caso de una pc y el de la supercomputadora, este último lo tomaremos como procesador 1 y el de la estación de trabajo como procesador 2.

Tareas por procesador

** Procesador 1

Realización de cálculos remotos:

- Dinámica molecular.
- Potenciales.
- Generación de hoyos.
- Generar corte

** Procesador 2

Realización de cálculos locales:

- Visualización de molécula.
- Generación de archivo de rayos X.
- Generación de archivos de hoyos.
- Impresiones
- Generación de imágenes
- Editor molecular
- Generar vecinos
- Análisis de coordinación
- Generar distancias
- Generar ángulos

9.2.1.3.4 Determinar la conectividad física

Para entablar las conexiones físicas entre ambos equipos, se hará uso de la red, y por lo tanto no se requerirá de equipo adicional para tal fin, es decir, la comunicación se establecerá a través de comandos de sistema operativo UNIX.

Estos se utilizarán cuando el sistema requiera enviar o recibir datos a la supercomputadora, y serán monitoreados por el sistema a través del monitor de procesos, para saber cuando los resultados estén disponibles para que el sistema haga el uso adecuado de ellos.

9.2.1.4 Manejo de almacenamiento de datos

El manejo del almacenamiento de datos se realizará a través del uso de archivos, ya que el sistema lee los datos de los archivos de configuración para cada molécula, y así mismo los escribe, por ejemplo se tienen archivos de:

- ** Contenedor de partículas (archivo de configuración de partículas)
- ** Contenedor de hoyos (archivo de configuración de hoyos)
- ** Contenedor de potencial (archivo de configuración de potenciales)
- ** Contenedor de dipolos (archivo de configuración de dipolos)
- ** Contenedor de rayos X (archivo de configuración de rayos X)
- ** Contenedor de conexiones (archivo de configuración de conexiones)
- ** Contenedor de configuración de dinámica t (archivo de configuración de dinámica)
- ** Contenedor de configuración de dinámica t-dt
- ** Contenedor de inicia dinámica (archivo de configuración de inicia dinámica)
- ** Contenedor de imagen (archivo de imagen)

9.2.1.5 Escoger la implementación de control en software

Dentro del sistema, se realizará el control de todos los procesos y de todos los cálculos, a través de software. es decir, el sistema será implementado completamente por medio de software.

La metodología seleccionada para la realización del sistema será OMT con programación orientada a objetos, y será efectuada en C++, en una plataforma UNIX, IRIX ó LINUX específicamente.

El control se llevará a cabo de la siguiente manera:

Control de procesos:

Las clases serán

- Monitor de procesos
- Proceso
 - Proceso local
 - Proceso remoto

El control de la lectura de archivos se llevará a cabo con las clases de lectura de archivos de configuración:

- Objetos de contenedores

El control de la visualización será llevado a cabo por los objetos derivados de las clases gráficas.

9.2.1.6 Manejo de condiciones de borde

Moléculas demasiado grandes: Una de las condiciones de borde de mayor importancia que hay que contemplar dentro de nuestro sistema es el manejo de los archivos de configuraciones de moléculas que constan de un número muy grande de partículas, ya que es necesario definir un tamaño óptimo debido a las limitaciones de la estación de trabajo en la cual se ejecutara el sistema.

El SVDM, es un sistema impulsado por eventos, esto es, será manejado por eventos que se efectúan sobre la escena o la interfaz gráfica. El programa será ejecutado sin ningún parámetro, y enseguida el usuario indicará al programa el nombre del archivo, haciendo uso de las diferentes opciones que ofrecerá la interfaz gráfica del sistema.

Una vez efectuada la visualización, el sistema esperará a que el usuario realice un evento como el presionar un botón del ratón para llevar a cabo una acción, que puede ser, modificar la gráfica actual, o realizar la escritura de un nuevo archivo de partículas, etc.

En tanto el sistema no reciba alguna orden por parte del usuario, permanecerá visualizando la molécula seleccionada.

9.2.1.7 Decidir entre distintas prioridades

Prioridades:

En el caso del presente sistema la prioridad mayor es la de atender a la visualización gráfica, es decir, el proceso local de mostrar la molécula en tiempo real, éste proceso será tomado como el principal, y se tratará de mejorar la calidad de la misma, es decir, el desempeño de los procesos de graficación será la prioridad mayor dentro del sistema.

A continuación se tendrá la atención a los distintos eventos que sean realizados por parte del usuario como órdenes y comandos.

Por último, el sistema prestará la atención a los procesos que sean creados tanto local como remotamente, ya que estos constituirán las órdenes del usuario.

9.2.1.8 Arquitecturas

El sistema de visualización de dinámica molecular, funciona de manera similar a más de un tipo de arquitecturas:

Transformación continua: esto es por que el sistema recibe inicialmente un archivo de configuración, pero el funcionamiento del sistema cambia en función con las entradas por parte del usuario, así como también cambian los datos procesados y por tanto los resultados entregados por parte del sistema. Es decir, es un sistema en el cual las salidas dependen, de manera activa, de las entradas que cambian y son actualizadas frecuentemente.

Dentro de esta parte, se especifica que la definición de los objetos y clases fueron determinados después de haber realizado el *MODELO DE OBJETOS* del análisis del sistema.

Por otro lado, el *MODELO DINÁMICO* del análisis del sistema nos indica brevemente como funcionaría el sistema en casos definidos.

Finalmente, el comportamiento del sistema se refleja en los diagramas de flujo de datos del *MODELO FUNCIONAL* del análisis del sistema.

Interfaz Interactiva: Este tipo de arquitectura es tal vez la que más corresponde al SVDM, ya que éste siempre estará esperando a que el usuario realice alguna acción o haga una solicitud de proceso o de modificación a las características iniciales de la molécula.

Simulación dinámica: Por otra parte, el sistema será capaz de visualizar moléculas que cambian en el tiempo. esto se refiere a las moléculas que son animadas, es decir, configuraciones moleculares que tienen un comportamiento diferente de un instante a otro, el cual se simula a través de una pequeña animación.

La arquitectura de **administración de transacción** también corresponde a nuestro sistema de visualización, ya que dependiendo de las acciones que realice el usuario sobre una configuración molecular determinada, previamente leída e interpretada, será necesario que el sistema actualice los archivos de configuración o visualice algunas otras opciones (leer archivos), como la visualización de potenciales o la visualización de hoyos, rayos x, etc.

9.2.2 Diseño de objetos

Las clases, los objetos y sus relaciones fueron determinadas durante la realización del modelo de objetos, y en esta parte del diseño se definen los detalles de las mismas, así como también las asociaciones que hay entre ellas dentro del sistema.

Las clases definitivas son las siguientes:

CLASES		
1. Objeto gráfico A	18. Modificador de propiedades	38. Monitor de procesos
2. Objeto estático A	19. Escena gráfica	39. Proceso A
3. Caja	20. Estructura cristalina	40. Proceso local
4. Eje	21. Celda	41. Proceso remoto
5. Plano de corte	22. Proceso cálculo A	42. Impresiones A
6. Objeto dinámico A	23. Cálculo remoto	43. Impresión de archivo de configuración
7. Esfera A	24. Cálculo local	44. Impresión de imagen
8. Dipolo	25. Cálculo generado por partículas A	45. Puntero 3D
9. Átomo	26. Cálculo imagen	46. Editor molecular
10. Hoyo	27. Cálculo de magnitudes y ángulos	47. Contenedor de conexiones
11. Conexión	28. Cálculo de dinámica	48. Contenedor de rayos x
12. Plano potencial	29. Contenedor de datos A	49. Contenedor de configuración de dinámica
13. Objeto animado A	30. Contenedor de partículas	50. Contenedor de inicia dinámica
14. Sistema de cavidades (Hoyos)	31. Contenedor de hoyos	
15. Molécula	32. Contenedor de potencial	
16. Superficie potencial	33. Contenedor de dipolos	
17. Indicador de selección	34. Contenedor de imagen	
	35. Contenedor de configuración de dinámica t	
	36. Contenedor de configuración de dinámica t-dt	
	37. Contenedor configuración de inicia dinámica	

9.2.2.1 Combinar los tres modelos

En esta parte, se hace coincidir lo realizado en el análisis (modelos de objetos, dinámico y funcional) con la estructura del programa, es decir, a partir de ahora se traducirá de un diagrama que describe el comportamiento del sistema, así como el de cada uno de los objetos que forman parte de este, a su correspondiente organización física del programa final (se debe dar una interpretación a los diagramas de estado, y analizar más a fondo los DFD's).

El modelo de objetos, nos describe todas y cada una de las clases que formarán parte del sistema, es decir, las clases de las cuales se derivan cada uno de los elementos que forman el sistema, como son los objetos visuales: partículas y las conexiones, y los objetos abstractos, pero que forman parte del sistema y que auxilian o sirven de "recipientes" para los datos que requieren las operaciones del sistema, tales como los objetos contenedores de datos.

Los diagramas de flujo de datos representan las operaciones e interacciones existentes entre los elementos del sistema, es decir, son la descripción de las funciones de los objetos.

9.2.2.2 Diseñar los algoritmos

9.2.2.2.1 Escoger los algoritmos

Los algoritmos de los procesos mas importantes del sistema son los siguientes:

9.2.2.2.1.1 Algoritmos de visualización

9.2.2.2.1.1.1 Dibujar átomos

El primer proceso que se realiza en este módulo es el de la lectura del “archivo de visualización” que se obtuvo a partir de el programa de la dinámica molecular. Este es un archivo en código ASCII, para facilitar su manipulación.

El archivo de la molécula, necesaria para la visualización tiene el siguiente formato:

```

titulo
boxx boxy boxz gama
NC
NA
e x y z q t
e x y z q t

```

y cada variable se describe a continuación:

titulo: cadena de caracteres que identifica a la estructura que se visualiza. y sirve para distinguir entre los diferentes archivos de partículas y para poder saber las condiciones bajo las cuales se obtuvo el resultado.

Las variables reales: *boxx*, *boxy*, y *boxz*, definen el tamaño de la caja computacional y son utilizadas para iniciar ciertas variables que permiten observar a una distancia razonable toda la estructura, es decir, que con éstas variables se define la matriz de transformación que va a mapear las coordenadas cartesianas a coordenadas de pantalla.

gama, indica el ángulo de inclinación que tienen las paredes laterales de la caja con respecto al eje *x* y se utiliza para definir las características romboédricas.

Con la DM se obtienen las posiciones de los átomos en un instante de tiempo *T*. Pero si se toma un reporte de dichas posiciones en varios instantes de tiempo y se despliegan en pantalla, entonces se puede observar la evolución del comportamiento de las partículas de la estructura molecular durante el tiempo simulado, es decir, que será posible observar una animación de la molécula en estudio. Un número entero *NC*, indica cuántas muestras fueron tomadas en el tiempo para generar el archivo, es decir, indica el número de cuadros de la “filmación” del proceso.

Es necesario indicar al inicio de cada cuadro el número de átomos, *NA* que contiene dicho cuadro, esto se debe a que aunque el número de átomos en cada cuadro es constante, este puede variar cuando dentro de la DM se centra una esfera de un determinado radio de corte, lo que implica que al estar en movimiento los átomos, entran y salen constantemente de la esfera, con lo cual varía el número de partículas por cuadro.

La etiqueta *e*, es únicamente un valor entero que identifica cada átomo y sirve para diferenciarlos entre sí.

x, *y*, y *z* son las coordenadas en el espacio de los átomos dentro de la caja computacional.

La carga *q*, de las partículas se representa mediante el color de las esferas. Se utiliza un valor para hacer una correspondencia entre ésta y una gama de colores predefinida. Las unidades de la carga no son importantes ya que el color se obtiene mediante una interpolación entre los valores extremos de las cargas para cada átomo.

El tipo *t*, es un entero que indica el número atómico de la partícula.

Las unidades de longitud de la caja así como las de las posiciones de los átomos deben ser ángstroms (Å). Las coordenadas de las partículas tienen como referencia el centro de la caja computacional, es decir, que las coordenadas (0,0,0) equivalen al centro de la caja.

Los archivos deberán ser entregados, por la interfaz, al sistema de visualización de la DM. En caso de ser archivos de un formato no válido, el sistema producirá un aviso de error de formato.

Si el archivo cumple con las características del formato del sistema de visualización, entonces se vacía la información a las estructuras correspondientes, que son cada uno de los datos necesarios para la generación de los objetos de tipo átomo que serán creados en la memoria principal de la estación de trabajo.

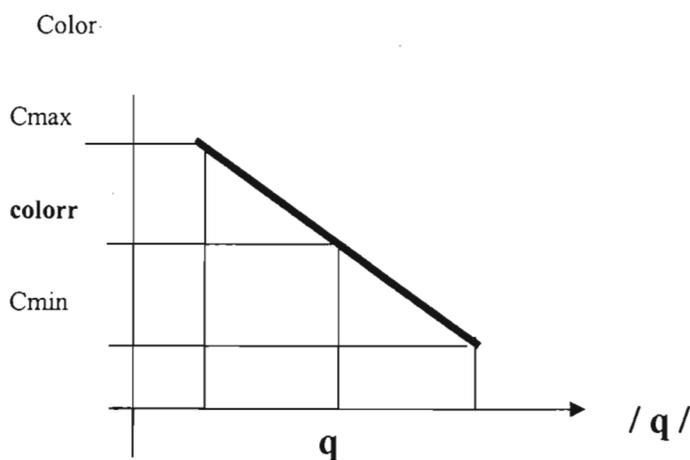
El sistema cuenta con una estructura (en un archivo de inclusión, *atomo.h*) donde están predefinidas las características gráficas de las partículas. Éstas últimas están ordenadas en función del número atómico, de tal forma que la variable *t* del archivo de visualización es el índice para acceder a la estructura. Esta estructura contiene para cada elemento de la tabla periódica la siguiente información:

```
char *nombre;
float radio, qmin, qmax;
short colorvar, colormax, colormin, colorr, colorg, colorb;
```

donde *nombre* es una cadena de 2 caracteres que definen el símbolo químico de cada átomo. El *radio* se especifica mediante un cifra en angstroms. *qmax* y *qmin* son los valores de carga extremos que se obtienen al leer el archivo de visualización para cada tipo de átomo. *colorvar* es el color que se va a variar (1 rojo, 2 verde, 3 azul), *colormax* y *colormin* son los valores que delimitan el intervalo en el cual varía *colorvar*, y *colorr*, y *colorg* y *colorb* que son las constantes de inicio del color de los átomos.

Después de leído el archivo de visualización, el sistema dibuja las partículas. Por cada partícula se dibuja una esfera centrada en las coordenadas señaladas por *x*, *y*, *z*, y con un radio que se obtiene en función de su número atómico. El color de la esfera se establece en función de la carga de la partícula.

Para obtener el color de la esfera, se hace una interpolación entre los límites de colores marcados en la estructura que contiene las características de las partículas, luego este valor es asignado a la componente de la terna *rgb* indicada por *colorvar*. Por ejemplo, si *colorvar* es 1 entonces las componentes *colorg* y *colorb* quedarán como constantes, y el valor de *colorr* variará entre las constantes *Cmax* y *Cmin* en función de su carga. Una forma simple de explicar cómo se obtiene el valor del color variable es por medio de la siguiente gráfica.



de esta manera, la componente *colorr* se obtendrá por medio de:

$$color = \frac{(Cmax - Cmin) * (qmax - q)}{(qmax - qmin)}$$

El propósito de variar sólo una de las componentes de la terna *rgb* es que las esferas de una misma especie se vean más brillantes o más opacas pero que no tengan un cambio brusco de color, puesto que se confundirían con otras especies

de partículas. De esta manera la partícula con más carga (positiva o negativa) tendrá una tonalidad más oscura que una con menor carga.

Obtenidas todas las características de la esfera (posición, radio y color), se dibuja por medio de una función de la biblioteca gráfica, la cual nos permite crear las esferas con la definición que deseemos. Este procedimiento se repite para todas las partículas y para todos los cuadros de la simulación.

9.2.2.2.1.1.2 Dibujar conexiones

Como parte importante del proceso de visualización, se encuentra la capacidad de ver las partículas entre sí en función de la distancia del enlace químico. De esta manera, podemos ver la “estructura de alambre” que proporciona valiosa información.

Gracias a este módulo, se puede observar la coordinación de ciertas partículas de interés, las cavidades, verificar distancias, etc. El proceso consiste en tomar cada elemento de la lista y la etiqueta de cada uno de sus vecinos. Conociendo la etiqueta de una partícula y las de sus vecinos, se obtienen las coordenadas, con lo que se procede a dibujar un elemento, en este caso un cilindro, para conectar a cada par de partículas.

Los cilindros se dibujan utilizando una función de `opengl(mesa)` y para colocarlos adecuadamente es necesario dibujarlos a partir del centro de una de las esferas a conectar y posteriormente orientarlos hacia las coordenadas de la otra esfera. Esto es debido a que la orientación inicial de todos los cilindros es hacia las coordenadas del eje Z, que corresponde a la posición desde la cual se encuentra el usuario.

9.2.2.2.1.1.3 Dibujar hoyos

Consiste en tomar la información generada por el módulo buscar hoyos, y dibujar esferas en la posición y con el tamaño indicados. Los hoyos se dibujan de color blanco con el objeto de distinguirlos de los átomos.

Dependiendo de las características de las estructuras a analizar, se observa una diferente cantidad de hoyos en las partículas. Si los hoyos generados en una estructura son demasiados, entonces al momento de visualizarlos junto con la estructura, se perderían detalles importantes. Es por esto que se hace uso de las propiedades de la esfera y se altera su transparencia para permitir seguir viendo en su totalidad la estructura y tener bien localizados los hoyos.

9.2.2.2.1.1.4 Dibujar curvas de potencial

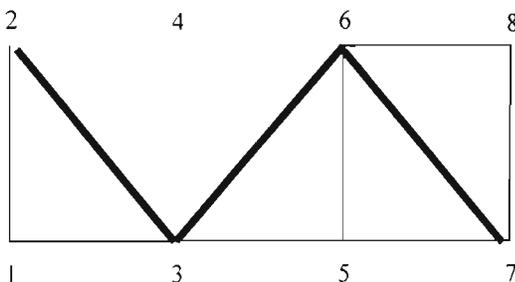
Los cálculos de potenciales son realizados por el módulo subsidiario de calcular potenciales, el cual entrega un archivo que se utiliza para unir los puntos formando una malla de distintos tonos, de esta forma se pueden ver planos con regiones que, dependiendo del color, darán información visual del valor del potencial electrostático dentro de la estructura. Los valores de potencial se transfieren a una matriz dinámica tridimensional con el propósito de tener acceso a la información identificando los planos de potencial por medio de índices. Debido a que las ternas del archivo de potenciales, el cual tiene la extensión “.pot”, van de la arista de coordenadas negativas a la arista con coordenadas positivas y a que el número de puntos de potencial es constante, es que en la matriz sólo se almacenan los valores de potencial sin sus coordenadas. Como se conocen las longitudes de la caja y el número de particiones que se le hicieron en cada lado para obtener los puntos de cálculo de potencial, se puede acceder a cualquier plano de potencial encontrando los índices correspondientes. Por ejemplo, si se desea observar el plano de potencial en $x=6.5$, entonces el índice x de la matriz se calcula por:

$$indx = \frac{(NPX * 6.5) + boxx / 2}{boxx}$$

donde NPX es el número de partículas sobre $boxx$ e $indx$ el redondeo del resultado de la ecuación.

El siguiente paso es hacer una “malla” donde se varían tanto la y como la z desde $-boxy/2$ a $boxy/2$ y desde $-boxz/2$ a $boxz/2$ respectivamente con x fija en 6.5. Se utiliza una malla triangular que es un conjunto de triángulos formados

por una serie de puntos. Por ejemplo, si tenemos un conjunto de 8 puntos y el punto 4 está marcado como punto sin potencial por estar situado dentro de una partícula, entonces la unión de los puntos sería la siguiente:



El color de la malla varía según el potencial en cada punto. El color para cada punto es evaluado en función de su potencial. Si el potencial es positivo entonces se interpola entre el azul (b) más oscuro para el potencial menos positivo y el potencial y el azul más claro para el mayor potencial. En caso de ser negativo el potencial, entonces el color se interpola entre rojo (r) oscuro para los menos negativos y rojo brillante para los más negativos. A continuación se resume este proceso:

$$v = 0$$

Si $pot > entonces$

$$r = 0$$

$$b = \frac{255}{(q_{max} - q_{min})} * (m [indx][y][z] - pot_{min})$$

en caso contrario

$$b = 0$$

$$r = \frac{255}{(q_{max} - q_{min})} * (pot_{min} - m [indx][y][z])$$

Al tener el valor del color en cada uno de los puntos, la función de la biblioteca gráfica que une los puntos, automáticamente llena el triángulo formado interpolando el color interno entre los tres colorees límites. Es decir, que automáticamente desvanece el color interno, píxel por píxel desde un extremo a los otros dos.

9.2.2.2.1.2 Algoritmos de realización de cálculos

9.2.2.2.1.2.1 Buscar vecinos

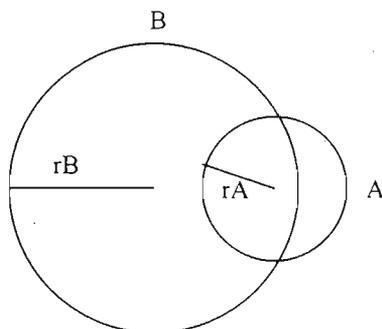
Este módulo genera nueva información a partir del archivo de visualización. Sin embargo, a diferencia de los otros módulos intermedios (hoyos, potenciales y volumen), éste es interactivo y solo se ejecuta en la estación de trabajo durante el proceso de visualización. Esto es debido a que, mientras que para una estructura promedio sólo existe un conjunto de hoyos que definen un espacio vacío, un conjunto de puntos de potencial electrostático y un volumen, por otro lado los vecinos de los átomos varían según el radio de conexión que se especifique.

El funcionamiento consiste en posicionarse en cada una de las partículas y en base a una distancia dada, llamada radio de conexión, comparar la distancia con el resto de las partículas de la estructura. De esta comparación, se seleccionan las que se encuentran a una distancia menor o igual al radio de conexión y se forma una lista dinámica para cada partícula. En la lista no se almacenan las posiciones de los vecinos por ser ésta una información redundante, y contiene únicamente las etiquetas de los átomos vecinos.

9.2.2.2.1.2.2 Buscar hoyos

El método para encontrar el conjunto de hoyos de una estructura, se basa en una sucesión de puntos aleatorios uniformemente distribuidos en el espacio, que es generada dentro de la caja computacional. Una esfera es centrada en cada

punto y su radio toma el valor de la distancia entre su centro y la superficie de la partícula más cercana. Si la nueva esfera que representa un hoyo está contenida en otro hoyo entonces se descarta, y si previamente existía un hoyo contenido en la nueva esfera, entonces este último se elimina, si ninguno de los casos anteriores sucede, se añade a la colección un nuevo hoyo y se actualiza. El criterio de contención empleado es el siguiente: Dados dos hoyos A y B. A está contenido en B si $d_{AB} < r_B$ y $r_A < r_B$, de la siguiente manera:



La generación de puntos aleatorios en el espacio, que cubren densamente el volumen de la caja, se realiza tomando tres números aleatorios sucesivos uniformemente distribuidos sobre el intervalo $[-L, L]$ donde L es el lado de la caja computacional. Si los números aleatorios son independientes, entonces los puntos del espacio constituidos por las triadas así formadas, cubren densamente el volumen de la caja que contiene a las partículas para un número suficientemente grande de triadas.

El número de triadas aleatorias que podrán ser centros de hoyos, está dado por la variable entera *alea*, cuyo valor generalmente se coloca en 20,000. Este valor fue obtenido por ensayo y error ya que, aproximadamente con 15,000 triadas aleatorias el algoritmo converge, es decir, con este número de hoyos, se llega a detectar un alto porcentaje del volumen libre contenido en la estructura.

Aunque el algoritmo continúa encontrando hoyos al aumentar el número de triadas aleatorias, los hoyos que se localizan constituyen pequeñas esferas que quedan en los intersticios, son demasiado pequeños y no son de gran interés.

El módulo tiene la opción de obtener el conjunto de hoyos de la estructura con condiciones de periodicidad ya se en dirección de uno, dos o de los tres ejes. Las condiciones de periodicidad se refieren a tomar a la estructura no como una estructura aislada, sino que se repite infinitamente, lo cual provoca que el criterio para asignarle una longitud a un hoyo cambie. En realidad, para encontrar los hoyos con condiciones de periodicidad en las tres direcciones de una estructura, sólo importaría las cajas adyacentes, para lo cual se toman las imágenes de las partículas en las cajas vecinas.

El archivo de salida que genera tiene el mismo nombre que el archivo de entrada, más la extensión .hoyo. El contenido de éste es una lista de las coordenadas y radios de cada hoyo encontrado en la estructura.

9.2.2.2.1.2.3 Calcular volumen y densidad

El volumen de un conjunto de átomos encerrados en una caja imaginaria, se obtiene midiendo el volumen de una sustancia que cabe dentro de la estructura, para así conocer el volumen que cubren las partículas y la densidad de la estructura. Con el fin de poder comparar los resultados de volumen y densidad experimentales, con los obtenidos en la simulación, es que se hace el cálculo sobre la estructura de poros, pretendiendo que los hoyos representan las especies absorbidas que pueden ser argón, nitrógeno o mercurio.

El procedimiento que se realiza para obtener el volumen vacío de la estructura consiste en partir la caja computacional en miles de pequeños prismas rectangulares, los cuales son considerados e incrementados al volumen vacío cuando su centro se encuentra dentro de al menos un hoyo.

$$\lim_{n \rightarrow \text{alea}} V_{\text{hoyos}}^n = V_{\text{vacío}}$$

Después de la generación de 200,000 puntos aleatorios el volumen no aumenta substancialmente no obstante que se encuentran más hoyos. Esto se debe a que los hoyos encontrados son muy pequeños y su volumen es casi despreciable. El algoritmo encuentra entre el 85 y 90 por ciento del volumen total dependiendo de la porosidad de la estructura.

9.2.2.2.1.3 Algoritmos de manejo de datos

9.2.2.2.1.3.1 Obtención de datos

El usuario puede proporcionar el nombre del archivo de datos al sistema a través de la interfaz gráfica. Una vez que el nombre de archivo ha sido proporcionado, el sistema procede a verificar si este existe, así como su formato, y si el archivo es correcto, en seguida lo abre, para comenzar a procesar los datos, si por el contrario no existe entrega un mensaje de error de archivo, ya sea si el archivo no existió, o si es de un formato incorrecto.

9.2.2.2.1.3.2 Lectura de archivos

Cuando el sistema conoce el nombre del archivo a abrir, procede a verificar el tipo del archivo, y su formato. Si el tipo de archivo y el formato son correctos, el sistema lee los datos, si no es así, se deberá traducir el formato, en caso de que no sea posible traducirlo, debido a que es de un formato desconocido, se entrega un mensaje de error, indicando que el formato es desconocido.

9.2.2.2.1.3.3 Almacenamiento de datos en memoria

Una vez que el sistema ha leído el archivo, se procede a colocar los datos en el o los vectores correspondientes:

- si es **archivo de configuración de partículas**, hoyos, en el vector de partículas.
- si es **archivo de configuración de hoyos**, en el vector de hoyos.
- si es **archivo de configuración de dipolos**, en el vector de dipolos.
- si es **archivo de configuración de rayos x**, en el vector de rayos x.
- si es **archivo de configuración de potenciales**, en el vector de potenciales.

ó entrega un vector de datos de configuración al subsistema de **realización de gráficas**.

ó entrega un vector de datos de configuración a **monitor de procesos**.

ó entrega el **archivo de configuración al usuario**

ó entrega el **archivo de configuración a monitor de procesos**.

Así mismo, cuando se requiere, el sistema entrega un vector de datos de configuración al subsistema de **realización de impresiones**.

9.2.2.2.1.3.4 Renombrar archivo de configuración

Para renombrar el archivo de configuración, primero el usuario elige el archivo correspondiente. Enseguida el sistema solicita al **usuario** el nuevo nombre del archivo y una vez que lo tiene verifica su sintaxis, para que sea un nombre aceptado por el sistema operativo.

Si es un nombre válido, el sistema cambia el nombre anterior por el nuevo nombre. Por el contrario, si no es válido, el sistema entrega un mensaje de error y solicita otra vez el nombre.

9.2.2.2.1.3.5 Guardar archivo de configuración

Primero el **usuario** elige el **archivo de configuración** a guardar, posteriormente el **sistema** guarda los datos en el **archivo de configuración** en disco, y finalmente notifica al **usuario** sobre el resultado de la operación.

9.2.2.2.1.3.6 Guardar con otro nombre archivo de configuración

El **usuario** elige la configuración de datos para guardarla en un **archivo de configuración** con otro nombre. Una vez realizado esto, el **sistema** solicita al **usuario** que introduzca un nuevo nombre para el **archivo de configuración**.

El **sistema** valida la sintaxis del nombre especificado y si no es válido vuelve a solicitarlo, si el nombre es válido, el **sistema** crea un nuevo **contenedor de datos**, en el cual va a copiar los datos de la configuración. Este nuevo **contenedor de datos** cambia el nombre a su **archivo de configuración** asociado.

El nuevo **contenedor de datos** guarda los datos en su **archivo de configuración**. Mientras el **contenedor de datos** original cierra su archivo de configuración sin guardar los cambios. Una vez realizado esto, el **sistema** elimina el **contenedor de datos** original de la memoria

Finalmente el **sistema** indica al **usuario** que la operación de guardar con otro nombre se llevó a cabo.

Los procesos de salvar los archivos de configuración son los mismos para todo tipo de archivo de configuración, lo único que cambia son los datos a almacenar.

9.2.2.2.1.4 Algoritmos de impresiones

El proceso de impresiones, es el siguiente: el usuario seleccionará lo que desea imprimir, es decir, si selecciona una impresión de una imagen actual, el sistema almacenará la información en una estructura con el formato de imagen que se predefina, y estos datos serán entregados al proceso de impresiones para realizar la impresión de la imagen, opcionalmente, el sistema convertirá el área seleccionada de la pantalla, en algún formato de imagen (gif, jpg, etc), para guardarla en disco.

Así mismo, el proceso de impresión de archivo de configuración, será similar: el proceso de realización de impresiones solicitará los datos del archivo especificado al contenedor de archivo de configuración correspondiente y con ellos la impresión será llevada a cabo.

9.2.2.2.2 Escoger estructuras de datos

Se requiere de una estructura de objetos de tipo átomos, la cual es una lista ligada, para facilitar el desplazamiento entre las diferentes partículas en caso de que se requiera buscar específicamente alguna de ellas.

De igual manera se requiere de estructuras similares para los objetos de tipo: Hoyos, dipolos, conexiones.

Se utilizarán estructuras de datos genéricas (*Template* en C++), según sea necesario, y para mayor compatibilidad de datos.

9.2.2.2.3 Definir clases internas y operaciones

Por el momento, no es necesario, ya que las clases y operaciones que fueron definidas en el análisis son las necesarias.

Las operaciones agregadas durante la fase de diseño son las necesarias para especificar el intercambio de los mensajes y valores requeridos por las diferentes clases.

Por ejemplo es necesario que haya operaciones para asignar un nuevo valor cuando esto sea requerido, o en su defecto deberá haber solo una operación para asignar los nuevos valores, dependiendo si al registrarse un cambio en un determinado valor, se deben modificar algunos otros.

De modo que se agregarán las operaciones, solo cuando sea conveniente.

9.2.2.2.4 Asignar responsabilidades para operaciones

En esta parte, los métodos fueron reducidos utilizando la herencia, llegando a eliminarse los métodos repetidos en algunas clases que se derivan de una misma clase base.

9.2.2.2.4.1 Módulo de visualización (Jerarquía de clases gráficas)

Nota: los nombres de las clases NO llevan acento debido a que es el nombre con el cual quedarán al codificar los objetos, es decir en lenguaje C++

Nombre de la clase :	Objeto_grafico
Instancias a métodos :	
	<ul style="list-style-type: none"> ** Ajustar configuración al tipo de caja: Ésta función se encarga de adecuar las características de la configuración actual y la disposición de los objetos dentro de la caja, según el tipo especificado para ésta.
Relaciones de herencia :	
	Clases derivadas de Objeto_grafico: Objeto_estatico Objeto_dinamico Plano_potencial

Nombre de la clase :	Objeto_estatico
Instancias a métodos :	
	<ul style="list-style-type: none"> ** Cambiar grosor de línea: Se encarga de modificar el grosor de la línea. ** Cambiar tipo de línea: Se encarga de modificar el tipo de la línea.
Relaciones de herencia :	
	Clases derivadas de Objeto_estatico: Caja Eje Clase base: Objeto_grafico

Nombre de la clase :	Eje
Instancias a métodos :	
<ul style="list-style-type: none"> ** Pinta eje: Ésta operación se encarga de dibujar un eje en el espacio tridimensional. ** Acceder a propiedades: Con ésta operación se puede acceder a los valores de las propiedades del objeto. ** Asignar propiedades: Con ésta operación se pueden modificar los valores de las propiedades del objeto. 	
Relaciones de herencia :	
Clase base: Objeto_estatico	

Nombre de la clase :	Caja
Instancias a métodos :	
<ul style="list-style-type: none"> ** Pinta caja: Ésta operación se encarga de dibujar la caja en el espacio 3D. ** Seleccionar tipo de caja: Se encarga de seleccionar el tipo de caja que será dibujada. ** Acceder a propiedades: Con ésta operación se puede acceder a los valores de las propiedades del objeto. ** Asignar propiedades: Con ésta operación se pueden modificar los valores de las propiedades del objeto. 	
Relaciones de herencia :	
Clase base: Objeto_estatico	

Nombre de la clase :	Objeto_dinamico
Instancias a métodos :	
<ul style="list-style-type: none"> ** Marcar por conjunto: Ésta propiedad se encarga de marcar a las partículas de una determinada región del área de trabajo. ** Marcar individual: Ésta función se encarga de marcar las partículas individualmente. ** Indicar tipo de selección: Ésta función se encarga de especificar de que manera se va a realizar la selección de objetos. ** Cambiar forma de visualización: Ésta función se encarga de modificar la manera en que se van a dibujar los objetos en la escena gráfica. ** Indicar estado de modificación: Sirve para informar el estado de modificación del objeto. 	
Relaciones de herencia :	
Clases derivadas de Objeto_dinamico:	
Esfera	
Conexión	
Clase base: Objeto_grafico	

Nombre de la clase :	Conexion
Instancias a métodos :	
<ul style="list-style-type: none"> ** Generar conexión inicial: Ésta función es el constructor de la clase. ** Pintar visualización de conexión: Sirve para dibujar la conexión en el espacio 3D. ** Liberar conexión de memoria: Ésta función es el destructor de la clase. ** Acceder a propiedades: Con ésta operación se puede acceder a los valores de las propiedades del objeto. ** Asignar propiedades: Con ésta operación se pueden modificar los valores de las propiedades del objeto. 	
Relaciones de herencia :	
Clase base: Objeto_dinamico	

Nombre de la clase :	Esfera
Instancias a métodos :	
<ul style="list-style-type: none"> ** Pinta esfera: Ésta operación se encarga de dibujar la esfera en el espacio tridimensional. 	
Relaciones de herencia :	
Clases derivadas de Esfera:	
Atomo	
Dipolo	
Hoyo	
Clase base: Objeto_dinamico	

Nombre de la clase :	Dipolo
Instancias a métodos :	
<ul style="list-style-type: none"> ** Generar dipolo inicial: Ésta función es el constructor de la clase. ** Pinta visualización de dipolo: Ésta operación se encarga de dibujar el dipolo en el espacio tridimensional. ** Acceder a propiedades: Con ésta operación se puede acceder a los valores de las propiedades del objeto. ** Asignar propiedades: Con ésta operación se pueden modificar los valores de las propiedades del objeto. ** Liberar dipolo de la memoria: Ésta función es el destructor de la clase. 	
Relaciones de herencia:	
Clase base: Esfera	
Tiene Comunicación con:	
Editor_molecular	
Relaciones con otras clases:	
Es parte de Molecula	

Nombre de la clase :	Atomo
Instancias a métodos :	
<ul style="list-style-type: none"> ** Generar átomo inicial: Ésta función es el constructor de la clase. ** Pintar visualización de átomo: Ésta operación se encarga de dibujar el átomo en el espacio ** Acceder a propiedades: Con ésta operación se puede acceder a los valores de las propiedades del objeto. ** Asignar propiedades: Con ésta operación se pueden modificar los valores de las propiedades del objeto. ** Liberar átomo de memoria: Ésta función es el destructor de la clase. 	
Relaciones de herencia:	
Clase base: Esfera	
Relaciones con otras clases:	
Es parte de Molecula	

Nombre de la clase :	Hoyo
Instancias a métodos :	
<ul style="list-style-type: none"> ** Generar hoyo inicial: Ésta función es el constructor de la clase. ** Pintar visualización de hoyo: Ésta operación se encarga de dibujar el hoyo en el espacio. ** Acceder a propiedades: Con ésta operación se puede acceder a los valores de las propiedades del objeto. ** Asignar propiedades: Con ésta operación se pueden modificar los valores de las propiedades del objeto. ** Liberar hoyo de memoria: Ésta función es el destructor de la clase. 	
Relaciones de herencia:	
Clase base: Esfera	
Relaciones con otras clases:	
Es parte de Sistema_de_cavidades	

Nombre de la clase :	Plano_potencial
Instancias a métodos :	
<ul style="list-style-type: none"> ** Pinta campos escalares: Ésta función se encarga de pintar los campos escalares en la escena gráfica. ** Acceder a propiedades: Con ésta operación se puede acceder a los valores de las propiedades del objeto. ** Asignar propiedades: Con ésta operación se pueden modificar los valores de las propiedades del objeto. 	
Relaciones de herencia:	
Clase base: Objeto_grafico	
Relaciones con otras clases:	
Es parte de Superficie_potencial	

Nombre de la clase :	Objeto_animado
Instancias a métodos :	<ul style="list-style-type: none"> ** Detener animación en cualquier cuadro: Ésta función se encarga de detener la animación del objeto animado. ** Invertir sentido de la animación: Ésta función se encarga de cambiar el sentido de la animación del objeto animado. ** Encender y apagar: Indica cuando se visualiza o no el objeto animado. ** Recibir estado de animación: Obtiene el estado actual de animación del objeto. ** Cambiar estado de animación: Sirve para modificar el estado actual de la animación. ** Proporcionar estado de animación: Notifica el estado actual de la animación. ** Indicar tipo de selección: Sirve para notificar el tipo de la selección a realizar. <p>Clases derivadas de Objeto_animado</p> <ul style="list-style-type: none"> Sistema_de_cavidades Molecula Superficie_potencial

Nombre de la clase :	Sistema_de_cavidades
Instancias a métodos :	<ul style="list-style-type: none"> ** Recibir vector de hoyos: Ésta función se encarga de recibir el vector de datos de hoyos. ** Generar sistema de cavidades inicial: Ésta función es el constructor de la clase. ** Pintar visualización de cavidades: Se encarga de dibujar los hoyos en la escena gráfica. ** Cambiar propiedades: Se encarga de modificar las siguientes propiedades del sistema de cavidades: <ul style="list-style-type: none"> o Cambiar color a hoyos: Se encarga de proporcionar un color diferente a los hoyos, o Cambiar transparencia a hoyos: Se encarga de modificar el grado de transparencia de los hoyos. o Cambiar calidad a hoyos: Se encarga de modificar la calidad de los hoyos. o Cambiar slices y stacks a hoyos: sirve para modificar la manera de visualizar los hoyos, ya sea en stacks o en slices. ** Devolver vector de hoyos: Ésta función devuelve el vector con los datos de hoyos. ** Liberar sistema de cavidades de memoria: Ésta función es el destructor de la clase. <p>Relaciones de herencia:</p> <p>Clase base: Objeto_animado</p> <p>Relaciones con otras clases:</p> <p>Tiene objetos de la clase Hoyo</p>

Nombre de la clase :	Molécula
-----------------------------	-----------------

Instancias a métodos :

- ** Recibir vector de partículas: Ésta función recibe los datos de las partículas.
- ** Recibir vector de dipolos: Ésta función recibe los datos de los dipolos.
- ** Recibir vector de conexiones: Ésta función recibe los datos de las conexiones.
- ** Generar Molécula inicial: Es el constructor de la clase.
- ** Pintar visualización de molécula: Realiza la visualización de la molécula seleccionada.
- ** Fraccionar molécula: Ésta función se encarga de dividir la molécula para manipular las fracciones por separado.
- ** Mover molécula: Desplazar la molécula de posición original.
- ** Conectar átomos:
 - o Lee tabla de conectividad: Ésta función lee la configuración de la tabla de conectividad.
 - o Valida conexiones: Ésta función verifica que sea una conexión válida.
 - o Establecer tipo de conexión: Ésta función define el tipo de conexión, (por especie, no por especie, sin conexión)
- ** Modificar lista de átomos: Ésta función se encarga de cambiar las propiedades de los átomos de la molécula.
- ** Modificar lista de dipolos: Ésta función se encarga de cambiar las propiedades de los dipolos de la molécula.
- ** Modificar lista de conexiones: Ésta función se encarga de cambiar las propiedades de los conexiones de la molécula.
- ** Unir molécula: Se encarga de unir moléculas.
- ** Devolver vector de partículas: Se encarga de entregar el vector de datos de partículas a los objetos que lo requieran.
- ** Devolver vector de dipolos: Se encarga de entregar el vector de datos de partículas a los objetos que lo requieran.
- ** Devolver vector de conexiones: Se encarga de entregar el vector de datos de partículas a los objetos que lo requieran.
- ** Cambiar estado de edición: Se encarga de modificar el estado de edición de la molécula actual.
- ** Cambiar propiedades:
 - o Modificar radio de átomos: Se encarga de modificar el valor del radio de los átomos.
 - o Cambiar color a objetos: Se encarga de modificar la propiedad del color de los objetos.
 - o Cambiar transparencia a objetos: Sirve para modificar la propiedad de transparencia de los objetos.
 - o Cambiar calidad a objetos: Sirven para modificar la propiedad de calidad de los objetos.
 - o Cambiar slices y stacks a objetos: Cambia el valor de las propiedades "slices" y "stacks" de los objetos (objetos derivados de la clase esfera)
- ** Indicar estado de modificación: Sirve para notificar el estado de modificación de los objetos seleccionados.
- ** Liberar molécula de memoria: Es el destructor de la clase.

Relaciones de herencia:

Clase base: Objeto_animado

Relaciones con otras clases:

Tiene objetos de la clase Atomo, Dipolo y Conexion

Nombre de la clase :	Superficie_potencial
Instancias a métodos :	
<ul style="list-style-type: none">** Recibir vector de potenciales: Se encarga de recibir el vector de datos de potenciales.** Generar superficie potencial inicial: Es el constructor de la clase.** Pintar visualización de superficie potencial: Se encarga de dibujar la superficie de potencial.** Liberar superficie potencial de memoria: Es el destructor de la clase.** Acceder a propiedades: Con ésta operación se puede acceder a los valores de las propiedades del objeto.** Asignar propiedades: Con ésta operación se pueden modificar los valores de las propiedades del objeto.	
Relaciones con otras clases:	
Tiene objetos de la clase Plano_potencial	

Nombre de la clase :	Estructura_cristalina
Instancias a métodos :	
<ul style="list-style-type: none">** Multiplicar celda: Realiza los cálculos de multiplicar una celda base o celda unitaria.** Crear estructura cristalina : Es el constructor de la clase.** Pintar estructura cristalina: Se encarga de dibujar las repeticiones de la celda base o unitaria.** Cambiar factores de multiplicación: Permite modificar los factores por los que se multiplica una celda.** Liberar estructura de la memoria: Es el destructor de la clase.	
Relaciones con otras clases:	
Celda Superficie_potencial	

Nombre de la clase :	Escena_grafica
Instancias a métodos :	
<ul style="list-style-type: none"> ** Generar escena gráfica inicial: Es el constructor de la clase. ** Modificar objetos de escena gráfica: <ul style="list-style-type: none"> o Modificar molécula: <ul style="list-style-type: none"> ▪ Cambiar radio a todos los átomos de una molécula: sirve para modificar el radio de todos los átomos de una molécula. ▪ Cambiar color solo a objetos seleccionados: sirve para modificar la propiedad de color de los objetos seleccionados. ▪ Cambiar transparencia a todos los hoyos en una molécula: sirve para modificar la propiedad de transparencia de los hoyos de una molécula. ▪ Cambiar calidad a objetos seleccionados: sirve para modificar la propiedad de calidad de los objetos seleccionados. ▪ Cambiar slices y stacks a objetos seleccionados: sirve para modificar las propiedades de "slices" y "stacks" de los objetos (objetos derivados de la clase esfera). o Modificar ejes: Se encarga de cambiar los ejes de la escena gráfica. o Modificar puntero 3D: Sirve para modificar la forma del puntero 3D. o Modificar fondo: Sirve para cambiar el color del fondo de la escena gráfica. ** Deshacer cambios hechos en la escena: Sirve para deshacer los cambios realizados en la escena gráfica. ** Rehacer cambios: Sirve para volver a efectuar los cambios de la escena gráfica. ** Cambiar perspectiva: Intercambia la escena entre perspectiva o vista ortogonal. ** Asignar estado de animación: Indica el estado de animación seleccionado al objeto animado. ** Cambiar transformación espacial <ul style="list-style-type: none"> o Rotación: Realiza una rotación en la molécula o Traslación: Realiza traslación en la molécula o Escalación: Realiza un escalamiento en la molécula ** Visualizar ejes: Ésta función se encarga de dibujar los ejes en la escena gráfica. ** Visualizar estructura cristalina: Se encarga de dibujar la estructura cristalina. ** Visualizar puntero 3D: Ésta función se encarga de dibujar el puntero 3D en la escena gráfica. ** Liberar escena gráfica de la memoria: Es el destructor de la clase. 	
Relaciones con otras clases:	
<ul style="list-style-type: none"> Tiene objetos de la clase <ul style="list-style-type: none"> Eje Estructura_cristalina Puntero_3D 	

Nombre de la clase :	Celda
Instancias a métodos :	
<ul style="list-style-type: none"> ** Generar celda inicial: Es el constructor de la clase. ** Pintar celda: Se encarga de dibujar la celda en la escena gráfica. ** Liberar celda de la memoria: Es el destructor de la clase. 	
Relaciones con otras clases:	
<ul style="list-style-type: none"> Tiene objetos de la clase <ul style="list-style-type: none"> Molecula Caja Sistema_de_cavidades 	

Nombre de la clase :	Modificador_de_propiedades
Instancias a métodos :	
<ul style="list-style-type: none"> ** Identificar propiedad y objeto: Se encarga de identificar a que objeto se refiere. ** Cambiar propiedad: Se encarga de modificar la propiedad indicada ** Deshacer cambios: Se encarga de deshacer las operaciones realizadas en la escena. ** Rehacer cambios: Se encarga de repetir las operaciones realizadas en la escena. ** Notificar a Editor: Se encarga de informar al editor molecular de todos los cambios y estados de los cambios. 	
Relaciones con otras clases:	
Todas las clases gráficas	

Nombre de la clase :	Indicador_de_seleccion
Instancias a Métodos :	
<ul style="list-style-type: none"> ** Marcar objetos seleccionados: Realiza el marcado de los objetos o el área seleccionada. ** Identificar objetos seleccionados: Se encarga de identificar los objetos que fueron seleccionados. ** Obtener lista de objetos seleccionados: Genera una lista de los objetos seleccionados. ** Obtener lista de etiquetas de objetos seleccionados: Reúne las etiquetas de los objetos seleccionados. ** Regresar lista de objetos seleccionados: Entrega la lista generada de objetos seleccionados. ** Regresar lista de etiquetas de objetos seleccionados: Se encarga de devolver la lista que contiene las etiquetas de los objetos seleccionados. 	
Relaciones con otras clases:	
Todas las clases gráficas.	

9.2.2.2.4.2 Módulo de clases de cálculos

Nombre de la clase :	Proceso_calculo
Instancias a métodos :	
<ul style="list-style-type: none"> ** Generar vector de datos: Genera los datos del cálculo indicado 	
Relaciones de herencia:	
Clases derivadas de Proceso_calculo	
Calculo_de_imagen	
Calculo_de_dinamica	
Calculo_generado_por_particulas	
Calculo_de_magnitudes_y_angulos	
Calculo_local	
Calculo_remoto	

Nombre de la clase :	Calculo_de_imagen
Instancias a métodos :	
<ul style="list-style-type: none"> ** Generar imagen: Ésta función se encarga de realizar el cálculo para la obtención de una imagen de la molécula, partir de un área seleccionada (algoritmo de calculo de imagen). ** Devolver vector de imagen: Esta función entrega un vector de datos de la imagen calculada, a las clases que lo soliciten (contenedores de impresión, o de archivos). 	
Relaciones de herencia:	
Clases derivadas de Calculo_de_imagen	
ninguna	

Nombre de la clase :	Calculo_de_dinamica
Instancias a métodos :	
<ul style="list-style-type: none"> ** Generar dinámica: Ésta función se encarga de enviar los archivos requeridos para la generación del archivo de partículas, estos archivos son el archivo de configuración de dinámica y el de configuración de inicia dinámica. ** Devolver archivos de datos: Se encarga de devolver el archivo de datos, que contiene los datos obtenidos por la función generar dinámica. 	
Relaciones de herencia:	
Clases derivadas de Calculo_de_dinamica	
Ninguna	

Nombre de la clase :	Calculo_generado_por_particulas
Instancias a métodos :	
<ul style="list-style-type: none"> ** Obtener vector de partículas: Recibe el vector de datos de partículas ** Devolver vector de datos: Entrega el vector de datos de partículas 	
Relaciones de herencia:	
Clases derivadas de Calculo_generado_por_particulas	
Ninguna	

Nombre de la clase :	Calculo_de_magnitudes_y_angulos
Instancias a métodos :	
<ul style="list-style-type: none"> ** Generar distancias: Ésta función calcula las distancias solicitadas entre los objetos seleccionados. ** Generar ángulos: Ésta función se encarga de calcular los ángulos entre los objetos seleccionados. ** Mostrar resultados: Sirve para mostrar los resultados de los cálculos de magnitudes o ángulos. 	
Relaciones de herencia:	
Clases derivadas de Calculo_de_dinamica	
Ninguna	

Nombre de la clase :	Calculo_local
Instancias a métodos :	
<ul style="list-style-type: none">** Generar rayos x: Ésta función se encarga de realizar los cálculos de rayos x de la molécula especificada.** Generar vecinos: Ésta función se encarga de realizar los cálculos de rayos x de la partícula especificada.** Generar análisis de coordinación: Ésta función se encarga de realizar los cálculos del análisis de coordinación de la molécula especificada.** Generar archivo de datos: Se encarga de almacenar los datos generados por alguno de los cálculos locales, por si es requerido posteriormente.	
Relaciones de herencia:	
Clases derivadas de Calculo_Local	
Ninguna	

Nombre de la clase :	Calculo_remoto
Instancias a métodos :	
<ul style="list-style-type: none">** Generar dipolos: Se encarga de calcular los datos de dipolos** Generar hoyos: Se encarga de calcular los datos de hoyos** Generar campo escalar: Se encarga de calcular los datos de campo escalar** Generar corte: Ésta función se encarga de realizar un corte en la escena gráfica para mostrar las partículas que se encuentren sobre ese plano.	
Nota: Esta clase hace un llamado a un programa externo	
Relaciones de herencia:	
Clases derivadas de Calculo_remoto	
Ninguna	

9.2.2.2.4.3 Módulo de clases de monitor de procesos

Nombre de la clase :	Monitor_de_procesos
Instancias a métodos :	<ul style="list-style-type: none"> ** Generar proceso: Se encarga de crear el nuevo proceso. ** Asociar cálculo con proceso: Sirve para indicarle al proceso el tipo de cálculo que realizará. ** Asigna tipo de proceso: Le indica al proceso si es un proceso Local o Remoto. ** Identificar procesos residentes: Función que realiza el monitoreo de los procesos que se están ejecutando. ** Activar proceso: Realiza la activación del proceso indicado <ul style="list-style-type: none"> o Añadir proceso a tabla de procesos: Coloca el identificador de proceso en una tabla de procesos residentes. ** Obtener estado del proceso: Se encarga de obtener el estado actual del proceso especificado. ** Desplegar estado del proceso: Visualiza el estado actual del proceso indicado. ** Suspender proceso: Se encarga de suspender el proceso indicado. ** Avisos del proceso: Se encarga de manejar los avisos del proceso actual. ** Cerrar proceso: Se encarga de eliminar el proceso de la tabla, así como de terminarlo.
Relaciones con otras clases:	
Tiene objetos del tipo:	
	Proceso_local
	Proceso_remoto

Nombre de la clase :	Proceso
Instancias a métodos :	<ul style="list-style-type: none"> ** Generar proceso inicial: Es el constructor de la clase. ** Generar estado del proceso: Se encarga de entregar el estado actual del proceso ** Obtener identificadores del proceso: Obtiene el o los números identificadores del proceso ** Ejecutar cálculo asociado: Ejecuta el cálculo correspondiente al proceso. ** Esperar resultados de cálculo: Monitorea el cálculo para recibir los resultados. ** Recibir datos de cálculo asociado: Obtiene los datos generados por el cálculo correspondiente. ** Suspender cálculo: Detiene la ejecución del calculo asociado. ** Notificar mensajes a Monitor: Sirve para Indicarle al monitor de procesos el estado de todos los procesos actuales, así como los mensajes de error generados. ** Mostrar avance y estado de cálculo asociado: Indica gráficamente el grado de avance de un cálculo así como su estado actual. ** Liberar proceso de memoria: Es el destructor de la clase.
Relaciones de herencia:	
Clases derivadas de Proceso	
	Proceso_local
	Proceso_remoto

Nombre de la clase :	Proceso_remoto
Instancias a métodos :	<ul style="list-style-type: none"> ** Ejecutar cálculo remoto: Función que ejecuta el cálculo o programa que se encuentra en una máquina remota. ** Enviar Archivos de Entrada y de Salida: Se encarga de enviar los archivos a la máquina donde se va a realizar el proceso remoto. ** Esperar resultados de cálculo: Espera la terminación exitosa de un cálculo remoto para la obtención de los resultados. ** Recibir archivos de entrada y salida: Se encarga de recibir los archivos de la máquina donde se realizó el proceso remoto. ** Suspender cálculo: Sirve para detener el cálculo por un tiempo determinado. ** Abortar cálculo: Sirve para cancelar la ejecución del cálculo asociado.
Relaciones de herencia:	Clases derivadas de Proceso_remoto

Nombre de la clase :	Proceso_local
Instancias a métodos :	<ul style="list-style-type: none"> ** Ejecutar cálculo local: Es la función que ejecuta el cálculo o programa externo que se encuentra en la misma máquina del sistema. ** Enviar vector de datos de entrada: Se encarga de enviar los datos de entrada al cálculo correspondiente. ** Esperar resultados de cálculo: Espera la terminación exitosa de un cálculo local para la obtención de los resultados. ** Recibir vector de datos de salida: Se encarga de recibir los datos del cálculo realizado. ** Suspender cálculo: Sirve para detener el cálculo por un tiempo determinado. ** Abortar cálculo: Sirve para cancelar la ejecución del cálculo asociado.
Relaciones de herencia:	Clases derivadas de Proceso_local

9.2.2.2.4.4 Módulo de clases de impresiones

Nombre de la clase :	Impresiones
Instancias a métodos :	<ul style="list-style-type: none"> ** Identificar tipo de documento: Se encarga de obtener el tipo del documento a imprimir. ** Verificar estado de impresora: Se encarga de ver si el estado de la impresora es idóneo para imprimir. ** Poner en línea impresora: Se encarga de establecer la comunicación con la impresora. ** Recibir vector de datos (T) : Recibe los datos que se van a imprimir. ** Notificar resultado de impresión: Se encarga de informar el resultado de la impresión. ** Notificar tipo de error: Se encarga de informar el tipo de problema que ha ocurrido.
Relaciones de herencia:	Clases derivadas de Impresiones Impresion_de_imagen Impresión_de_archivo_de_configuracion

Nombre de la clase :	Impresion_de_archivo_de_configuracion
Instancias a métodos :	<ul style="list-style-type: none">** Recibir vector de configuración: Recibe el vector de datos de configuración a imprimir.** Verificar tipo de datos: Se encarga de verificar que los datos tengan el formato correcto.** Imprimir archivo de configuración: Envía los datos del archivo de configuración indicado a imprimir.
Relaciones de herencia:	Clase base Impresiones

Nombre de la clase :	Impresion_de_imagen
Instancias a métodos :	<ul style="list-style-type: none">** Recibir vector de imagen : Recibe el vector de información de la imagen a imprimir.** Identificar formato (verificar tipo de imagen): Se encarga de identificar el formato de la imagen.** Imprimir imagen. Envía la imagen a la impresora.
Relaciones de herencia:	Clase base Impresiones

9.2.2.2.4.5 Módulo de clases de editor molecular

Nombre de la clase :	Editor_molecular
Instancias a métodos :	
<ul style="list-style-type: none"> ** Identificar selección y acción: Se encarga de obtener el tipo de selección y la acción adecuados para realizar una edición sobre una molécula. ** Obtener etiquetas de objetos seleccionados: Obtiene las etiquetas de los objetos de la molécula a los que se les va a realizar la edición. ** Ejecutar acción: Realiza el tipo de edición o acción solicitado sobre la escena gráfica o la molécula. <ul style="list-style-type: none"> o Copiar objetos en escena: Se encarga de hacer copias de los objetos seleccionados sobre la escena gráfica. o Pegar objetos en escena: Se encarga de pegar las copias de los objetos en la escena gráfica. o Cortar objetos de escena: Se encarga de quitar los objetos seleccionados de la escena gráfica. o Unir objetos de escena (átomos o moléculas): Se encarga de unir los objetos seleccionados. o Deshacer cambios: Deshace los cambios realizados por el editor. o Rehacer cambios: Repite los cambios deshechos por el editor. o Agrupar objetos de escena: Ésta función se encarga de asociar los objetos seleccionados en la escena gráfica, como si éstos fueran un solo objeto. o Mostrar tabla periódica: Se encarga de activar la tabla periódica. o Asignar propiedades para mostrar elemento sobre una ventana: Se encarga de indicar que propiedades del elemento serán mostradas en la ventana. o Generar átomo temporal: Se encarga de crear un objeto temporal del tipo átomo. o Agregar átomo sobre una molécula: Se encarga de adicionar un átomo a la molécula indicada. o Generar conexión temporal: Se encarga de crear un objeto temporal del tipo conexión. o Agregar conexión a escena: Se encarga de adicionar una conexión a la molécula indicada. o Generar dipolo temporal: Se encarga de crear un objeto temporal del tipo dipolo. o Agregar dipolo a escena: Se encarga de adicionar un dipolo a la molécula indicada. o Agregar molécula sobre celda: Se encarga de adicionar una molécula a la celda computacional actual. o Identificar errores: Se encarga de clasificar los errores que ocurren y notificarlos. o Efectuar actualización del editor: Actualiza la configuración del editor molecular. ** Actualizar vector de historia: Genera un vector de historia con las operaciones y los datos sobre los que se realiza la edición. 	
Relaciones con otras clases:	
Tiene objetos de la clase	
Molécula	
Atomo	
Conexion	
Dipolo	

Nombre de la clase :	Puntero_3D
Instancias a métodos :	
<ul style="list-style-type: none"> ** Obtener coordenadas: Obtiene las coordenadas de la posición del objeto seleccionado por el puntero 3D ** Cambiar tipo de puntero: Cambia el tipo de puntero con el que se seleccionan los objetos. ** Cambiar color al puntero: Cambia el color del puntero 3D. 	
Relaciones de herencia:	
Ninguna	

9.2.2.2.4.6 Módulo de clases de contenedores

Nombre de la clase :	Contenedor_de_datos
Instancias a métodos :	
**	Verificar tipo de archivo de datos: Se encarga de verificar el tipo del archivo que fue leído, revisando la extensión del nombre del mismo.
**	Obtener nombre de archivo de datos: Se encarga de recibir el nombre del archivo de datos para el contenedor.
**	Verificar nombre de archivo de datos: Sirve para revisar que el nombre del archivo sea válido.
**	Verificar formato de archivo de datos: Sirve para revisar que el formato del archivo sea válido.
**	Leer archivo de datos: Ésta función se encarga de leer el archivo y almacenar los datos en un vector.
**	Actualizar vector: Cambia los datos anteriores por los más recientes.
**	Traducir datos: Ésta función se encarga de convertir los datos de formatos de otros programa al formato de grafica (SVDM).
**	Renombrar archivo de datos: Se encarga de cambiar el nombre al archivo de datos.
**	Guardar archivo de datos: Se encarga de colocar los datos del vector de configuración actual en el archivo de datos correspondiente.
**	Cerrar archivo de datos: Cierra el archivo de datos.
**	Generar vector de datos de configuración actual: Prepara el vector de configuración con los datos más actuales.
**	Devolver vector de datos: Ésta función entrega los datos a las clases que lo soliciten.
**	Devolver archivo de datos: Ésta función proporciona el archivo de datos a los procesos que lo solicitan.
**	Notificar mensajes de error: Se encarga de proporcionar los mensajes de error al usuario.
<p>Nota: Los métodos de verificar pueden ser implantados como una función los tres o cada uno puede ser una función por si solo.</p>	
Relaciones de herencia:	
<p>Clases derivadas de Contenedor_de_datos</p> <ul style="list-style-type: none"> Contenedor_de_hoyos Contenedor_de_particulas Contenedor_de_imagen Contenedor_de_potencial Contenedor_de_dipolos Contenedor_de_configuracion_de_dinamica_t Contenedor_de_configuracion_de_dinamica_t-dt Contenedor_de_inicia_dinamica 	

Nombre de la clase :	Contenedor_de_particulas
Instancias a métodos :	<ul style="list-style-type: none">** Verificar tipo de archivo de partículas: Se encarga de verificar que el tipo del archivo de partículas sea el correcto.** Verificar nombre de archivo de partículas: Sirve para revisar que el nombre del archivo de partículas sea válido.** Verificar formato de archivo de partículas: Función para revisar que el formato del archivo de datos sea de partículas.** Leer archivo de partículas: Se encarga de leer el archivo de partículas y almacenar los datos en un vector.** Actualizar vector: Sirve para actualizar los datos del vector de partículas.** Traducir datos de partículas: Ésta función se encarga de convertir los datos de formatos de otros programa al formato de grafica (SVDM).** Guardar archivo de partículas: Se encarga de colocar los datos del vector de configuración actual de partículas en el archivo de partículas correspondiente.** Generar vector de datos de configuración actual de partículas: Prepara el vector de configuración con los datos más recientes de partículas.** Devolver vector de partículas: Ésta función entrega los datos de partículas a las clases que lo soliciten.** Devolver archivo de partículas: Ésta función proporciona el archivo de partículas a los procesos que lo solicitan.** Proporcionar tipo y datos de caja: Entrega el tipo de caja, los valores ladoa, ladob, ladoc y los ángulos o ángulo a las rutinas de visualización y de cálculos que lo soliciten.** Obtener nombre de archivo: Se encarga de recibir el nombre del archivo de partículas para el contenedor.** Cerrar archivo anterior sin guardar cambios: Sólo cierra el archivo sin vaciar el vector de datos de partículas en éste.** Crear archivo con nuevo nombre: Genera un archivo con el nombre indicado y lo asocia con el contenedor en uso.** Mostrar datos para modificación: Muestra el vector de datos de partículas actual para que el usuario lo modifique manualmente.** Notificar mensajes de error: Se encarga de proporcionar los mensajes de error al usuario.
Relaciones de herencia:	
Clases base:	Contenedor_de_datos

Nombre de la clase :	Contenedor_de_hoyos
Instancias a métodos :	
**	Verificar tipo de archivo de hoyos: Se encarga de verificar que el tipo del archivo de hoyos sea correcto.
**	Verificar nombre de archivo de hoyos: Sirve para revisar que el nombre del archivo de hoyos sea válido.
**	Verificar formato de archivo de hoyos: Función para revisar que el formato del archivo de datos sea de hoyos.
**	Leer archivo de hoyos: Se encarga de leer el archivo de hoyos y almacenar los datos en un vector.
**	Actualizar vector: Sirve para actualizar los datos del vector de hoyos.
**	Guardar archivo de hoyos: Se encarga de colocar los datos del vector de configuración actual de hoyos en el archivo de hoyos correspondiente.
**	Generar vector de datos de configuración actual de hoyos: Prepara el vector de configuración con los datos más actuales de hoyos.
**	Devolver vector de hoyos: Ésta función entrega los datos de hoyos a las clases que lo soliciten.
**	Devolver archivo de hoyos: Ésta función proporciona el archivo de hoyos a los procesos que lo solicitan.
**	Obtener nombre de archivo: Se encarga de recibir el nombre del archivo de hoyos para el contenedor.
**	Cerrar archivo anterior sin guardar cambios: Sólo cierra el archivo sin vaciar el vector de datos de hoyos en éste.
**	Crear archivo con nuevo nombre: Genera un archivo con el nombre indicado y lo asocia con el contenedor en uso.
**	Mostrar datos para modificación: Muestra el vector de datos de hoyos actual para que el usuario lo modifique manualmente.
**	Notificar mensajes de error: Se encarga de proporcionar los mensajes de error al usuario.
Relaciones de herencia:	
Clases base: Contenedor_de_datos	

Nombre de la clase :	Contenedor_de_imagen
Instancias a métodos :	
**	Actualizar vector: Sirve para reemplazar los datos del vector de imagen con los más actuales.
**	Generar vector de datos de configuración actual de imagen: Prepara el vector de configuración con los datos más actuales de imagen.
**	Guardar archivo de imagen: Se encarga de colocar los datos del vector de configuración actual de imagen en el archivo de imagen correspondiente.
**	Devolver vector de imagen: Ésta función entrega los datos de imagen a las clases que lo soliciten.
**	Devolver archivo de imagen: Ésta función proporciona el archivo de imagen al usuario
**	Obtener nombre de archivo: Se encarga de recibir el nombre del archivo de imagen para el contenedor.
**	Cerrar archivo anterior sin guardar cambios: Sólo cierra el archivo sin vaciar el vector de datos de imagen en éste.
**	Crear archivo con nuevo nombre: Genera un archivo con el nombre indicado y lo asocia con el contenedor en uso.
**	Notificar mensajes de error: Se encarga de proporcionar los mensajes de error al usuario.
Relaciones de herencia:	
Clases base: Contenedor_de_datos	

Nombre de la clase :	Contenedor_de_dipolos
Instancias a métodos :	
**	Verificar tipo de archivo de dipolos: Se encarga de verificar que el tipo del archivo de dipolos sea correcto.
**	Verificar nombre de archivo de dipolos: Sirve para revisar que el nombre del archivo de dipolos sea válido.
**	Verificar formato de archivo de dipolos: Función para revisar que el formato del archivo de datos sea de dipolos.
**	Leer archivo de dipolos: Se encarga de leer el archivo de dipolos y almacenar los datos en un vector.
**	Actualizar vector: Sirve para reemplazar los datos del vector de dipolos con los más actuales.
**	Guardar archivo de dipolos: Se encarga de colocar los datos del vector de configuración actual de dipolos en el archivo de dipolos correspondiente.
**	Generar vector de datos de configuración actual de dipolos: Prepara el vector de configuración con los datos más actuales de dipolos.
**	Devolver vector de dipolos: Ésta función entrega los datos de dipolos a las clases que lo soliciten.
**	Devolver archivo de dipolos: Ésta función proporciona el archivo de dipolos a los procesos que lo solicitan.
**	Obtener nombre de archivo: Se encarga de recibir el nombre del archivo de dipolos para el contenedor.
**	Cerrar archivo anterior sin guardar cambios: Sólo cierra el archivo sin vaciar el vector de datos de dipolos en éste.
**	Crear archivo con nuevo nombre: Genera un archivo con el nombre indicado y lo asocia con el contenedor en uso.
**	Mostrar datos para modificación: Muestra el vector de datos de dipolos actual para que el usuario lo modifique manualmente.
**	Notificar mensajes de error: Se encarga de proporcionar los mensajes de error al usuario.
Relaciones de herencia:	
Clases base: Contenedor_de_datos	

Nombre de la clase :	Contenedor_de_potencial
Instancias a métodos :	
**	Verificar tipo de archivo de potencial: Se encarga de verificar que el tipo del archivo de potencial sea correcto.
**	Verificar nombre de archivo de potencial: Sirve para revisar que el nombre del archivo de potencial sea válido.
**	Verificar formato de archivo de potencial: Función para revisar que el formato del archivo de datos sea de potencial.
**	Leer archivo de potencial: Se encarga de leer el archivo de potencial y almacenar los datos en un vector.
**	Actualizar vector: Sirve para actualizar los datos del vector de potencial.
**	Guardar archivo de potencial: Se encarga de colocar los datos del vector de configuración actual de potencial en el archivo de potencial correspondiente.
**	Generar vector de datos de configuración actual de potencial: Prepara el vector de configuración con los datos más actuales de potencial.
**	Devolver vector de potencial: Ésta función entrega los datos de potencial a las clases que lo soliciten.
**	Devolver archivo de potencial: Ésta función proporciona el archivo de potencial a los procesos que lo solicitan.
**	Obtener nombre de archivo: Se encarga de recibir el nombre del archivo de potencial para el contenedor.
**	Cerrar archivo anterior sin guardar cambios: Sólo cierra el archivo sin vaciar el vector de datos de potencial en éste.
**	Crear archivo con nuevo nombre: Genera un archivo con el nombre indicado y lo asocia con el contenedor en uso.
**	Mostrar datos para modificación: Muestra el vector de datos de potencial actual para que el usuario lo modifique manualmente.
**	Notificar mensajes de error: Se encarga de proporcionar los mensajes de error al usuario.
Relaciones de herencia:	
Clases base: Contenedor_de_datos	

Nombre de la clase :**Contenedor_de_conexiones****Instancias a métodos :**

- ** Verificar tipo de archivo de conexiones: Se encarga de verificar que el tipo del archivo de conexiones sea el correcto.
- ** Verificar nombre de archivo de conexiones: Sirve para revisar que el nombre del archivo de conexiones sea válido.
- ** Verificar formato de archivo de conexiones: Función para revisar que el formato del archivo de datos sea de conexiones.
- ** Leer archivo de conexiones: Se encarga de leer el archivo de conexiones y almacenar los datos en un vector.
- ** Actualizar vector: Sirve para actualizar los datos del vector de conexiones.
- ** Guardar archivo de conexiones: Se encarga de colocar los datos del vector de configuración actual de conexiones en el archivo de conexiones correspondiente.
- ** Generar vector de datos de configuración actual de conexiones: Prepara el vector de configuración con los datos más actuales de conexiones.
- ** Devolver vector de conexiones: Ésta función entrega los datos de conexiones a las clases que lo soliciten.
- ** Devolver archivo de conexiones: Ésta función proporciona el archivo de conexiones a los procesos que lo solicitan.
- ** Obtener nombre de archivo: Se encarga de recibir el nombre del archivo de conexiones para el contenedor.
- ** Cerrar archivo anterior sin guardar cambios: Sólo cierra el archivo sin vaciar el vector de datos de conexiones en éste.
- ** Crear archivo con nuevo nombre: Genera un archivo con el nombre indicado y lo asocia con el contenedor en uso.
- ** Mostrar datos para modificación: Muestra el vector de datos de conexiones actual para que el usuario lo modifique manualmente.
- ** Notificar mensajes de error: Se encarga de proporcionar los mensajes de error al usuario.

Relaciones de herencia:

Clases base: Contenedor_de_datos

Nombre de la clase : Contenedor_de_configuración_de_dinamica_t

Instancias a métodos :

- ** Verificar tipo de archivo de configuración de dinámica t: Se encarga de verificar que el tipo del archivo de configuración de dinámica t sea correcto.
- ** Verificar nombre de archivo de configuración de dinámica t: Sirve para revisar que el nombre del archivo de configuración de dinámica t sea válido.
- ** Verificar formato de archivo de configuración de dinámica t: Función para revisar que el formato del archivo de datos sea de configuración de dinámica t.
- ** Leer archivo de configuración de dinámica t: Se encarga de leer el archivo de configuración de dinámica t y almacena los datos en un vector.
- ** Actualizar vector: Sirve para actualizar los datos del vector de configuración de dinámica t.
- ** Guardar archivo de configuración de dinámica t: Se encarga de colocar los datos del vector de configuración actual de dinámica t en el archivo de configuración de dinámica t correspondiente.
- ** Generar vector de datos de configuración actual de configuración de dinámica t: Prepara el vector de configuración con los datos más actuales de configuración de dinámica t.
- ** Devolver vector de configuración de dinámica t: Ésta función entrega los datos de configuración de dinámica t a las clases que lo soliciten.
- ** Devolver archivo de configuración de dinámica t: Ésta función proporciona el archivo de configuración de dinámica t a los procesos que lo solicitan.
- ** Obtener nombre de archivo: Se encarga de recibir el nombre del archivo de configuración de dinámica t para el contenedor.
- ** Cerrar archivo anterior sin guardar cambios: Sólo cierra el archivo sin vaciar el vector de datos de configuración de dinámica t en éste.
- ** Crear archivo con nuevo nombre: Genera un archivo con el nombre indicado y lo asocia con el contenedor en uso.
- ** Mostrar datos para modificación: Muestra el vector de datos de configuración de dinámica t actual para que el usuario lo modifique manualmente.
- ** Notificar mensajes de error: Se encarga de proporcionar los mensajes de error al usuario.

Relaciones de herencia:

Clases base: Contenedor_de_datos

Nombre de la clase : `Contenedor_de_configuración_de_dinámica_t-dt`

Instancias a métodos :

- ** Verificar tipo de archivo de configuración de dinámica t-dt: Se encarga de verificar que el tipo del archivo de configuración de dinámica t-dt sea correcto.
- ** Verificar nombre de archivo de configuración de dinámica t-dt: Sirve para revisar que el nombre del archivo de configuración de dinámica t-dt sea válido.
- ** Verificar formato de archivo de configuración de dinámica t-dt: Función para revisar que el formato del archivo de datos sea de configuración de dinámica t-dt.
- ** Leer archivo de configuración de dinámica t-dt: Se encarga de leer el archivo de configuración de dinámica t-dt y almacenar los datos en un vector.
- ** Actualizar vector: Sirve para actualizar los datos del vector de configuración de dinámica t-dt.
- ** Guardar archivo de configuración de dinámica t-dt: Se encarga de colocar los datos del vector actual de configuración de dinámica t-dt en el archivo de configuración de dinámica t-dt correspondiente.
- ** Generar un vector de datos actual de configuración de dinámica t-dt: Prepara el vector de configuración con los datos más recientes de configuración de dinámica t-dt.
- ** Devolver vector de configuración de dinámica t-dt: Ésta función entrega los datos de configuración de dinámica t-dt a las clases que lo soliciten.
- ** Devolver archivo de configuración de dinámica t-dt: Ésta función proporciona el archivo de configuración de dinámica t-dt a los procesos que lo solicitan.
- ** Obtener nombre de archivo: Se encarga de recibir el nombre del archivo de configuración de dinámica t-dt para el contenedor.
- ** Cerrar archivo anterior sin guardar cambios: Solo cierra el archivo sin vaciar el vector de datos de configuración de dinámica t-dt en éste.
- ** Crear archivo con nuevo nombre: Genera un archivo con el nombre indicado y lo asocia con el contenedor en uso.
- ** Mostrar datos para modificación: Muestra el vector de datos de configuración de dinámica t-dt actual para que el usuario lo modifique manualmente.
- ** Notificar mensajes de error: Se encarga de proporcionar los mensajes de error al usuario.

Relaciones de herencia:

Clases base: `Contenedor_de_datos`

Nombre de la clase :	Contenedor_de_inicia_dinamica
Instancias a métodos :	
	<ul style="list-style-type: none">** Verificar tipo de archivo de inicia dinámica: Se encarga de verificar que el tipo del archivo de inicia dinámica sea correcto.** Verificar nombre de archivo de inicia dinámica: Sirve para revisar que el nombre del archivo de inicia dinámica sea válido.** Verificar formato de archivo de inicia dinámica: Función para revisar que el formato del archivo de datos sea de inicia dinámica.** Leer archivo de inicia dinámica: Se encarga de leer el archivo de inicia dinámica y almacenar los datos en un vector.** Actualizar vector: Sirve para actualizar los datos del vector de inicia dinámica.** Guardar archivo de inicia dinámica: Se encarga de colocar los datos del vector de configuración actual de inicia dinámica en el archivo de inicia dinámica correspondiente.** Generar vector de datos de configuración actual de inicia dinámica: Prepara el vector de configuración con los datos más actuales de inicia dinámica.** Devolver vector de inicia dinámica: Ésta función entrega los datos de inicia dinámica a las clases que lo soliciten.** Devolver archivo de inicia dinámica: Ésta función proporciona el archivo de inicia dinámica a los procesos que lo soliciten.** Obtener nombre de archivo: Se encarga de recibir el nombre del archivo de inicia dinámica para el contenedor.** Cerrar archivo anterior sin guardar cambios: Sólo cierra el archivo sin vaciar el vector de datos de inicia dinámica en éste.** Crear archivo con nuevo nombre: Genera un archivo con el nombre indicado y lo asocia con el contenedor en uso.** Mostrar datos para modificación: Muestra el vector de datos de inicia dinámica actual para que el usuario lo modifique manualmente.** Notificar mensajes de error: Se encarga de proporcionar los mensajes de error al usuario.
Relaciones de herencia:	
	Clases base: Contenedor_de_datos

Nombre de la clase :	Contenedor_de_rayos_x
Instancias a,métodos :	<ul style="list-style-type: none"> ** Verificar tipo de archivo de rayos x: Se encarga de verificar que el tipo del archivo rayos x sea correcto. ** Verificar nombre de archivo de rayos x: Sirve para revisar que el nombre del archivo de rayos x sea válido. ** Verificar formato de archivo de rayos x: Función para revisar que el formato del archivo de datos sea de rayos x. ** Leer archivo de rayos x: Se encarga de leer el archivo de rayos x y almacenar los datos en un vector. ** Actualizar vector: Sirve para actualizar los datos del vector de rayos x. ** Guardar archivo de rayos x: Se encarga de colocar los datos del vector de configuración actual de rayos x en el archivo de rayos x correspondiente. ** Generar vector de datos de configuración actual de rayos x: Prepara el vector de configuración con los datos más actuales de rayos x. ** Devolver vector de rayos x: Ésta función entrega los datos de rayos x a las clases que lo solicitan. ** Devolver archivo de rayos x: Ésta función proporciona el archivo de rayos x a los procesos que lo solicitan. ** Obtener nombre de archivo: Se encarga de recibir el nombre del archivo de rayos x para el contenedor. ** Cerrar archivo anterior sin guardar cambios: Sólo cierra el archivo sin vaciar el vector de datos de rayos x en éste. ** Crear archivo con nuevo nombre: Genera un archivo con el nombre indicado y lo asocia con el contenedor en uso. ** Mostrar datos para modificación: Muestra el vector de datos de rayos x actual para que el usuario lo modifique manualmente. ** Notificar mensajes de error: Se encarga de proporcionar los mensajes de error al usuario.
Relaciones de herencia:	
	Clases base: Contenedor_de_datos

9.2.2.3 Optimizar el diseño

9.2.2.3.1 Añadir asociaciones redundantes

El sistema lee los datos y los traduce, interpretando y llenando las estructuras de datos indicadas, así como los archivos de contenedores, los cuales serán enviados a las operaciones y clases según sean requeridos.

Se debe evitar caer en ciclos repetitivos, cuando se pueden usar índices o formas de direccionamiento más rápidas o eficientes a los datos que sean de uso frecuente.

Nota:

Se deben proveer índices para operaciones frecuentes y costosas con aciertos bajos ya que tales operaciones son ineficientes usando una implementación de ciclos anidados.

9.2.2.3.2 Reorganizar el orden de ejecución

Al leer los datos del archivo, es necesario almacenarlos en la estructura de datos de manera secuencial, esto es, los datos correspondientes a cada partícula deben ser almacenados, pero se debe considerar que es mejor almacenar los datos de las partículas más cercanas en las posiciones iniciales, y posteriormente las que se encuentren en los siguientes planos, de manera que sea más rápida la graficación.

Esto es, dependiendo de lo que se quiera realizar, es decir, dentro de un ciclo anidado, se debe realizar primero lo que requiera menos tiempo de proceso y posteriormente lo que requiera mayor tiempo.

9.2.2.3 Guardar atributos derivados

Es necesario cuidar que los datos que sean derivados de otros datos puedan guardarse en su forma computada. en caso de que sean requeridos nuevamente, para este propósito se utilizarán en el sistema los contenedores de datos, que serán actualizados según sea requerido, esto es, se utilizarán los objetos y clases de Contenedores.

Los datos que serán almacenados en los contenedores, para su uso según lo sea requerido por los diferentes objetos y componentes del sistema:

- Datos de configuración (éstos son los datos que se leerán directamente de archivos de configuración):
 - datos de partículas,
 - de hoyos,
 - de conexiones,
 - de dipolos,
 - de potenciales,
 - de rayos x,
 - de configuración de dinámica
 - de inicialización de dinámica
 - archivo de definición de elementos (atomo.h)
- Datos obtenidos durante la realización de cálculos (éstos serán entregados por los procesos de cálculos):
 - cálculos de manipulación de objetos
 - operaciones realizadas (para vector de historia)
 - cálculo de imágenes (para impresión y salvar imágenes)

9.2.2.4 Implementar el control

- Lectura de archivos y almacenamiento de los datos (control secuencial)
- Cálculo de datos necesarios (control secuencial)
 - Generación de datos auxiliares
- Graficación de resultados (impulsado por eventos)
- Generación de procesos interactivos (hilos de control)
- Entrega de impresiones (control secuencial)

9.2.2.5 Ajustar la herencia

9.2.2.5.1 Reorganizar las clases

Las operaciones que se identificaron como necesarias en varias clases se agruparon en una clase superior o clase padre, ejemplos de clases padre son, dentro de las clases gráficas: objeto dinámico y objeto estático.

Todos los atributos que correspondían a los atributos similares fueron heredados de clases superiores. Esto es, las operaciones similares fueron heredadas de clases padres.

9.2.2.5.2 Abstraer comportamiento común

Aquí fue necesario identificar operaciones entre clases “hermanas”, es decir, clases que se derivan de una misma superclase, para reconocer el comportamiento común y dejar así las especializaciones únicamente para las clases que así lo requirieran.

9.2.2.5.3 Delegación

La delegación consiste en tomar la operación de un objeto y mandarla a otro objeto que está relacionado con el primer objeto. Siendo delegadas al segundo objeto, sólo las operaciones importantes. Esto se puede manejar como delegar las responsabilidades de una operación a una o más operaciones similares de otro objeto, que pueden realizar esa actividad.

En el caso del sistema éste tipo de manejo de herencia se utilizará en el monitor de procesos.

En éste caso se utilizó únicamente herencia sencilla, no existen clases con herencia múltiple.

De esa manera la herencia fue asignada y ajustada correctamente.

9.2.2.6 Diseñar asociaciones

Con respecto a este punto, se formuló una estrategia para implantar las asociaciones, se realizaron las asociaciones como relaciones entre clases, es decir, que clase u objeto se relacionaba con que otra y así se obtuvieron las asociaciones correspondientes.

Las clases abstractas tienen relaciones sólo con las clases que se derivan de ellas. Aquí se tomaron en cuenta las asociaciones en una dirección, que por lo general vienen de clases superiores, y para cuyo tipo de relaciones no es necesario crear ninguna clase nueva al representar dichas asociaciones, las cuales son hasta cierto punto implícitas y necesarias.

Así mismo, las clases abstractas (clases base) no tienen asociación directa de ningún tipo con ninguna otra clase. Interactúan a través de las clases que de ellos se derivan (clases terminales).

En esta parte únicamente se tomarán en cuenta las asociaciones funcionales y de agregación entre las clases (clases que contengan uno o más atributos del tipo de otra clase determinada), que son las asociaciones de diseño, es decir, las asociaciones que definen el comportamiento y comunicación necesaria entre los diferentes objetos que forman parte del sistema.

Debido a que cualquier objeto puede ser conceptualizado, como una agregación de sus atributos, es decir, "un atributo es parte de un objeto" y sus tiempos de vida son también enlazados, cuando un objeto es creado o destruido. así mismo, sus atributos, las asociaciones de agregación serán omitidas por que se refieren únicamente a asociaciones de las clases con sus respectivos atributos. Únicamente se mencionarán las asociaciones de agregación de las clases con sus atributos objeto, es decir, clases que tengan uno o más atributos de algún tipo de otra clase definida, debido a que éstas son asociaciones de agregación que se pueden interpretar como funcionales.

Existe un tipo de asociación llamada por encapsulación y consiste en que una clase posee un atributo, ya sea un valor, o un objeto atributo, al cual sólo se puede tener acceso a través de dicho objeto. De éste tipo de asociaciones se pueden identificar cuando un determinado objeto requiere de cambiar el valor de un atributo de algún otro objeto o de un atributo objeto en sí, y no puede acceder a éste sino a través del objeto al cual pertenece dicho atributo (atributo u objeto atributo), debido que este último se encuentra encapsulado dentro del objeto.

Un tipo de éstas asociaciones son las que corresponden al funcionamiento del Monitor de Procesos, ya que cualquier objeto que requiera tener acceso a los estados de los procesos, o el mismo usuario, lo hará por medio del monitor de procesos.

9.2.2.6.1 Revisar las asociaciones entre clases

A continuación se elaboró una lista describiendo todas las relaciones funcionales entre los diferentes objetos, con el fin de encontrar las asociaciones que podrían ser representadas mejor como clases en sí.

9.2.2.6.1.1 Asociaciones de las clases gráficas

Clase abstracta:	
Nombre de la clase:	Objeto_grafico
Tiene relaciones con:	
Relaciones de herencia	Clases derivadas de Objeto_grafico
	Objeto_estatico
	Objeto_dinamico
	Plano_potencial
Clase abstracta:	
Nombre de la clase:	Objeto_estatico
Tiene relaciones con:	
Relaciones de herencia	Clases derivadas de Objeto_estatico
	Caja
	Eje
	Clase base: Objeto_grafico
Clases terminales:	
Nombre de la clase:	Eje
Tiene relaciones con:	
Relaciones de herencia	Clase base: Objeto_estatico
Relaciones con otras clases	Los Ejes forman parte de Escena_grafica
Nombre de la clase:	Caja
Tiene relaciones con:	
Relaciones de herencia	Clase base: Objeto_estatico
Relaciones con otras clases	La Caja define los límites de la Celda
	La Caja forma parte de la Celda
Clase abstracta:	
Nombre de la clase:	Objeto_dinamico
Tiene relaciones con:	
Relaciones de herencia	Clases derivadas de Objeto_dinamico
	Esfera
	Conexion
	Clase base: Objeto_grafico

Clases terminales:

Nombre de la clase: **Conexion**

Tiene relaciones con:

- Relaciones de herencia
 - Clase base: Objeto_dinamico
- Relaciones con otras clases
 - Una Conexión une a dos Atomos

Nombre de la clase: **Esfera**

Tiene relaciones con:

- Relaciones de herencia
 - Clases derivadas de Esfera
 - Atomo
 - Hoyo
 - Dipolo
- Clase base: Objeto_dinamico

Nombre de la clase: **Atomo**

Tiene relaciones con:

- Relaciones de herencia
 - Clase base: Esfera
- Relaciones con otras clases
 - Un Atomo es parte de Molecula
 - Un Atomo es una Esfera
 - Dos Atomos pueden o no ser unidos por una Conexion

Nombre de la clase: **Hoyo**

Tiene relaciones con:

- Relaciones de herencia
 - Clases base: Esfera
- Relaciones con otras clases (agregación)
 - Un Hoyo es parte de Sistema_de_cavidades
 - Un Hoyo es una Esfera
 - Un Hoyo forma parte de una Molecula

Nombre de la clase: **Dipolo**

Tiene relaciones con:

- Relaciones de herencia
 - Clase base: Esfera
- Relaciones con otras clases
 - Un Dipolo es parte de una Molecula
 - Un Dipolo corresponde a un Atomo

Nombre de la clase: **Plano_potencial**

Tiene relaciones con:

- Relaciones de Herencia
 - Clases base: Objeto_grafico
- Relaciones con otras clases
 - Plano_potencial es parte de Superficie_de_potencial

Clase abstracta:	
Nombre de la clase:	Objeto_animado
Tiene relaciones con:	<ul style="list-style-type: none"> Relaciones de herencia <ul style="list-style-type: none"> Clases derivadas de Objeto_animado <ul style="list-style-type: none"> Sistema_de_cavidades Molecula Superficie_potencial
Clases terminales:	
Nombre de la clase:	Sistema_de_cavidades
Tiene relaciones con:	<ul style="list-style-type: none"> Relaciones de herencia <ul style="list-style-type: none"> Clase base: Objeto_animado Relaciones con otras clases <ul style="list-style-type: none"> Sistema_de_cavidades está formado por objetos de tipo Hoyo Sistema_de_cavidades se forma dentro de la Escena_grafica
Nombre de la clase:	Molecula
Tiene relaciones con:	<ul style="list-style-type: none"> Relaciones de herencia <ul style="list-style-type: none"> Clase base: Objeto_animado Relaciones con otras clases <ul style="list-style-type: none"> Tiene uno o más objetos de las clases: <ul style="list-style-type: none"> Atomo Dipolo Conexión Una o más Moléculas son parte de Escena_grafica La Molecula genera una Superficie_potencial
Nombre de la clase:	Superficie_potencial
Tiene relaciones con:	<ul style="list-style-type: none"> Relaciones de herencia <ul style="list-style-type: none"> Clase base: Objeto_animado Relaciones con otras clases <ul style="list-style-type: none"> Superficie_potencial está compuesta por Planos potenciales
Nombre de la clase:	Estructura_cristalina
Tiene relaciones con:	<ul style="list-style-type: none"> Relaciones con otras clases <ul style="list-style-type: none"> Celda Superficie_potencial Una Estructura_cristalina es una repetición de Moleculas Una Escena_grafica forma parte de una Estructura_cristalina
Nombre de la clase:	Escena_grafica
Tiene relaciones con:	<ul style="list-style-type: none"> Relaciones con otras clases <ul style="list-style-type: none"> Está formada por objetos de los siguientes tipos: <ul style="list-style-type: none"> Estructura_Cristalina, Puntero_3D Una Escena_Grafica tiene tres Ejes coordenados Una Escena_Grafica tiene una Celda Una Escena_Grafica tiene una o más Moleculas de diferente tipo

Clases terminales:**Nombre de la clase:** **Celda**

Tiene relaciones con:

Relaciones con otras clases

Tiene objetos de la clase:

Molecula

Caja

Sistema_de_cavidades

Plano_potencial

La Celda forma parte de la Escena_grafica

La Celda está delimitada por la Caja

La Celda se puede ajustar a la configuración del tipo de la Caja

Nombre de la clase: **Indicador_de_seleccion**

Tiene relaciones con:

Relaciones con otras clases

Modificador_de_propiedades

Indicador_de_seleccion sirve en el Editor_molecular para seleccionar los distintos objetos gráficos

Indicador_de_selección sirve en la Escena_grafica para seleccionar los diversos objetos gráficos

Nombre de la clase: **Modificador_de_propiedades**

Tiene relaciones con:

Relaciones con otras clases

Modificador_de_propiedades utiliza las selecciones realizadas por el

Indicador_de_seleccion

Modificador_de_propiedades cambia los valores especificados de las propiedades de los objetos gráficos indicados

Modificador_de_propiedades notifica los cambios al Editor_molecular

9.2.2.6.1.2 Asociaciones de las clases de cálculos**Clase abstracta:****Nombre de la clase:** **Proceso_calculo**

Tiene relaciones con:

Relaciones de herencia

Clases derivadas de Proceso_calculo

Calculo_de_imagen

Calculo_de_dinamica

Calculo_generado_por_particulas

Calculo_de_magnitudes_y_angulos

Calculo_local

Calculo_remoto

Clases terminales:	
Nombre de la clase:	Calculo_de_imagen
Tiene relaciones con:	
Relaciones de herencia	Clase base: Proceso_calculo
Relaciones con otras clases	Calculo_de_imagen proporciona los datos de la Imagen generada al Contenedor_de_imagen, Monitor_de_procesos activa el Calculo_de_imagen
Nombre de la clase:	Calculo_de_dinamica
Tiene relaciones con:	
Relaciones de herencia	Clase base: Proceso_calculo
Relaciones con otras clases	Calculo_de_dinamica genera los datos de la dinámica molecular y los coloca en los contenedores de datos, Monitor_de_procesos activa el Calculo_de_dinamica
Nombre de la clase:	Calculo_de_magnitudes_y_angulos
Tiene relaciones con:	
Relaciones de herencia	Clase base: Proceso_calculo
Relaciones con otras clases	Monitor_de_procesos activa el Calculo_de_magnitudes_y_angulos
Nombre de la clase:	Calculo_generado_por_particulas
Tiene relaciones con:	
Relaciones de herencia	Clase base: Proceso_calculo
	Clases derivadas
	Calculo_local, Calculo_remoto
Relaciones con otras clases	Calculo_generado_por_particulas proporciona los datos de los cálculos especificados a los contenedores correspondientes
	Monitor_de_procesos activa el Calculo_generado_por_particulas
Nombre de la clase:	Calculo_local
Tiene relaciones con:	
Relaciones de herencia	Clase base: Calculo_generado_por_particulas
Relaciones con otras clases	Proceso_local envía el vector de datos a Calculo_local, para generar el cálculo de Rayos x, de vecinos, de análisis de coordinación y la generación del archivo de datos.
	Monitor_de_procesos activa el Calculo_local.
Nombre de la clase:	Calculo_remoto
Nota: Esta clase hace un llamado a un programa externo	
Tiene relaciones con:	
Relaciones de herencia	Clase base: Calculo_generado_por_particulas
Relaciones con otras clases	Proceso_remoto envía el vector de datos a Calculo_remoto para generar el cálculo de dipolos, de hoyos, de campo escalar y de corte.
	Monitor_de_procesos activa el Calculo_remoto.

9.2.2.6.1.3 Asociaciones de las clases de monitor de procesos

Clase terminal:	
Nombre de la clase:	Monitor_de_procesos
Tiene relaciones con:	<ul style="list-style-type: none"> Relaciones con otras clases <ul style="list-style-type: none"> Tiene objetos del tipo <ul style="list-style-type: none"> Proceso_local Proceso_remoto Monitor_de_procesos verifica el estado de los procesos locales y remotos Monitor_de_procesos permite ejecutar procesos locales y remotos
Clase abstracta:	
Nombre de la clase:	Proceso
Tiene relaciones con:	<ul style="list-style-type: none"> Relaciones de herencia <ul style="list-style-type: none"> Clases derivadas de Proceso <ul style="list-style-type: none"> Proceso_local Proceso_remoto
Clases terminales:	
Nombre de la clase:	Proceso_remoto
Tiene relaciones con:	<ul style="list-style-type: none"> Relaciones de herencia <ul style="list-style-type: none"> Clase base: Proceso Relaciones con otras clases <ul style="list-style-type: none"> Proceso_remoto recibe el archivo de datos de Calculo_remoto Proceso_remoto envía el archivo de datos de partículas a Calculo_remoto para generar el cálculo de dipolos, de hoyos, de campo escalar y de corte. Monitor_de_procesos ejecuta el Proceso_remoto Proceso_remoto ejecuta el Calculo_remoto asociado Proceso_remoto notifica los mensajes a Monitor_de_procesos
Nombre de la clase:	Proceso_local
Tiene relaciones con:	<ul style="list-style-type: none"> Relaciones de herencia <ul style="list-style-type: none"> Clase base: Proceso Relaciones con otras clases <ul style="list-style-type: none"> Proceso_local recibe el vector de datos de Calculo_local Proceso_local envía el vector de datos de partículas a Calculo_local para generar el cálculo de rayos x, de vecinos, de análisis de coordinación y la generación del archivo de datos Monitor_de_procesos ejecuta el Proceso_local Proceso_local ejecuta el Calculo_local asociado Proceso_local notifica los mensajes a Monitor_de_procesos

9.2.2.6.1.4 Asociaciones de las clases de impresiones

Clase abstracta:	
Nombre de la clase:	Impresiones
Tiene relaciones con:	<ul style="list-style-type: none"> Relaciones de herencia <ul style="list-style-type: none"> Clases derivadas de Impresiones <ul style="list-style-type: none"> Impresion_de_imagen Impresion_de_archivo_de_configuracion
Clases terminales:	
Nombre de la clase:	Impresion_de_archivo_de_configuracion
Tiene relaciones con:	<ul style="list-style-type: none"> Relaciones de herencia <ul style="list-style-type: none"> Clase base: Impresiones Relaciones con otras clases <ul style="list-style-type: none"> Impresiones recibe un vector de datos del contenedor del archivo de configuración
Nombre de la clase:	Impresion_de_imagen
Tiene relaciones con:	<ul style="list-style-type: none"> Relaciones de herencia <ul style="list-style-type: none"> Clase base: Impresiones Relaciones con otras clases <ul style="list-style-type: none"> Impresiones recibe un vector de datos del contenedor del archivo de imagen

9.2.2.6.1.5 Asociaciones de las clases de editor molecular

Clases terminales:	
Nombre de la clase:	Editor_molecular
Tiene relaciones con:	<ul style="list-style-type: none"> Relaciones con otras clases <ul style="list-style-type: none"> Tiene uno o más atributos objeto de las clases: <ul style="list-style-type: none"> Molecula Atomo Conexion Dipolo
Nombre de la clase:	Puntero_3D
Tiene relaciones con:	<ul style="list-style-type: none"> Relaciones con otras clases <ul style="list-style-type: none"> Puntero_3D permite seleccionar atomos Puntero_3D permite seleccionar hoyos Puntero_3D permite seleccionar conexiones Puntero_3D permite seleccionar dipolos

9.2.2.6.1.6 Asociaciones de las clases de contenedores

Clase abstracta:	
Nombre de la clase:	Contenedor_de_datos
Tiene relaciones con:	
Relaciones de herencia	Clases derivadas de Contenedor_de_datos
	Contenedor_de_hoyos
	Contenedor_de_particulas
	Contenedor_de_imagen
	Contenedor_de_potencial
	Contenedor_de_dipolos
	Contenedor_de_configuracion_de_dinamica (t o t-dt)
	Contenedor_de_inicia_dinamica
	Contenedor_de_rayos_x
Clases terminales:	
Nombre de la clase:	Contenedor_de_particulas
Tiene relaciones con:	
Relaciones de herencia	Clase base: Contenedor_de_datos
Relaciones con otras clases	Contenedor_de_particulas proporciona la configuración para representar a la molécula (relaciones con clases gráficas).
Nombre de la clase:	Contenedor_de_hoyos
Tiene relaciones con:	
Relaciones de herencia	Clase base: Contenedor_de_datos
Relaciones con otras clases	Contenedor_hoyos proporciona la configuración para representar el sistema de cavidades (relaciones con clases gráficas).
Nombre de la clase:	Contenedor_de_imagen
Tiene relaciones con:	
Relaciones de herencia	Clase base: Contenedor_de_datos
Relaciones con otras clases	Contenedor_de_imagen proporciona la configuración para realizar la impresion_de_imagen.
Nombre de la clase:	Contenedor_de_dipolos
Tiene relaciones con:	
Relaciones de herencia	Clase base: Contenedor_de_datos
Relaciones con otras clases	Contenedor_de_dipolos proporciona la configuración para dibujar dipolos (relaciones con clases graficas).
	Contenedor_de_dipolos proporciona los datos necesarios para realizar la Impresion_de_archivo_de_configuracion de dipolos.

Clases terminales:

Nombre de la clase: **Contenedor_de_potencial**

Tiene relaciones con:

Relaciones de herencia

Clase base: Contenedor_de_datos

Relaciones con otras clases

Contenedor_de_potencial proporciona la configuración para representar el plano_potencial

Contenedor_de_potencial proporciona los datos necesarios para realizar la Impresion_de_archivo_de_configuracion de potenciales.

Nombre de la clase: **Contenedor_de_conexiones**

Tiene relaciones con:

Relaciones de herencia

Clase base: Contenedor_de_datos

Relaciones con otras clases

Contenedor_de_conexiones proporciona la configuración para dibujar las conexiones

Contenedor_de_conexiones proporciona los datos necesarios para realizar la Impresion_de_archivo_de_configuracion de conexiones

Nombre de la clase: **Contenedor_de_configuración_de_dinamica (t o t-dt)**

Tiene relaciones con:

Relaciones de herencia

Clase base: Contenedor_de_datos

Relaciones con otras clases

Contenedor_de_configuración_de_dinamica proporciona la configuración de entrada (en conjunto con el contenedor de inicia dinámica), para generar la dinámica molecular a través de los cálculos remotos.

Contenedor_de_configuración_de_dinamica proporciona los datos necesarios para realizar la Impresion_de_archivo_de_configuracion de dinámica.

Nombre de la clase: **Contenedor_de_inicia_dinamica**

Tiene relaciones con:

Relaciones de herencia

Clase base: Contenedor_de_datos

Relaciones con otras clases

Contenedor_de_inicia_dinamica proporciona la configuración de entrada (en conjunto con el contenedor de configuración de dinámica) para generar la dinámica molecular a través de los cálculos remotos.

Contenedor_de_inicia_dinamica proporciona los datos necesarios para realizar la Impresion_de_archivo_de_configuracion de inicia dinámica.

Nombre de la clase: **Contenedor_de_rayos_x**

Tiene relaciones con:

Relaciones de herencia

Clase base: Contenedor_de_datos

Relaciones con otras clases

Contenedor_de_rayos_x envía los datos de rayos x para que sean visualizados por la clase de calculo_local.

Es posible también usar otras clases intermedias para representar las asociaciones que tengan demasiados atributos, por ejemplo:

- La asociación entre molécula y partículas.
- La asociación entre molécula y dipolos.
- La asociación entre molécula y conexiones.
- La asociación entre sistema de cavidades y hoyos.

Éstas se pueden optimizar haciendo uso de clases intermedias:

- Entre Molécula y átomo, se puede usar la clase: `lista_de_atomos`. 1-n
- Entre Molécula y dipolos, se puede hacer uso de la clase: `lista_de_dipolos`. 1-n
- Entre Molécula y conexiones, se puede usar la clase: `lista_de_conexiones`. 1-n
- Entre sistema de cavidades y hoyos se puede hacer uso de la clase: `lista_de_hoyos`. 1-n

En una asociación de 1-n, los atributos son parte del lado n, en éste caso, los atributos serán parte de átomo, dipolo, conexión, hoyo.

En el caso presente es posible que no se usen asociaciones ya que las clases correspondientes a éstas vienen implícitas como atributos de las clases implicadas, como son la clase de molécula y la de sistema de cavidades, y no es necesario que sean clases como tales.

9.2.2.7 Representar objetos

La representación de los objetos, es decir, el tipo de datos que corresponden a los atributos de cada objeto, se realizó en base a las necesidades de cada clase, así como también tratando de minimizar el espacio reservado en memoria para dichas clases.

En esta parte se podrán usar los tipos de datos de plantillas para c++ (los *templates*), de manera que en la clase base se define el “molde” de tipo *template* y en las clases derivadas se definen las estructuras correspondientes para que los datos tipo plantilla correspondan a los de cada objeto.

Los atributos de los contenedores de datos corresponden al formato característico de cada tipo de archivo.

Los tipos de datos de los atributos de la clase se enumeran a continuación del nombre de cada clase:

9.2.2.7.1 Módulo de visualización

Nombre de la clase:	Objeto_grafico
Propiedades:	<ul style="list-style-type: none"> • Color, arreglo entero de 3 localidades: int color[3]. • Etiqueta, identificador entero corto: int etiqueta. • Origen, arreglo flotante de coordenadas en X, Y, Z: float origen[3]: origen[0] es X, origen[1] es Y, origen[2] es Z.

Nombre de la clase:	Plano_potencial
Propiedades:	<ul style="list-style-type: none"> • T_plan, valor de punto flotante, que indica en que plano de el eje Y se va a dibujar el potencial: float t_plano. • Número de partículas por plano, valor entero que indica el número de partículas que hay por plano: int parti_X_plano (NPX). • Máximo potencial, valor de punto flotante que indica el máximo potencial: float potmax. • Mínimo potencial, valor de punto flotante que indica el mínimo potencial: float potmin. • Tsup, valor entero que indica sobre cual de los 3 planos se dibujará el plano de potencial: int tsup. • Rango de potenciales, es un número real que indica el rango que delimitará a los potenciales: float rangopot. • Alto: va a ser tomado por el alto de la caja, el cual va a ser el lado de la caja en el eje Y: float alto. • Ancho: va a ser tomado por el ancho de la caja, el cual va a ser el lado de la caja en el eje X: float ancho • Largo: va a ser tomado por el largo de la caja, el cual va a ser el lado de la caja en el eje Z: float largo.

Nombre de la clase:	Objeto_estatico
Propiedades:	<ul style="list-style-type: none"> • Grosor de línea, valor de punto flotante que indica el grueso de la línea: float grosor_linea. • Tipo de línea, valor real que indica el tipo de la línea de los objetos estáticos: float tipo_linea.

Nombre de la clase:	Caja
Propiedades:	<ul style="list-style-type: none"> • Tipo de caja, valor entero que indica el tipo de caja de la molécula: int Tipo_caja. • Lados, número entero que indica el número de lados de la caja dependiendo de su tipo: short int lados. • Angulo a, valor real que indica el ángulo alfa de la caja computacional: float angulo_a. • Angulo b, valor real que indica el ángulo beta de la caja computacional: float angulo_b. • Angulo g, valor real que indica el ángulo gamma de la caja computacional: float angulo_g. • Longitud de lado, valor real que indica el largo de los lados de la caja: float longitud_lado. • Longitud de ancho, valor real que indica el ancho de la caja: float longitud_ancho. • Longitud de alto, valor real que indica el alto de la caja: float longitud_alto.

Nombre de la clase:	Eje
Propiedades:	<ul style="list-style-type: none"> • Punto X, coordenada destino en X, valor de punto flotante que indica la segunda coordenada en X de la conexión: • Punto Y, coordenada destino en X, valor de punto flotante que indica la segunda coordenada en X de la conexión: • Punto Z, coordenada destino en X, valor de punto flotante que indica la segunda coordenada en X de la conexión: float puntos_destinos[3]: puntos_destinos[0] es punto_X, puntos_destinos[1] es punto_Y, puntos_destinos[2] es punto_Z. • Longitud, valor de punto flotante que indica la longitud de los ejes: float longitud.

Nombre de la clase:

Objeto_dinamico

Propiedades:

- Calidad, valor de punto flotante que indica la calidad de definición del objeto; **float calidad.**
- Material: define el material para dibujar los objetos tridimensionales, como una matriz de valores que definen las propiedades del material: **struct material {int GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE; float material[4]}**
- Selección, (Grupo-Caja, Grupo-Esfera, Lista, Individual, Todos, No Selección), valor entero que indica que tipo de selección se va realizar: **int seleccion.**
- Transparencia, valor entero corto que indica si el objeto es transparente o no: **short int transparencia.**
- Slices, Divisiones alrededor del eje Z, valor entero que indica el número de divisiones alrededor de Z: **int slices.**
- Stacks, Divisiones a lo largo del eje Z, valor entero que indica el número de divisiones a lo largo de Z: **int stacks.**
- Radio, valor flotante que indica el radio del objeto y depende del tipo de éste: **float radio.**
- Visualización (Visibilidad [Visible-No Visible], Calidad [Con calidad – Sin calidad], Densidad [transparente – Difuminado], Estructura [Sólido – Malla]), Es un arreglo con 4 valores enteros cortos, que define la manera en que será visualizado el objeto dinámico:
int visualizacion[4]: visualizacion[0] = 1 ó 0, visualizacion[1] = 1 ó 0, visualizacion[2] = 1 ó 0, visualizacion[3] = 1 ó 0
- Modificación (Modificado – No Modificado), valor entero corto que indica si es modificado o no el objeto: **short int modificado.**

Nombre de la clase:

Esfera

Propiedades:

- Molécula asociada, número entero, correspondiente a la etiqueta que identifica a la molécula a la cual pertenece la partícula, hoyo, o dipolo: **int molecula_asociada.**
- Visualización (Representación [Esfera – Puntos]), define como será representado el objeto del tipo esfera, es un valor entero corto: **short int representación.**

Nombre de la clase:

Dipolo

Propiedades:

- Carga, es un valor entero corto que indica la carga que tiene la partícula: **short int carga.**
- Partícula Asociada, número entero que identifica a la partícula asociada con el dipolo: **int partícula_asociada.**
- Distancia, valor real que indica la distancia entre la partícula y su momento eléctrico: **float distancia.**
- Angulo, valor real que indica el ángulo existente entre la partícula y su momento eléctrico: **float angulo.**
- Visualización (Representación [Elipse – Puntos]), define como será representado el objeto del tipo esfera, es un valor entero corto: **short int representación.**
- Edición (Editado, Agrupado, Agregado, Unido, No Editado), valor entero corto que indica el estado de edición del objeto de tipo dipolo : **short int edición.**
- Excentricidad, valor que indica la excentricidad del dipolo y que debe ser diferente de 1: **float excentricidad**

Nombre de la clase:	Atomo
Propiedades:	<ul style="list-style-type: none">• Creado (Temporal, No temporal), indica de que tipo fue creado el objeto átomo: short int creado.• Tipo de elemento, entero corto que indica de que tipo de elemento es la partícula: int tipo_elemento.• Carga, valor entero corto que indica la carga que tiene la partícula: short int carga.• Selección (Especie), valor entero corto que indica el tipo de selección aplicada: short int selección.• Conexión (Especie, No Especie, Sin Conexión), valor entero corto que indica si el objeto átomo ésta o no conectado: short int conexión.• Visualización (Representación [Esfera – Punto]), define cómo será representado el objeto del tipo esfera y es un valor entero corto: short int representacion.• Edición (Editado, Agrupado, Agregado, Unido, No Editado), valor entero corto que indica el estado de edición del objeto del tipo dipolo : short int edicion.

Nombre de la clase:	Hoyo
Propiedades:	<ul style="list-style-type: none">• Lista de vecinos, estructura de datos que almacena valores enteros de identificadores de átomos (etiquetas): int *lista_vecinos (uso de memoria dinámica).• Visualización (Representación [Esfera – Puntos]), define como será representado el objeto del tipo esfera y es un valor entero corto: short int representacion.

Nombre de la clase:	Conexión
Propiedades:	<ul style="list-style-type: none">• longitud, valor real que indica la longitud de la conexión: float longitud.• Punto X, es la coordenada en X del punto final de la conexión (destino): float punto_x.• Punto Y, es la coordenada en X del punto final de la conexión (destino): float punto_y• Punto Z, es la coordenada en X del punto final de la conexión (destino): float punto_z.• Edición (Editado, Agrupado, Agregado, Unido, No Editado): valor entero corto que indica el estado de edición del objeto del tipo conexión: short int edicion.• Conectada (con 1 átomo – con 2 átomos – no conectada), es un valor entero corto que indica si ésta conectada con un átomo, con dos o con ninguno: short int conectada.• Válida (Es válida – No es válida), un valor entero corto que indica si es una conexión válida o no lo es: short in valida.• Visualización (Representación [Cilindro – Línea]), define cómo será representado el objeto del tipo conexión y es un valor entero corto: short int representacion.

Nombre de la clase:

Celda

Propiedades:

- Número de Moléculas, valor entero que indica el número de moléculas que forman parte de la Celda: **float numero_de_moleculas**.
- Lista de Moléculas, lista de objetos del tipo molécula: **molecula *lista_moleculas**.
- Caja, objeto del tipo caja que delimitará el espacio computacional: **Caja CajaComp**.
- Sistema de cavidades, objeto del tipo sistema de cavidades: **Sistema_de_cavidades Sist_Cav**.
- Visualización (Celda - Celda unitaria - Estructura cristalina), valor entero corto que indica el estado de la visualización de la celda dentro de la escena gráfica: **short int visualizacion**.

Nombre de la clase:

Objeto_animado

Propiedades:

- Número de cuadros, valor entero que indica el numero de cuadros que tiene la animación de la molécula: **int n_cuadros**.
- Lista de datos por cuadro, puntero a una estructura que almacena los datos para cada cuadro de la animación: **template lista_datos_x_cuadro[n_cuadros]**.
- Sentido de animación (Avance, Pausa, Retroceso), valor entero corto que indica en que sentido va la animación: **short int sentido_anim**.
- Número de cuadro actual, valor entero que indica en que número de cuadro ésta actualmente la animación: **int n_cuadro_actual**.
- Visualización (Visibilidad [Visible - No Visible], Calidad [Con calidad - Sin calidad]), es un arreglo con 2 valores enteros cortos, que define la manera en que será visualizado el objeto animado: **int visualizacion[2]: visualizacion[0] = 1 ó 0, visualizacion[1] = 1 ó 0**,
- Estructura [Sólido - Malla]), es un valor entero que indica de que tipo será la estructura del objeto animado: **short int estructura**.
- Modificación (Modificado - No Modificado), es un valor entero que indica si el objeto es modificado o no: **short int modificacion**
- Selección (Seleccionado - No seleccionado), es un valor entero corto que indica si el objeto ésta o no seleccionado, se usa principalmente cuando se tiene más de un objeto del tipo molécula: **short int seleccion**.

Nombre de la clase:

Molécula

Propiedades:

- Nombre, cadena de caracteres que identifica a la molécula: **char *nombre**.
- Descripción, cadena de caracteres que describe brevemente a la molécula: **char *descripción**.
- Lista de átomos, lista de objetos del tipo atomo, es un vector: **atomo *lista_atomos**.
- Lista de dipolos, lista de objetos del tipo dipolos, es un vector: **dipolo *lista_dipolos**.
- Lista de conexiones, lista de objetos del tipo conexion, es un vector: **conexion *lista_conexiones**.
- Numero de conexiones, valor entero que indica el número de conexiones de la molécula: **int num_conex**.
- Numero de Átomos, valor entero que indica el número de átomos de la molécula: **int num_atom**.
- Numero de Dipolos, valor entero que indica el número de dipolos de la molécula: **int num_dip**.
- Conexión (Por especie – No por Especie – Sin conexión), es un valor entero corto que indica de que manera se conectan las partículas: **short int conexion**.
- Edición (Editada – Agregada – Unida - No Editada), es un valor entero corto que indica el estado de edición de la molécula: **short int edicion**.
- Visualización (Representación [Esferas - Cilindros – Puntos - Líneas], Densidad [transparente – Difuminado]), es un arreglo con 2 valores enteros cortos, que define la manera en que será visualizada la molécula: **int visualizacion[2]: visualizacion[0] = 1 ó 0, visualizacion[1] = 1 ó 0**.

Nombre de la clase:

Sistema_de_cavidades

Propiedades:

- Molécula asociada, es un identificador entero corto de la molécula asociada: **int mol_asociada**.
- Número de hoyos, número entero que indica la cantidad de hoyos existentes en la molécula: **int n_hoyos**.
- Lista de hoyos, lista de objetos de tipo hoyo: **hoyo *lista_hoyos**.
- Visualización (Representación [Esfera – Punto], Densidad [transparente – Difuminado]), es un arreglo con 2 valores enteros cortos, que define la manera en que será visualizada la molécula: **int visualizacion[2]: visualizacion[0] = 1 ó 0, visualizacion[1] = 1 ó 0**.

Nombre de la clase:

Superficie_potencial

Propiedades:

- Número de Planos, número entero que dice cuantos planos de potencial tiene la molécula: **int n_planos**.
- Lista de planos potenciales, lista de objetos de tipo plano potencial: **Plano_potencial *lista_de_planos**.
- Estructura cristalina asociada, número entero corto para identificar la estructura cristalina perteneciente a la molécula: **int estructura_asociada**.
- Visualización (Representación [Superficie – Plano], Selección (Plano – Superficie – No seleccionada), es un arreglo con 2 valores enteros cortos, que define la manera en que será visualizada la molécula: **int visualizacion[2]: visualizacion[0] = 1 ó 0, visualizacion[1] = 1 ó 0**.

Nombre de la clase:

Escena_grafica

Propiedades:

- Fondo, número entero largo, hexadecimal de 3 bytes: **long int fondo**.
- Transformación espacial (En Rotación – En Translación – En Escalamiento), es un valor entero corto que indica el tipo de transformación espacial que se realiza: **short int transformacion_espacial**.
- Perspectiva, es un valor entero corto que indica si la vista de la escena tiene la perspectiva activada o no (Activada - Desactivada): **short int perspectiva**.
- Modificación (Molécula - Ejes – Puntero 3D – Fondo – Celda), es un valor entero corto que indica que es lo que se va a modificar dentro de la escena gráfica: **short int modificacion**.
- Ejes, son 3 objetos de tipo Eje: **Eje Eje_X, Eje_Y, Eje_Z**.
- Estructura cristalina, objeto del tipo estructura cristalina: **Estructura_cristalina estruc_crist**.
- Puntero 3D, objeto del tipo Puntero_3D: **Puntero_3D puntero**.

Nombre de la clase:

Puntero_3D

Propiedades:

- Color, arreglo de tres valores enteros que definirá las variaciones de color en rojo, verde y azul. para dar color al puntero de ratón: **int color[3]: color[0] es el valor rojo, color[1] es el valor rojo, color[2] es el valor rojo**.
- Origen x, coordenada en X del cursor, número real: **float origen_x**.
- Origen y, coordenada en Y del cursor, número real: **float origen_y**.
- Origen z, coordenada en Z del cursor, número real: **float origen_z**.
arreglo correspondiente a los tres valores de las coordenadas del cursor, float origen[3]: origen[0] es origen_x, origen[1] es origen_y, origen[2] es origen_z.
- Tipo, cadena de caracteres para identificar el tipo de ayuda que se solicita: **char *tipo**.

9.2.2.7.2 Módulo de cálculos

Nombre de la clase:

Proceso_calculo

Propiedades:

- Nombre, es una cadena de caracteres para nombrar el proceso: **char *nombre**.
- Tipo, es un valor entero corto que sirve para identificar el tipo del proceso: **short int tipo**.
- Vector de datos (T), es el vector que contiene los datos necesarios para realizar el cálculo especificado, es de tipo *template*: **T **vector_datos**.

Nombre de la clase:

Calculo_de_imagen

Propiedades:

- Vector de datos de imagen, es el vector que contiene la información necesaria para generar la imagen y es de tipo *template*: **T **vector_imagen**.
- Tipo de Imagen, valor entero corto que indica de que tipo es la imagen: **short int tipo_imagen**.

Nombre de la clase: **Calculo_de_dinamica**

Propiedades:

- Vector de datos, es el vector que contiene los datos necesarios para el cálculo de la dinámica y es de tipo *template*: **T**vector_dinamica**.
- Proceso remoto asociado, es el identificador del proceso de dinámica generado: **short int id_proceso_R_asociado**.

Nombre de la clase: **Calculo_generado_por_particulas**

Propiedades:

- Vector de datos de partículas, vector de datos que almacena la información de las partículas y es de tipo *template*: **T**datos**.
- Número de partículas, es un valor entero que indica el número de partículas de la molécula: **int n_particulas**.
- Tipo de cálculo, valor entero corto que sirve para identificar (verificar y diferenciar) el tipo de cálculo: **int tipo_calculo**.
- Vector de datos generado, es el vector de datos que se genera cuando se realiza un cálculo. y es de tipo *template* (puede ser de diferente tipo): **T*datos_generados**.

Nombre de la clase: **Calculo_de_magnitudes_y_angulos**

Propiedades:

- Lista de datos seleccionados, es un vector que contiene todos los datos de los objetos seleccionados y puede ser una de dos estructuras que definen magnitud: **struct magnitud {float x1,y1,z1,x2,y2,z2;}** o: **struct magnitud {float x1,y1,z1,x2,y2,z2, x3,y3,z3;}**
- Tipo de selección, es un valor entero corto que indica de que tipo será la selección a realizar: **short int tipo_de_seleccion**.

Nombre de la clase: **Calculo_remoto**

Propiedades:

- Proceso remoto asociado, valor entero corto que indica el número identificador del proceso: **int proceso_R_asociado**.

Nombre de la clase: **Calculo_local**

Propiedades:

- Proceso local asociado, valor entero corto que indica el número identificador del proceso: **int proceso_L_asociado**.

9.2.2.7.3 Módulo de editor molecular

Nombre de la clase:	Editor_molecular
Propiedades:	<ul style="list-style-type: none"> • Vector de partículas, es el vector de datos de partículas que contiene el editor molecular y es una copia del vector de partículas inicial: T **vector_parti_editor; • Tipo de selección, es un valor entero corto que indica el tipo de selección que se va a realizar: short int tipo_selección. • Tipo de acción, es un valor entero corto que indica el tipo de acción realizada por el Editor molecular sobre los objetos seleccionados: short int tipo_accion. • Vector de objetos gráficos (T), es el vector de datos que contiene toda la información de los objetos gráficos temporales que usa el editor molecular: T *vector_grafico; • Vector de historia, es el vector de datos que contiene la información de todas las acciones realizadas: struct *vector_historia {partis **partil; hoyos **hoyol; conexs **conexl; dips **dip1;} y las estructuras son struct partil {particula parttemp; short int accion; short int contraccion}, struct hoyos {particula parttemp; short int accion; short int contraccion}, struct conexs {particula parttemp; short int accion; short int contraccion}, struct dips {particula parttemp; short int accion; short int contraccion}

Nombre de la clase:	Estructura_cristalina
Propiedades:	<ul style="list-style-type: none"> • Nombre, cadena de caracteres que indica el nombre de la estructura: char *nombre • Lista de celdas, arreglo de datos de objetos de tipo celda: Celda *lista_celdas. • Número de celdas, valor entero corto que indica el número de celdas: int n_celdas. • Lista de planos potenciales, arreglo de datos que almacena la información de los planos potenciales: Plano_potencial *lista_potenciales.

Nombre de la clase:	Indicador_de_seleccion
Propiedades:	<ul style="list-style-type: none"> • Tipo de objeto seleccionado, es un valor entero corto que indica de que tipo es el objeto seleccionado: short int tipo_objeto. • Modo de selección, es un valor entero corto que indica de que manera se está realizando la selección: short int modo_seleccion. • Lista de etiquetas de objetos seleccionados, es un arreglo de estructuras que contiene las etiquetas de los objetos que han sido seleccionados: struct etiqueta {short int cont (contador para las etiquetas); char tipo} etiqueta *lista_etiquetas.

Nombre de la clase:	Modificador_de_propiedades
Propiedades:	<ul style="list-style-type: none"> • Lista de etiquetas de objetos seleccionados, vector que almacena las etiquetas de los objetos de la escena que se han seleccionado: struct etiqueta {short int cont; char tipo} etiqueta *lista_etiquetas. • Tipo de selección, entero corto que indica el tipo de selección que se realizará: short int tipo_selección. • Propiedad a cambiar, valor entero corto que indica el tipo de propiedad a ser modificada: short int propiedad. • Vector de historia, lista de los datos que cambiaron y de los objetos cuyas propiedades han cambiado: struct historia {etiqueta_compuesta etiq_hist; short int propiedad; T valor anterior;}

9.2.2.7.4 Módulo de monitor de procesos

Nombre de la clase: **Monitor_de_procesos**

Propiedades:

- Acción sobre proceso de cálculo, es un entero corto que indica la operación que se va a realizar sobre el proceso: **short int accion.**
- Tabla de procesos, es una estructura de datos que contiene información importante sobre el estado y desempeño de todos los procesos de cálculo ejecutados: **struct tabla {short int proceso, short int estado} tabla *tabla_procesos.**
- Tipo de mensaje, valor entero corto que indica el tipo de mensaje emitido por el monitor o hacia el monitor: **short int tipo_de_mensaje.**
- Número de procesos locales, valor entero, es el número de procesos locales monitoreados en ese momento: **int no_procesos_locales.**
- Lista de procesos locales, lista que hace referencia a los procesos que se están ejecutando localmente: **int *lista_procesos.**
- Número de procesos remotos, valor entero, es el número de procesos remotos monitoreados en ese momento: **int no_procesos_remotos.**
- Lista de procesos remotos, lista que hace referencia a los procesos que se están ejecutando remotamente: **int *lista_proceso_remoto.**

Nombre de la clase: **Procesos**

Propiedades:

- Estado del proceso, valor entero corto que sirve para indicar en que estado se encuentra el proceso. (Activo, Residente, Suspendido, Perdido, Terminado): **short int estado_proceso.**
- Nombre del proceso, cadena que sirve para identificar y diferenciar los procesos: **char *nombre_proceso.**
- Tipo de proceso, valor entero corto que indica el tipo de proceso (Local -Remoto): **short int tipo_proceso.**
- Fecha de inicio, estructura que almacena la fecha en que inicia el proceso: **struct fecha {char dia[3], mes[3], año[4]} fecha fecha_inicio.**
- Hora de inicio, estructura que almacenan la hora en que inicia el proceso: **struct hora {char hora[2], minuto[2], segundo[2]} hora hora_inicio.**
- Fecha de terminación, estructura que almacena la fecha en que termina el proceso: **fecha fecha_terminacion.**
- Hora de terminación, estructura que almacenan la hora en que termina el proceso: **hora hora_terminacion.**
- Número de proceso, valor entero corto que se utiliza para identificar al proceso: **short int no_proceso.**

Nombre de la clase: **Proceso_local**

Propiedades:

- Vector de datos de entrada, estructura de datos *template*, que almacena los datos que se envían al proceso local especificado: **T *vector_datos_entrada.**
- Vector de datos de salida, estructura de datos *template*, que almacena los datos que salen del proceso local especificado: **T *vector_datos_salida.**
- Tipo de proceso, valor entero corto que indica de que tipo es el proceso actual (Remoto-Local): **short int tipo_proceso.**
- Tipo de cálculo, entero corto que indica de que tipo es el cálculo indicado: **short int tipo_calculo.**
- Numero de elementos, valor entero que indica el número de elementos que contiene el vector de datos: **int no_elementos.**

Nombre de la clase: **Proceso_remoto**

Propiedades:

- Dirección IP, cadena de caracteres que representa el identificador de Red: **char Dir_IP[15]**.
- Login, cadena de caracteres que indica el nombre del usuario que será utilizado para iniciar el proceso en el equipo remoto: **char *login**.
- "Password", cadena de caracteres que indica la clave de acceso del usuario: **char *password**.
- Puerto, número entero corto que indica el puerto por el cual se establecerá la comunicación: **int puerto**.
- Nombre del archivo de entrada, cadena de caracteres que indica el nombre del archivo de entrada para el proceso remoto: **char *nom_arch_ent**.
- Nombre del archivo de salida, cadena de caracteres que indica el nombre del archivo de salida: **char *nom_arch_sal**.
- Tipo de proceso, valor entero corto que indica de que tipo es el proceso actual (Remoto - Local): **short int tipo_proceso**.
- Tipo de cálculo, valor entero corto identificador de tipo de cálculo: **short int tipo_calculo**.
- Número de elementos, valor entero corto que indica el número de elementos: **int no_elementos**.

9.2.2.7.5 Módulo de contenedores

Nombre de la clase: **Contenedor_de_datos**

Propiedades:

- Nombre del archivo, es una cadena que contiene el nombre del archivo de datos: **char *nombre_archivo**.
- Tipo del archivo, valor entero que indica el tipo del archivo de datos: **short int tipo_archivo**.
- Vector de datos: éste vector de datos está predefinido para esperar distintos tipos dependiendo del contenedor de datos específico. **T *vector_datos**.
- Número de elementos, valor entero corto que indica el número de elementos que contiene el archivo: **short int no_elementos**.

Nombre de la clase: **Contenedor_de_particulas**

Propiedades:

- Lado x, valor real de un lado de la caja: **float lado_x**.
- Lado y, valor real de un lado de la caja: **float lado_y**.
- Lado z, valor real de un lado de la caja: **float lado_z**.
- Ángulo alfa, valor real del ángulo alfa de la caja: **float angulo_alfa**.
- Ángulo beta, valor real del ángulo alfa de la caja: **float angulo_beta**.
- Ángulo gama, valor real del ángulo alfa de la caja: **float angulo_gama**.
- Número de cuadros, valor entero corto que es el número de cuadros: **short int no_cuadros**.
- Número de partículas, valor entero que indica el número de partículas: **int no_particulas**.
- Vector de datos, es un vector que contiene los siguientes datos:
 - Etiqueta, valor entero corto que identifica a la partícula: **short int etiqueta**.
 - X, valor x de las coordenadas de la partícula: **float x**.
 - Y, valor y de las coordenadas de la partícula: **float y**.
 - Z, valor z de las coordenadas de la partícula: **float z**.
 - Carga, valor real que indica la carga del elemento: **float q**.
 - Tipo, valor entero que indica el tipo del elemento: **int tipo**.

Nombre de la clase: **Contenedor_de_hoyos**

Propiedades:

- Número de cuadros, valor entero corto que indica el número de cuadros: **short int no_cuadros.**
- Número de hoyos, valor entero que indica el número de hoyos: **int no_hoyos.**
- Archivo de partículas asociado, es una cadena de caracteres que indica el nombre del archivo de partículas a la cual pertenecen los hoyos: **char *archivo_parti_asociado.**
- Vector de datos, se definirá una estructura con los siguientes campos:
 - Etiqueta, valor entero que identifica a cada hoyo: **short int etiqueta.**
 - X, valor flotante x de las coordenadas del hoyo: **float x.**
 - Y, valor flotante y de las coordenadas del hoyo: **float y.**
 - Z, valor flotante z de las coordenadas del hoyo: **float x.**
 - Radio, valor flotante del radio del hoyo: **float radio.**

Nombre de la clase: **Contenedor_de_imagen**

Propiedades:

- Formato de Imagen, estructura de datos que contiene la información de la imagen: **struct formato_imagen { datos característicos de cada tipo de imagen };**
- Calidad de Imagen, valor real que indica la calidad de la imagen: **float calidad.**
- Largo (píxeles), valor real que indica el largo de la imagen: **float largo.**
- Ancho (píxeles), valor real que indica el ancho de la imagen: **float largo.**

Nombre de la clase: **Contenedor_de_potencial**

Propiedades:

- Número de cuadros, valor entero que indica el número de cuadros: **short int no_cuadros.**
- Número de potenciales, valor entero que indica el número de planos de potencial: **int no_potenciales**
- Archivo de partículas asociado, cadena de caracteres que indica el archivo asociado de partículas: **char *archiv_asoc_parti.**
- Máximo potencial, es el valor del máximo potencial: **float potmax.**
- Mínimo potencial, es el valor del mínimo potencial: **float potmin.**
- Vector de datos, es una estructura de datos que contiene la siguiente información:
 - Etiqueta, valor entero que identifica al plano de potencial: **int etiqueta.**
 - T_plano: valor que indica en que plano de el eje Y se va a dibujar el potencial: **float t_plano.**
 - Número de partículas por plano, es el número de partículas que hay por plano: **int parti_X_plano (NPX).**
 - Máximo Potencial, valor que indica el máximo potencial: **float potmax.**
 - Mínimo Potencial, valor que indica el mínimo potencial: **float potmin.**
 - Tsup, valor que indica sobre cual de los 3 planos se dibujará el plano de potencial: **int tsup.**
 - Rango de Potenciales, valor que indica el rango que delimitará a los potenciales: **float rangopot.**
 - Alto: es el alto de la caja, el cual va a ser el lado de la caja en el eje Y: **float alto.**
 - Ancho: es el ancho de la caja, el cual va a ser el lado de la caja en el eje X: **float ancho.**
 - Largo: es el largo de la caja, el cual va a ser el lado de la caja en el eje Z: **float largo.**

Nombre de la clase:**Contenedor_de_dipolos****Propiedades:**

- Número de dipolos, valor entero que indica el número de dipolos: **int no_dipolos.**
- Número de cuadros, valor entero que indica el número de cuadros: **int no_cuadros.**
- Archivo de partículas asociado, cadena de caracteres, que almacena el nombre del archivo de partículas asociado: **char *archivo_asociado.**
- Vector de datos, es la estructura de datos que contiene los siguientes datos:
 - Etiqueta, valor entero que identifica al dipolo: **int etiqueta.**
 - X1, valor real, x de las coordenadas de la esfera asociada: **float x1.**
 - Y1, valor real, y de las coordenadas de la esfera asociada: **float y1.**
 - Z1, valor real, z de las coordenadas de la esfera asociada: **float z1.**
 - X2, valor real, x del dipolo: **float x2.**
 - Y2, valor real, y del dipolo: **float y2.**
 - Z2, valor real, z del dipolo: **float z2.**
 - Carga, valor real que representa la carga del elemento: **float q.**

La siguiente clase (Contenedor_de_inicia_dinamica) ésta formada por las variables que constituyen el archivo **inicia.loc**, que es el archivo de configuración utilizado para generar la dinámica molecular, y el cual se envía a la CRAY para generar los datos de las distintas configuraciones moleculares. Los atributos están distribuidos en 8 estructuras. debido a que al pasar los datos al programa en lenguaje de programación fortran, éste los maneja dentro de 8 arreglos

Nombre de la clase:

Contenedor_de_inicia_dinamica

Propiedades:

- **struct A**
 - KINIT : Valor entero corto que sirve para leer el tipo de configuración al tiempo t y t-dt: **short int KINIT**
 - ITMAX : Número máximo de iteraciones (configuraciones): **int ITMAX**
 - IRE : Número de pasos que promedia para hacer el reporte: **int IRE**
 - IMV : Cada cuantos pasos guarda una imagen del sistema: **int IMV**
 - IAN : Cada cuantos pasos promedia los ángulos: **int IAN**
 - IAF : Cada cuantos pasos almacena R para calcular función automática c: **int IAF**
 - TFIN : La temperatura que se espera tener durante el proceso: **float TFIN**.
 - INAP : Número total de partículas en el sistema: **int INAP**.
 - IFCHD : Número de pasos de expansión (+) o compresión (-) del sistema: **int IFCHD**
 - FDENS : Densidad final después de la compresión o expansión: **int FDENS**
 - IQST : Cada cuantos pasos obtiene las estadísticas de carga: **int IQST**.
- **struct B**
 - IZERO : Número de configuración al que se reinician a cero los contadores: **int IZERO**.
 - IBKUP : Índice auxiliary (dummy): **int IBKUP**.
 - I3BF : Valor entero corto que indica si calcula las fuerzas de 3 cuerpos o no: **short int I3BF**.
 - ITCIN : Frecuencia de llamadas a therma (control de temperatura): **int ITCIN**.
 - IRENO : Valor entero corto que sirve para llamar a renorm (reescala las velocidades): **short int IRENO**.
 - IINT : Valor entero corto para indicar si se usan las rutinas de interacción de salto de rana o de VERNET: **short int IINT**.
 - IRECH : Número de pasos en los que se modifican los parámetros de potencial: **int IRECH**.
 - IVEL : Número de pasos después de los cuales se guardarán todas las velocidades: **int IVEL**.
 - DT : Valor flotante que indica el paso de integración de las ecuaciones de movimiento: **float DT**.
- **struct C**
 - JUMP : Es un multiplicador que permite modificar IRE para comparar distintos pasos de integración: **int JUMP**.
 - ITCAN : Valor entero corto que indica: 0 sin control de temperatura ni presión, 1 con control de temperatura usando renormalización de velocidades, 2 con control de temperatura usando control canónico y 3 con control de temperatura y presión usando control canónico e isobárico: **short int ITCAN**.
- **struct D**
 - TQUE : Temperatura de salto (usualmente igual a TFIN): **float TQUE**.
 - PEXT : Valor flotante que indica la presión que se espera tener durante el proceso: **float PEXT**.
 - ICONI : Índice auxiliar (dummy): **int ICONI**.
 - ICONF : Índice auxiliar (dummy): **int ICONF**.
- **struct E**
 - NTIPOS : Número de especies diferentes en el sistema: **int NTIPOS**.
 - Ptle : Nombre de la partícula de tipo K: **char *nombre_part**.
 - Rmtpcl(Ki): Masa de la partícula Ki: **float Rmtpcl**.
 - Nformls: número de fórmulas completas en la composición del sistema usado para calcular la energía en Kjoules/mol: **int Nformls**.
- **struct F**
 - PTNCL : Nombre del potencial para la nva-ésima interacción: **char * nombre_pot**
 - Npmax : Número de parámetros que aparecen en el potencial: **int Npmax**.
 - RCP : El valor del parámetro K-ésimo para la interacción nva-ésima: **double RCP**.
- **struct G**
 - NFOCOS : Es el número de termostatos que van controlar la temperatura del sistema (cuando ITCAN es 2 o 3): **short int NFOCOS**.
 - KNK (Ki) : Es el número de foco que controla la temperatura de la especie K: **short int KNK**.

Contenedor_de_inicia_dinamica (continua)

- **Taut** (knf) : Valor flotante que indica el periodo de oscilador para el foco knf: **double Taut**
- **Sedacan** (knf): Condición inicial para la integración de las ecuaciones de movimiento de los focos: **float Sedacan**.
- **TauP** : Periodo de oscilador para el pistón que controla presión (cuando ITCAN es 3): **float TauP**.
- **Eptpn** : Condición inicial para integrar las ecuaciones de movimiento del pistón: **float Eptpn**.
- **struct H** : Variables necesarias para comparar $g(r)$ con el experimento por el momento son fijas:
 - **lrdfnw** : Variable auxiliar si ITCAN es 1 para indicar que debe leer mc y coef: **short int lrdfnw**.
 - **mc** (10) : Arreglo de 10 localidades que sirve para renormalizar $g(r)$: **int mc[10]**.
 - **coef**(4) : Arreglo de 4 localidades que debe de leer después de leer las variables canónicas. si ITCAN es 2: **float coef[4]**.

La siguiente clase toma sus atributos del archivo de configuración **confin00**, el cual contiene los datos de la configuración inicial.

Nombre de la clase: **Contenedor_de_configuración_de_dinamica_t**

Propiedades:

- Densidad inicial, valor real que indica la densidad base: **float densidad_init**.
 - Temperatura inicial, valor real que indica la densidad base: **float temper_init**.
 - **Arreglo de estructuras que contiene los siguientes valores:**
 - **struct config_init** {
 - Número de partícula, valor entero que indica el numero de la partícula: **int no_particula**.
 - **RX(I)** : Arreglo de valores reales Rx: **float rxI**.
 - **RY(I)** : Arreglo de valores reales Ry: **float ryI**.
 - **RZ(I)** : Arreglo de valores reales Rz: **float rzI**.
 - **VX(I)** : Arreglo de datos flotantes Vx: **float vxI**.
 - **VY(I)** : Arreglo de datos flotantes Vy: **float vyI**.
 - **VZ(I)** : Arreglo de datos flotantes Vz: **float vzI**.
 - **ITYP(I)** : Valor entero corto que indica el tipo de partícula: **short int ltyp**.
- }; **config_init [108];**

La siguiente clase es la correspondiente a los datos del archivo de configuración **confin.loc**, el cual contiene los datos de la configuración en un tiempo t

Nombre de la clase: **Contenedor_de_configuración_de_dinamica_t-dt**

Propiedades:

- Densidad : Valor real que indica la densidad en un tiempo t: **float densidad_t**.
- Temperatura : Valor real que indica la densidad base: **float temper_t**.
- **RX(I)** : Arreglo de 35 tercias de valores reales Rx: **float rxI[3][35]**.
- **RY(I)** : Arreglo de 36 tercias de valores reales Ry: **float ryI[3][36]**.
- **RZ(I)** : Arreglo de 36 tercias de valores reales Rz: **float rzI[3][36]**.
- **RX0** : Arreglo de 36 tercias de valores reales Rx0: **float rx0[3][36]**.
- **RY0** : Arreglo de 36 tercias de valores reales Ry0: **float ry0[3][36]**.
- **RZ0** : Arreglo de 36 tercias de valores reales Rz0: **float rz0[3][36]**.
- **Q(I)** : Arreglo de 37 tercias de valores reales Qi: **float Qi[3][37]**.
- **VX(I)** : Arreglo de 37 tercias de valores reales VXi: **float vxI[3][37]**.
- **VY(I)** : Arreglo de 36 tercias de valores reales VYi: **float vyI[3][36]**.
- **VZ(I)** : Arreglo de 18 grupos de 6 de enteros cortos reales Vz: **short int vzI[6][18]**.

Contenedor_de_configuración_de_dinamica_t-dt (continua)

- ITYP(I) : Arreglo de datos reales I_{typ} de 36 localidades: **float I_{typ}[3][36]**.
- V_itcan : Valor que indica si el valor de itcan es 2 o 3: **short int V_itcan**.
- V_itcan1 : Valor que indica si el valor de itcan es 3: **short int V_itcan1**.
- Sedacan : Condición inicial en un tiempo t para la integración de las ecuaciones de movimiento de los focos : **float Sedacan**.
- Sedacan0 : Siguiete condición en un tiempo t para la integración de las ecuaciones de movimiento de los focos: **float Sedacan0**.
- Tinst : Valor real: **float Tinst**.
- Eptpn : Condición inicial en un tiempo t para integrar las ecuaciones de movimiento del pistón: **float Eptpn**.
- Eptn : Valor real: **float Eptn**.
- Vol : Valor real que indica el volumen: **float Vol**.
- Eptpn0 : Condición siguiente en un tiempo t para integrar las ecuaciones de movimiento del pistón: **float Eptpn**.
- Eptn0 : Valor real: **float Eptn0**.
- Vol0 : Valor real que indica el volumen siguiente: **float Vol0**.
- Pinst : Valor real: **float Pinst**.
- Dens0 : Valor real que indica la densidad 0: **float Dens0**.
- Dens : Valor real que indica la densidad: **float Dens**.

9.2.2.7.6 Módulo de impresiones**Nombre de la clase: Impresiones****Propiedades:**

- Nombre, cadena de caracteres para identificador de la impresión: **char *nombre**
- Vector de datos, estructura tipo *template* que almacena los datos a imprimir: **T *vector_datos**.
- Tipo de dato a imprimir, es un valor entero corto que indica que tipo de datos se van a imprimir: texto, para archivo de configuraciones o binario para imágenes: **short int tipo**.
- Identificador de impresión, valor entero corto que sirve para identificar las impresiones: **short int ID_impresion**.
- Calidad de Impresión, valor real que indica el grado de calidad de impresión: **float calidad**.

Nombre de la clase: Impresion_de_archivo_de_configuracion**Propiedades:**

- Tipo de archivo de configuración, valor entero que sirve para diferenciar entre el tipo de archivo de configuración: **char *tipo_archivo_configuración**.

Nombre de la clase: Impresion_de_imagen**Propiedades:**

- Formato de imagen, estructura de datos de tipo *template* que almacena el formato de la imagen: **T *formato_imagen**.

9.2.2.8 Empacar en módulos

9.2.2.8.1 Ocultar información

Se definió la interfaz pública de cada clase, y se decidió qué atributos y operaciones son accesibles fuera de la clase, dependiendo de las necesidades y relaciones con las clases externas.

La información privada, es conveniente que sea únicamente la que vean las clases padres, así como las clases derivadas, para evitar modificaciones indeseadas y accesos a información no relevante en las diferentes clases.

9.2.2.8.2 Coherencia de entidades

En esta parte, las clases, operaciones y módulos deben encajar para una meta común. Una operación debe hacer una cosa bien, y no debe mezclar a la vez política e implementación.

Política: Tomar decisiones que dependen del contexto.

Implementación: Ejecución del algoritmo totalmente especificado.

9.2.2.8.3 Construir módulos

La conectividad del modelo de objetos se puede usar como guía para dividir en módulos. Los módulos deben ser definidos para que sus interfaces sean mínimas y bien especificadas.

Los módulos son los siguientes:

- **Interacción con el usuario:** Éste módulo es el correspondiente a la interfaz gráfica del Sistema de Visualización de la Dinámica Molecular
- **Control de datos:** Éste módulo corresponde a los clases de contenedores de datos, las cuales distribuirán los datos entre las diferentes clases.
- **Realización de cálculos:** Éste módulo corresponde a las clases de cálculos, tanto locales como remotos.
- **Realización de gráficas:** Corresponde a las clases gráficas que visualizarán los objetos dentro del Sistema.
- **Control y monitoreo de procesos:** Corresponde a las clases del monitor de procesos y se encarga de gestionar los diferentes procesos del sistema.
- **Realización de impresiones:** Corresponde a las clases de impresiones y se encarga de realizar las diferentes impresiones solicitadas por el sistema.

9.2.2.9 Consideraciones finales

Correcciones finales realizadas al análisis y diseño:

Se decidió quitar las funciones de traducción de archivos, en los contenedores de dipolos, hoyos, conexiones, rayos X, ya que únicamente se van a traducir los archivos de configuración de partículas.

Se eliminó el contenedor de manual, ya que se llegó a la conclusión de que no era necesario.

Debido a lo anterior también se eliminó la clase de impresión de manual.

Se revisaron los métodos del análisis y se eliminaron algunos debido al funcionamiento y requerimientos del sistema.

CAPITULO VI

**FUNCIONAMIENTO DEL SISTEMA DE
VISUALIZACIÓN DE DINÁMICA MOLECULAR**

10. CAPÍTULO VI FUNCIONAMIENTO DEL SISTEMA DE VISUALIZACIÓN DE DINÁMICA MOLECULAR.

10.1 Especificaciones y requerimientos

El sistema de visualización de dinámica molecular fue programado y diseñado bajo una arquitectura accesible y escalable, utilizando para su programación herramientas de software libre. Para el desarrollo del sistema se utilizó:

- El sistema operativo Linux Red Hat en su versión 6.2. (2.2.14-5.0).
- GTK+ (Gimp ToolKit) biblioteca para la creación de interfaces gráficas de usuario para el Sistema X Window del GTK Team, versión 1.2.7 de Febrero del 2000.
- GLib, biblioteca que incluye rutinas de soporte para C como listas, árboles, manejo de memoria, entre otras cosas, del GLib Team, versión 1.2.7 de Febrero del 2000.
- El conjunto de clases de VDK (The Visual Development Kit) y VdkBuilder (herramienta para construir aplicaciones gui) de Mario Motta versión 1.0.6 de Octubre de 1999.
- El componente gtkglarea de Janne Löff versión 1.2.2 de febrero de 1999 para la visualización de la Escena Gráfica.
- MESA (clon de Open GL para Linux) de Brian Paul versión 3.1 de julio de 1999 para la creación de los objetos 3D.

Todas éstas herramientas se utilizaron para aprovechar todas las ventajas que en conjunto ofrecían frente a las herramientas de desarrollo comerciales.

Todas fueron compiladas y configuradas para poder desarrollar nuestra aplicación con los comandos configure, make y make install de sus respectivos paquetes.

Para compilar el código de nuestro sistema que denominamos vdkgrafica, se utilizo el comando make de linux con su respectivo makefile.

Requerimientos de hardware:

- Computadora personal Intel Pentium III a 866 Mhz o superior,
- Memoria de 250 Mb de RAM o superior,
- Disco duro de 10 GB,
- Tarjeta de video Sis 630,
- Monitor con resolución 1024 x 768.
- Ratón

10.2 Aplicación y limitaciones

El sistema de visualización que presentamos fue desarrollado para facilitar el análisis de los resultados de la simulación numérica de dinámica molecular que se investiga en el Laboratorio de Simulación de Materiales de la UNAM, simulación que representa fenómenos fisicoquímicos y estructurales de los materiales que son de interés para los investigadores, entre los procesos que se pueden estudiar se encuentran los relacionados con el cambio en el radio y carga de los átomos, la formación de cavidades entre las estructuras y el análisis de potenciales electrostáticos entre otros, es importante indicar que el sistema representa la visualización de los procesos simulados antes mencionados de una manera similar tal como se verían a través de un microscopio electrónico que permitiera observar los átomos de una molécula.

Debido a la complejidad y al número de horas invertidas para la reestructuración del sistema, se decidió no incluir algunas características del diseño en el desarrollo de la actual versión, dejándolas para un siguiente desarrollo, como son los módulos del Editor molecular, el Monitor de procesos de cálculo y el módulo de Inicialización-configuración de dinámica molecular

10.3 Descripción general y funcionamiento

Las opciones que aparecen en la siguiente tabla en azul son las acciones activas de los módulos desarrollados que pueden utilizarse en ésta versión, las opciones en gris son de módulos que no fueron desarrollados por complicaciones de tiempo pero que su análisis y diseño se encuentra en el capítulo V.

Activado	Activado	Desactivado	Desactivado	Activado	Activado	Desactivado	Activado
Archivo	Ver	<i>Selección</i>	<i>Edición</i>	Propiedades	Transformación	<i>Cálculos</i>	Ayuda
Abrir	Átomos	Átomos	Cortar	Color	Traslación	<i>Dinámica Molecular</i>	Manual
Guardar Imagen	Hoyos	Hoyos	Copiar	Radio	Rotación	<i>Análisis de Coordinación</i>	Créditos
Salir	Dipolos	Dipolos	Pegar	Material	Escalamiento	<i>Distancia promedio entre vecinos</i>	
							<i>Distancia entre partículas</i>
	Conexiones	Conexiones	Agregar	Transparencia			<i>Encontrar hoyos</i>
							<i>Generar rayos x</i>
	Molécula	Todos	Agrupar	Calidad			<i>Generar campo escalar</i>
							<i>Generar estructura cristalina</i>
	Caja	Molécula	Desagrupar	Estilo			
	Ejes		Unir				
	Etiquetas		Desunir				
	Potenciales		Deshacer Edición				
	Superficie		Rehacer Edición				
	Rayos X						
	Perspectiva						

Figura 10.3

10.3.1 Elementos de la interfaz de usuario

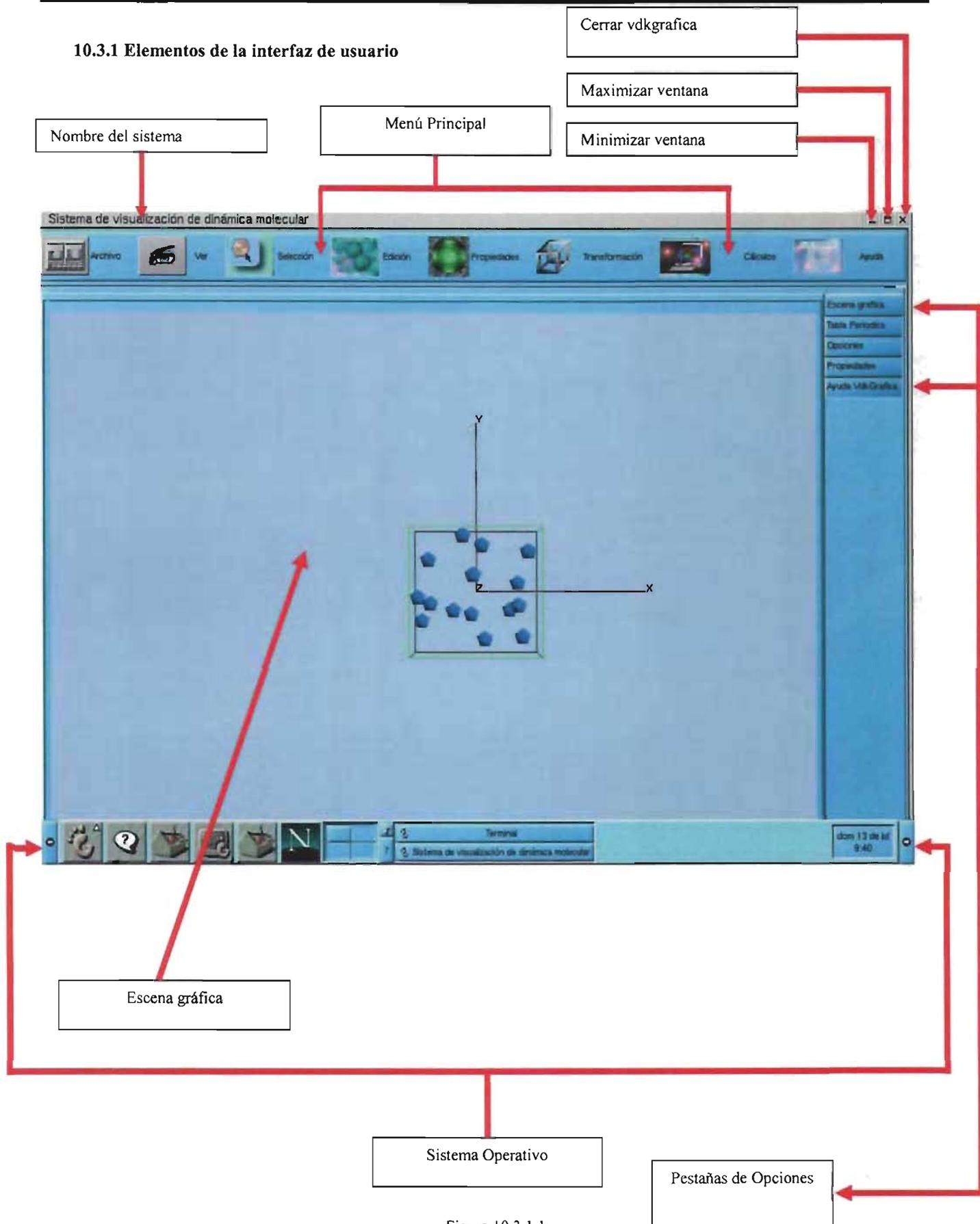


Figura 10.3.1.1

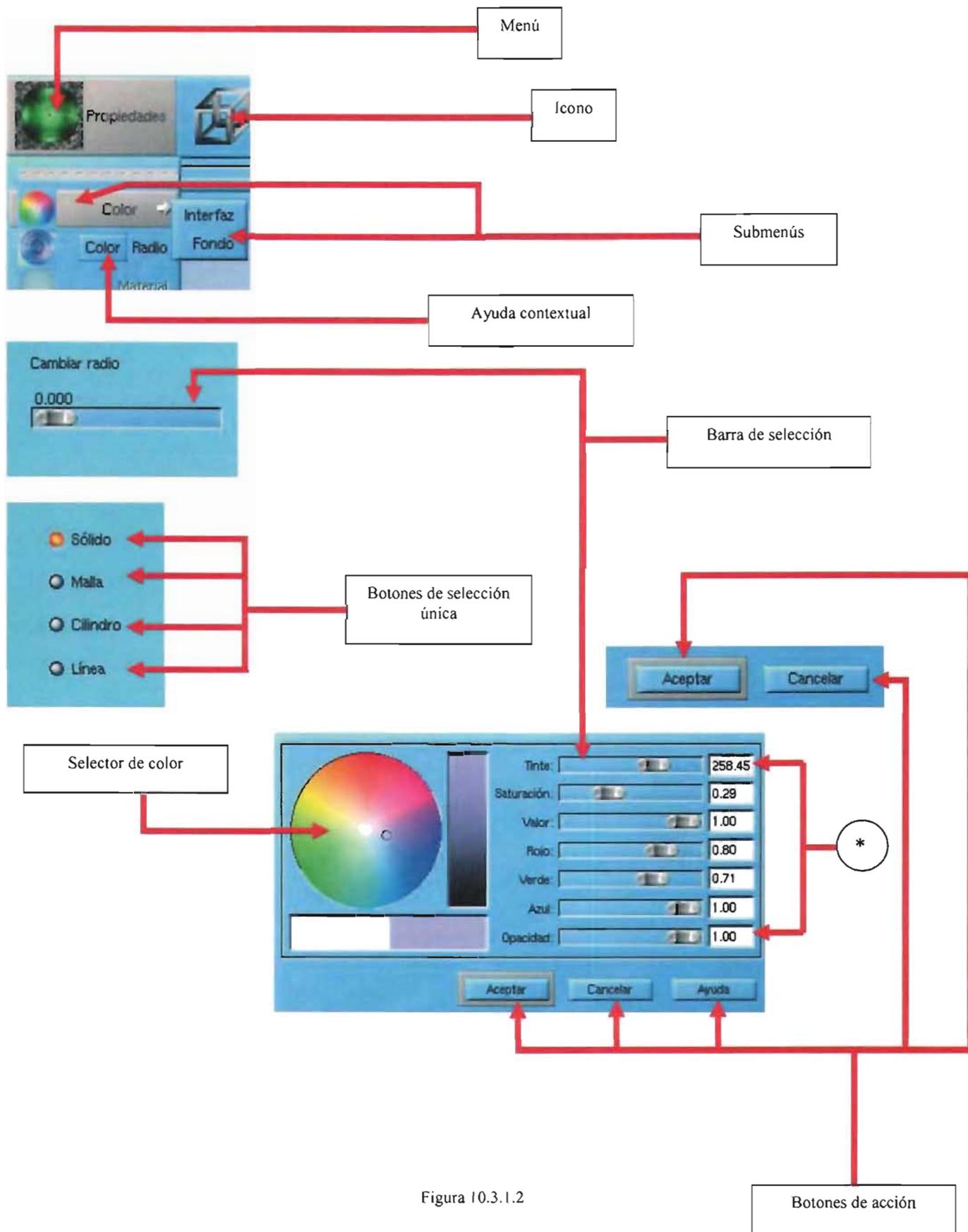


Figura 10.3.1.2

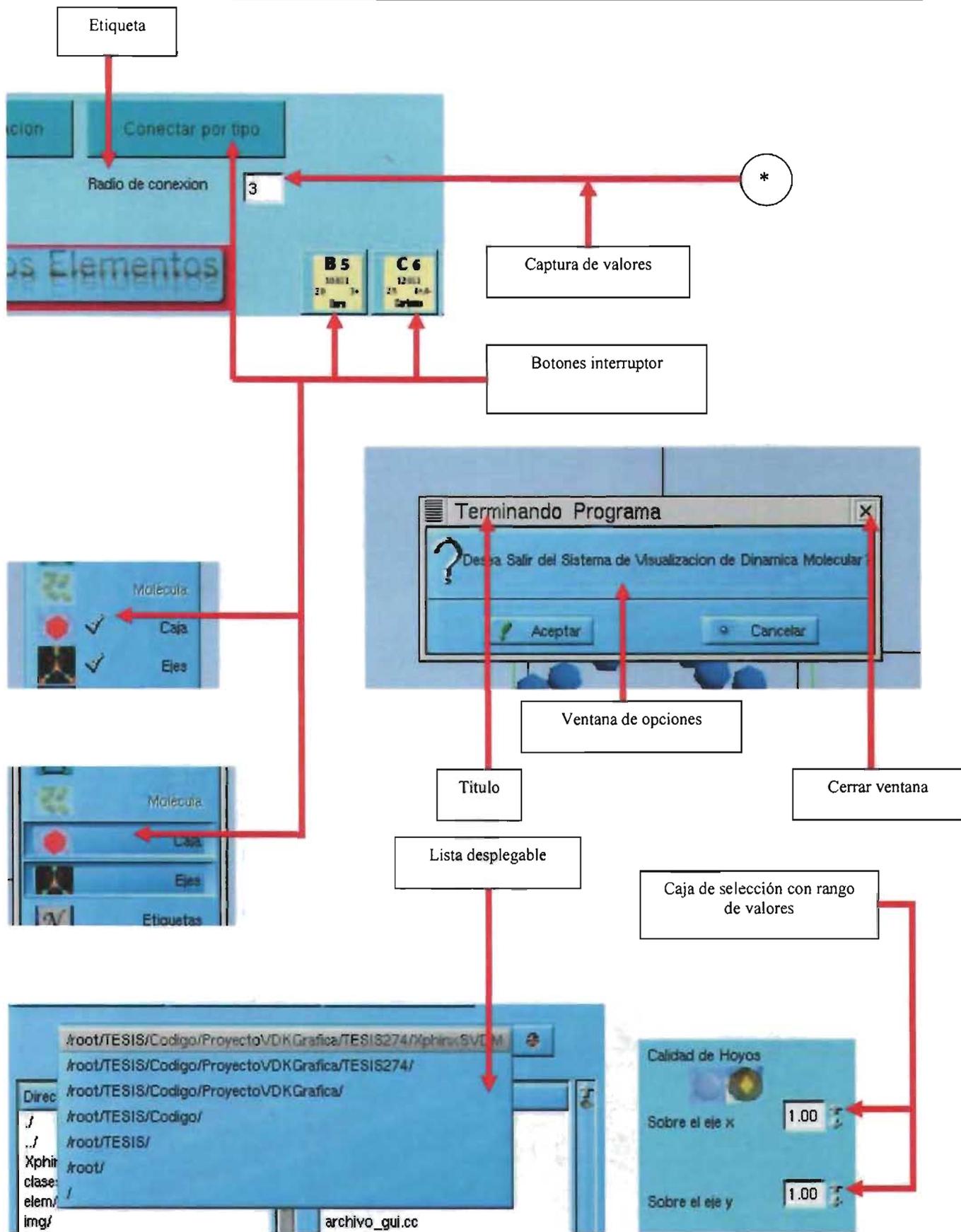


Figura 10.3.1.3

10.3.2 Ejecución del sistema



Figura 10.3.2.1

El SVDMM(Sistema de Visualización de Dinámica Molecular) es un programa desarrollado para ejecutarse en el ambiente gráfico que proporciona Linux, el cual puede ser GNOME o KDE. El programa se ejecuta desde una terminal como se muestra en la Figura 6.2.1



Figura 10.3.2.2



Figura 10.3.2.3

En la siguiente imagen se observa el selector de archivos, el cual permite al usuario escoger el archivo a visualizar



Figura 10.3.2.4

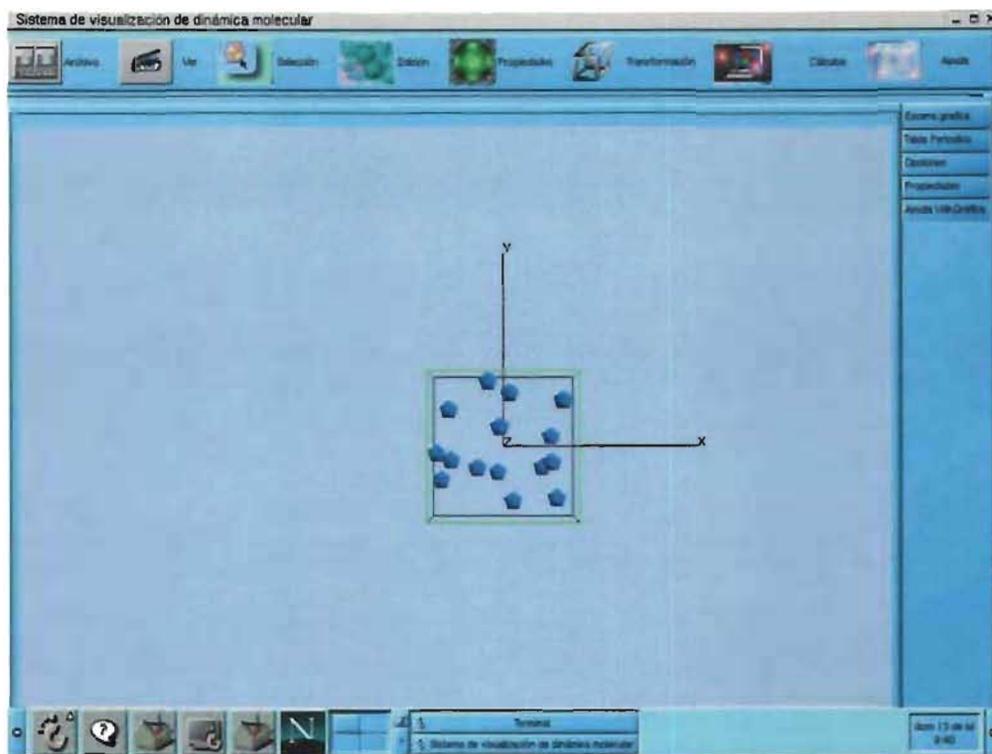


Figura 10.3.2.5

Una vez iniciado el sistema aparece la interfaz gráfica que permitirá manipular la visualización y mostrar en 3D la representación de los datos numéricos. En la Figura 6.2.2 tenemos la primera vista del sistema, desde ésta pantalla el usuario debe iniciar seleccionando el menú **Archivo**→**Abrir** para que el sistema, por medio del selector de archivos, permita elegir el archivo de datos de la dinámica molecular que se desea visualizar, cómo se aprecia en las Figuras 6.2.3 y 6.2.4. Es importante mencionar que los archivos provienen de un proceso de cálculo numérico complejo. Una simulación de Dinámica Molecular se ejecuta en la supercomputadora CRAY de la UNAM, misma que tarda de una a 20 horas de tiempo de procesador dependiendo de sus características, aproximadamente se generan 20 archivos con resultados uno de los cuales contiene la posición, tipo y carga de cada átomo, éste archivo es el que principalmente se visualiza en el sistema cómo puede verse en la Figura 6.2.5. Otros de los archivos que alimentan al SVDM son el archivo de conexiones, el de hoyos y el de potenciales (En el anexo I se muestran los formatos correspondientes a estos archivos).

10.3.3 Descripción de menús, submenús y opciones del sistema.

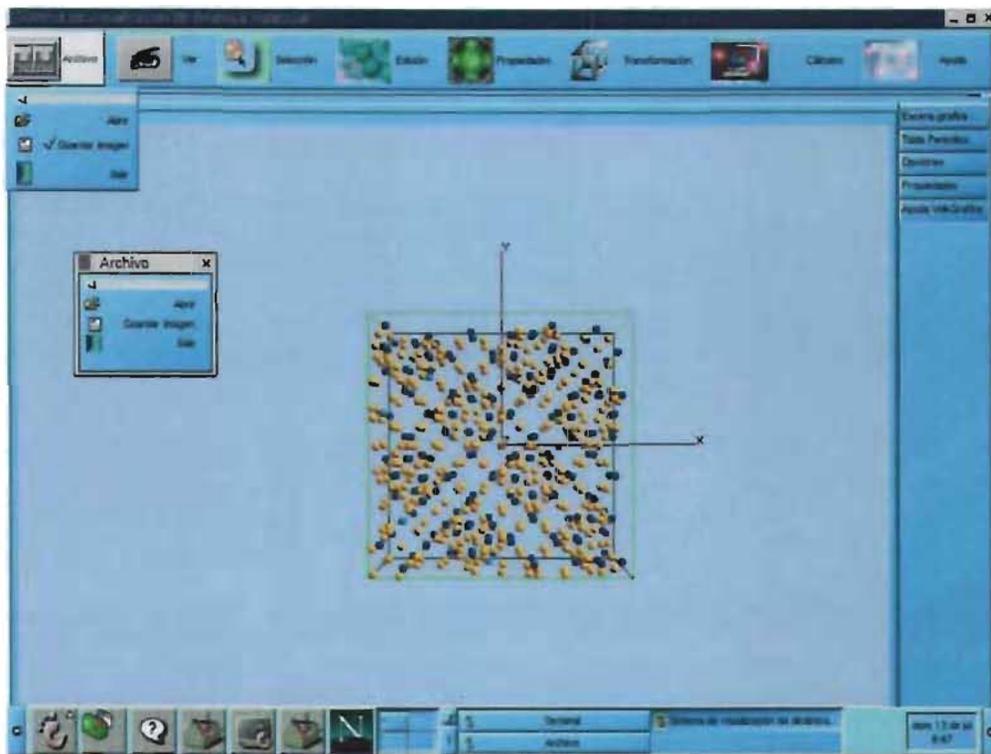


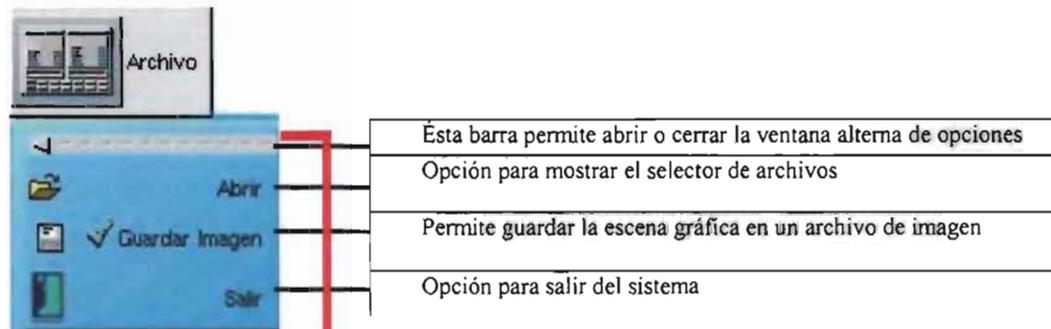
Figura 10.3.3.1

10.3.3.1 Menú Archivo

El primer submenú del menú principal contiene las opciones activadas:

Abrir, Guardar imagen y Salir.

Todos los submenús tienen la opción de “desprender” una ventana de opciones de manera que es posible tenerlas disponibles enfrente de la ventana principal una vez cerrado el menú.



Al elegir la opción para salir del sistema aparece la siguiente ventana que permite cancelar la acción de salir.



10.3.3.2 Menú Ver

El segundo submenú del menú principal contiene las opciones activadas: Átomos, Hoyos, Conexiones, Caja, Ejes, Etiquetas, Potenciales y Superficie.

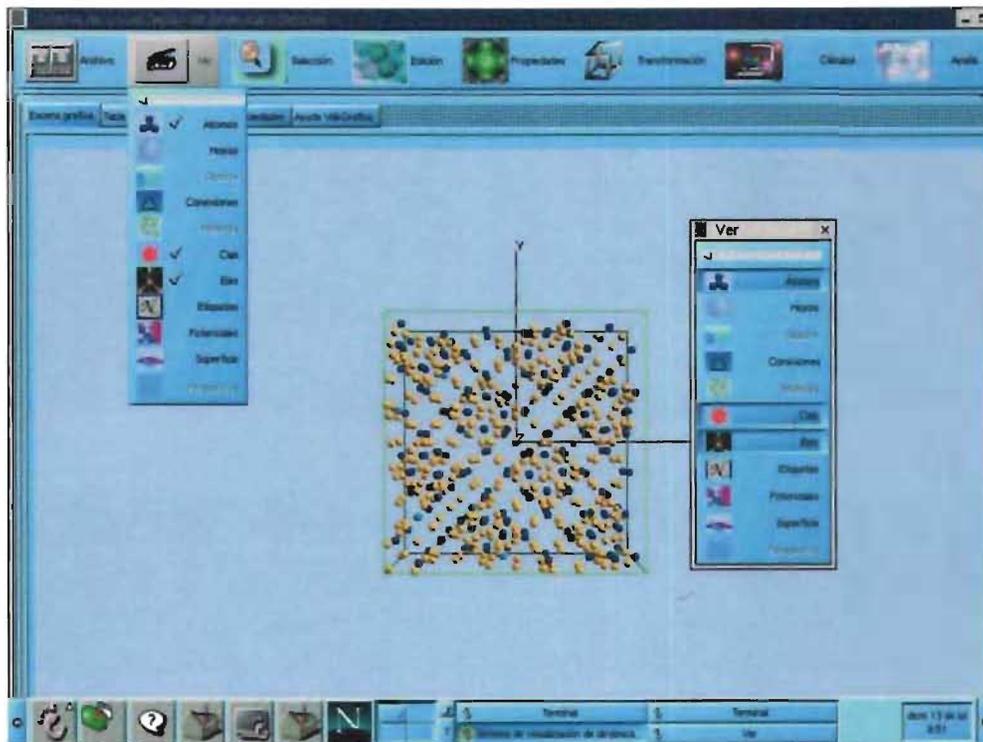
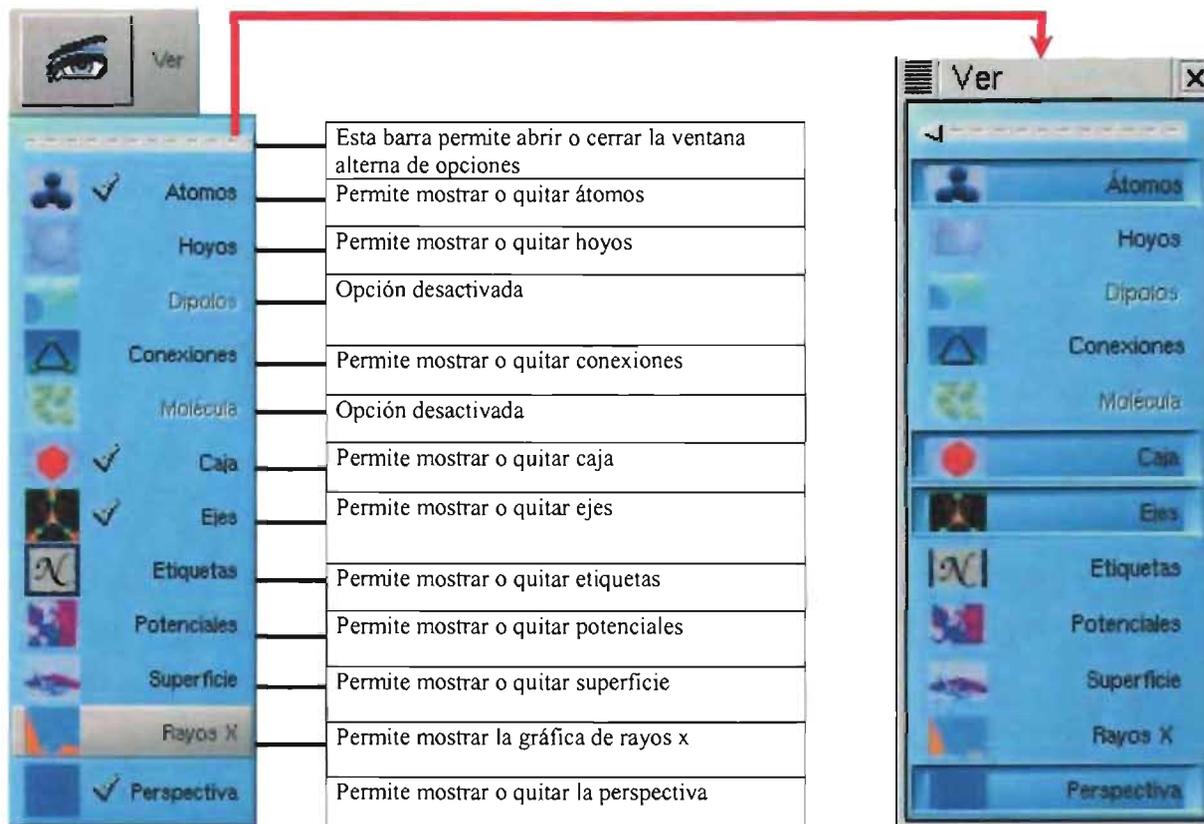


Figura 10.3.3.2



10.3.3.3 Menú Selección

El tercer submenú del menú principal contiene opciones que pertenecen al módulo Editor molecular, módulo que no fue integrado en ésta versión del SVDM, las siguientes opciones están desactivadas:

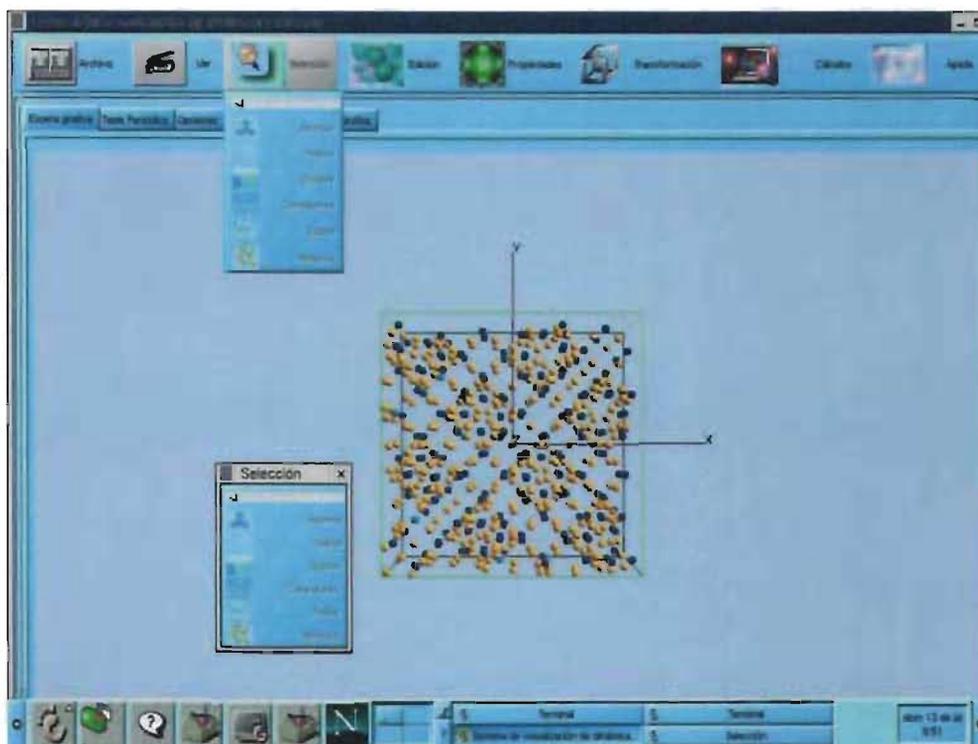
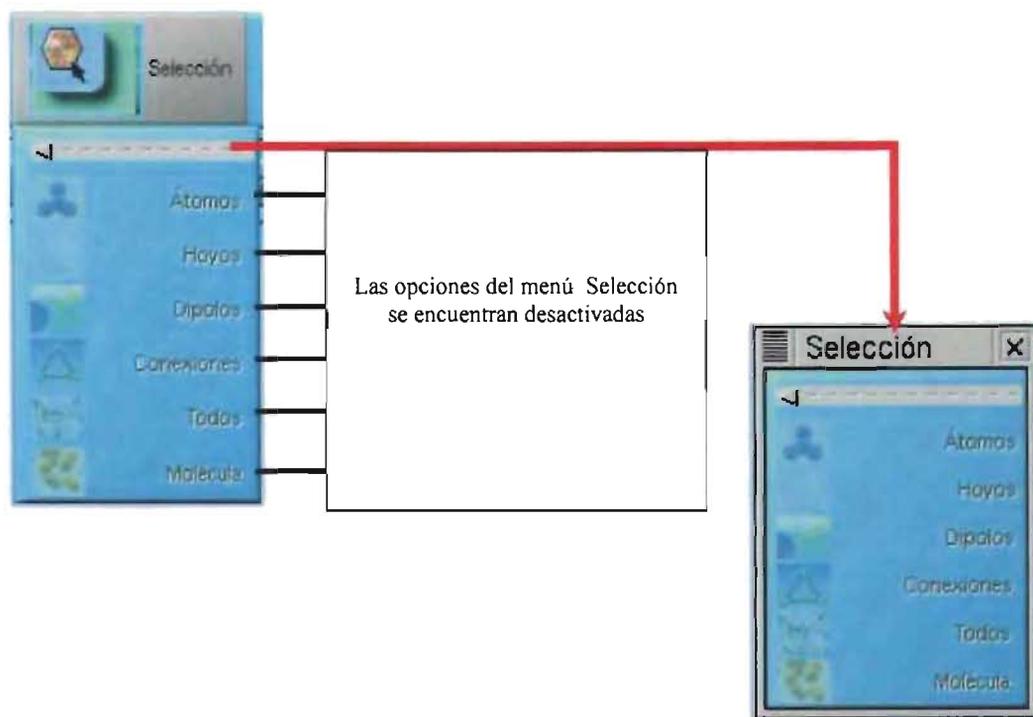


Figura 10.3.3.3

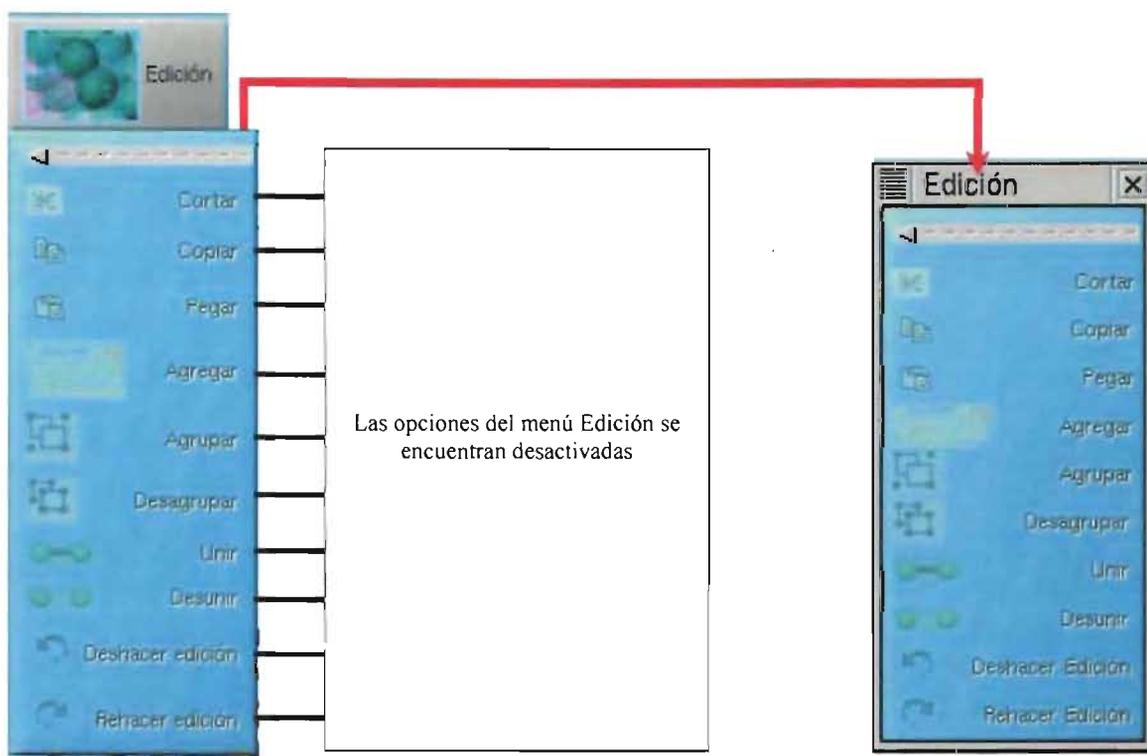


10.3.3.4 Menú Edición

El cuarto submenú del menú principal contiene opciones que pertenecen al módulo Editor molecular, el cual no fue integrado en ésta versión del SVDM, por lo tanto las siguientes opciones están desactivadas:



Figura 10.3.3.4



10.3.3.5 Menú Propiedades

El quinto submenú del menú principal contiene las opciones activadas:

Color->Interfaz, Color->Fondo, Radio, Transparencia, Calidad y Estilo.

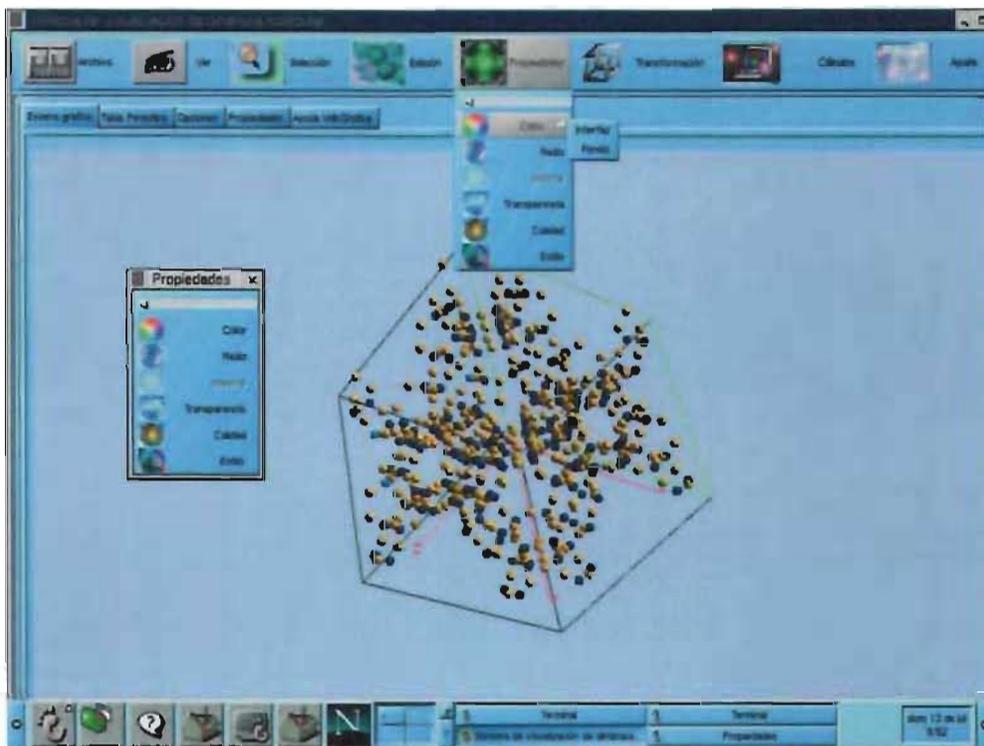


Figura 10.3.3.5



10.3.3.6 Menú Transformación

El sexto submenú del menú principal contiene las opciones activadas:

Traslación, Rotación y Escalamiento.

Éste submenú permite realizar movimientos de la celda unitaria dentro del espacio tridimensional.

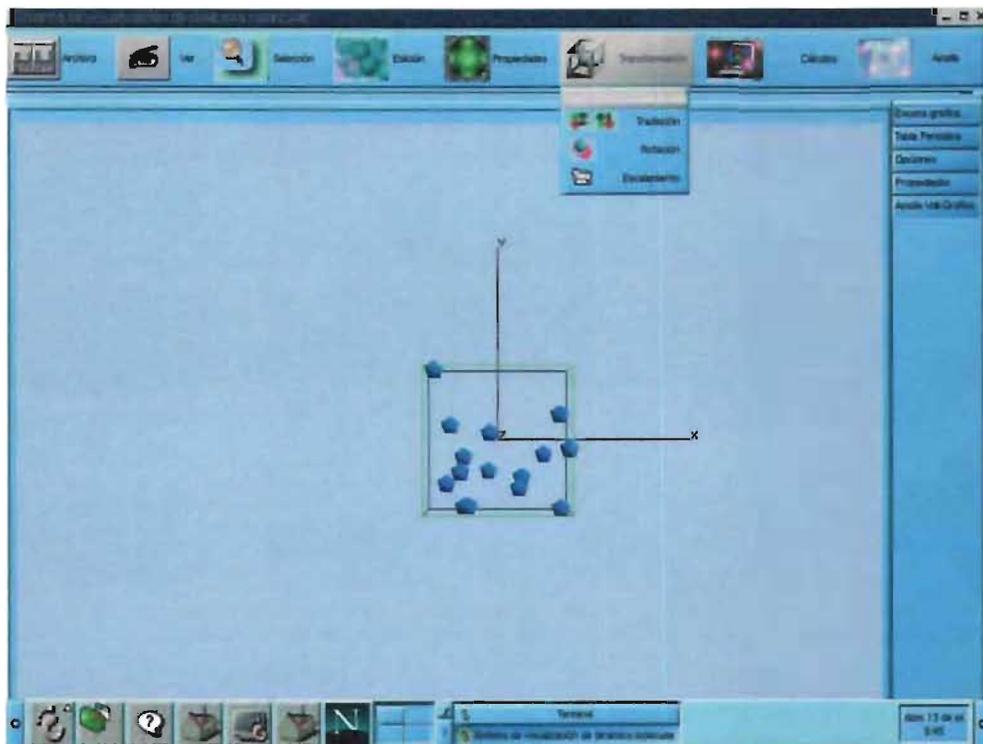
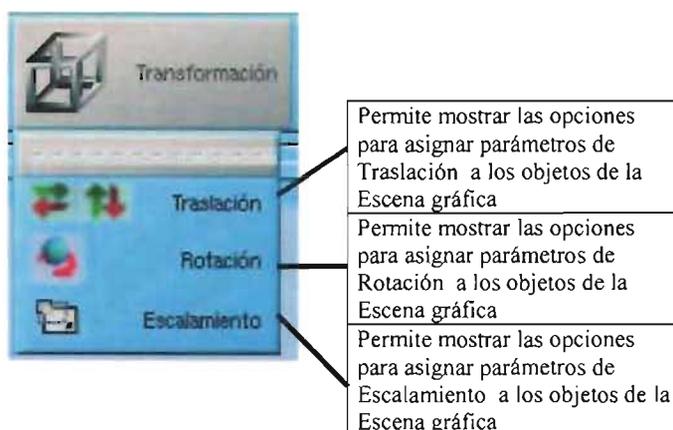


Figura 10.3.3.6



10.3.3.7 Menú Cálculos

El siguiente submenú también corresponde a un módulo que no se integro en ésta versión del sistema, por lo tanto las opciones están deshabilitadas

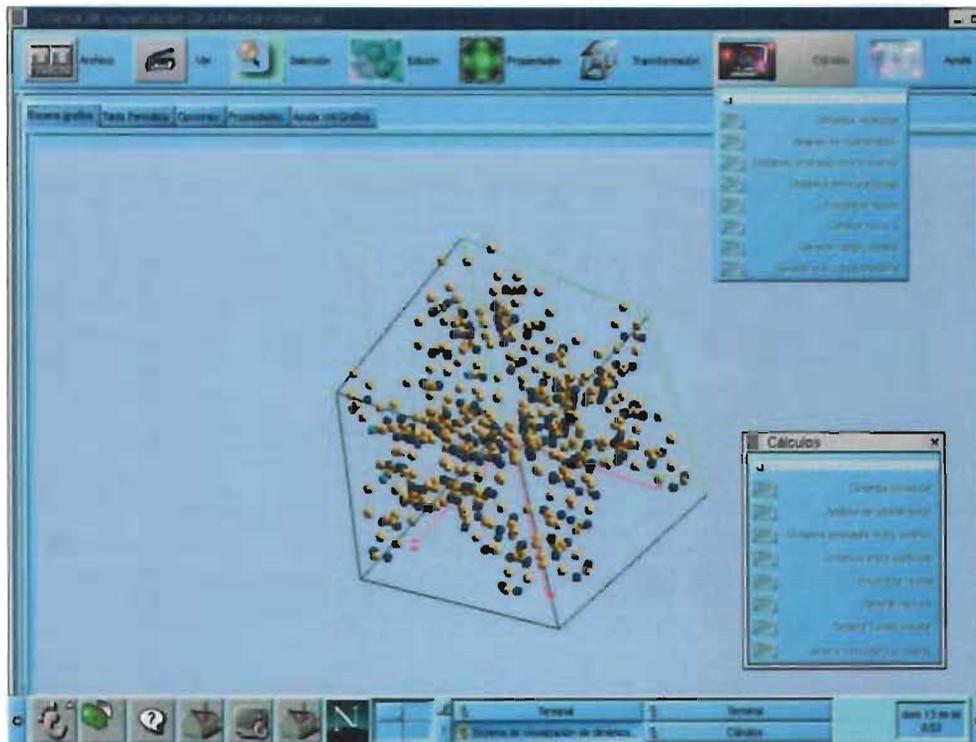
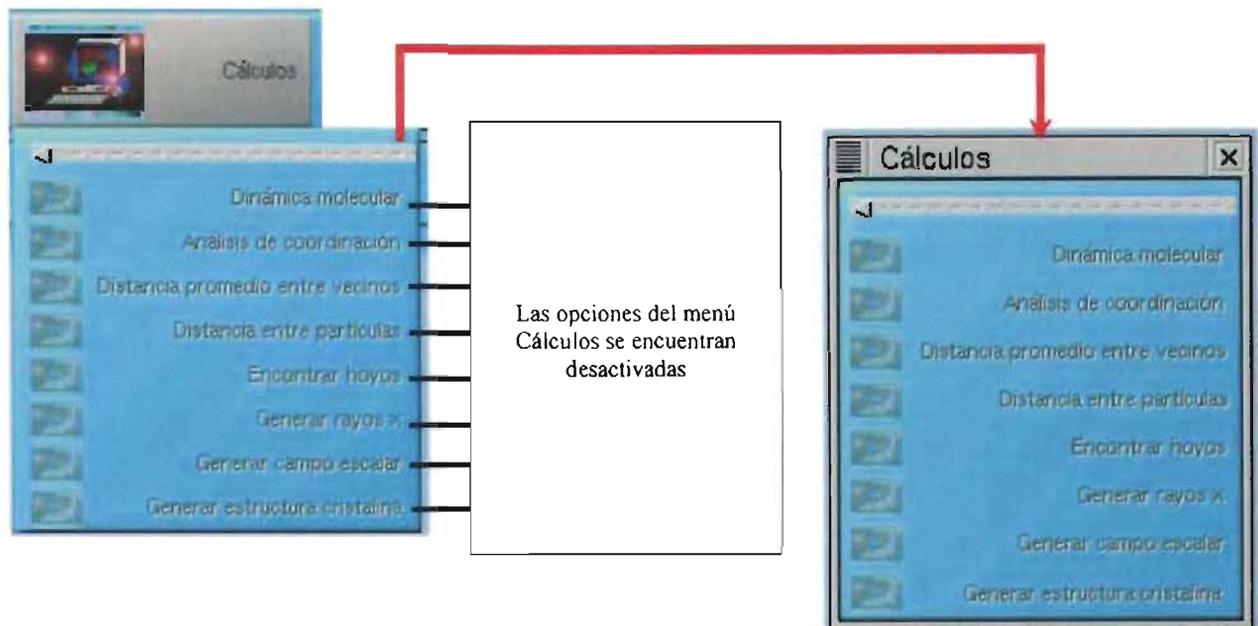


Figura 10.3.3.7

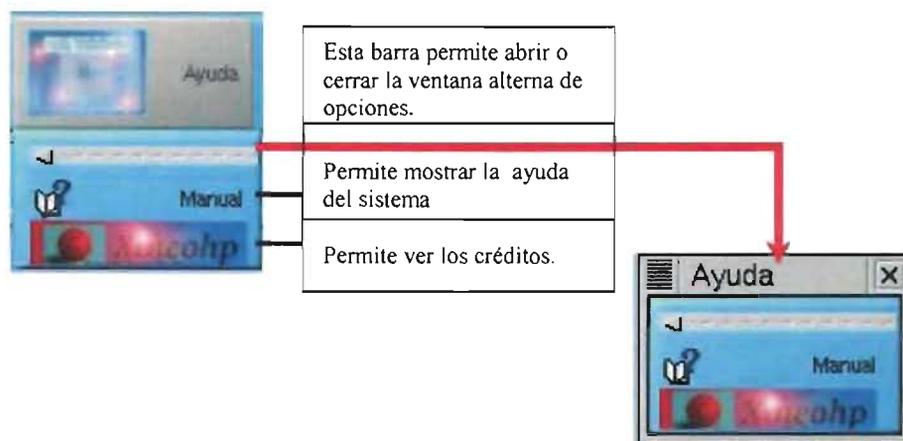


10.3.3.8 Menú Ayuda

El último submenú corresponde a la ayuda del sistema y permite abrir una ventana con los créditos del mismo.



Figura 10.3.3.8



10.3.3.9 Submenú de activación de menú principal y pestañas de módulos.

Este submenú se activa dando un clic con el botón derecho del ratón sobre la zona de pestañas y permite omitir el menú de modo que se aprecie únicamente la escena gráfica.

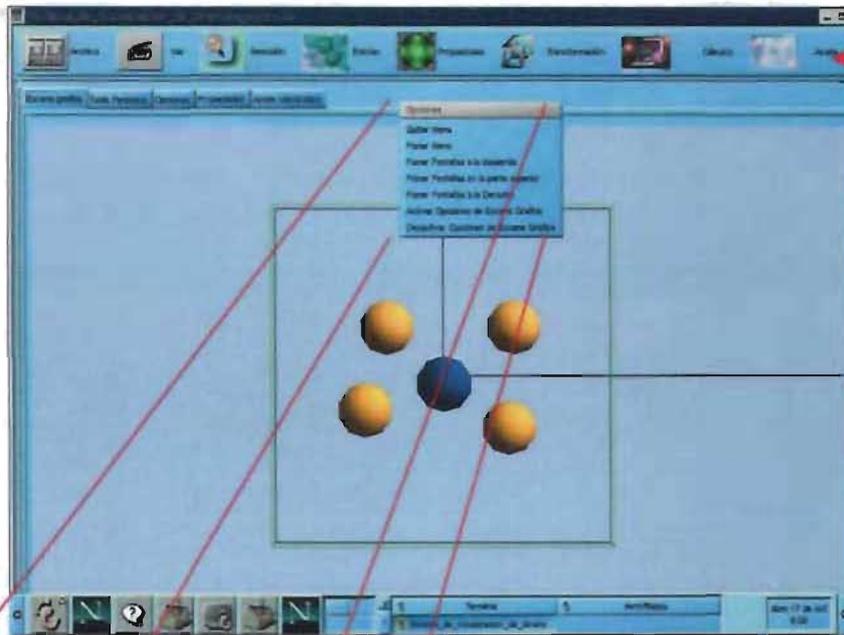
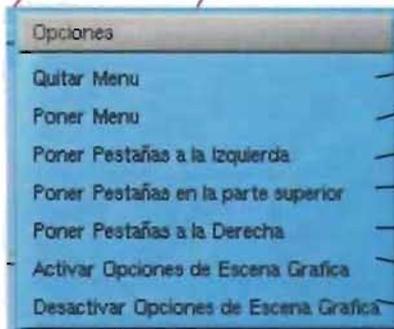


Figura 10.3.3.9



- Permite hacer no visible el menú principal.
- Permite mostrar el menú principal.
- Opción para mostrar las pestañas de acceso a los módulos en la parte izquierda del sistema.
- Opción para mostrar las pestañas de acceso a los módulos en la parte superior del sistema.
- Opción para mostrar las pestañas de acceso a los módulos en la parte inferior del sistema.
- Permite activar el menú de opciones de EG
- Permite desactivar el menú de opciones de EG

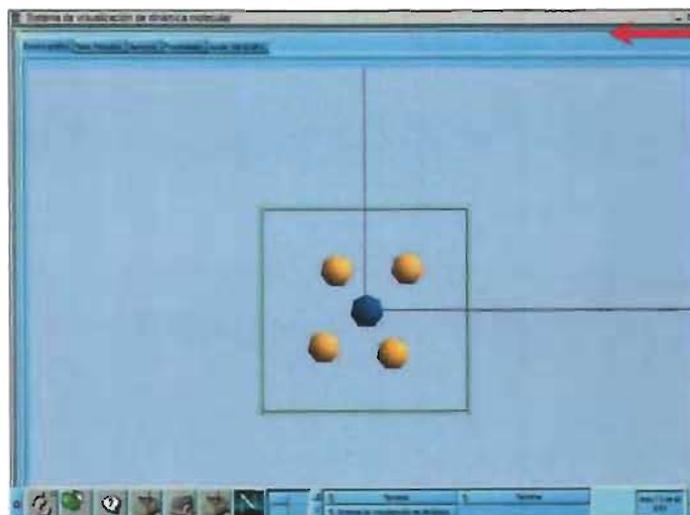


Figura 10.3.3.10

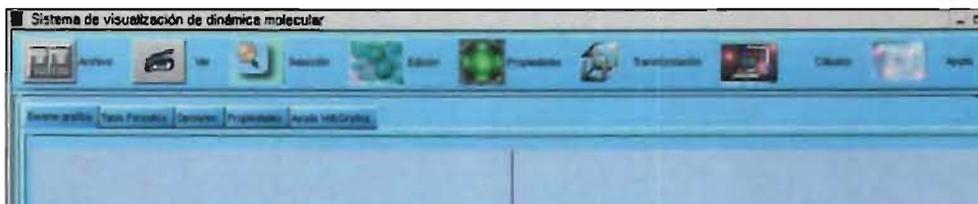


Figura 10.3.3.11

Las pestañas de acceso a los módulos principales pueden colocarse en la posición más conveniente para manipular la escena gráfica.

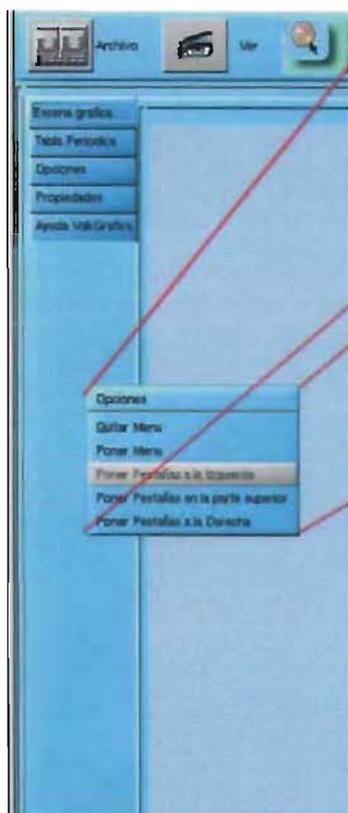


Figura 10.3.3.12

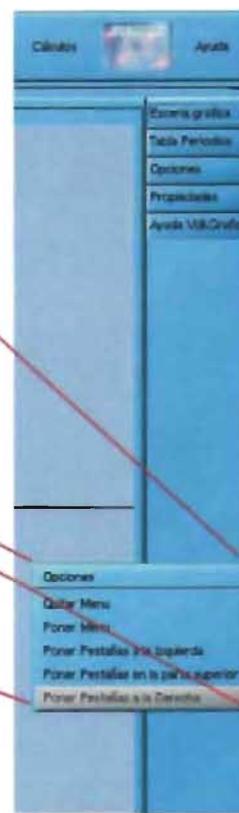
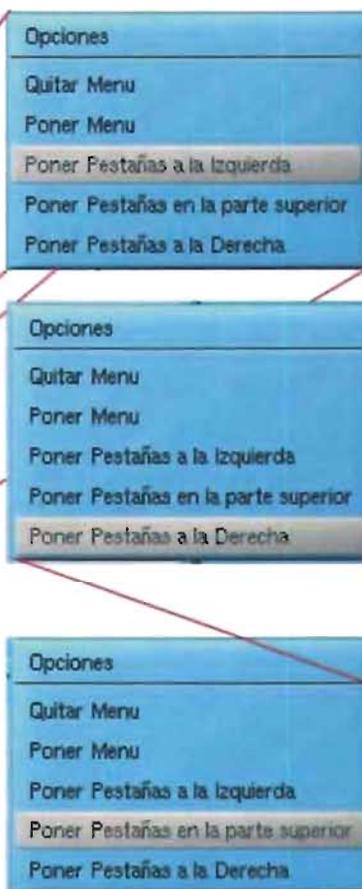


Figura 10.3.3.14

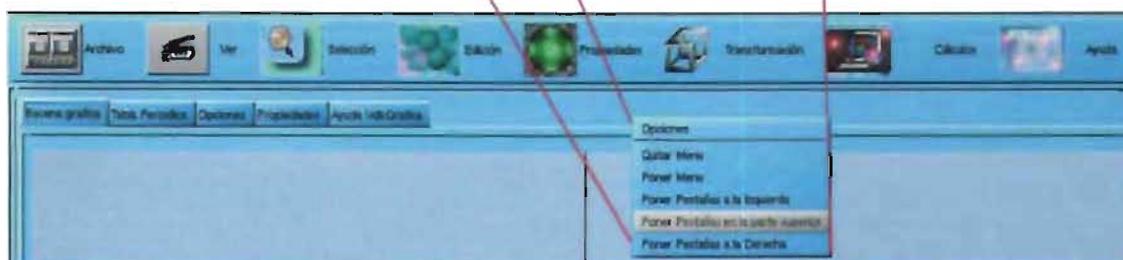


Figura 10.3.3.13

10.3.3.10 Submenú contextual sobre la escena gráfica.

Éste submenú aparece al dar un clic con el botón derecho sobre la Escena gráfica (Figura 10.3.1.1),
Las opciones disponibles son:



Figura 10.3.3.15

Permite detener la animación por cuadros de la molécula.	Parar Animación
Permite iniciar la animación por cuadros de la molécula.	Iniciar Animación
Esta opción sirve para desactivar el zoom.	Quitar Zoom
Esta opción sirve para activar el zoom.	Poner Zoom
Permite indicarle al sistema que la animación de los cuadros será en orden ascendente.	Animación Hacia Adelante
Permite indicarle al sistema que la animación de los cuadros será en orden descendente.	Animación Hacia Atrás
Esta opción sirve para desactivar la animación con el ratón.	Parar Animación con Ratón
Esta opción sirve para activar la animación con el ratón.	Animación con Ratón
Permite indicarle al sistema que se usaran los valores iniciales de la configuración del sistema para los objetos gráficos en la escena gráfica.	Configuración Inicial
Permite indicarle al sistema que se usaran los valores personalizados de la configuración del sistema para los objetos gráficos en la escena gráfica.	Configuración Personalizada
Permite activar la etiqueta de número de cuadros	Activar Número de cuadros
Permite desactivar la etiqueta de número de cuadros	Desactivar Número de cuadros

10.3.4 Descripción de los módulos principales.

10.3.4.1 Escena gráfica



Figura 10.3.4.1

La escena gráfica es la parte principal del sistema de visualización de dinámica molecular ya que es el área de trabajo donde se manipulan y observan las configuraciones de las dinámicas simuladas, y en la que aparecen representados en 3D los datos que han alimentado al sistema. En ésta zona se muestran los objetos de la molécula de acuerdo a las propiedades seleccionadas por el usuario, además se pueden manipular usando el ratón para rotar, trasladar o escalar la celda unitaria.

La mitad superior de la escena gráfica es una zona donde usando los botones izquierdo y derecho al mismo tiempo se aleja el “zoom” sobre los objetos de la molécula.

La mitad inferior de la escena gráfica es una zona donde usando los botones izquierdo y derecho al mismo tiempo se acerca el “zoom” sobre los objetos de la molécula.

Sobre toda la escena gráfica se puede utilizar el ratón presionando el botón izquierdo para rotar la molécula.

10.3.4.2 Tabla periódica

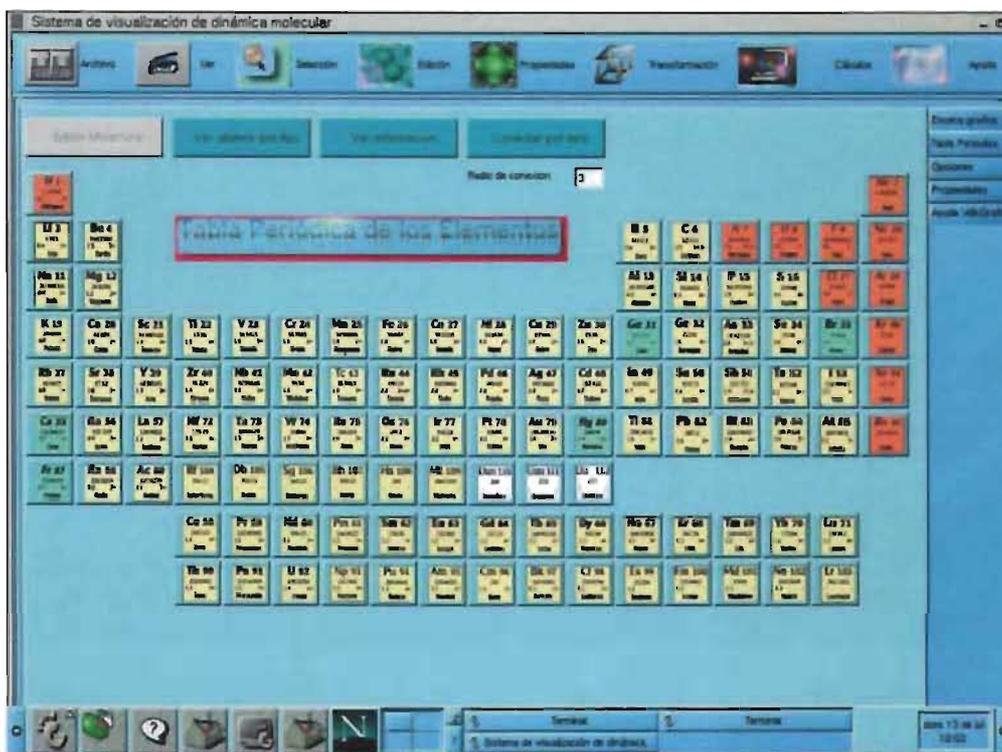


Figura 10.3.4.2.1

En este módulo se encuentran las opciones que involucran a los objetos de tipo átomo, se tiene un botón para cada elemento de la tabla periódica el cual al presionarse con el ratón responde de cierta manera dependiendo de que opción esta seleccionada. Se tienen cuatro opciones, la primera opción "Editor Molecular" se encuentra desactivada en esta versión del sistema. La segunda opción "Ver átomos por tipo" permite seleccionar los tipos de átomos que se mostraran visibles en la escena gráfica (ver Figura 10.3.4.2.3, 10.3.4.2.4). La tercera opción "Ver información" permite llamar una ventana con información referente al elemento de la tabla periódica seleccionado (ver Figuras 10.3.4.2.5, 10.3.4.2.6). La cuarta opción "Conectar por tipo" permite indicar que átomos serán conectados entre si al asignar el radio de conexión (ver Figuras 10.3.4.2.7 y 10.3.4.2.8). La caja de captura permite asignar el radio de conexión entre átomos.

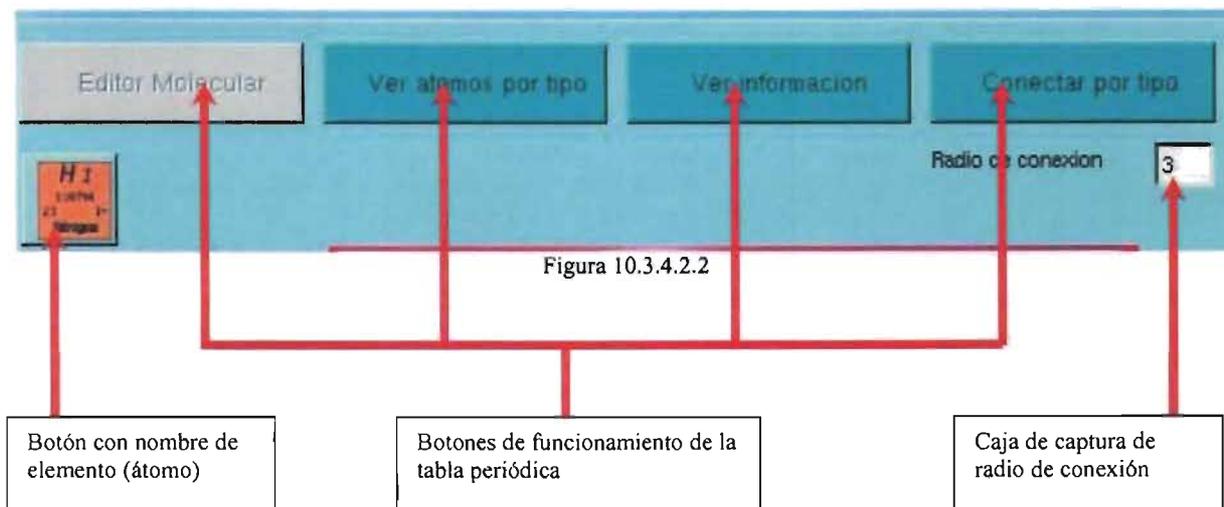


Figura 10.3.4.2.2

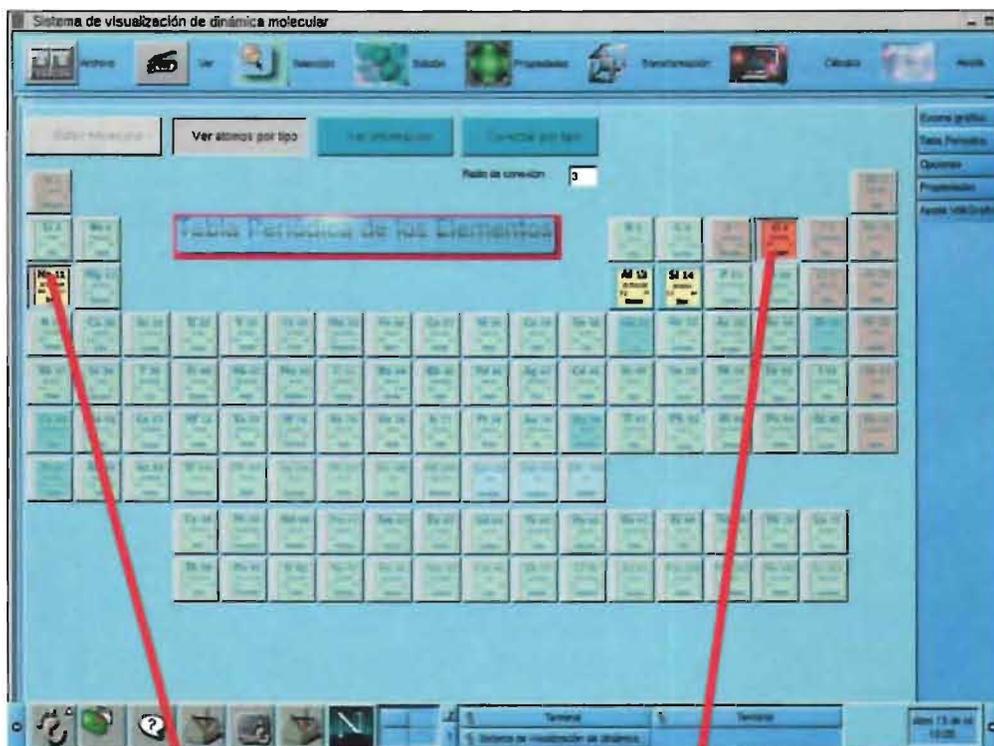


Figura 10.3.4.2.3

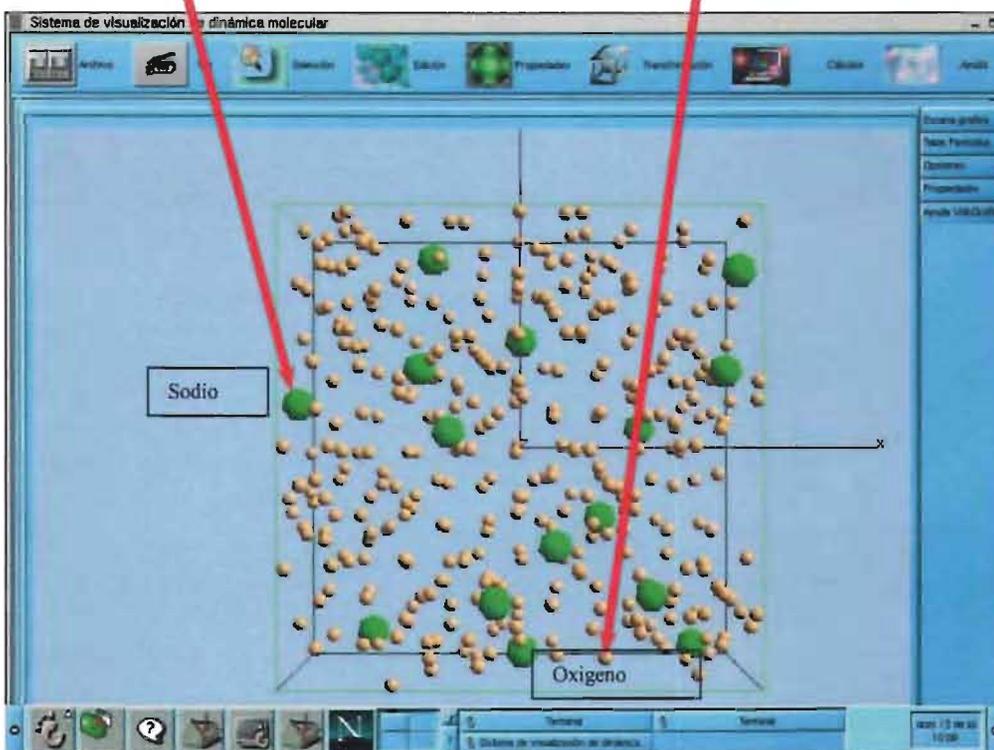


Figura 10.3.4.2.4

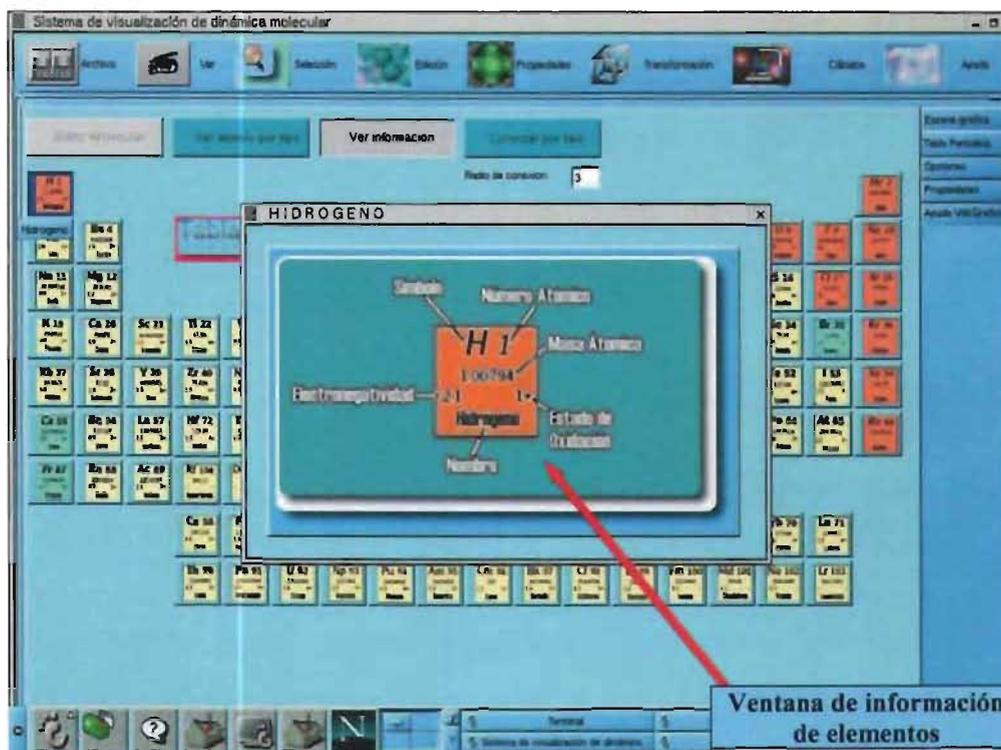


Figura 10.3.4.2.5

Ventana de información de elementos
Símbolo
Número atómico
Masa atómica
Estado de oxidación
Nombre
Electronegatividad

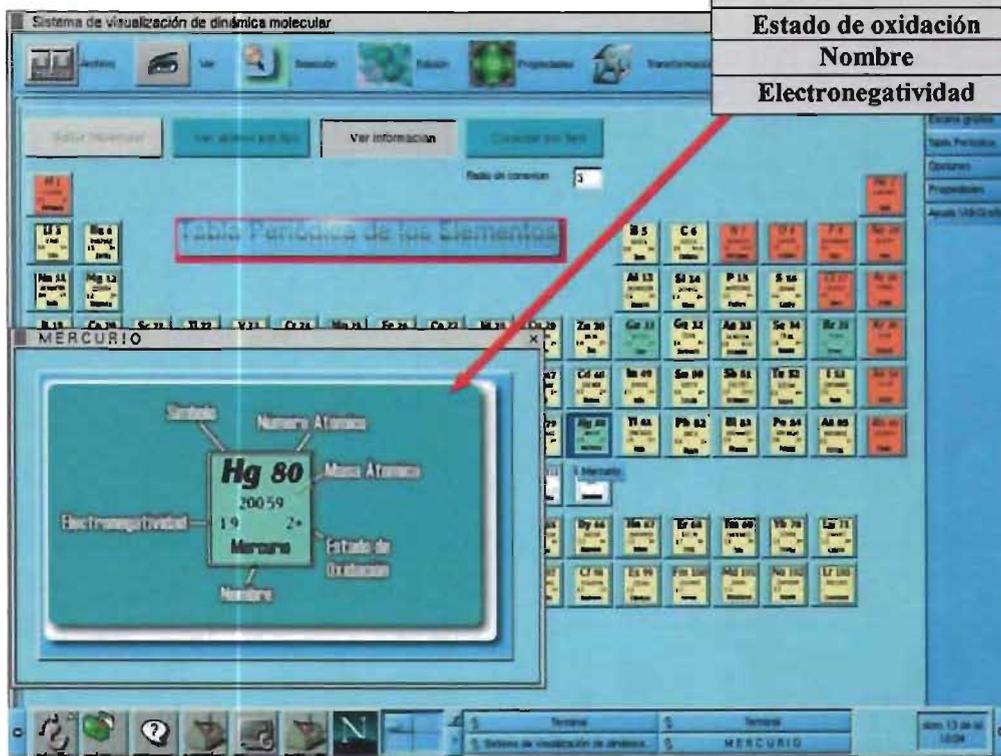


Figura 10.3.4.2.6

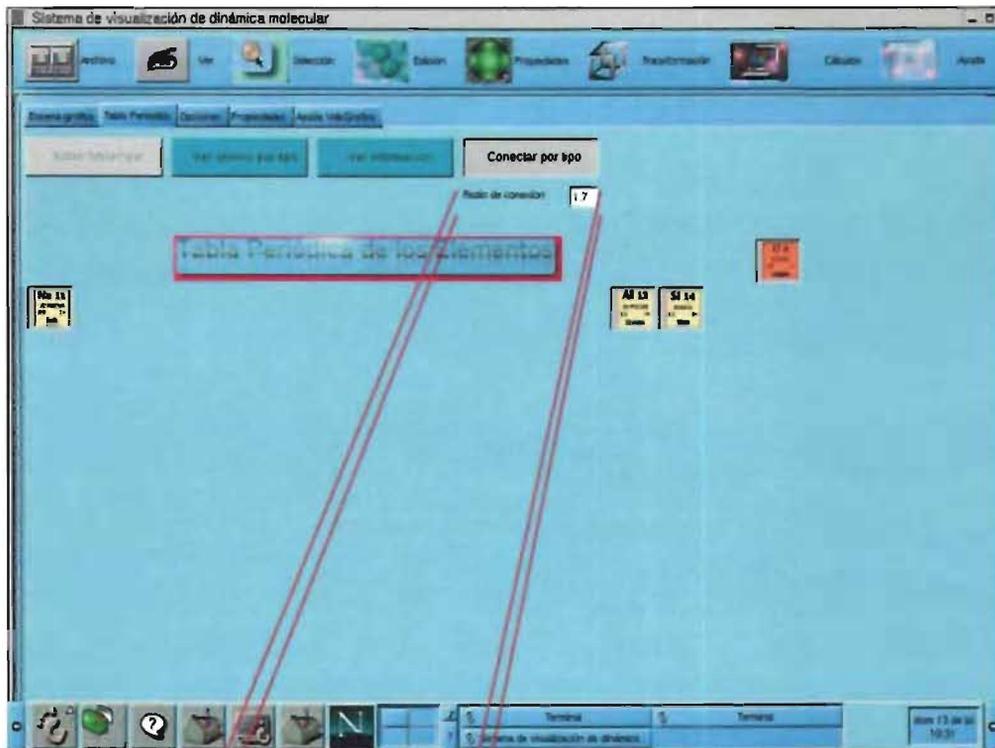


Figura 10.3.4.2.7

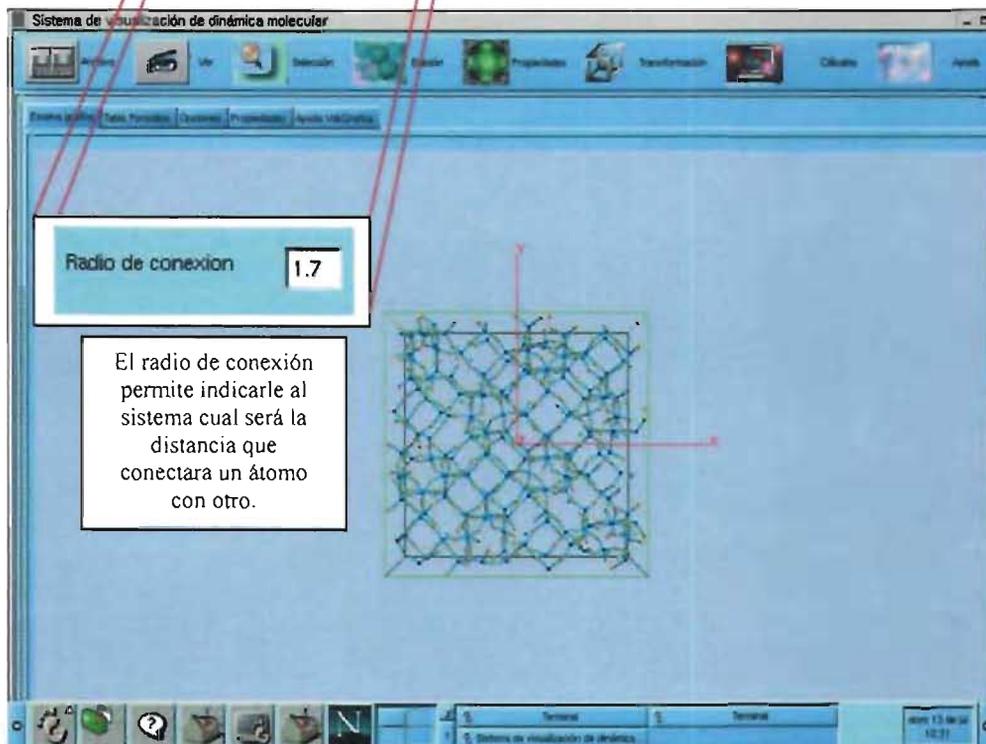


Figura 10.3.4.2.8

10.3.4.3 Opciones

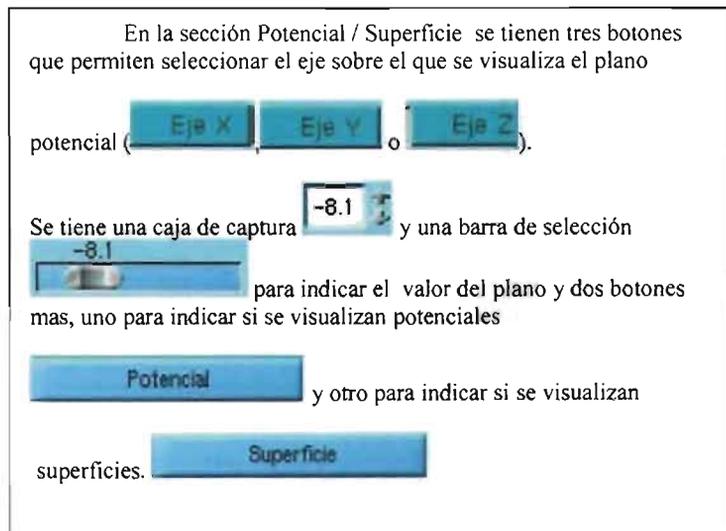


Figura 10.3.4.3.1

En éste lugar se encuentran agrupadas las opciones que permiten manipular la visualización de los potenciales y permiten asignar valores a las transformaciones de la escena gráfica.



Figura 10.3.4.3.2



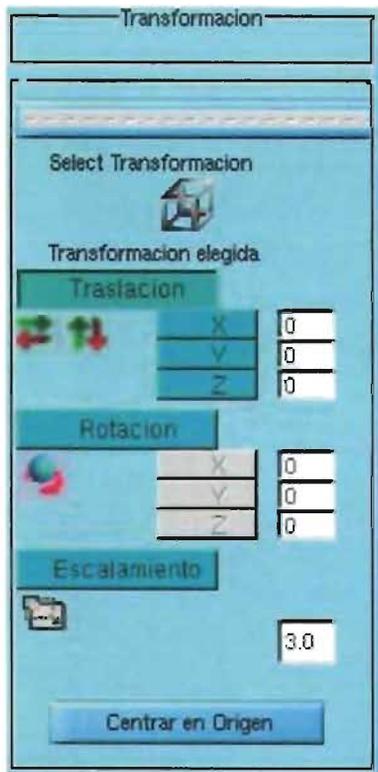


Figura 10.3.4.3.3

En la sección Transformación se tienen tres opciones principales que indican la transformación **Traslación**, **Rotación** y **Escala** y tres opciones que indican sobre que eje se efectúa la operación **X**, **Y** y **Z**, además de cajas de captura de valores que permiten fijar exactamente el valor que se utilizara en la transformación sobre la escena gráfica.

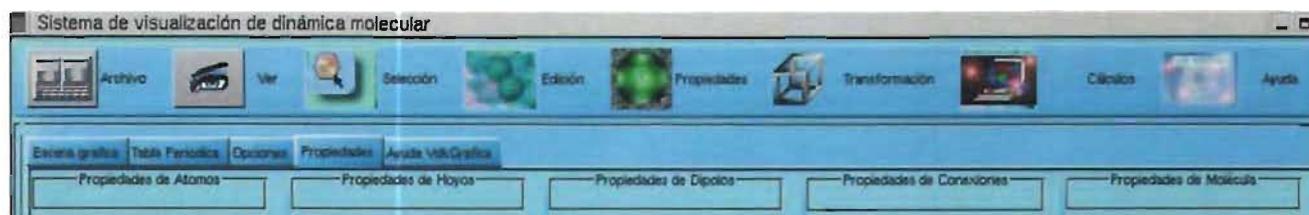
Existe además un botón **Centrar en Origen** que permite regresar la escena gráfica a la posición original antes de las transformaciones efectuadas.

Las siguientes transformaciones están implementadas para hacer uso del teclado y facilitar la manipulación de la escena gráfica.

Tecla	Transformación
x	Trasladar en el eje x
X	Trasladar en el eje -x
y	Trasladar en el eje y
Y	Trasladar en el eje -y
m	Zoom en el eje z
M	Zoom en el eje -z
Flecha arriba	Rotar en el eje x
Flecha abajo	Rotar en el eje -x
Flecha derecha	Rotar en el eje y
Flecha izquierda	Rotar en el eje -y
p	Incrementa el número de plano potencial
P	Decrementa el número de plano potencial

Para hacer uso de las funciones del teclado solo hay que presionar la tecla ENTER sobre la escena gráfica.

10.3.4.4 Propiedades



10.3.4.4.1 Propiedad Radio

Al seleccionar el menú **Propiedades-> Radio** se activan algunas de las opciones siguientes:



Radio de átomos: Ésta sección es para indicar al sistema los valores de radio para los elementos de tipo átomo.



Radio de dipolos: Ésta sección se encuentra desactivada en esta versión del sistema.



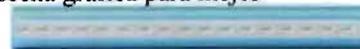
Radio de conexiones: Esta sección es para indicar al sistema los valores de radio para los elementos de tipo conexión.



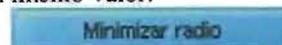
Radio de molécula: Esta sección se encuentra desactivada en esta versión del sistema.

Opciones:

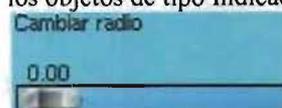
Este botón permite abrir una ventana con las mismas opciones pero sobre la escena gráfica para mejor manipulación y visualización de los objetos gráficos.



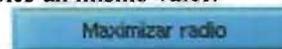
Este botón permite minimizar el radio de los elementos indicados asignándoles un mismo valor.



Esta barra de desplazamiento permite cambiar el valor del radio para asignárselo a los objetos de tipo Indicado.



Este botón permite maximizar el radio de todos los elementos indicados asignándoles un mismo valor.



10.3.4.4.2 Propiedad Transparencia

Al seleccionar el menú **Propiedades-> Transparencia** se activan algunas de las opciones siguientes:



Transparencia de átomos: Ésta sección es para indicar al sistema los valores de transparencia para los elementos de tipo átomo.



Transparencia de hoyos: Ésta sección es para indicar al sistema los valores de transparencia para los elementos de tipo hoyo.



Transparencia de dipolos: Ésta sección se encuentra desactivada en esta versión del sistema.

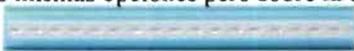


Transparencia de conexiones: Ésta sección es para indicar al sistema los valores de transparencia para los elementos de tipo conexión.

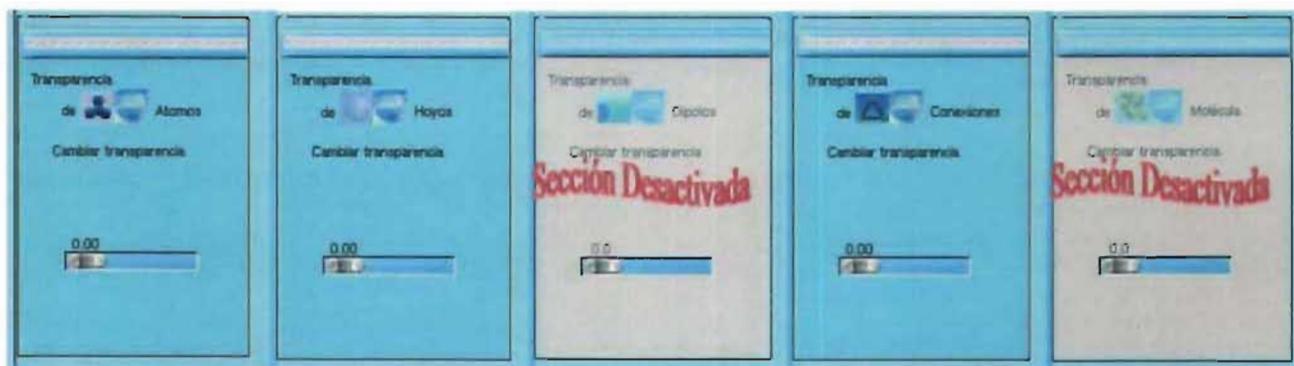
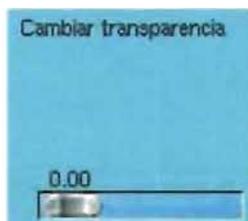


Transparencia de molécula: Ésta sección se encuentra desactivada en ésta versión del sistema.

Opciones:

Éste botón permite abrir una ventana con las mismas opciones pero sobre la escena gráfica para mejor manipulación y visualización de los objetos gráficos. 

Ésta barra de desplazamiento permite cambiar el valor de la transparencia para asignárselo a los objetos de tipo indicado.



10.3.4.4.3 Propiedad Calidad

Al seleccionar el menú **Propiedades-> Calidad** se activan algunas de las opciones siguientes:



Calidad de átomos: Ésta sección es para indicar al sistema los valores de calidad para los elementos de tipo átomo.



Calidad de hoyos: Ésta sección es para indicar al sistema los valores de calidad para los elementos de tipo hoyo.



Calidad de dipolos: Ésta sección se encuentra desactivada en ésta versión del sistema.



Calidad de conexiones: Ésta sección es para indicar al sistema los valores de calidad para los elementos de tipo conexión.



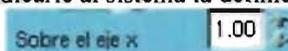
Calidad de molécula: Ésta sección se encuentra desactivada en ésta versión del sistema.

Opciones:

Éste botón permite abrir una ventana con las mismas opciones pero sobre la escena gráfica para mejor manipulación y visualización de los objetos gráficos.



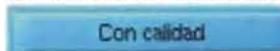
Ésta caja de captura de valores permite indicarle al sistema la definición de calidad en el eje x.



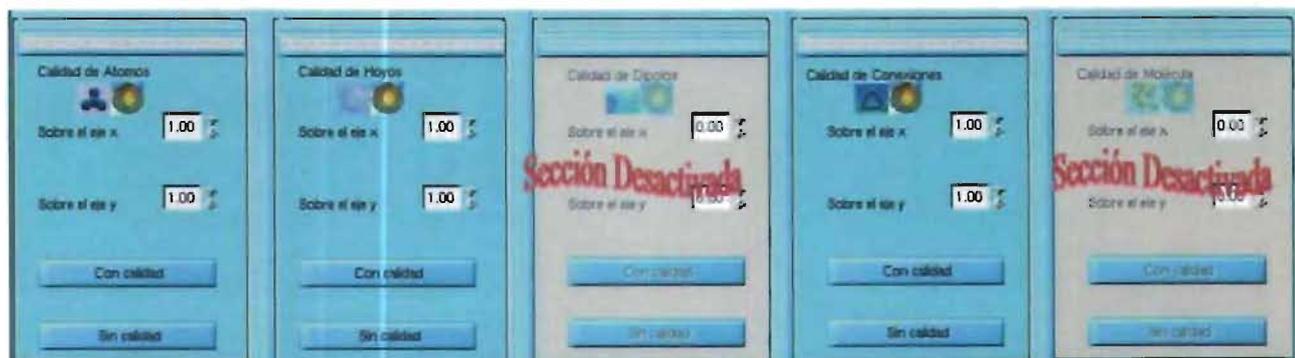
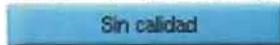
Ésta caja de captura de valores permite indicarle al sistema la definición de calidad en el eje y.



Éste botón permite poner máxima calidad a todos los elementos indicados asignándoles un mismo valor.



Éste botón permite poner mínima calidad a todos los elementos indicados asignándoles un mismo valor.



10.3.4.4 Propiedad Estilo

Al seleccionar el menú Propiedades-> Estilo se activan algunas de las opciones siguientes:



Estilo de átomos: Ésta sección es para indicar al sistema los valores de estilo para los elementos de tipo átomo.



Estilo de hoyos: Ésta sección es para indicar al sistema los valores de estilo para los elementos de tipo hoyo.



Estilo de dipolos: Ésta sección se encuentra desactivada en ésta versión del sistema.



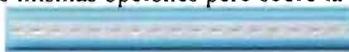
Estilo de conexiones: Ésta sección es para indicar al sistema los valores de estilo para los elementos de tipo conexión.



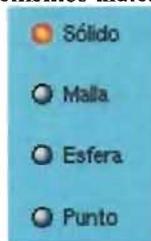
Estilo de molécula: Ésta sección se encuentra desactivada en ésta versión del sistema.

Opciones:

Éste botón permite abrir una ventana con las mismas opciones pero sobre la escena gráfica para mejor manipulación y visualización de los objetos gráficos.



Ésta sección de botones radio es para seleccionar el estilo de visualización de los elementos indicados.



10.3.4.4.5 Propiedad Color

Color interfaz gráfica



La opción para cambiar el color a la interfaz (que esté resaltada en la imagen de arriba) llama a un cuadro de dialogo de selección de color como se ve en la figura 10.3.4.4.5.1 de la derecha. Desde ésta ventana se puede seleccionar un color para la interfaz gráfica usando la circunferencia, las barras de desplazamiento de Tinte, Saturación Valor, Rojo, Verde, Azul y Opacidad así como sus respectivas cajas de valores, para capturar el dato directamente.

Una vez seleccionado el color y ya que se ha presionado el botón aceptar, el dialogo de color desaparece y se aplica el nuevo color a toda la interfaz como se muestra en la figura 10.3.4.4.5.2

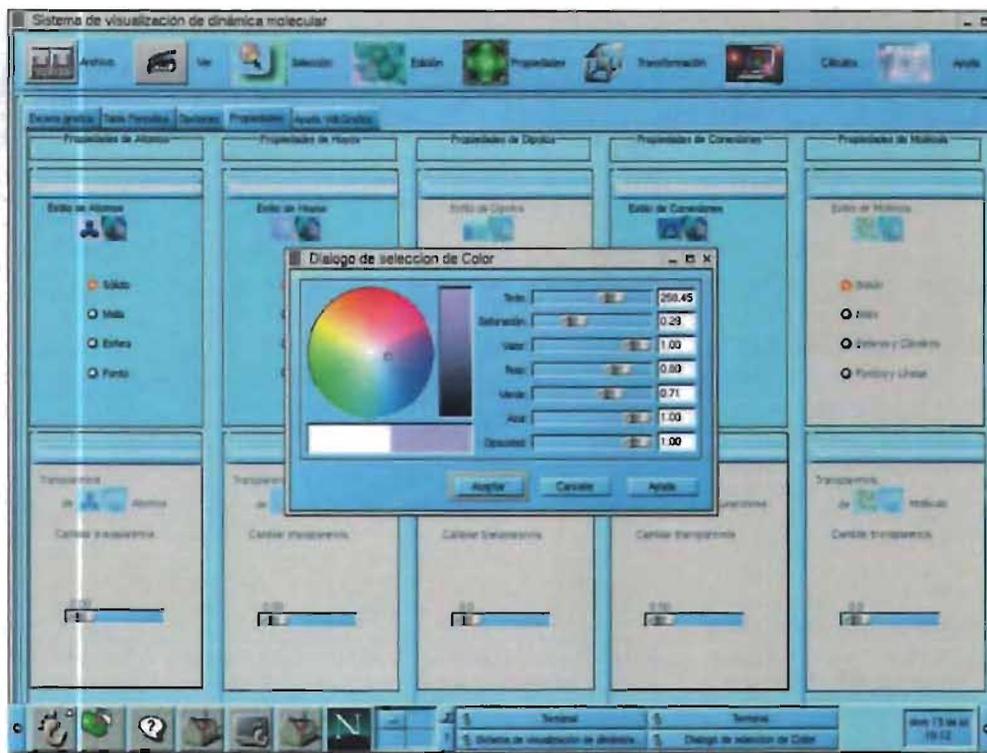


Figura 10.3.4.4.5.1

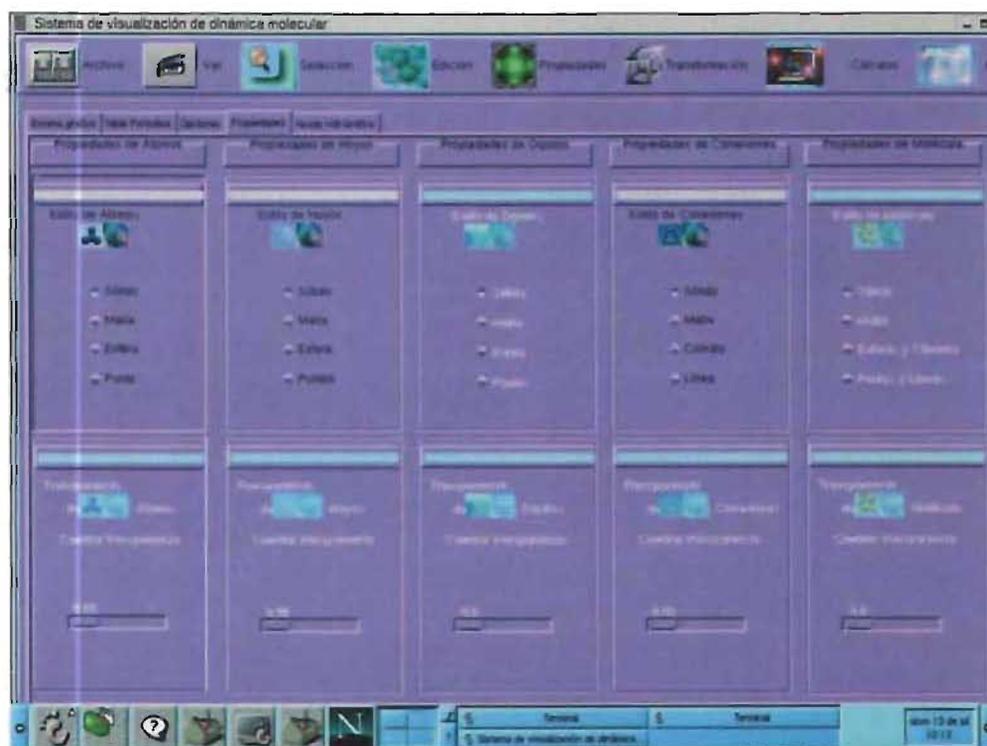


Figura 10.3.4.4.5.2

Color Fondo



La opción para cambiar el color al Fondo (que esta resaltada en la imagen de arriba) llama a un cuadro de dialogo de selección de color como se ve en la Figura 10.3.4.4.5.3 de la derecha. Desde esta ventana se puede seleccionar un color para el usando la circunferencia, las barras de desplazamiento de Tinte, Saturación Valor, Rojo, Verde, Azul y Opacidad así como sus respectivas cajas de valores, para capturar el dato directamente.

Una vez seleccionado el color y ya que se ha presionado el botón aceptar, el dialogo de color desaparece y se aplica el nuevo color al fondo como se muestra en la Figura 10.3.4.4.5.4



Figura 10.3.4.4.5.3

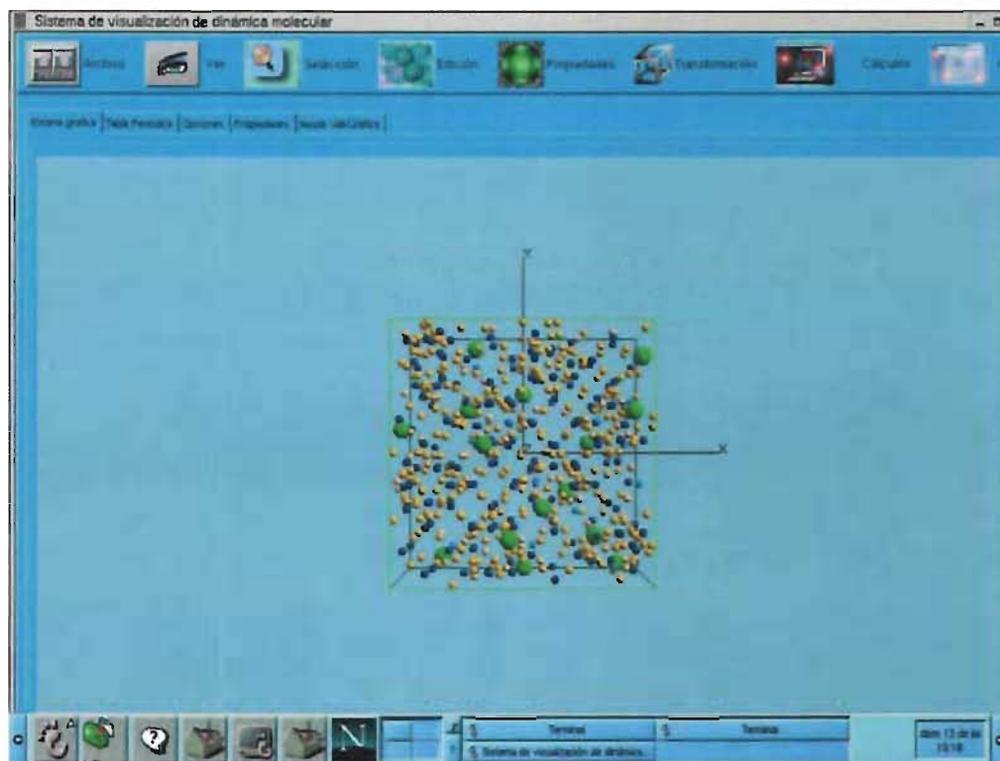


Figura 10.3.4.4.5.4

10.3.4.5 Ayuda

La ayuda que se muestra en el sistema es el mismo contenido de este capítulo, se puede consultar seleccionado el tema, delante de cada uno hay pequeños botones que indican la página o páginas con el material correspondiente a ese tema.

Las Figuras 10.3.4.5.1, 10.3.4.5.2 y 10.3.4.5.3 muestran un ejemplo de cómo se ve la ayuda desde el sistema.

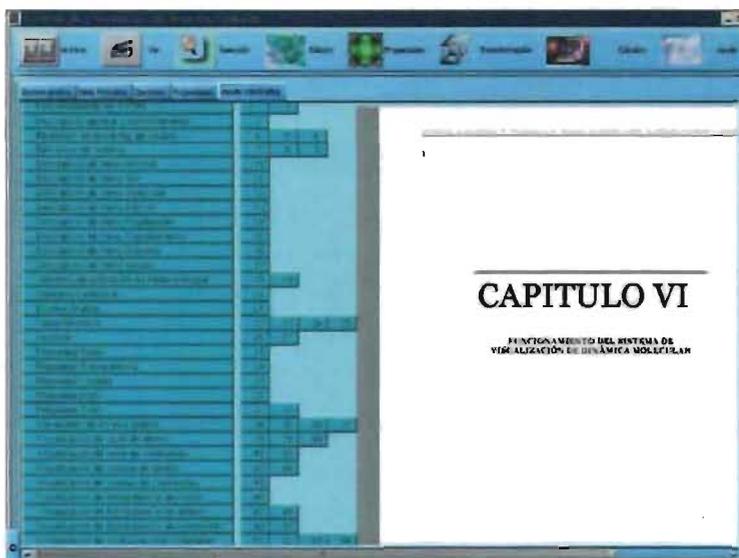


Figura 10.3.4.5.1

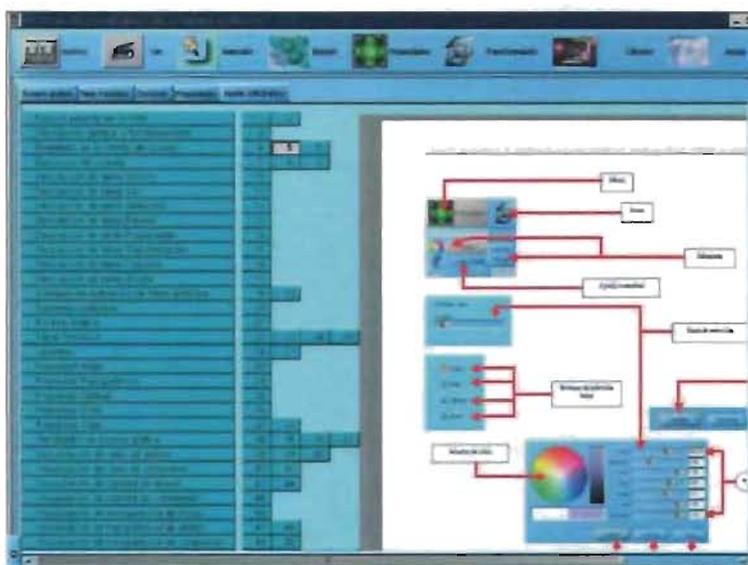


Figura 10.3.4.5.2



Figura 10.3.4.5.3

10.3.5 Funcionamiento del sistema

10.3.5.1 Ver objetos de escena gráfica

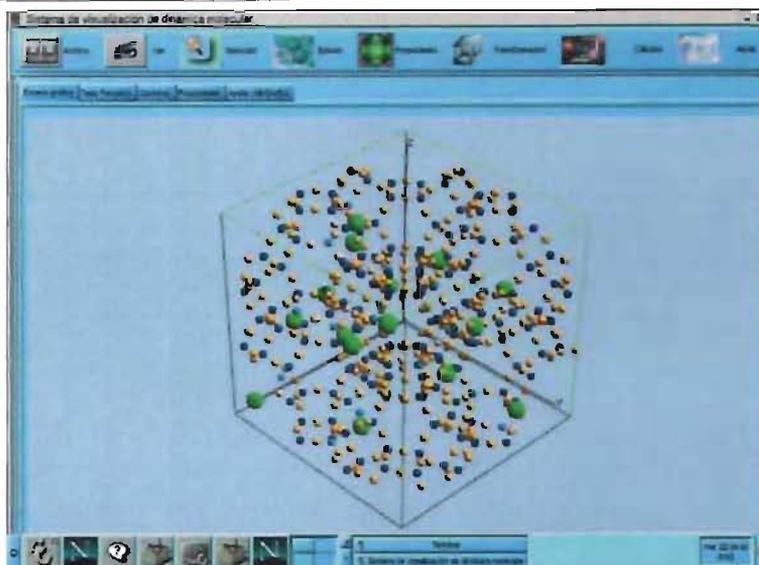


Figura 10.3.5.1.1

La Figura 10.3.5.1.1 muestra una configuración de átomos de Silicio, Aluminio, Oxígeno y Sodio (archivo de faujasita) con los Átomos, Caja y Ejes activados para su visualización como se muestra en el menú Ver en la imagen de arriba.



Figura 10.3.5.1.2



Figura 10.3.5.1.3

En la Figura 10.3.5.1.2 se muestran los hoyos o cavidades entre los átomos en Estilo Sólido. En la Figura 10.3.5.1.3 se pueden ver en Estilo Malla, los hoyos pertenecen a la configuración de átomos de la Figura 10.3.5.1.1.

La visualización de los hoyos se hace seleccionando la opción indicada en el menú Ver en la imagen de arriba.

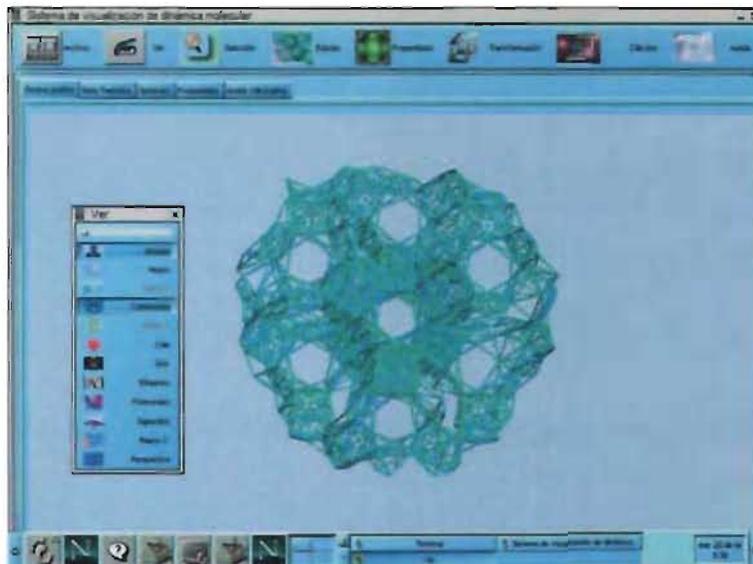


Figura 10.3.5.1.4

La Figura 10.3.5.1.4 muestra la visualización de las conexiones entre los átomos de la Figura 10.3.5.1.1 con un radio de conexión de 5.0 en la Figura 10.3.5.1.5 se ve la misma configuración pero con un radio de conexión de 1.7.

Para mostrar visibles las conexiones se debe seleccionar la opción correspondiente como se indica en el menú Ver en la imagen de arriba.

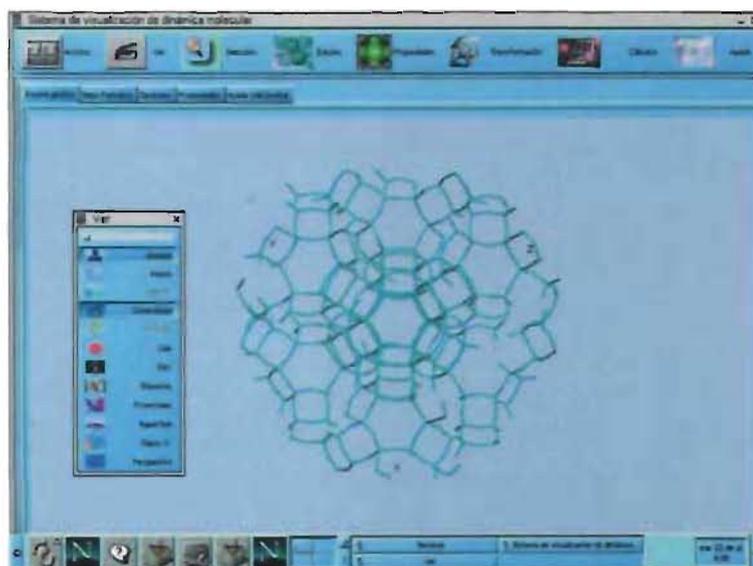


Figura 10.3.5.1.5

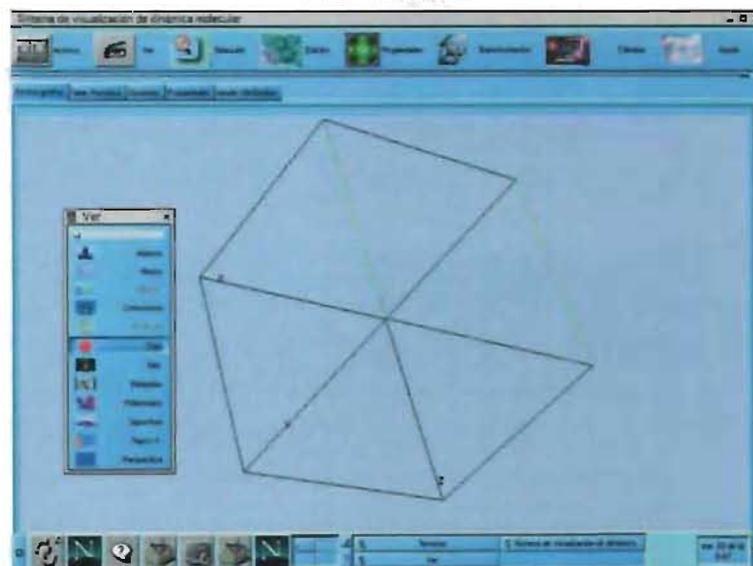


Figura 10.3.5.1.6

En la Figura 10.3.5.1.6 se muestra la caja o celda unitaria que contiene las moléculas y sus componentes para su estudio y manipulación. Para mostrarla seleccionar la opción correspondiente como se muestra en el menú Ver en la imagen de arriba.



La Figura 10.3.5.1.7 muestra la visualización de las etiquetas que identifican a cada átomo en particular de la configuración mostrada en la Figura 10.3.5.1.1 Para mostrar visibles las etiquetas seleccionar la opción correspondiente como se indica en el menú Ver en la imagen de arriba.

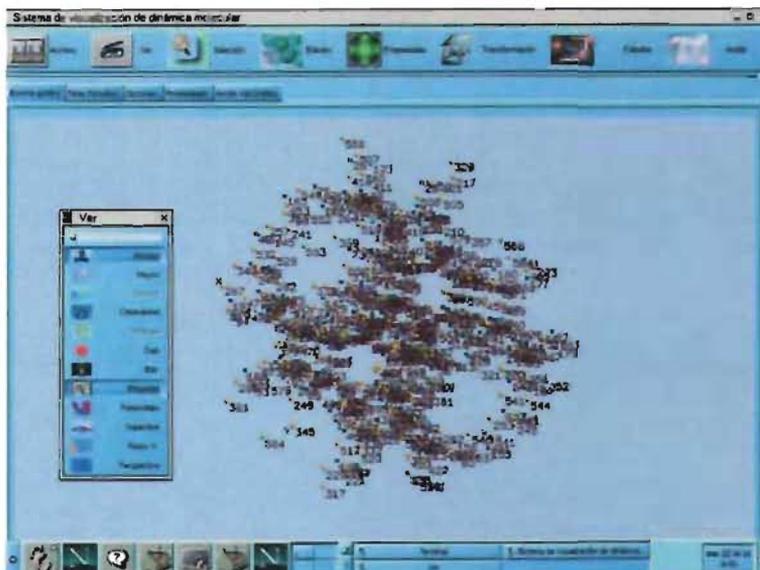


Figura 10.3.5.1.7

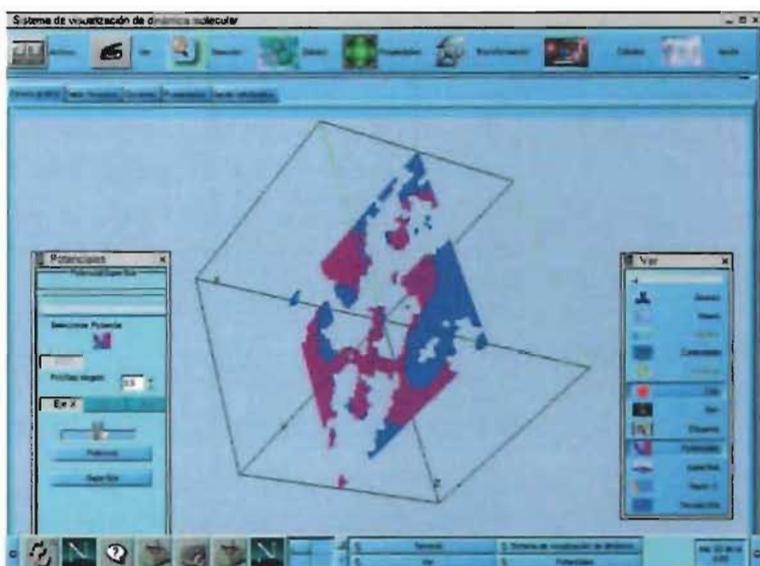


Figura 10.3.5.1.8



Las Figuras 10.3.5.1.8 y 10.3.5.1.9 muestran la visualización de planos potenciales en los ejes X (Eje X, Eje Y, Eje Z) y Y (Eje X, Eje Y, Eje Z) de la configuración mostrada en la Figura 10.3.5.1.1 Para mostrar visibles los planos potenciales seleccionar la opción correspondiente como se indica en el menú Ver en la imagen de arriba.

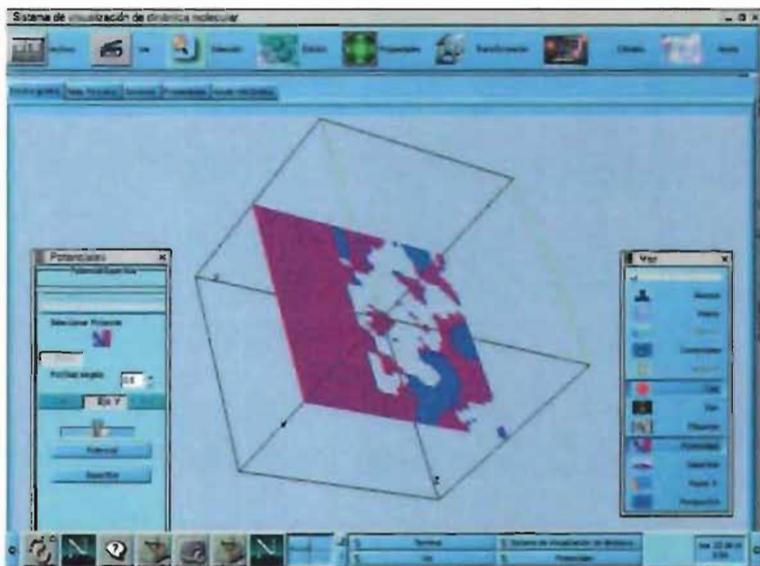


Figura 10.3.5.1.9

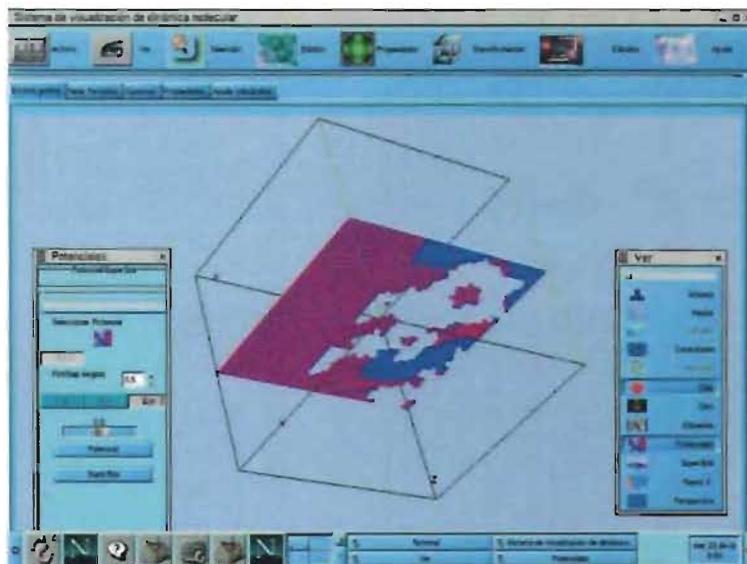
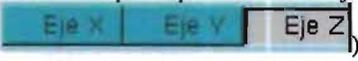


Figura 10.3.5.1.10

La Figura 10.3.5.1.10 muestra la visualización del plano potencial en el eje Z () de la configuración mostrada en la Figura 10.3.5.1.1 Para mostrar visible el plano potencial seleccionar la opción correspondiente como se indica en el menú Ver en la imagen de arriba.

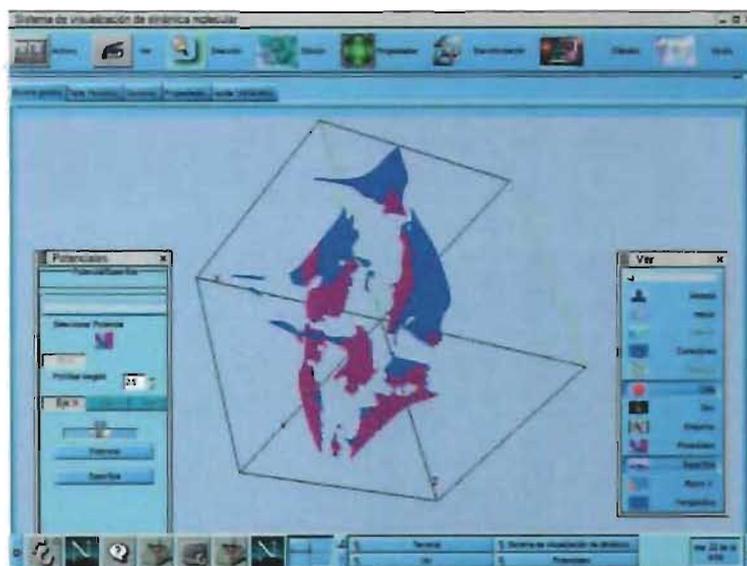
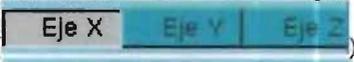


Figura 10.3.5.1.11



Las Figuras 10.3.5.1.11 y 10.3.5.1.12 muestran la visualización de superficies potenciales en los ejes X () y Y () de la configuración mostrada en la Figura 10.3.5.1.1 Para mostrar visibles las superficies potenciales seleccionar la opción correspondiente como se indica en el menú Ver en la imagen de arriba.

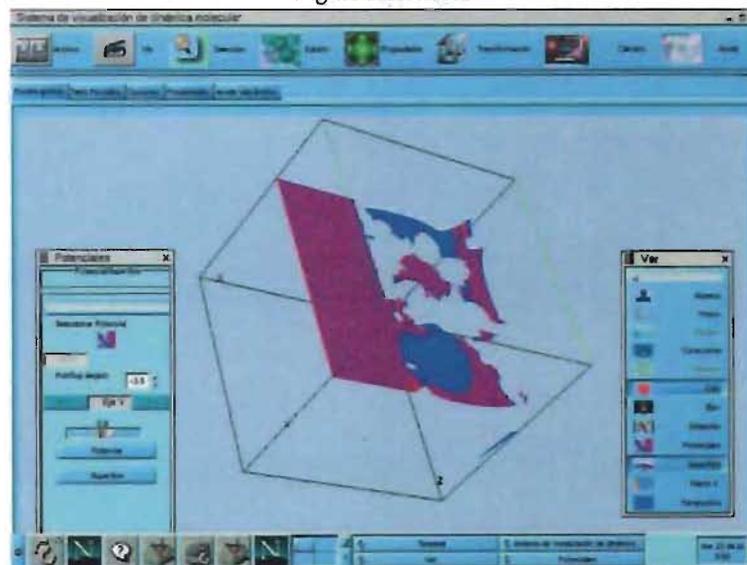
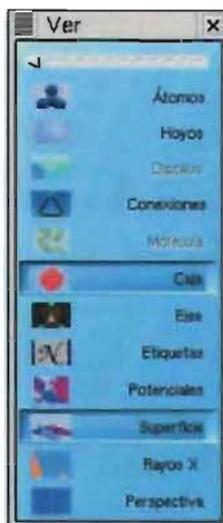


Figura 10.3.5.1.12



La Figura 10.3.5.1.13 muestra la visualización de la superficie potencial en el eje Z (Eje X | Eje Y | Eje Z) de la configuración mostrada en la Figura 10.3.5.1.1. Para mostrar visible la superficie potencial seleccionar la opción correspondiente como se indica en el menú Ver en la imagen de arriba.

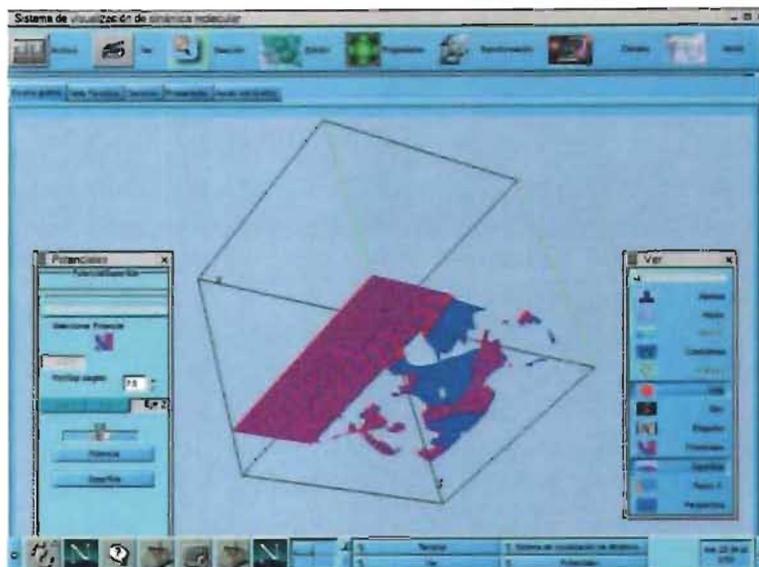


Figura 10.3.5.1.13

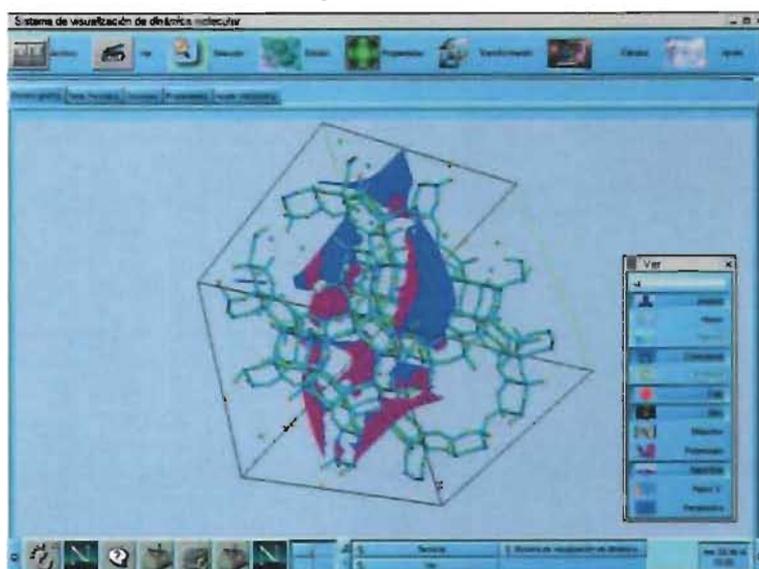


Figura 10.3.5.1.14



La Figura 10.3.5.1.14 muestra una visualización de átomos, conexiones, caja, ejes y superficie potencial de la configuración mostrada en la Figura 10.3.5.1.1

La Figura 10.3.5.1.15 muestra una visualización de átomos, hoyos, conexiones, caja y ejes de la configuración mostrada en la Figura 10.3.5.1.1

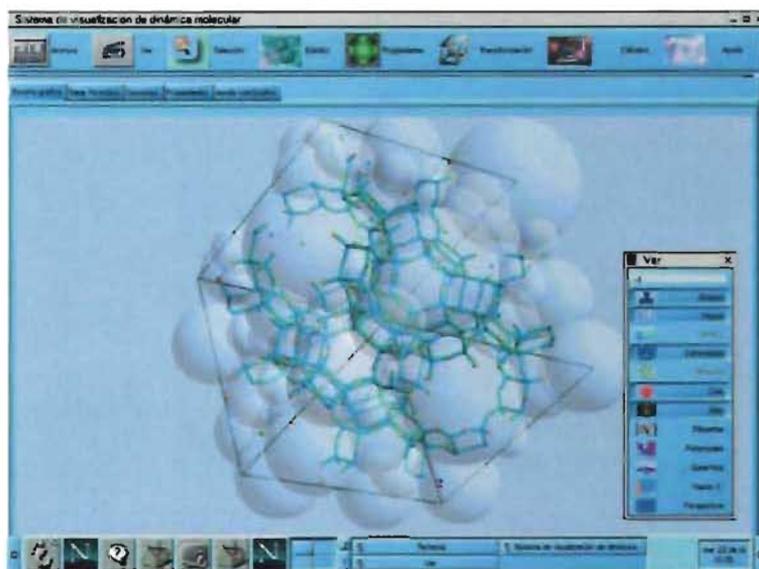
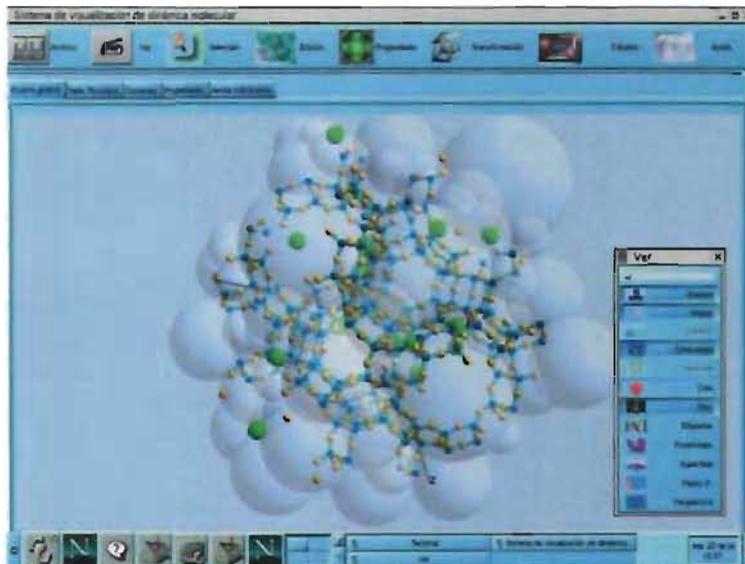


Figura 10.3.5.1.15



La Figura 10.3.5.1.16 muestra una visualización de átomos, hoyos, conexiones y ejes de la configuración mostrada en la Figura 10.3.5.1.1

Figura 10.3.5.1.16

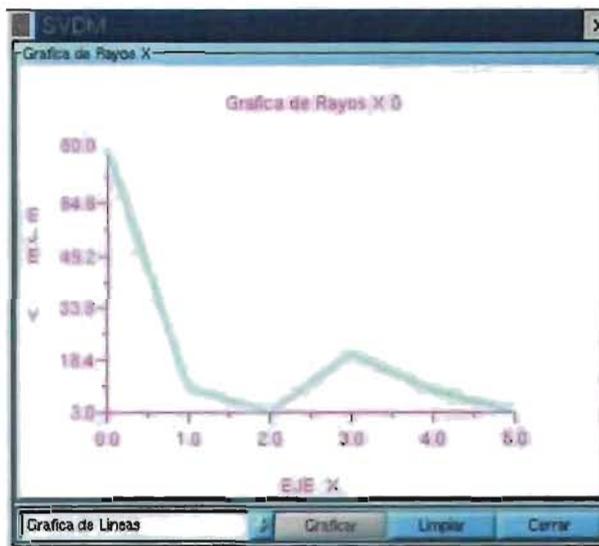


Figura 10.3.5.1.17

Las Figuras 10.3.5.1.17 y 10.3.5.1.18 muestran la gráfica de rayos x de la configuración mostrada en la Figura 10.3.5.1.1

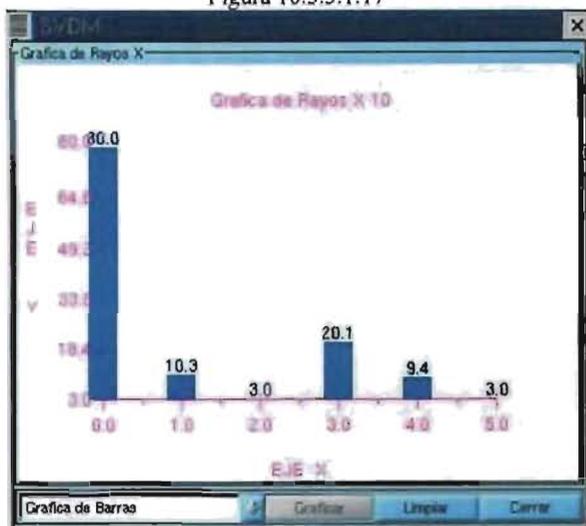


Figura 10.3.5.1.18

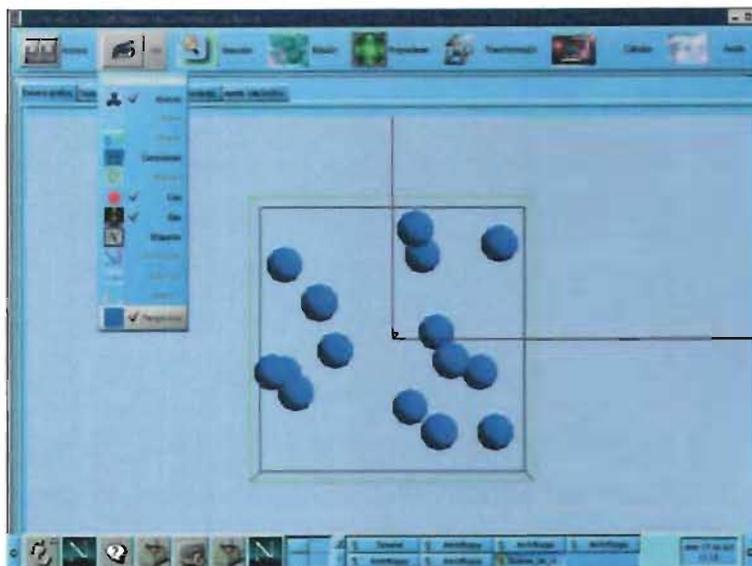


Figura 10.3.5.1.19

Las Figura 10.3.5.1.19 muestra la perspectiva activada

Las Figura 10.3.5.1.20 muestra la perspectiva desactivada

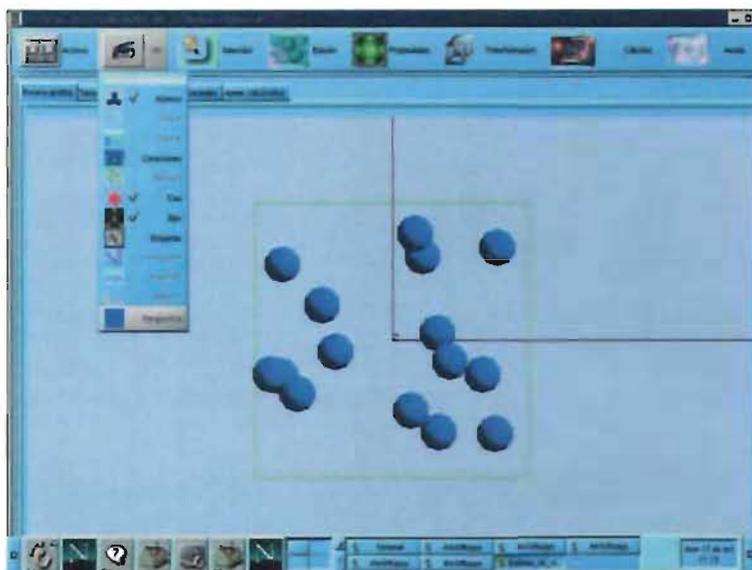


Figura 10.3.5.1.20

10.3.5.2 Visualización de propiedades en objetos gráficos

10.3.5.2.1 Radio de átomos

El radio de los átomos puede cambiarse utilizando los botones y la barra de desplazamiento en la ventana de ésta propiedad.

La Figura 10.3.5.2.1.1 muestra una visualización de átomos con radio maximizado.

La Figura 10.3.5.2.1.2 muestra una visualización de átomos con radio minimizado.



Figura 10.3.5.2.1.1

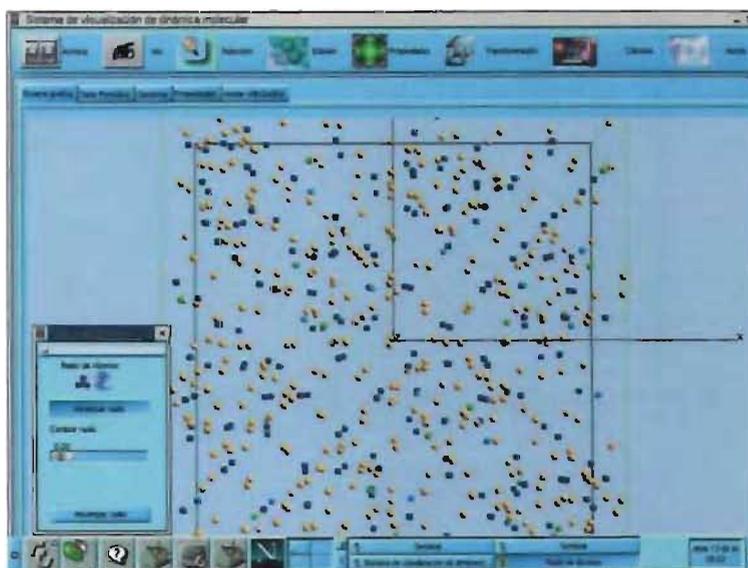


Figura 10.3.5.2.1.2

La Figura 10.3.5.2.1.3 muestra una visualización de átomos inicial con radio para cada tipo de átomo tomado del archivo de configuración atomo.h

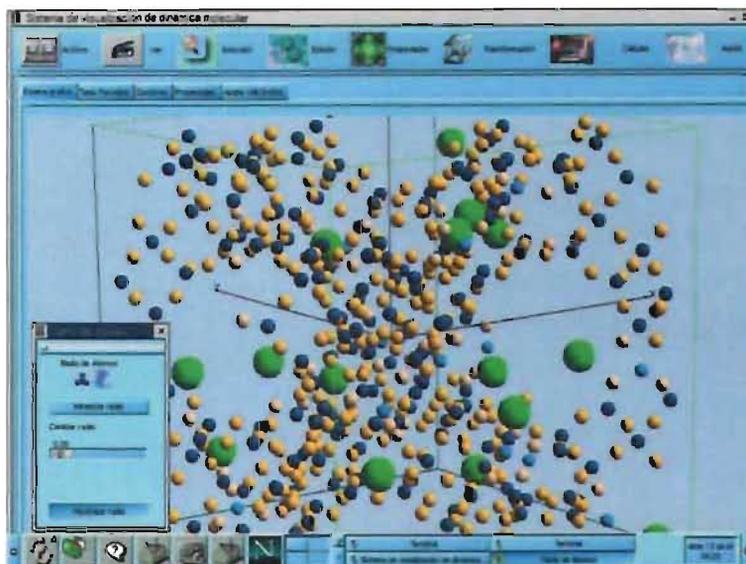


Figura 10.3.5.2.1.3

La Figura 10.3.5.2.1.4 muestra una visualización de átomos con radio de 0.05.

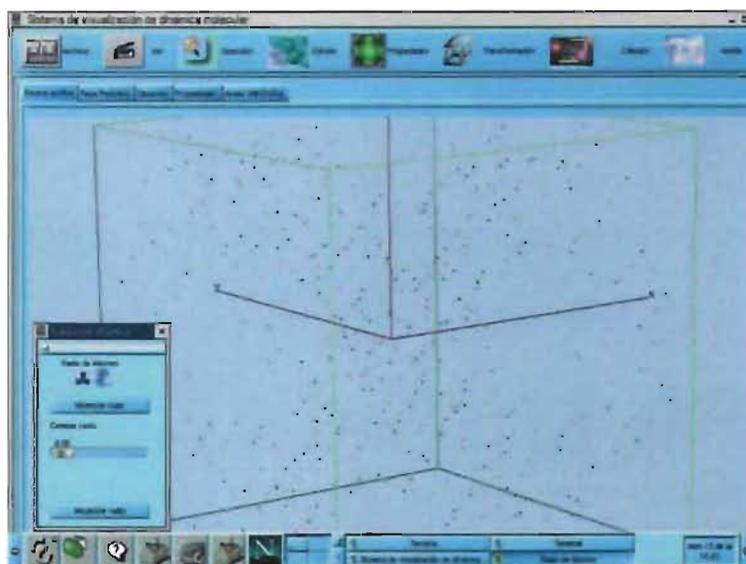


Figura 10.3.5.2.1.4

La Figura 10.3.5.2.1.5 muestra una visualización de átomos con radio de 0.25.

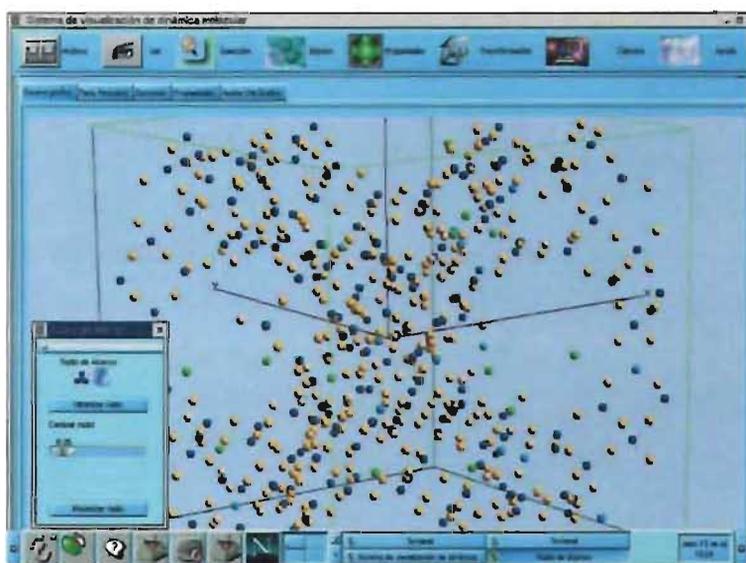


Figura 10.3.5.2.1.5

La Figura 10.3.5.2.1.6 muestra una visualización de átomos con radio de 0.56.

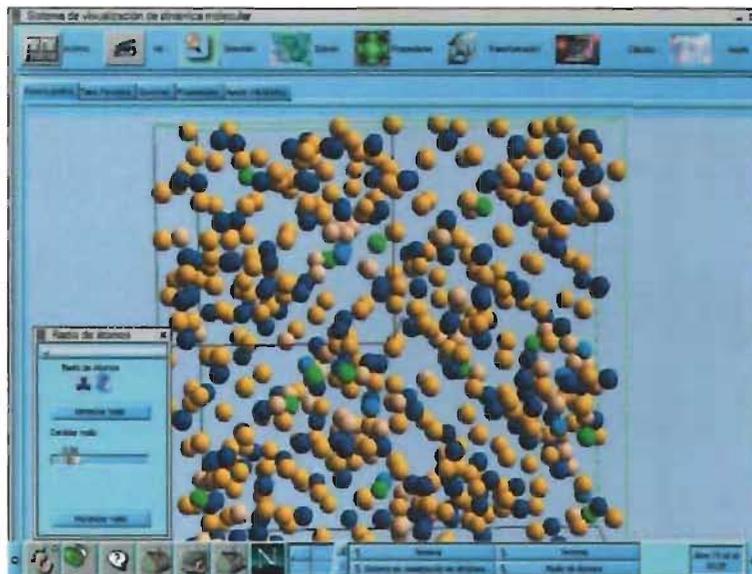


Figura 10.3.5.2.1.6

La Figura 10.3.5.2.1.7 muestra una visualización de átomos con radio de 1.11.

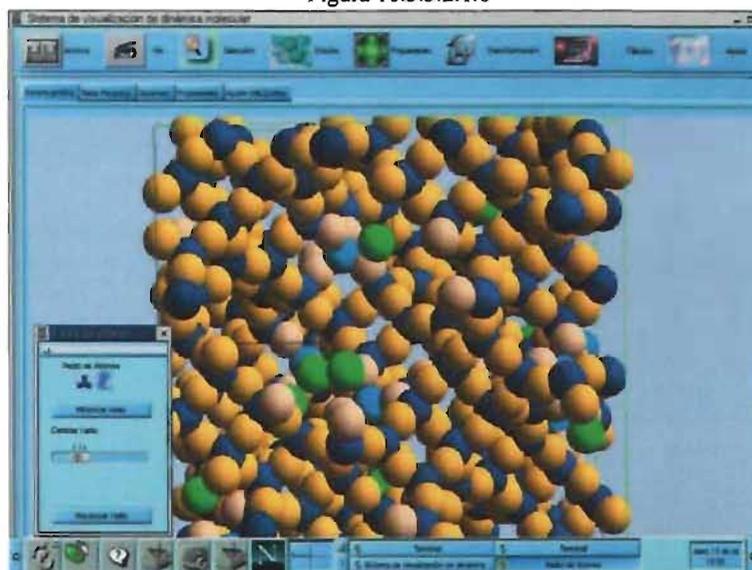


Figura 10.3.5.2.1.7

La Figura 10.3.5.2.1.8 muestra una visualización de átomos con radio de 2.27.

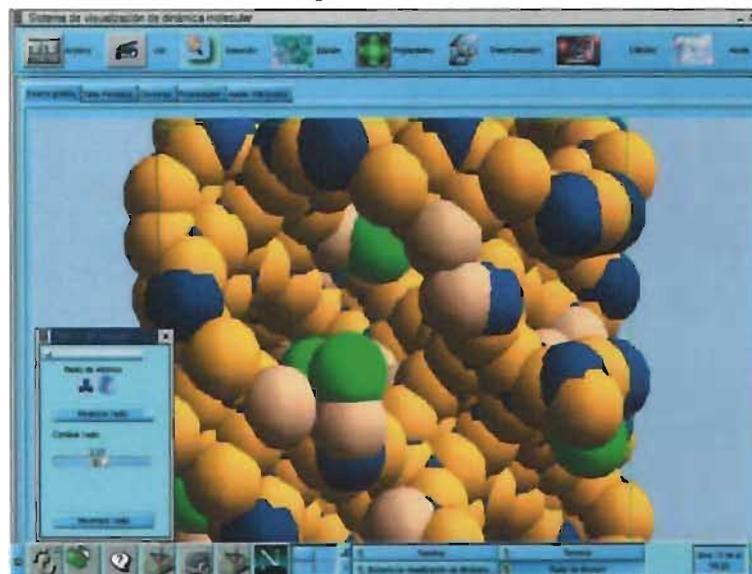


Figura 10.3.5.2.1.8

10.3.5.2.2 Radio de conexiones

El radio de las conexiones puede cambiarse utilizando los botones y la barra de desplazamiento en la ventana de ésta propiedad.



En la Figura 10.3.5.2.2.1 se muestra la configuración de conexiones por tipo, se ha seleccionado un radio de conexión entre vecinos de 1.7 para los átomos de Sodio, Aluminio, Silicio y Oxígeno.

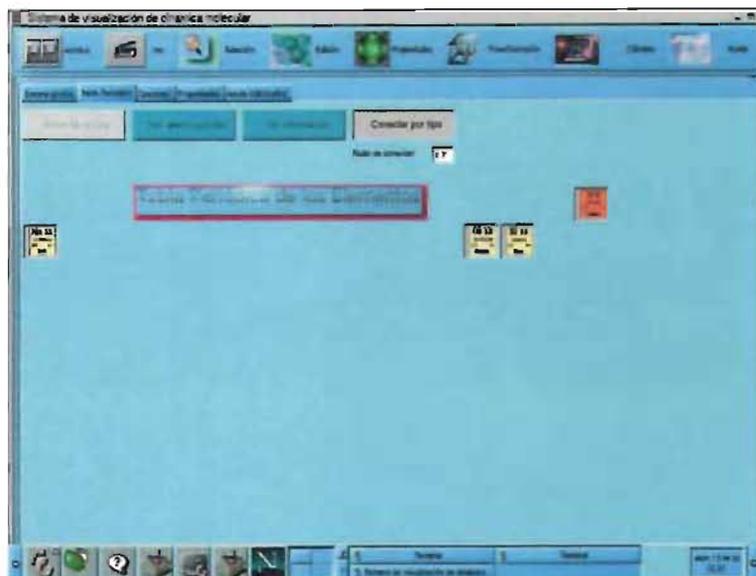


Figura 10.3.5.2.2.1

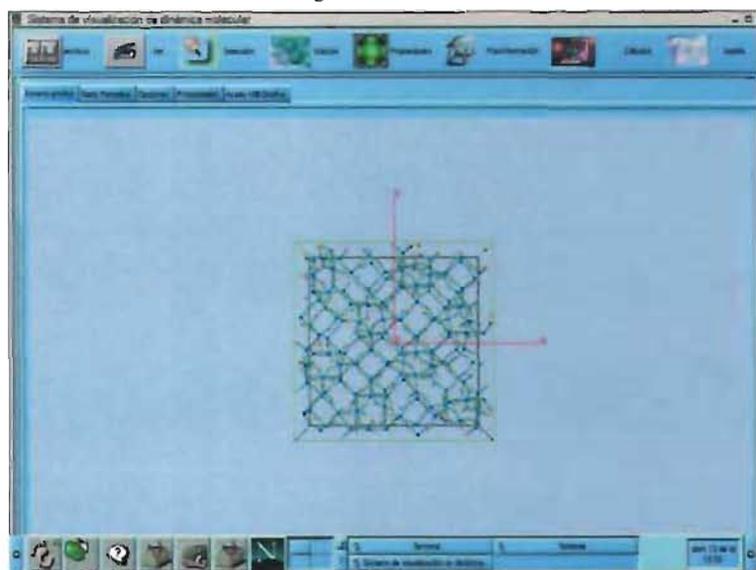
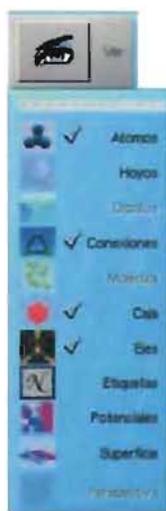


Figura 10.3.5.2.2.2



En la Figura 10.3.5.2.2.2 se muestra la visualización de átomos, conexiones, caja y ejes con el radio de conexión entre vecinos seleccionado antes.

En la Figura 10.3.5.2.2.3 se muestra la misma configuración pero con un acercamiento y con las conexiones con su radio minimizado (radio del prisma que forma el objeto conexión).

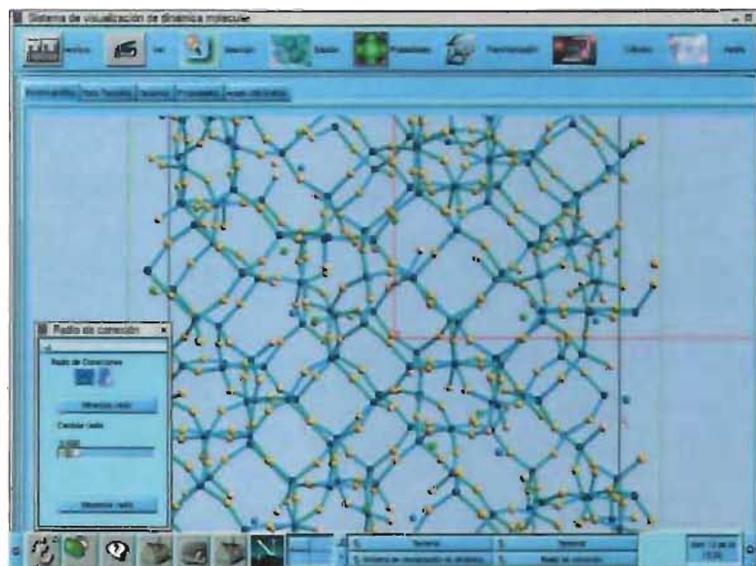


Figura 10.3.5.2.2.3

En la Figura 10.3.5.2.2.4 se muestra la configuración con un acercamiento y con las conexiones con su radio maximizado (radio del prisma que forma el objeto conexión).

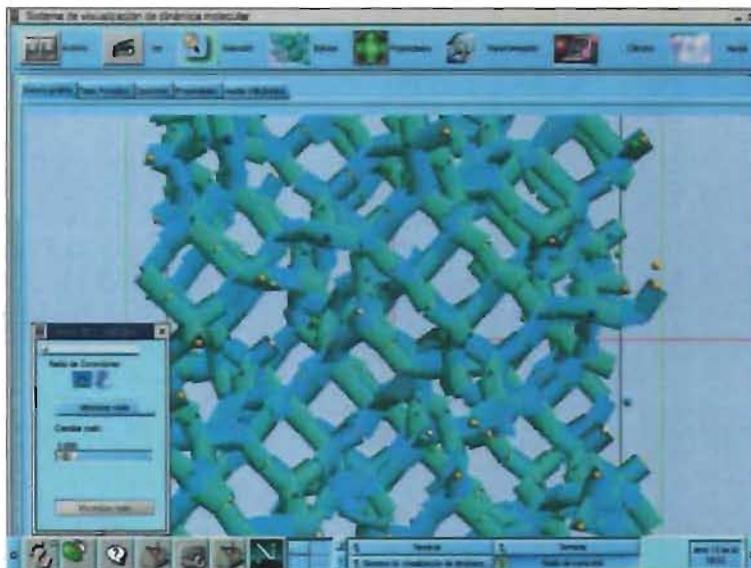


Figura 10.3.5.2.2.4

La Figura 10.3.5.2.2.5 muestra una visualización de conexiones con radio de 0.020.

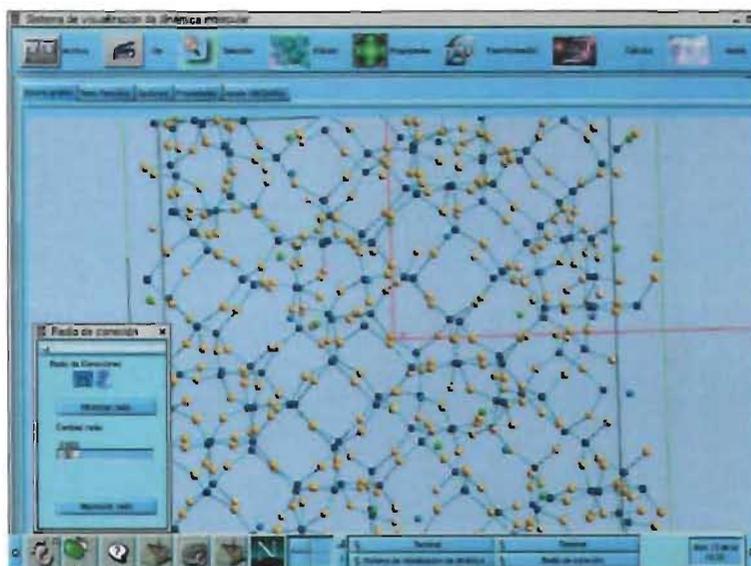


Figura 10.3.5.2.2.5

La Figura 10.3.5.2.2.6 muestra una visualización de conexiones con radio de 0.091.

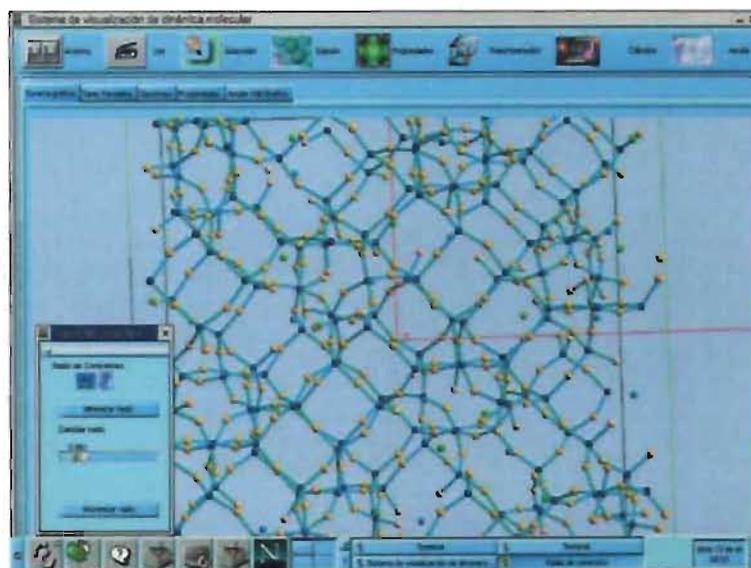


Figura 10.3.5.2.2.6

La Figura 10.3.5.2.2.7 muestra una visualización de conexiones con radio de 0.222.



10.3.5.2.3 Calidad de átomos

La calidad de los átomos puede cambiarse utilizando los botones y las cajas de captura en la ventana de ésta propiedad.

La Figura 10.3.5.2.3.1 muestra una visualización de átomos sin calidad.

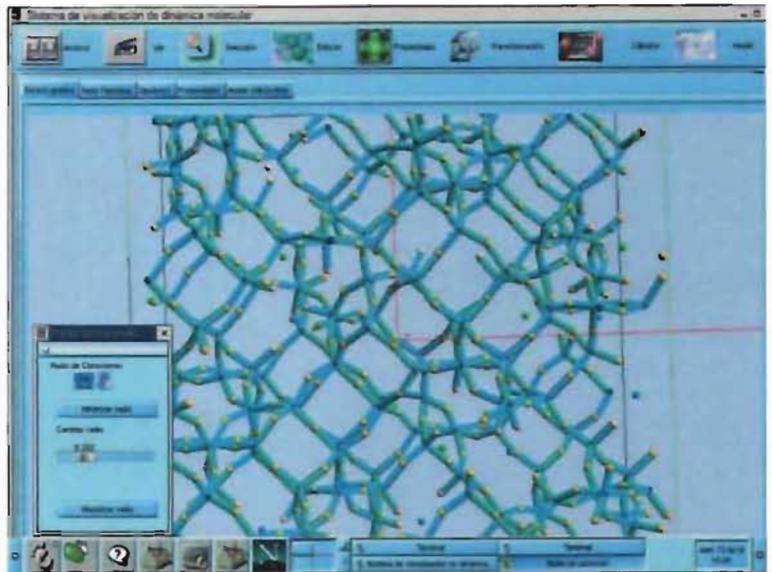


Figura 10.3.5.2.2.7

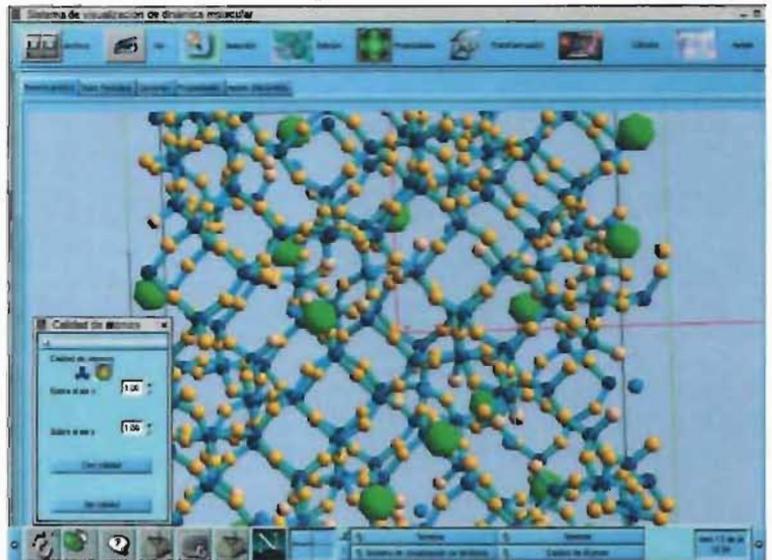


Figura 10.3.5.2.3.1

La Figura 10.3.5.2.3.2 muestra una visualización de átomos con calidad.

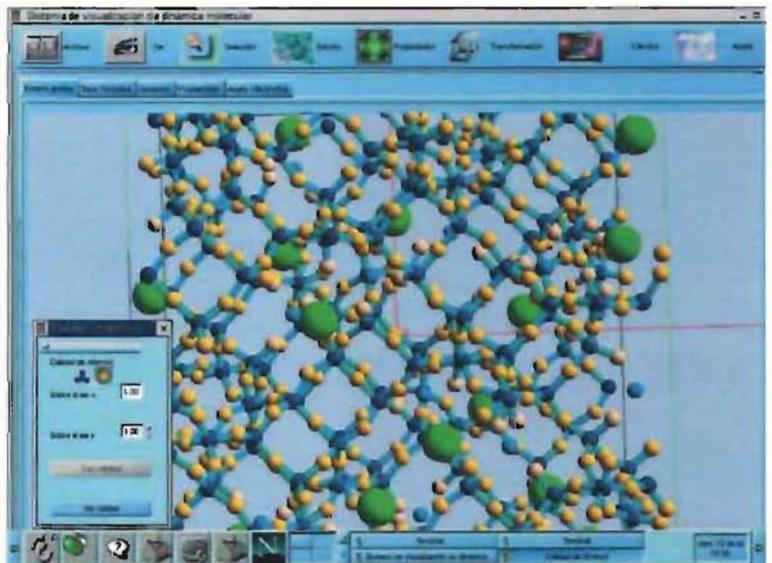


Figura 10.3.5.2.3.2

La Figura 10.3.5.2.3.3 muestra una visualización de átomos con calidad 1 sobre el eje x y calidad 1 sobre el eje y.

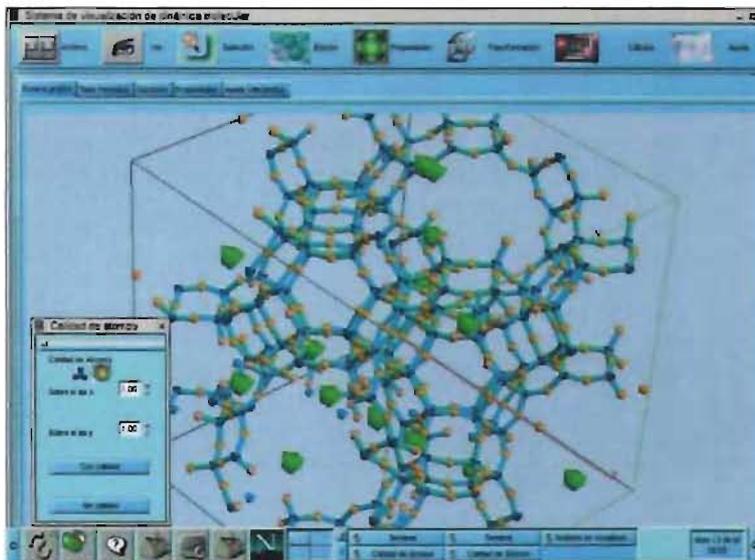


Figura 10.3.5.2.3.3

La Figura 10.3.5.2.3.4 muestra una visualización de átomos con calidad 7 sobre el eje x y calidad 2 sobre el eje y.

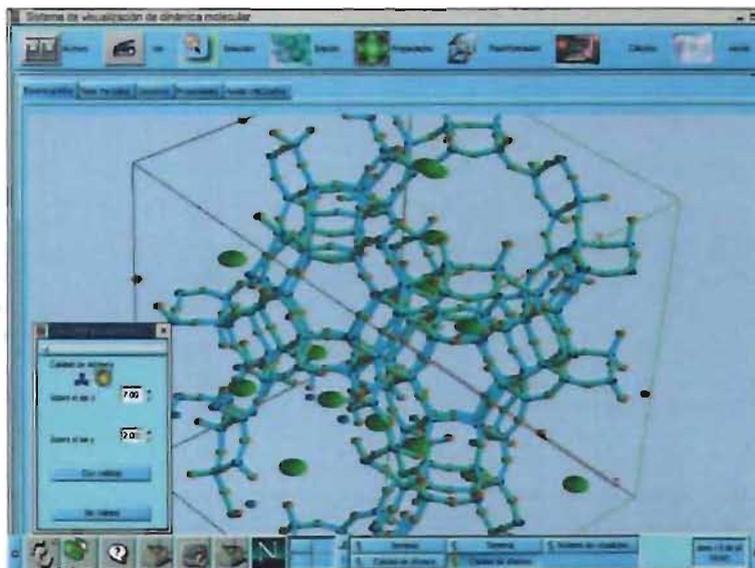


Figura 10.3.5.2.3.4

La Figura 10.3.5.2.3.5 muestra una visualización de átomos con calidad 3 sobre el eje x y calidad 4 sobre el eje y.

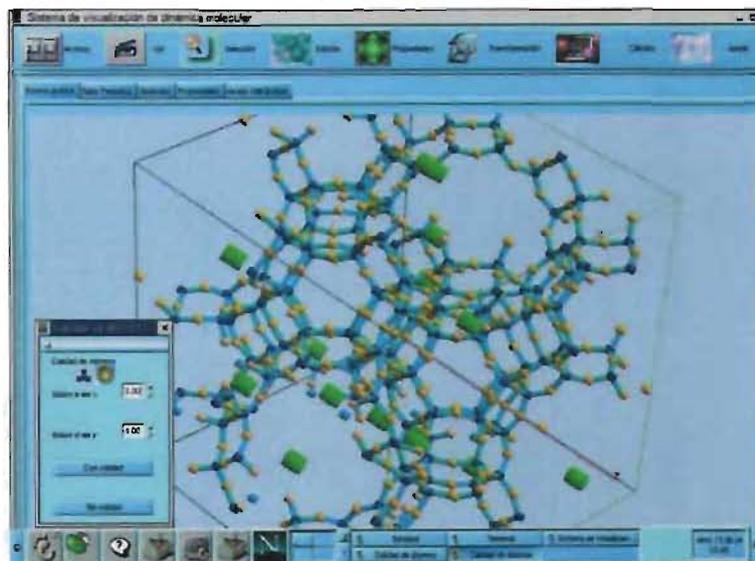


Figura 10.3.5.2.3.5

10.3.5.2.4 Calidad de conexiones

La calidad de las conexiones puede cambiarse utilizando los botones y las cajas de captura en la ventana de ésta propiedad.

La Figura 10.3.5.2.4.1 muestra una visualización de conexiones sin calidad.



La Figura 10.3.5.2.3.2 muestra una visualización de conexiones con calidad.

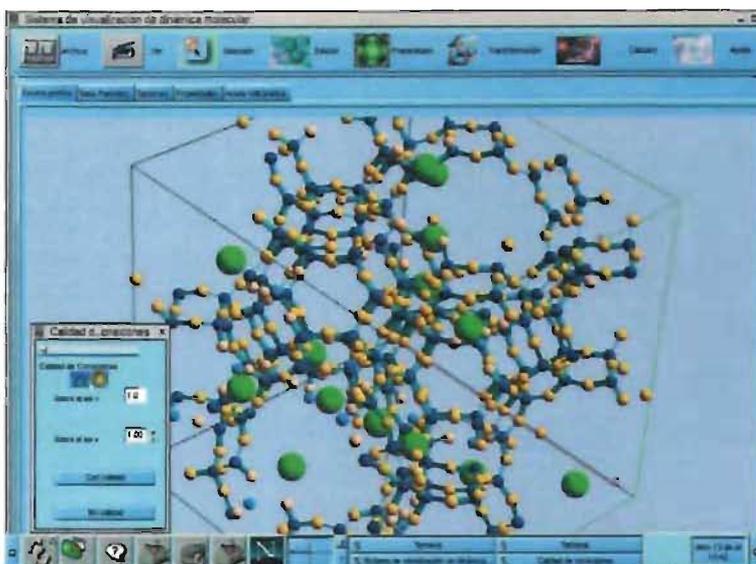


Figura 10.3.5.2.4.1

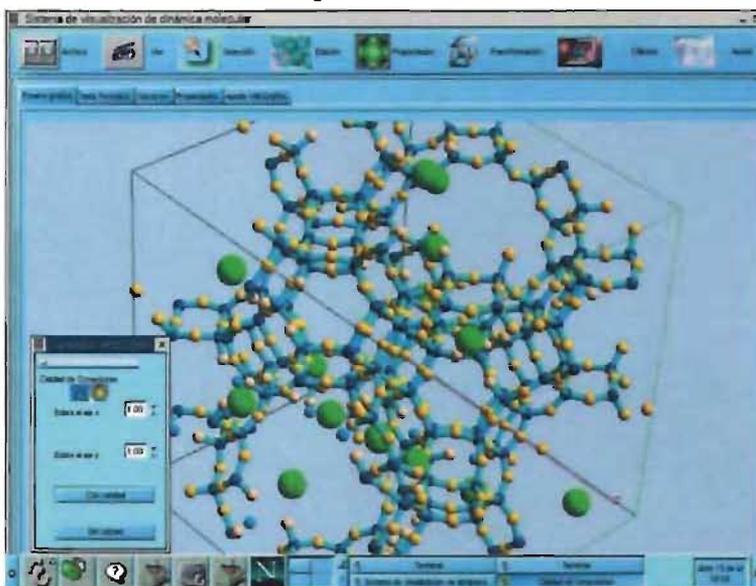


Figura 10.3.5.2.4.2

10.3.5.2.5 Transparencia de hoyos

La transparencia de los hoyos puede cambiarse utilizando la barra de desplazamiento en la ventana de ésta propiedad.

La Figura 10.3.5.2.5.1 muestra una visualización de hoyos con transparencia igual a 1.00.

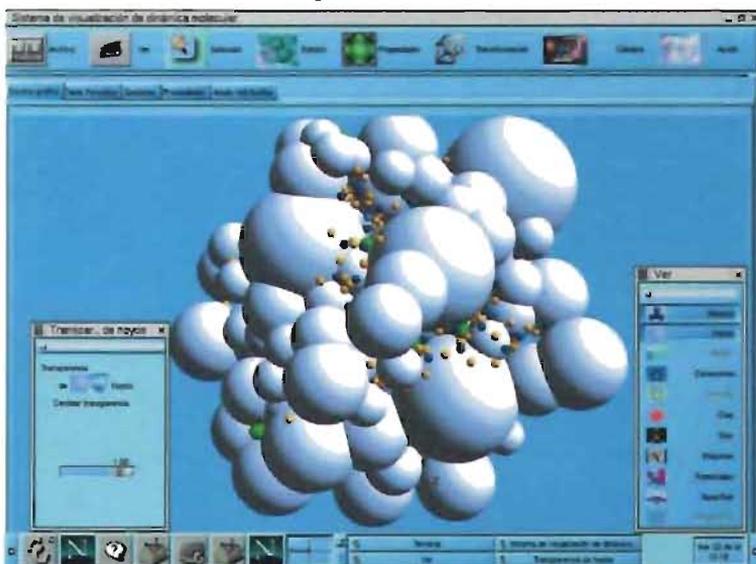


Figura 10.3.5.2.5.1

La Figura 10.3.5.2.5.2 muestra una visualización de hoyos con transparencia igual a 0.71.

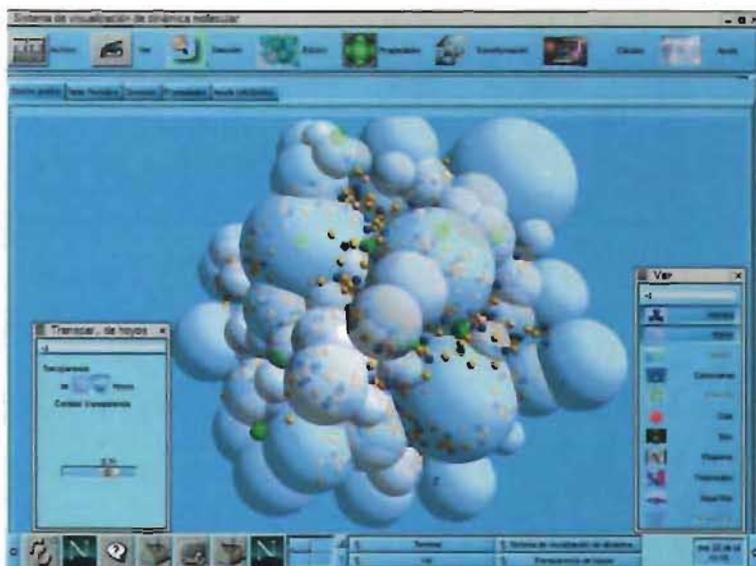
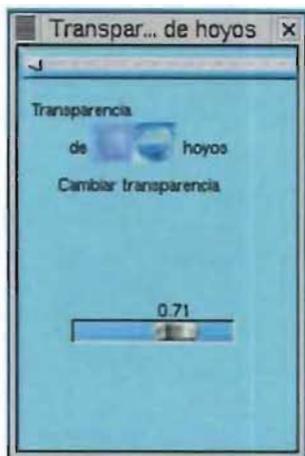


Figura 10.3.5.2.5.2

La Figura 10.3.5.2.5.3 muestra una visualización de hoyos con transparencia igual a 0.38.

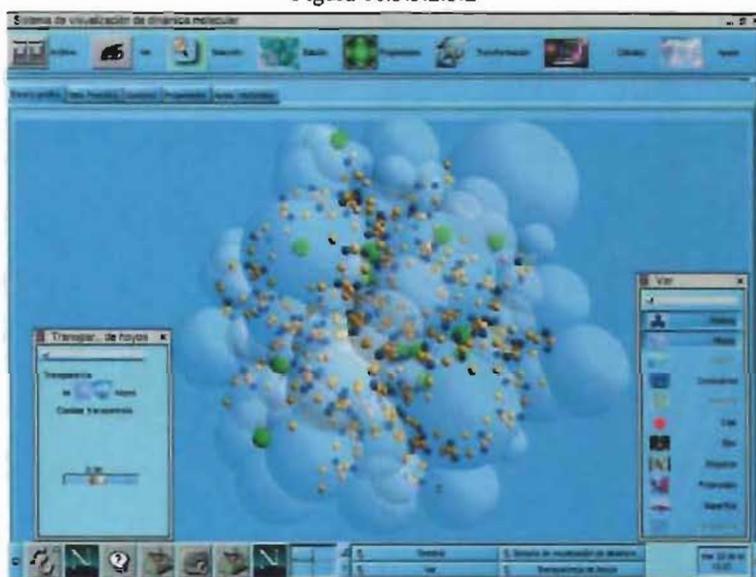


Figura 10.3.5.2.5.3

La Figura 10.3.5.2.5.4 muestra una visualización de hoyos con transparencia igual a 0.09.

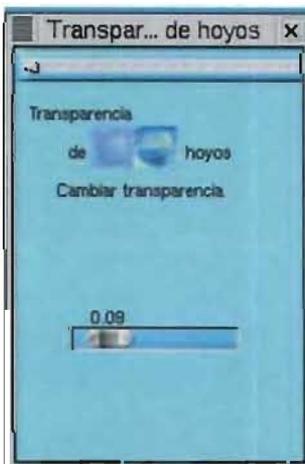


Figura 10.3.5.2.5.4

10.3.5.2.6 Transparencia de átomos

La transparencia de los átomos puede cambiarse utilizando la barra de desplazamiento en la ventana de ésta propiedad.

La Figura 10.3.5.2.6.1 muestra una visualización de átomos con transparencia igual a 1.00.



Figura 10.3.5.2.6.1

La Figura 10.3.5.2.6.2 muestra una visualización de átomos con transparencia igual a 0.65.



Figura 10.3.5.2.6.2

La Figura 10.3.5.2.6.3 muestra una visualización de átomos con transparencia igual a 0.34.

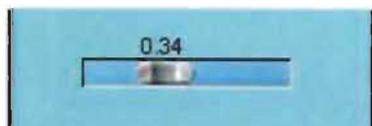


Figura 10.3.5.2.6.3

La Figura 10.3.5.2.6.4 muestra una visualización de átomos con transparencia igual a 0.06.



Figura 10.3.5.2.6.4

La Figura 10.3.5.2.6.5 muestra una visualización de átomos con transparencia igual a 0.18 y se pueden ver las activadas las conexiones.

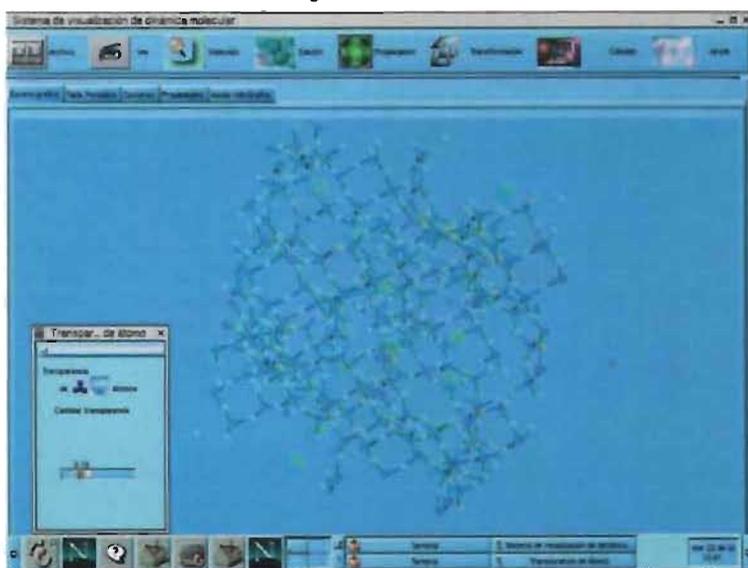
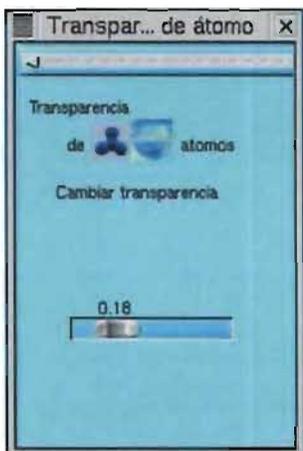


Figura 10.3.5.2.6.5

10.3.5.2.7 Transparencia de conexiones

La transparencia de las conexiones puede cambiarse utilizando la barra de desplazamiento en la ventana de ésta propiedad.

La Figura 10.3.5.2.7.1 muestra un acercamiento de la configuración de los átomos y sus conexiones.

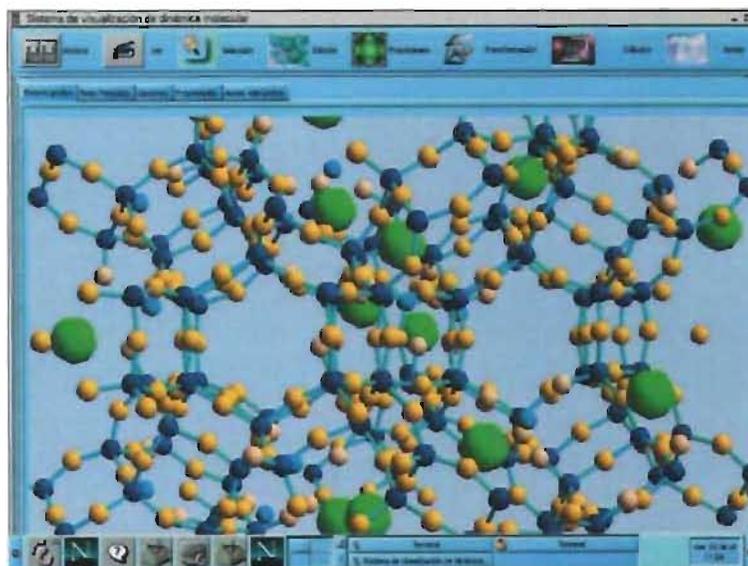


Figura 10.3.5.2.7.1

La Figura 10.3.5.2.7.2 muestra una visualización de átomos y sus conexiones, la transparencia de las conexiones tiene un valor igual a 1.00.

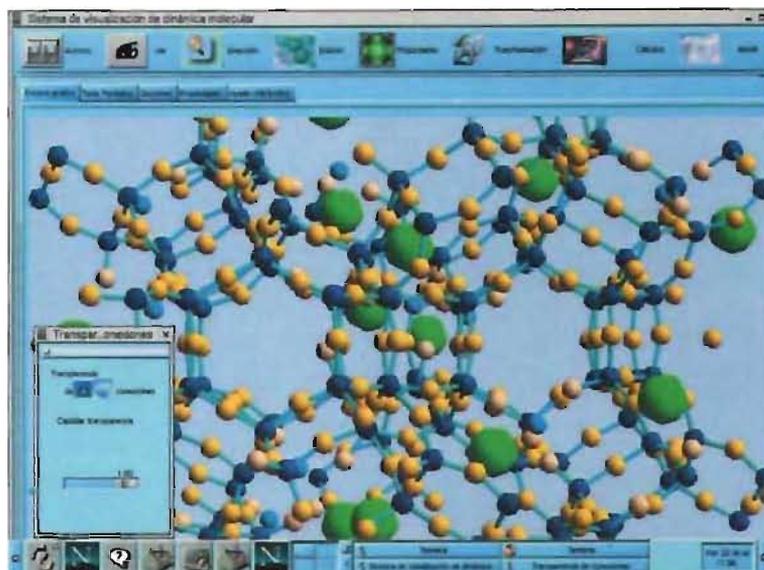


Figura 10.3.5.2.7.2

La Figura 10.3.5.2.7.3 muestra una visualización de átomos y sus conexiones, la transparencia de las conexiones tiene un valor igual a 0.75.

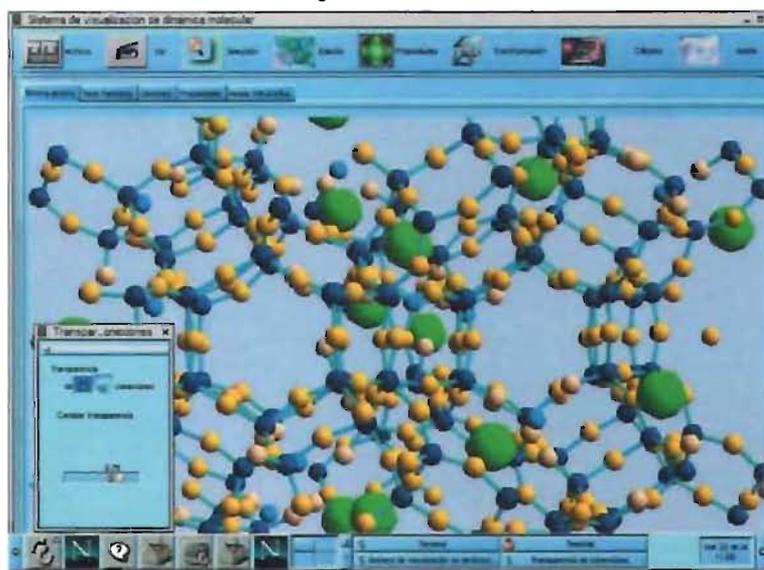


Figura 10.3.5.2.7.3

La Figura 10.3.5.2.7.4 muestra una visualización de átomos y sus conexiones, la transparencia de las conexiones tiene un valor igual a 0.46.

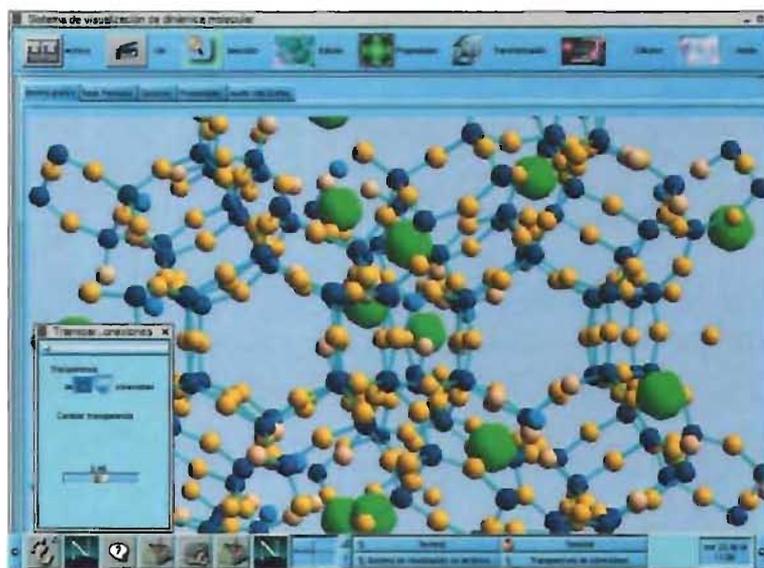


Figura 10.3.5.2.7.4

La Figura 10.3.5.2.7.5 muestra una visualización de átomos y sus conexiones, la transparencia de las conexiones tiene un valor igual a 0.46

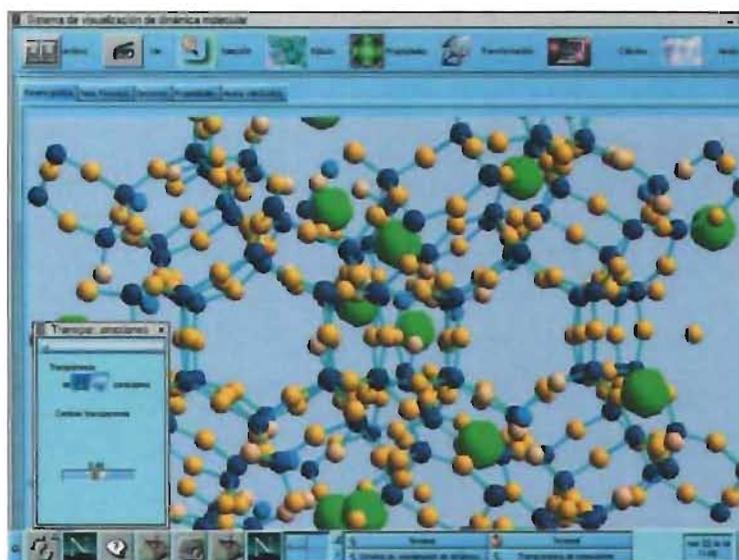


Figura 10.3.5.2.7.5

La Figura 10.3.5.2.7.6 muestra una visualización de átomos y sus conexiones, la transparencia de las conexiones tiene un valor igual a 0.18



Figura 10.3.5.2.7.6

La Figura 10.3.5.2.7.7 muestra una visualización de átomos, sus conexiones y sus hoyos



Figura 10.3.5.2.7.7

10.3.6 Visualización de configuraciones complejas.

El sistema de visualización de dinámica molecular diseñado y desarrollado por nosotros muestra una clara ventaja de la tecnología combinada del análisis y diseño orientado a objetos, la programación orientada a objetos y las herramientas de software libre como Gtk, Vdk, Vdkbuilder, y Mesa, pero sobre todo el Sistema Operativo Linux sobre el que se ejecuta el sistema de visualización, ya que éste permite el uso adecuado de los recursos del hardware,

aprovechándolos al máximo y dándole a nuestro sistema una velocidad de visualización y manipulación de configuraciones incluso de miles de átomos. Es importante aclarar que esto es sólo una muestra de lo que se puede lograr con una PC de tan solo 256 MB de memoria y un procesador Pentium III a 866 MHz. Las siguientes imágenes muestran la visualización de un archivo de partículas con 36684 átomos excepto la Figura 10.3.6.1 que muestra sólo una tercera parte de éstas.

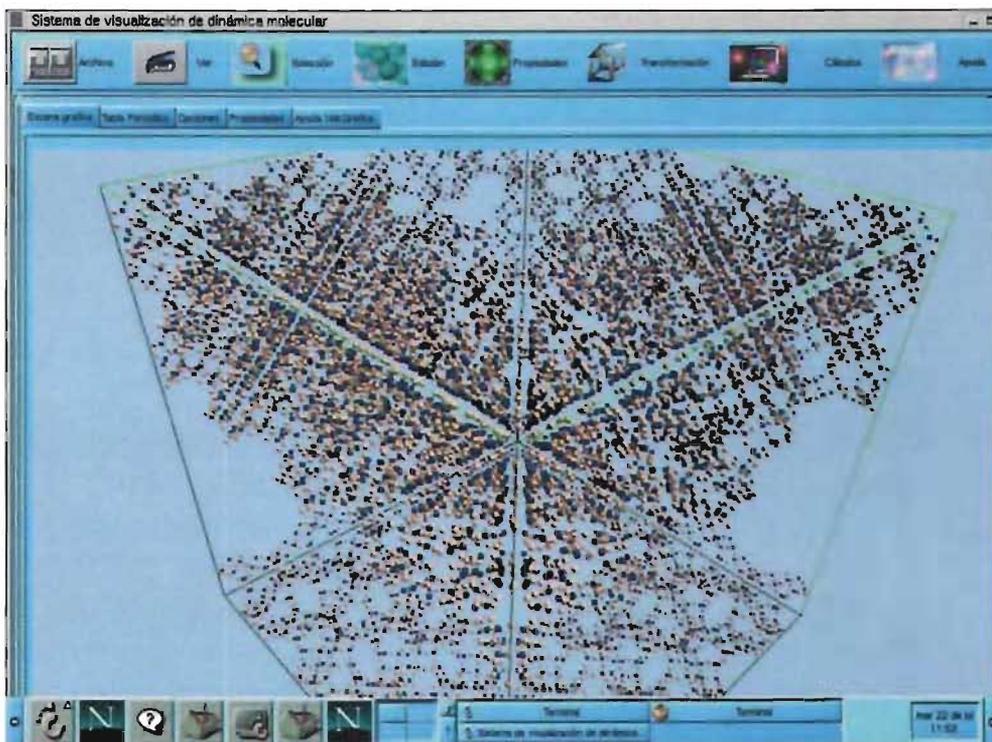


Figura 10.3.6.1

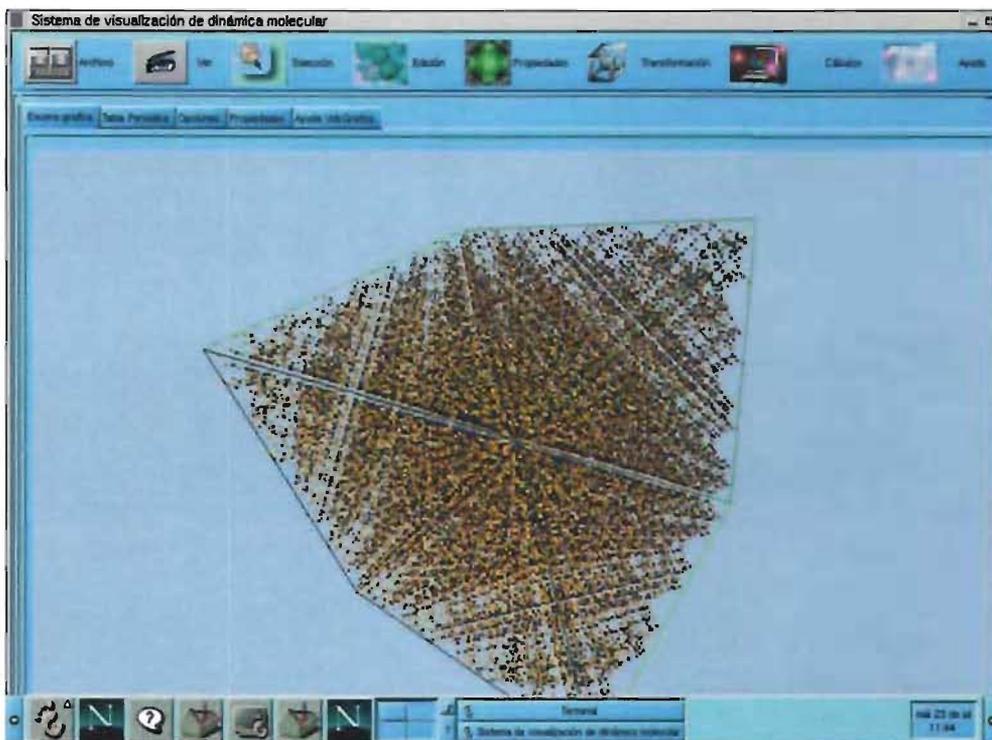


Figura 10.3.6.2

La Figura 10.3.6.2 Muestra la configuración de una Faujasita con 36684 con radio de átomos inicial del archivo atomo.h.

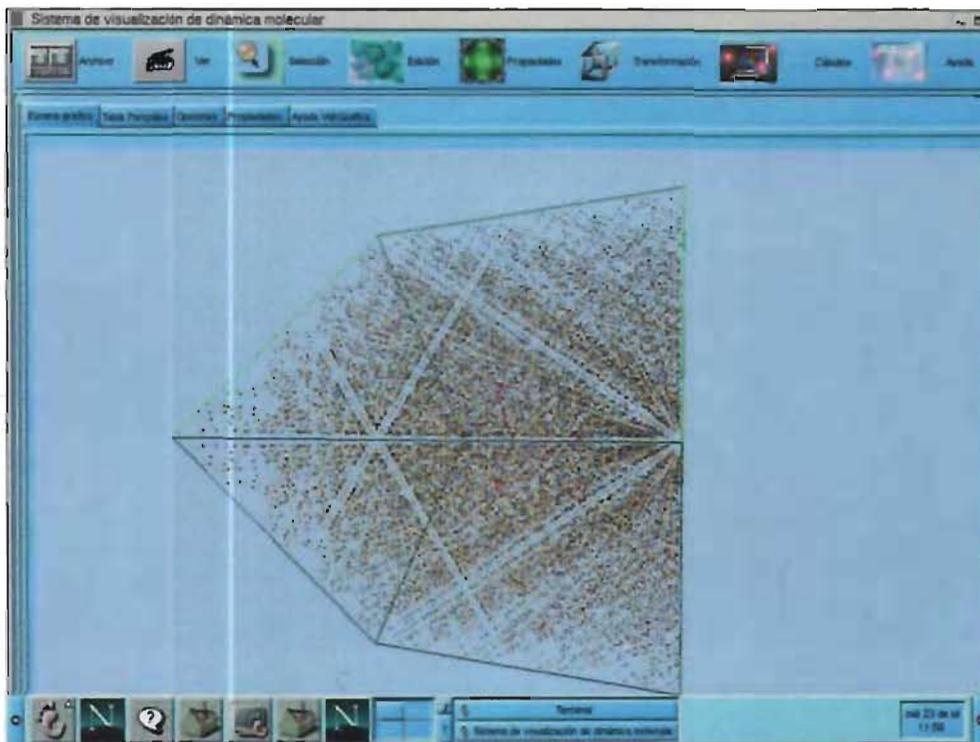


Figura 10.3.6.3

Muestra la configuración de una faujasita con 36684 átomos con radio minimizado.

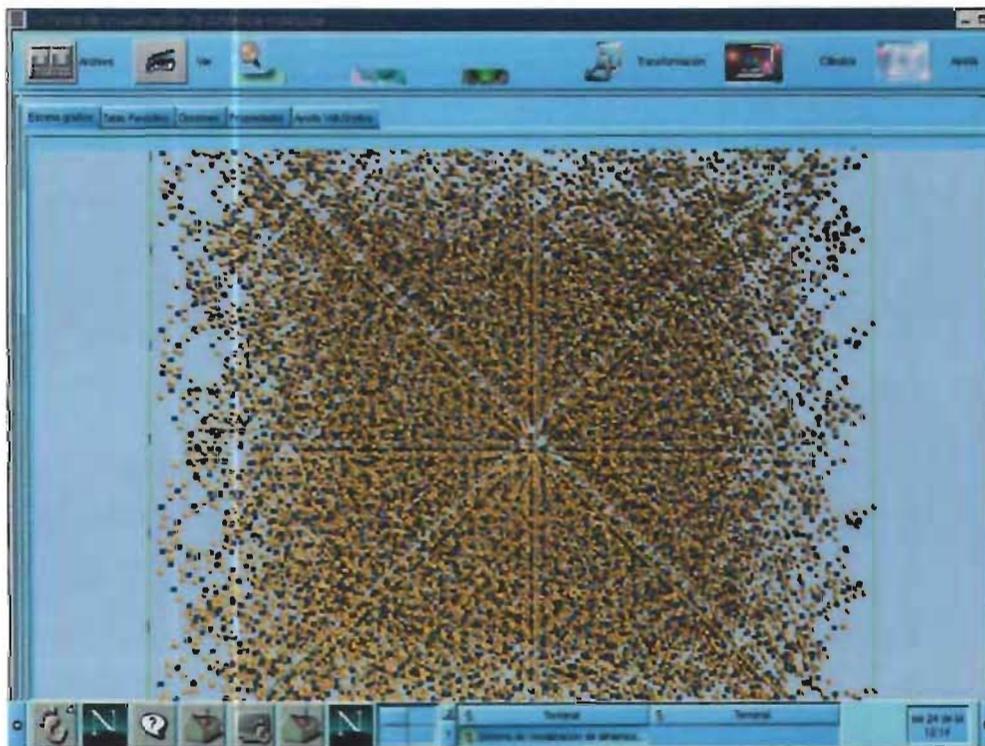


Figura 10.3.6.4

Muestra la configuración de una faujasita con 36684 átomos con un acercamiento en primer plano. Se pueden distinguir los dos tipos de átomos: silicios y oxígenos.

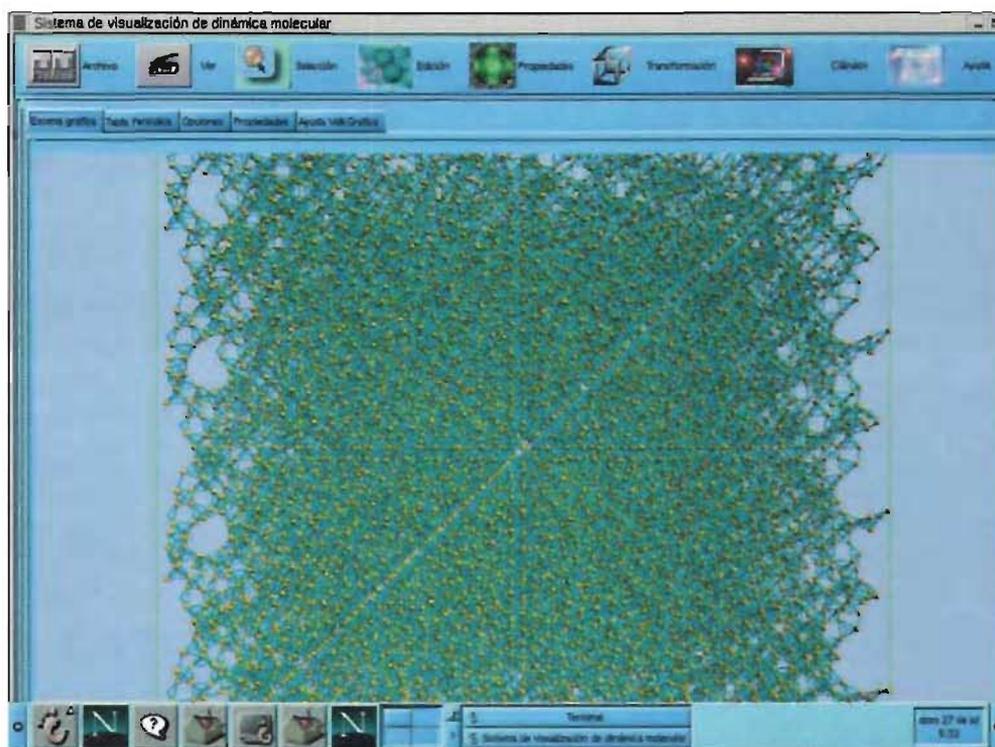


Figura 10.3.6.5

Muestra la configuración de una faujasita con 36684 átomos, con solo los átomos de silicio y sus conexiones.

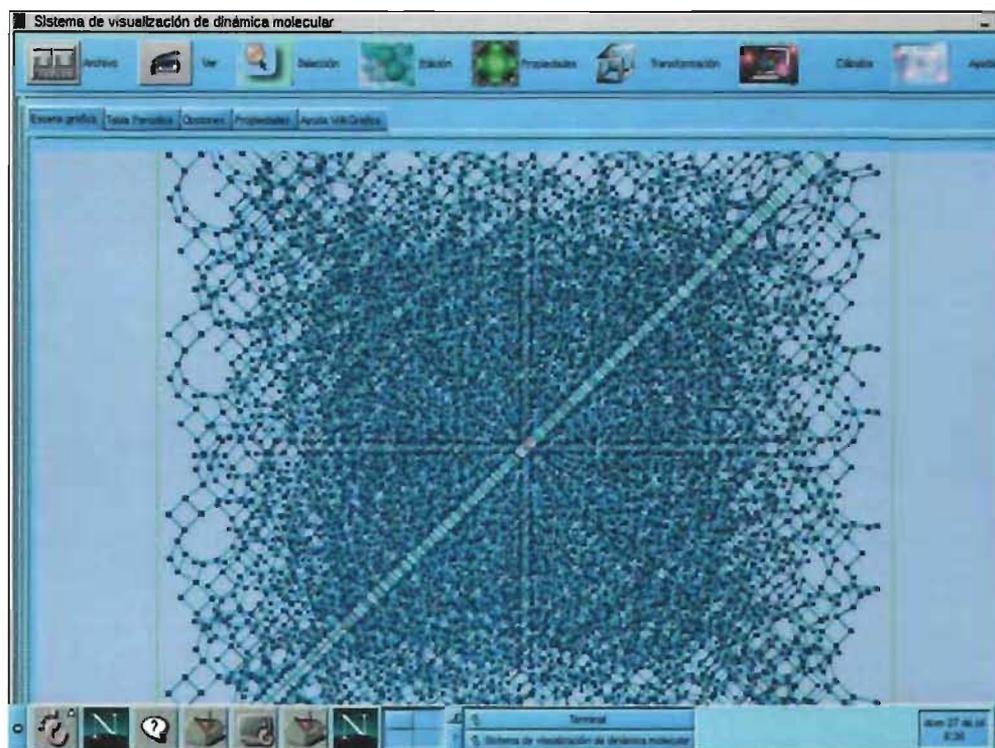


Figura 10.3.6.6

Muestra la configuración de una faujasita con 36684 átomos, con solo los átomos de oxígeno y sus conexiones.

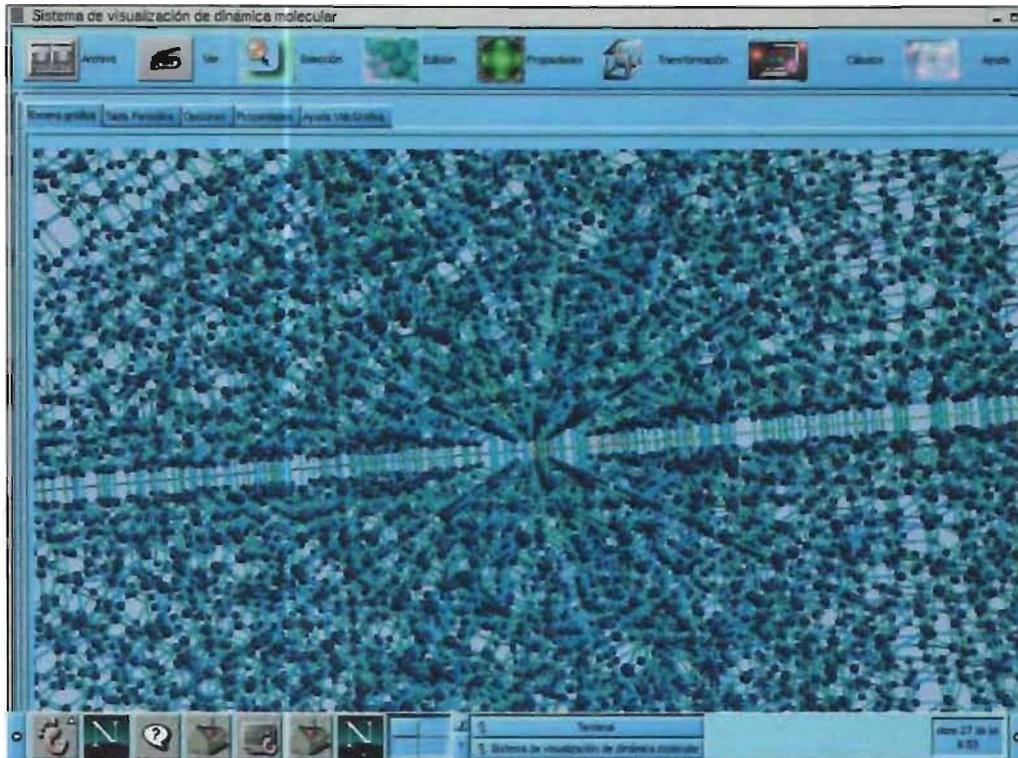


Figura 10.3.6.7

Muestra la configuración de una faujasita vista desde dentro. Se pueden ver los enlaces que forman las conexiones entre los átomos.

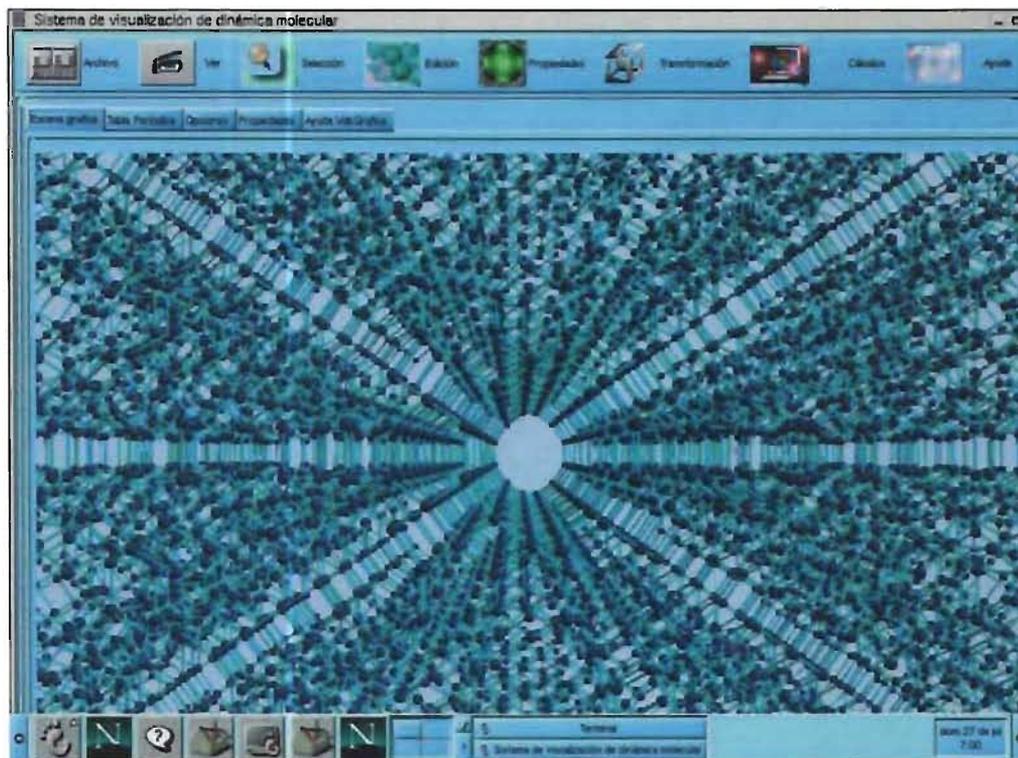


Figura 10.3.6.8

Muestra la configuración de una faujasita vista desde dentro pero rotada para mostrar la ordenación de los átomos y sus conexiones. Se puede ver como se forma un túnel en el centro de esta molécula.

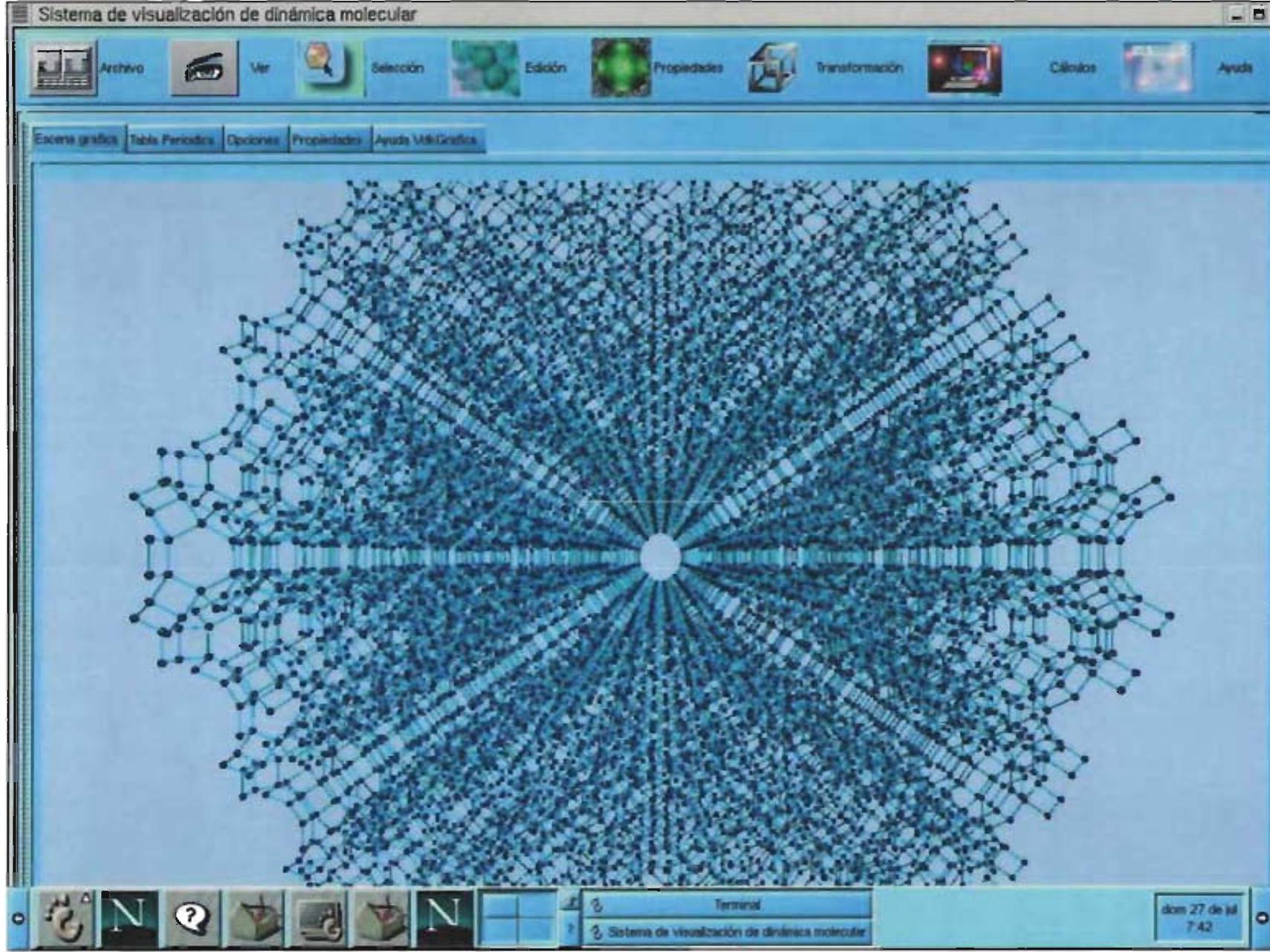


Figura 10.3.6.9 Átomos de oxígeno en una fajajista con sus conexiones.

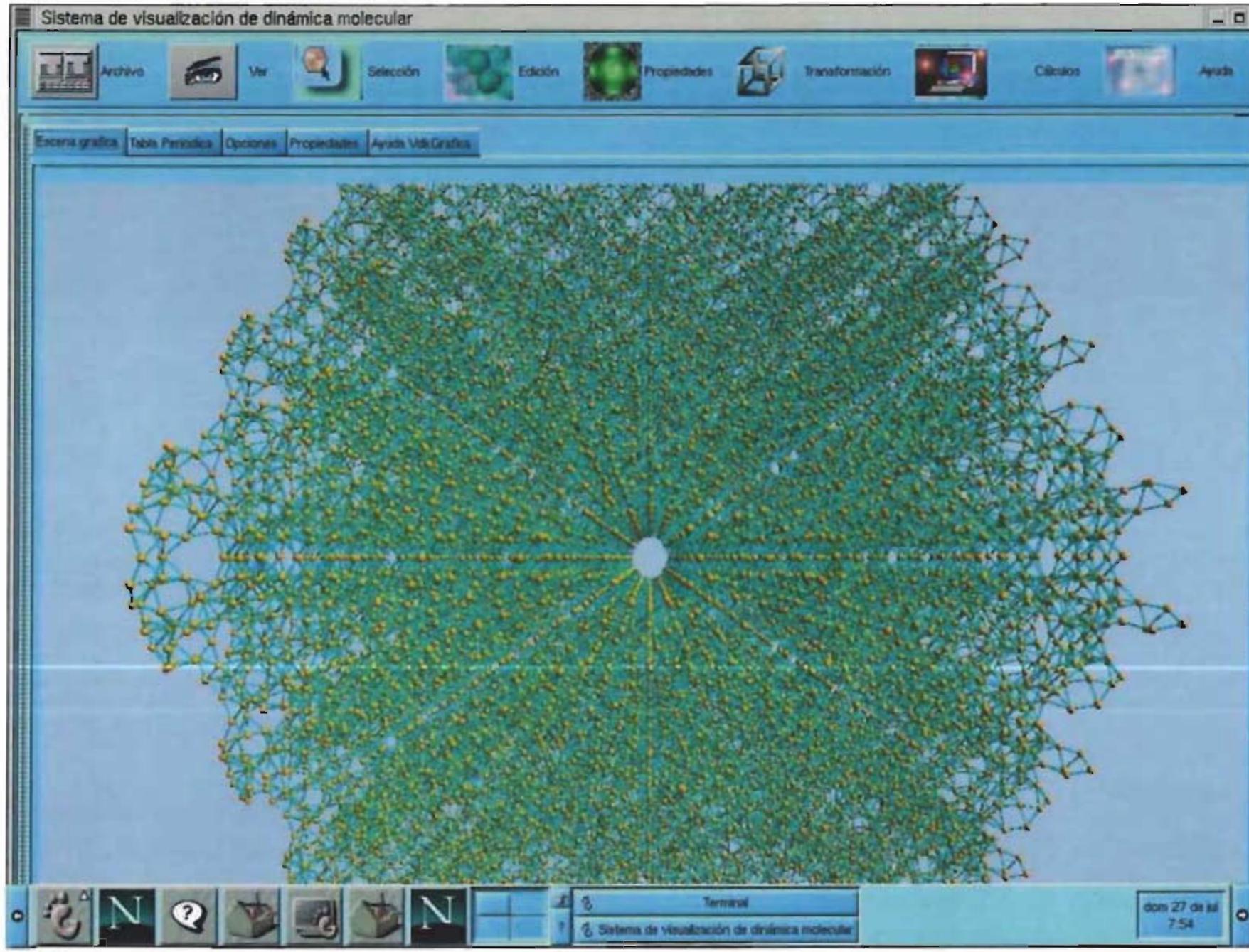


Figura 10.3.6.10 Átomos de silicio en una faujásita con sus conexiones.

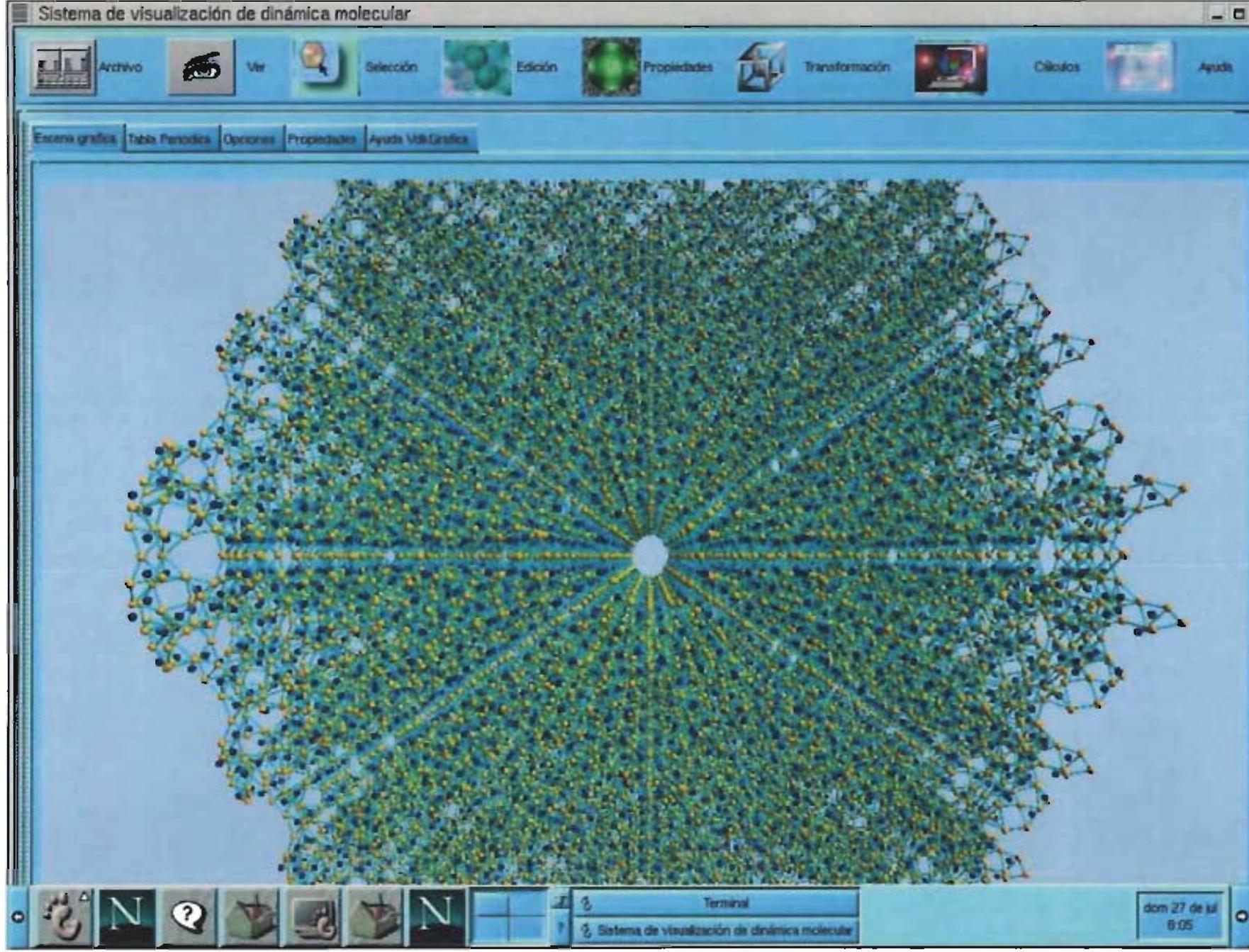


Figura 10.3.6.11 Átomos de oxígeno y silicio en una fajaasita, solo los átomos de silicio tienen conexiones.

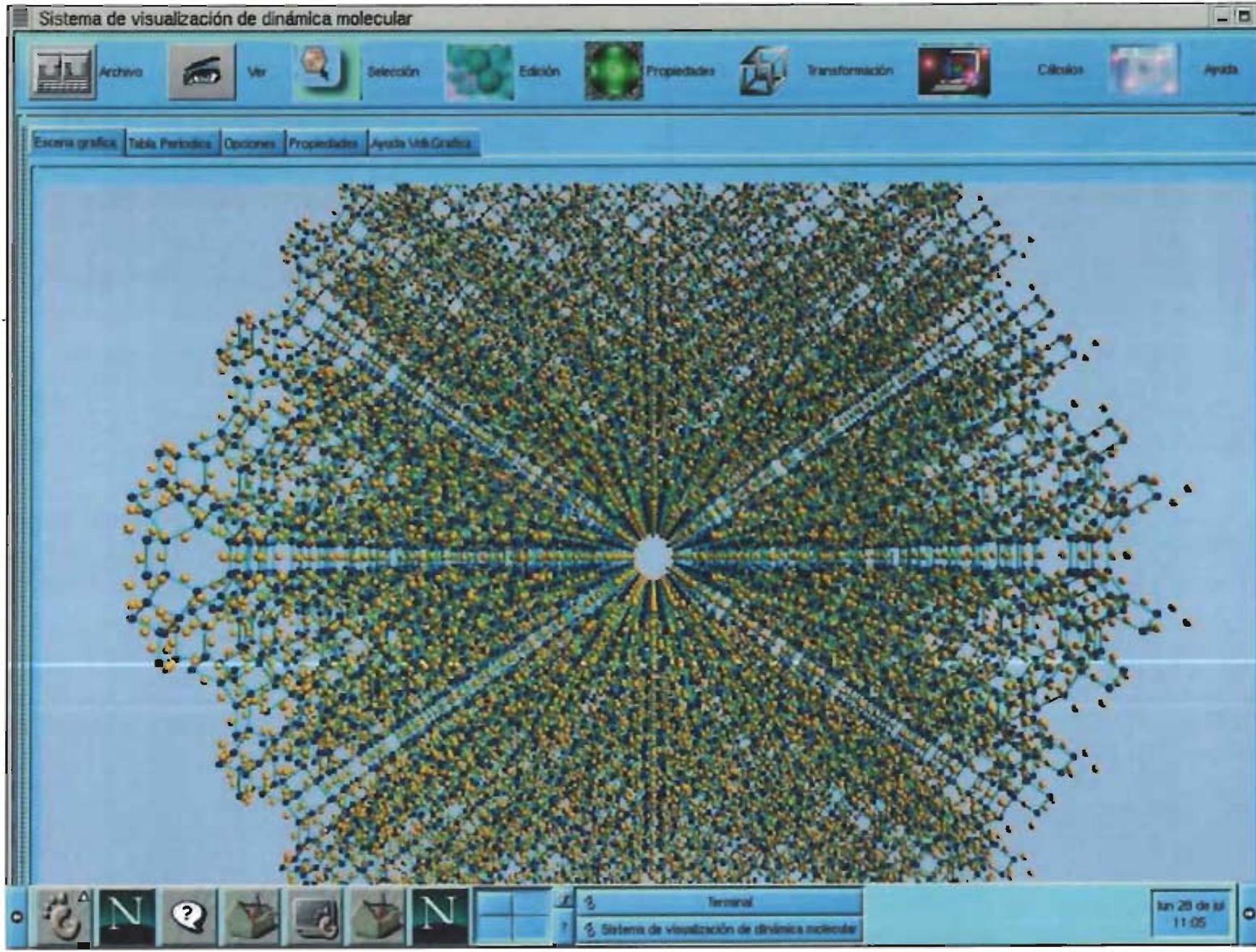


Figura 10.3.6.12 Átomos de oxígeno y silicio en una fajaquita, solo los átomos de oxígeno tienen conexiones.

10.4 Perspectivas de ampliación funcional

Las grandes ventajas que proporciona la nueva tecnología disponible en el campo de las computadoras personales y el desarrollo del sistema operativo Linux y su gama de herramientas de software libre, permitió trasladar la visualización que antes se hacía en costosas estaciones de trabajo con programas o sistemas costosos, a un ambiente más robusto, de mayor desempeño, con amplias perspectivas de desarrollo y compatibilidad entre diferentes plataformas pero a un costo sumamente menor, ya que no se pagan licencias ni del sistema operativo ni de las herramientas con que se desarrolla. Además se utilizaron técnicas de análisis y diseño, y herramientas de desarrollo Orientadas a Objetos para la construcción del sistema, permitiendo con esto disminuir el tiempo de visualización y facilitando una interacción entre el usuario y el sistema, por medio de una interfaz gráfica completamente diseñada para un mejor desempeño.

Debido a la complejidad del sistema, algunos módulos no se desarrollaron en ésta versión del mismo, las siguientes imágenes muestran los prototipos para los módulos “Monitor de procesos” (Figura 10.4.1), que despliega datos sobresalientes de los procesos locales o remotos; “Iniciación de dinámica molecular” (Figura 10.4.2), muestra una interfaz de captura para los archivos de inicialización de la dinámica molecular; “Editor molecular” (Figura 10.4.3), muestra la tabla periódica de los elementos que será utilizada para permitir la creación de átomos sobre la escena gráfica. En las Figuras 10.4.4 a 10.4.9 se muestra una herramienta que permite implementar la selección de objetos 3d sobre la escena gráfica y que por insuficiencia de tiempo no se incluyó en ésta versión del sistema.

Todos estos módulos constituyen un constante desarrollo del sistema y dada la naturaleza del mismo, requieren de recursos humanos, materiales y sobre todo de tiempo, el cual excede en demasía el asignado al desarrollo del presente trabajo. Es por eso que dichos módulos no fueron agregados en ésta versión, aunque son parte del análisis y diseño.

Las siguientes imágenes son una muestra de lo que se puede continuar construyendo siguiendo el análisis y diseño orientado a objetos que se ha elaborado en este trabajo.

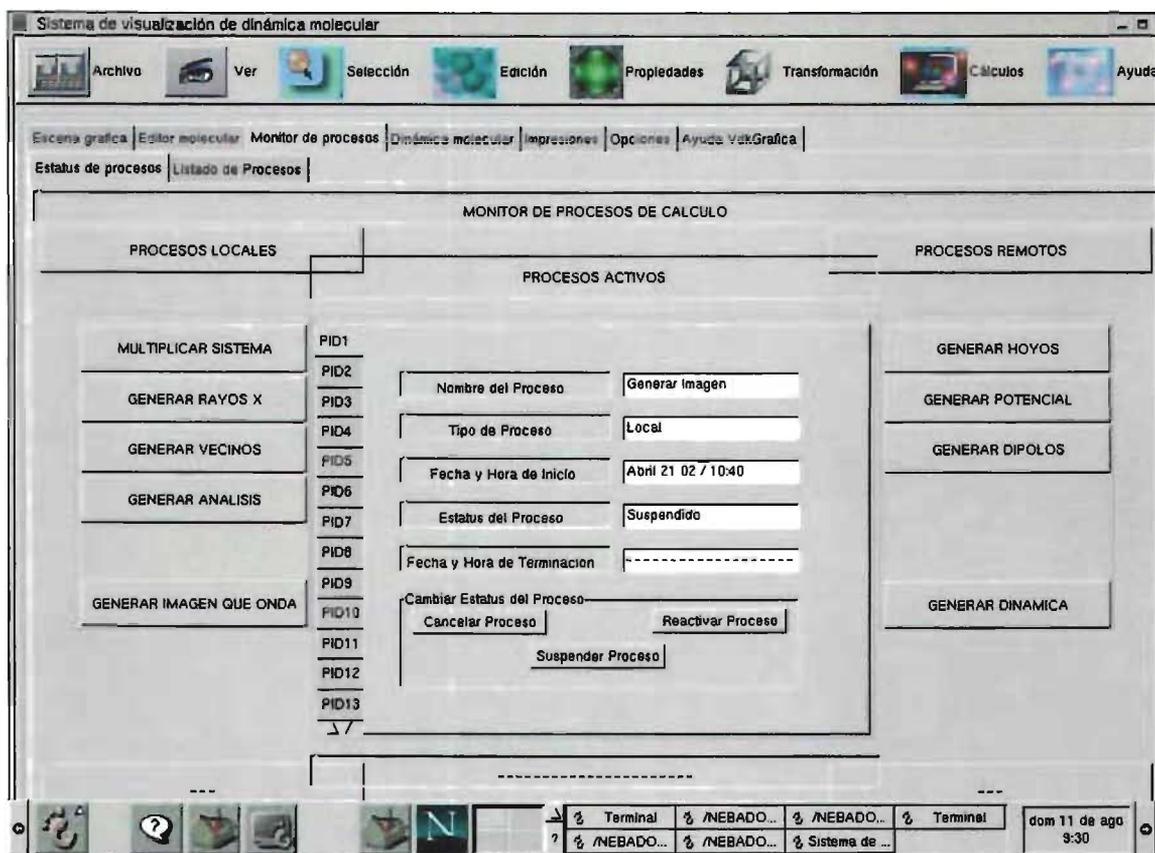


Figura 10.4.1

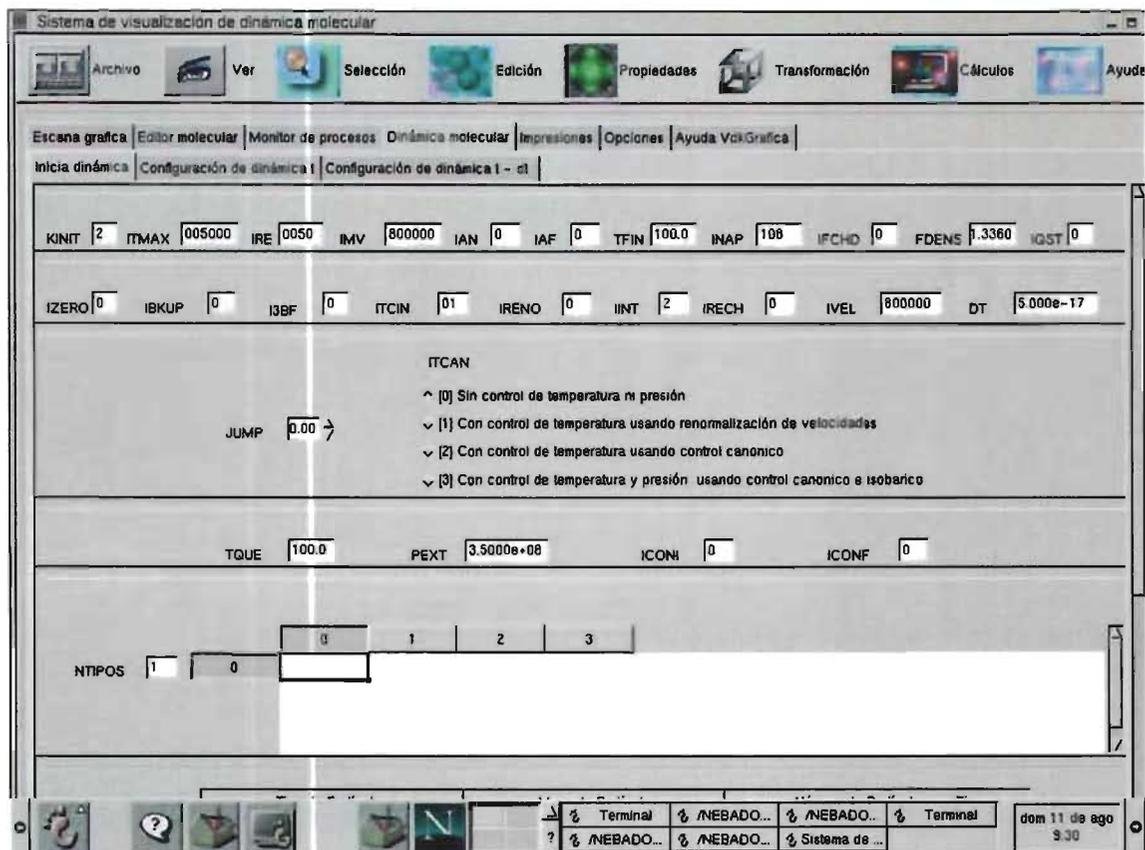


Figura 10.4.2

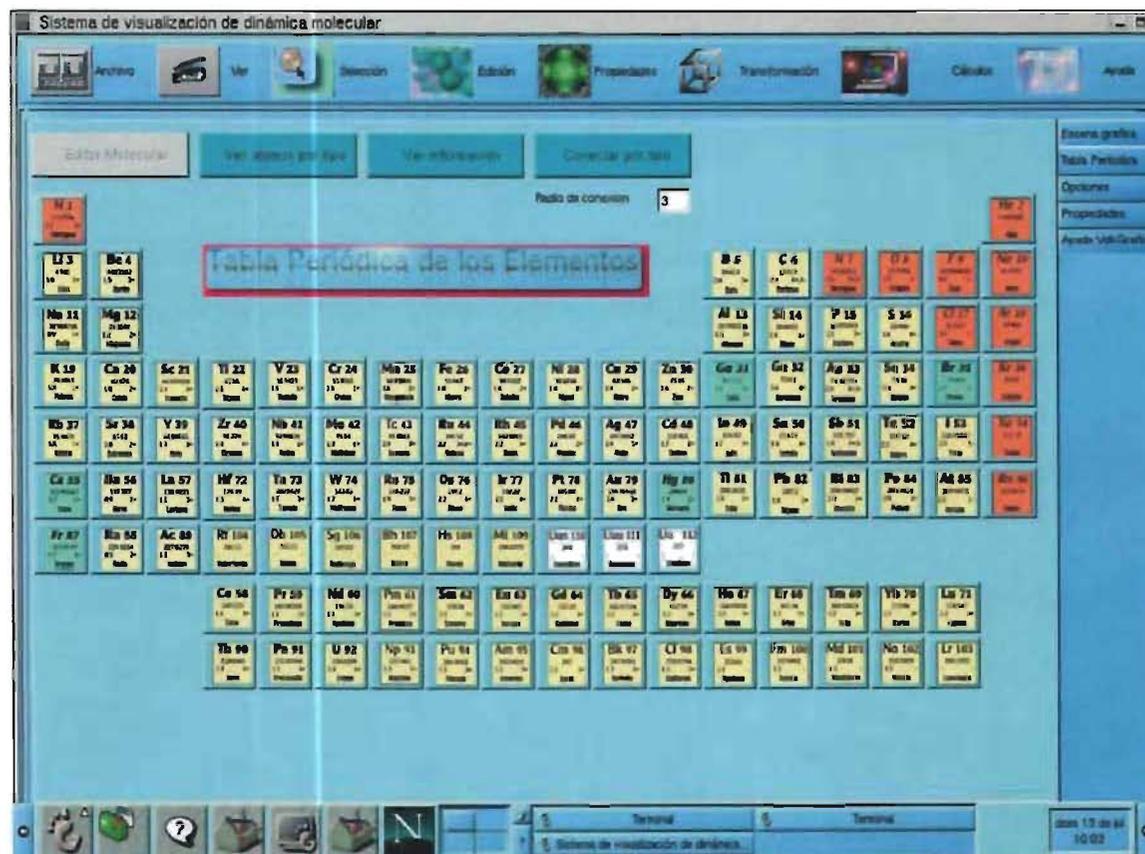


Figura 10.4.3

La Figura 10.4.4 muestra una visualización de átomos rotados sin calidad, con un átomo seleccionado en rojo por el cursor 3D que aparece en color verde.

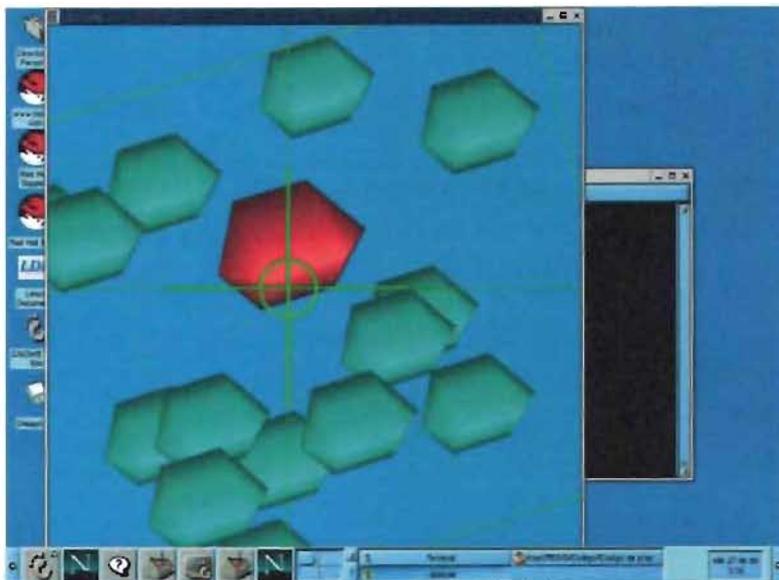


Figura 10.4.4

La Figura 10.4.5 muestra una visualización de átomos con un acercamiento sobre un átomo seleccionado en rojo por el cursor 3D que aparece en color verde.

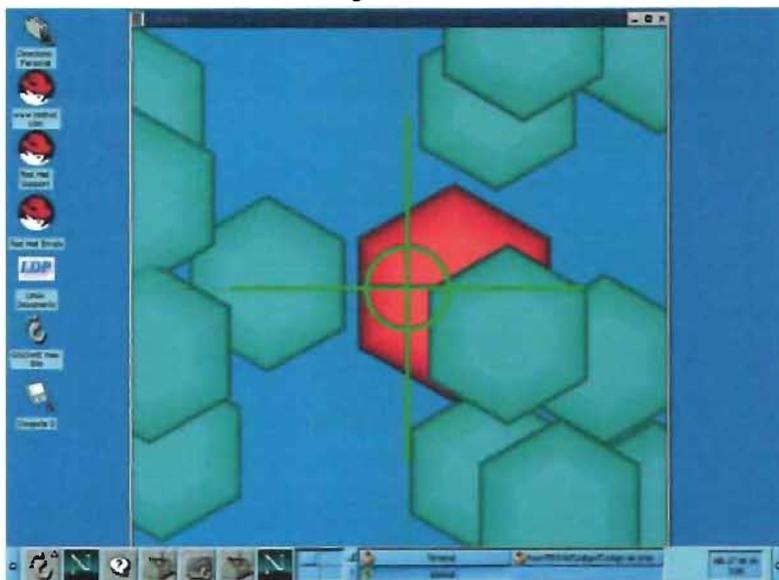


Figura 10.4.5

La Figura 10.4.6 muestra una visualización de átomos con calidad, con un átomo seleccionado en rojo por el cursor 3D.

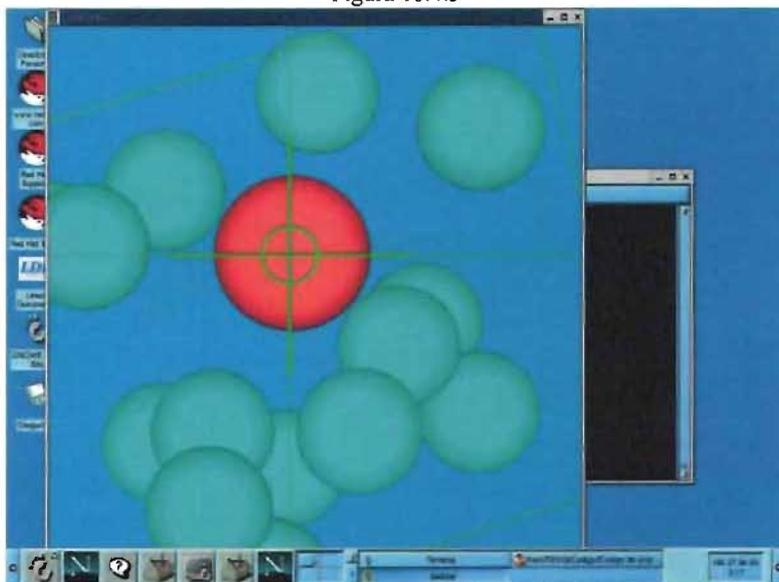


Figura 10.4.6

La Figura 10.4.7 muestra una visualización de átomos en estilo sólido y con un átomo seleccionado en rojo por el cursor 3D que aparece en color verde.

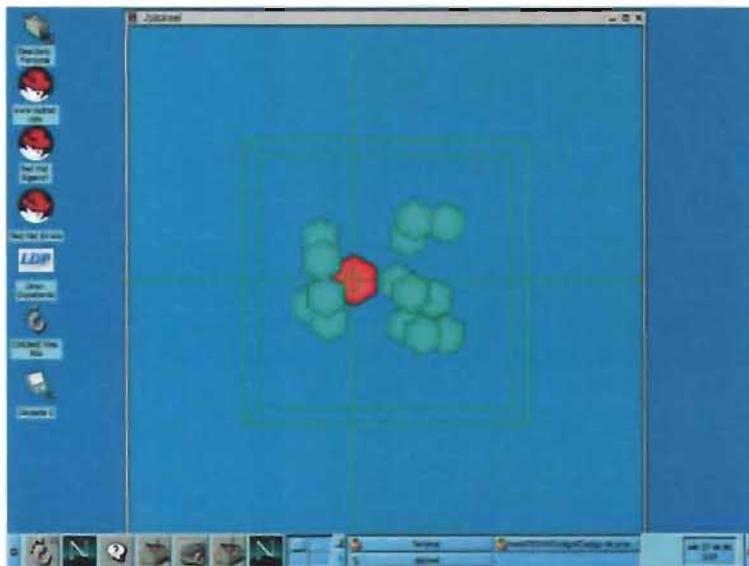


Figura 10.4.7

La Figura 10.4.8 muestra una visualización de átomos en estilo malla, con un átomo seleccionado en rojo por el cursor 3D que aparece en color verde.

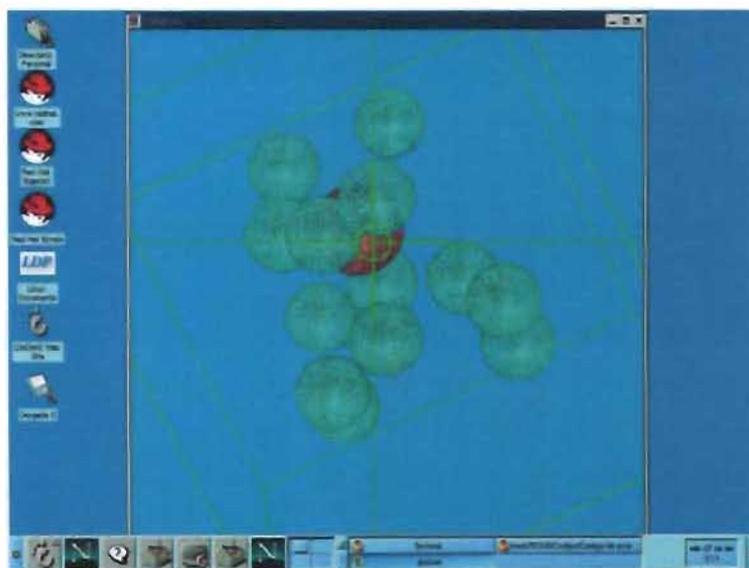


Figura 10.4.8

La Figura 10.4.9 muestra una visualización de átomos con calidad en estilo malla, en un acercamiento sobre un átomo seleccionado en rojo por el cursor 3D.

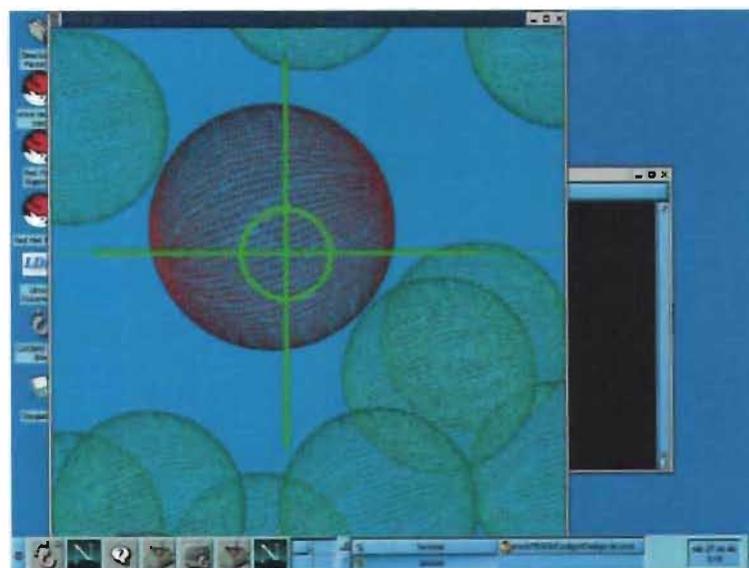


Figura 10.4.9

CONCLUSIONES

11. CONCLUSIONES

Con la realización del presente trabajo, se concluye que el desarrollo de sistemas con software libre es una opción viable para construir aplicaciones propias, incluso cuando las necesidades que se desean resolver implican cuestiones de alto desempeño de cómputo como lo es la simulación científica. Es muy importante que los programadores puedan desarrollar o modificar sus aplicaciones para que éstas realmente sirvan para los fines y necesidades específicas, cosa que no se puede lograr del todo con los sistemas comerciales, ya que muchos no permiten el acceso al código fuente o a los algoritmos utilizados, y si lo hacen, el costo de las licencias se vuelve totalmente excesivo. El *software* libre cambió este paradigma de cómputo, permitiendo reutilizar código y mejorar la calidad de las aplicaciones, pues el código y las herramientas de este tipo dan una mayor flexibilidad al desarrollador de sistemas informáticos para ajustar el software a sus propias necesidades. Esto permite que los desarrolladores adapten las aplicaciones más fácilmente a los recursos que tienen disponibles, de tal forma que es posible optimizar los equipos de cómputo de una manera más eficiente, sobre todo cuando no se cuenta con muchos recursos y la prioridad es resolver un problema específico.

El motivo que llevó a la elección de herramientas de software libre, para el desarrollo del sistema de visualización de dinámica molecular, fue el de demostrar que con el uso de estas herramientas de desarrollo era posible mejorar el desempeño de un sistema complejo y llevarlo al ámbito del cómputo personal, es decir, trasladar la visualización de dinámica molecular que originalmente se realizaba en costosas y potentes estaciones de trabajo a un entorno más económico pero a la vez más accesible y con el mismo o mejor desempeño.

En este trabajo concluimos que la elección de Linux como sistema operativo y de su entorno para el desarrollo de aplicaciones fue correcta, y que las ventajas de usar bibliotecas gráficas (Mesa, GTK), herramientas de desarrollo (VDK, VDKBuilder, C++), compiladores (*MakeFile*, *gcc*), herramientas de manipulación de imágenes para el diseño de la interfaz (Gimp), son enormes en comparación con otras aplicaciones comerciales para diseñar y desarrollar sistemas de visualización, (el simple ahorro en el gasto de las respectivas licencias basta como ejemplo).

Otra de las ventajas es la adaptabilidad de las aplicaciones que se desarrollan con software libre, ya que los sistemas pueden ser implantados de manera más fácil en diferentes plataformas, con muy pocos cambios en el código fuente o en algunos casos sin modificaciones. Linux es un sistema operativo realmente poderoso y muy completo, y cada vez tiene mejores programas y herramientas que lo hacen una verdadera opción para el desarrollo de aplicaciones de alto rendimiento. En el campo de la visualización científica no hay que desaprovechar todas las ventajas que ofrece dicho sistema operativo.

En la realización de este trabajo se hizo énfasis en el paradigma de programación orientada a objetos, especialmente en el seguimiento de una metodología de análisis, diseño (y posteriormente desarrollo) orientada a objetos. Al principio la elección de una metodología nos llevó a hacer una comparación entre las distintas metodologías más usadas para nuestro propósito, “Desarrollar un sistema de visualización de dinámica molecular usando un enfoque orientado a objetos”, es decir construir un sistema que utilizara las ventajas de dicho paradigma, para esto elegimos OMT en lugar de UML, en principio porque en el momento en que iniciábamos nuestro trabajo de tesis UML apenas comenzaba a conocerse y la información disponible que satisfacía nuestras necesidades era la de OMT, definitivamente porque cubría los aspectos necesarios a tomar en cuenta para el desarrollo de una aplicación tal como la que deseábamos construir; por lo mismo fue la elegida para realizar el análisis y el diseño de nuestro sistema, concluimos así que las tres partes que conforman el análisis y las dos que conforman el diseño en esta metodología, resultaron de gran utilidad para describir los componentes del sistema, sus eventos, los procesos y sus operaciones, así como para definir las estrategias de desarrollo más adecuadas.

Construir un sistema de cualquier tipo implica tiempo, recursos económicos y de cómputo, pero el principal factor es el factor humano. Para desarrollar un sistema de las características del descrito en este trabajo en un tiempo adecuado son necesarias más de dos personas, pues el ritmo con el que la tecnología avanza supera el tiempo de desarrollo de

aplicaciones, y se requiere investigación preliminar, un buen equipo de cómputo y comunicación entre el grupo de trabajo; y aun así teniendo todo esto, el desarrollo de una aplicación que pretenda competir con *software* comercial o al menos resolver un problema real requiere de un número grande de analistas, diseñadores y programadores, evaluadores de aplicaciones y documentadores. En nuestro caso para este trabajo de tesis nos dimos cuenta de dicha realidad, sin embargo el tiempo invertido en el análisis, diseño y posterior desarrollo de la aplicación fue sustancial, ya que logramos el propósito inicial: Construir un sistema de visualización de dinámica molecular, con tecnología orientada a objetos en un ambiente de *software* libre (Linux), y aprovechando al máximo los limitados recursos con que contábamos (computadoras personales).

BIBLIOGRAFIA

12. BIBLIOGRAFÍA

12.1 INTRODUCCIÓN

- Mc Cormick, B. H.
National Science Foundation Committee Report.
- Torrens Heeren, Rodrigo Javier.
Las ingenierías y las ciencias computacionales.
Talleres itinerantes en ciencias.
12 Agosto 1997.
- Visualization in scientific computing.
IEEE Computers Graphics, Nov-Dec 1987.

12.2 CAPITULO I

- Álvarez Noguera, Luis Javier.
Tesis de Doctorado.
“Estudios de Dinámica Molecular en dióxido de silicio”
- Coss Bú, Raúl.
Simulación un enfoque práctico
México 1985, Editorial LIMUSA, S.A de C.V.
pp. 12, 13.
- Dmontis, Pierfranco and B. Suffritti, Guieseppe.
Structure and Dynamics of Zeolites Investigated by Molecular Dynamics.
Chemical Reviews, 1997, Vol 97, No. 8
- Fox, David y Waite, Mitchell
Gráficos animados por computadora.
McGraw-Hiull 1983.
pp. 17.
- Freeman C.M. , Levine S. M.
Modeling of structure and reactivity in zeolites.
Zeolite Computer Graphics.
Academic Press 1992. pp. 133-135.
- Galison, Peter.
The disunity of science boundaries, context, and power.
Computer simulations and the Trading Zone.
Stanford, California 1996.
- Pang, Alex and Panendarm, Hans-Georg.
Visualization for Everyone
IEEE Computer Graphics. July-August 1998.
pp. 47, 48.
- Sharp, D.W.A.
The penguin Dictionary of chemistry
pp. 84,432.

- Subramanian, Kalpathi R. and Bashor, David P.
Multilevel Visualization of Spinal Reflex Circuit Simulations.
IEEE Computers Graphics. May-June 1997. pp. 6
- Van Nostrand's Scientific Encyclopedia
Edited by Douglas M. 1976.
pp. 1183.
- Weber, Jacques and Morgantini, Pierre Yves.
Molecular Computer Graphics in Chemistry
- Simulación:
http://www.imaginethatinc.com/sols_sim_def.html
<http://www.webopaedia.com/TERM/s/simulation.html>
<http://izta.matcuer.unam.mx/> Página del Laboratorio de Simulación de Materiales de la UNAM (Cuernavaca).

12.3 CAPITULO II

- Ellis, TMR.
A structured approach to fortran 77 programming.
University of Sheffield.
- Joyanes Aguilar, Luis
C++ a su alcance. un enfoque orientado a objetos.
McGraw-Hill 1994.
- Kartashey, Svetlana P. and Kartashey, Steven I.
Supercomputing Systems Architectures, Design, and Performance.
- Shah, Steve
Manual de administración de LINUX
McGraw Hill 2002.
- Introduction to IRIX.
Silicon Graphics Computer Systems 1997.
- Fortran: <http://www.strath.ac.uk/CC/Courses/fortran/computer.html>
- GTK: <http://www.gtk.org>
- Linux: <http://www.linux.org.mx>
<http://www.cofradia.org>
<http://es.tldp.org>
<http://www.maestrosdelweb.com>
- MESA: <http://www.mesa3D.org>
- Sistemas Operativos y UNIX
<http://www.sun.com/documentation>
<http://www.sgi.com>
<http://techpubs.sgi.com>
- Supercómputo: <http://www.psc.edu/general/mission.html>
- VDK:
Motta, Mario
VDK User Reference Manual
Version 1.2.4 - October 2000
<http://cvs.sourceforge.net/viewcvs.py/vdkbuilder/vdk/>

12.4 CAPITULO III

- Esteves, Julián y Dorantes, Victor Hugo.
UNAM. Computación Visual 98, Taller:
Construcción de interfaces gráficas con Tcl/tk y Xforms.
- Interfaces de interacción: <http://www.crnti.ed.uy/02cursos/hci99.ppt>
- Interfaces gráficas: <http://cs.cinvestav.mx/CursoVis/prinvisual.html>
<http://www.linux.cu/manual/basico-html/node106.html>
- Tcl/tk: <http://hegel.itc.ku.edu/topics/tcltk>
- Tcl: <http://tcl.sourceforge.net>
- Xforms: <http://world.std.com/~xforms>
<http://savannah-nongnu.org/download/xforms>
- PerlTK: <http://www.perltk.org>
<http://www.lns.cornell.edu/upvhp/ptk/ptkFAQ.html>

12.5 CAPITULO IV

- Joyanes Aguilar, Luis
C++ a su alcance, un enfoque orientado a objetos.
McGraw-Hill 1994.
- Weitzenfeld, Alfredo.
Paradigma Orientado a Objetos.
ITAM México Diciembre 1994.
- Metodologías OO:
 - OMT: http://exa.unne.edu.ar/depar/areas/informatica/anasisistem1/public_html/Temas/Temas_08.pdf
 - Método de Booch y Fusión de Coleman:
<http://pisuerga.inf.ubu.es/icruzado/ttc/Metodologias.pdf>
 - Yourdon: <http://ccc.inaoep.mx/~labvision/doo/proy/T32.pdf>
 - UML: <http://www.uml.org>

12.6 CAPITULO V

- Olivar Suárez, Isabel Laura.
Tesis de Licenciatura
“Una aplicación de la metodología OMT para diseño orientado a objetos”
UNAM México 1998.
- Weitzenfeld, Alfredo.
Paradigma Orientado a Objetos.
ITAM México Diciembre 1994.
- Winbland, Ánn L.
Edwards, Samuel D.
King, David R.
Software orientado a objetos.
Addison-Wesley Iberoamericana S. A. 1993.

12.7 CAPITULO VI

- Dunn, Fletcher and Parberry, Ian.
3D Math primer for graphics and game development.
Wordware Publishing, Inc. 2002.
- Harlow, Eric.
Desarrollo de aplicaciones Linux con GTK+ y GDK.
Prentice Hall. 1999.
- Kurt Wall, Et Al.
Programación en Linux Segunda edición.
Pearson Educación, S. A. 2001.
- Valiente Gómez, Santiago Igor.
UNAM. Computación Visual 98, Taller:
Programación con la Biblioteca Gráfica OpenGL.
- OpenGL : <http://www.opengl.org>
- VDKBuilder: <http://vdkbuilder.sourceforge.net/>
- GTKGIArea <http://englanders.cc/~jason/howtos.php?howto=gtkglarea>
<http://www.student.oulu.fi/~jlof/gtkglarea>