



UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGON

*SISTEMA PARA ADMINISTRAR EL
INVENTARIO DE EQUIPO DE
CÓMPUTO (FUNDACIÓN UNAM)*

TRABAJO DE TITULACIÓN

BAJO LA MODALIDAD DE “SEMINARIOS Y CURSOS DE ACTUALIZACIÓN Y CAPACITACIÓN PROFESIONAL”

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

PRESENTA:

Sergio Andrés Hernández González

ASESOR:

M. en C. Jesús Hernández Cabrera





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.

DEDICATORIA

A mis queridos padres Armando y Yolanda, por estar en todo momento conmigo y ser un ejemplo a seguir. Gracias por ser tan buenos.

A mis queridos hermanos Armando y Cintia.

A mis abuelitos Genoveva (q.e.p.d), Ignacio y Josefina por estar al pendiente de mí y por ser tan bondadosos.

A toda mi familia en general por su cariño.

ÍNDICE

INTRODUCCIÓN	6
OBJETIVO GENERAL	7
OBJETIVOS PARTICULARES.....	7
JUSTIFICACION	7
CONTENIDO	8
CAPITULO 1	9
MARCO TEÓRICO	9
1.1 El software	10
1.1.1 Características del Software.....	10
1.1.2 Componentes del Software	11
1.1.3 Aplicaciones del Software	11
1.1.4 Ingeniería de Software	12
1.1.5 Productos del Software	12
1.1.6 Características de los Productos del Software	13
1.1.7 Importancia de las Características del Producto	13
1.1.8 Procesos del Software	14
1.1.9 Paradigmas de Ingeniería de Software	15
1.1.9.1.1 Análisis	16
1.1.9.1.2 Diseño	17
1.1.9.1.3 Implantación	19
1.1.9.1.4 Prueba del Sistema	20
1.1.9.1.5 Mantenimiento	20
1.2 Teoría del Diseño Conceptual de Bases de Datos.....	21
1.2.1 Modelos de Bases de Datos	21
1.2.1.1 Modelo Entidad-Relación (E/R).....	22
1.2.1.2 Modelo Relacional.....	24
1.2.1.2.1 Reglas de Integridad	27
1.2.1.2.2 El proceso de Normalización.....	29
1.2.3 El Sistema Administrador de Base de Datos (DBMS).....	30
1.2.3.1 Arquitectura para un DBMS.....	31
1.2.4 Modelo Funcional.....	33
1.2.6 Metodología para desarrollar el Análisis de Sistemas de Bases de Datos	35
CAPITULO 2	36

ANÁLISIS DEL SISTEMA	36
2.1 Introducción	37
2.2 Definición y planteamiento del problema	37
2.3 Entrevista.....	37
2.4 Narrativa	37
2.5 Análisis de requerimientos.....	38
2.5.1 Requerimientos No Funcionales	38
2.5.2 Requerimientos Funcionales.....	39
2.6 Narrativas de Caso de Uso en Formato Expandido	39
2.6.1 Administrar Artículos.....	39
2.6.2 Dar de Alta Artículo	40
Caso de uso: Dar de alta artículo.....	40
2.6.3 Dar de Baja Artículo	41
2.6.4 Modificar Artículo.....	42
2.7 Análisis Costo/Beneficio	43
CAPÍTULO 3	47
DISEÑO DEL SISTEMA	47
3.1 Introducción	48
3.2 Modelo de Análisis del Sistema	48
3.3 Diagrama de Clases	52
3.4 Diagrama de Clase De Diseño	52
3.5 Diagrama de Secuencia	56
CAPITULO 4	63
ANÁLISIS Y DISEÑO DE LA BASE DE DATOS.....	63
4.1 Introducción	64
4.2 Modelo Funcional.....	64
4.2.1 Arquitectura del Sistema	64
4.3 Modelo Entidad-Relación (E/R)	67
4.4 Modelo de Flujo de Datos	69
4.5 Diccionario de Datos.....	71
4.6 Modelo Relacional.....	73
4.7 Normalización.....	74
4.8 Diseño de la Base de Datos.....	75
4.9 Crear Tablas	76
4.10 Etapas y Tiempo para el desarrollo del Sistema (Diagramas de Gantt y Pert).	78

CAPITULO 5	81
IMPLANTACIÓN DEL SISTEMA.....	81
5.1 Introducción	82
5.2 Breve explicación.....	82
5.3 Descripción del Sistema	84
5.4 Diseño de las Interfaces	84
5.4.2 Interfaz Menú Principal del Sistema.....	86
5.4.3 Interfaz Alta de Artículo	87
5.4.3 Interfaz Editar Artículo	89
5.4.5 Interfaz Baja Artículo.....	91
5.4.6 Interfaz Consultar Artículo	93
5.5 Plan de Implantación	95
5.6 Capacitación.....	96
5.7 Mantenimiento del sistema.....	97
CONCLUSIONES	98
GLOSARIO.....	100
BIBLIOGRAFIA.....	103
ANEXO (MANUAL DE USUARIO Y MANUAL DE ADMINISTRACIÓN).....	104

INTRODUCCIÓN

Actualmente existen diversas empresas, negocios, organizaciones, instituciones educativas, etc. Que se enfrentan ante la necesidad de administrar, automatizar y organizar su información, ante las exigencias y necesidades que día con día tienen que resolverse con eficiencia y rapidez. La tecnología y la ciencia avanzan a pasos agigantados que nos impulsan asumir nuevas responsabilidades y por tal motivo, es importante contar con las herramientas adecuadas y los conocimientos necesarios para responder a estas necesidades. Los sistemas de aplicación son un recurso vital para las necesidades de las organizaciones y empresas, los cuales deben de automatizar el método organizacional de la información.

Las Bases de Datos representan la esencia para lograr la organización, automatización y control de la información, las cuales se han visto sorprendidas por la revolución que se ha dado, más sin embargo, los logros han sido más que satisfactorios.

Específicamente en la mayoría de las instituciones educativas que cuentan con una infraestructura de procesamiento de información, llevan a cabo el proceso de inventario por medio de formatos en papel, dicho proceso en muchas ocasiones es confuso, tedioso y en el peor de los casos "incongruente", por lo tanto se requiere aprovechar la tecnología que existe para automatizar el procesamiento de dicha información.

En la Facultad de Estudios Superiores Aragón existe una organización llamada Fundación UNAM, la cual tiene a su cargo 6 salas de cómputo, cada sala lleva a cabo el proceso de inventario de equipo de cómputo en formato de manera tradicional para después archivarlos en pequeñas y sencillas estructuras de almacenamiento, lo cual lo hace de una manera poco eficiente.

Por lo que la institución quiere actualizar este sistema a una base de datos con la que se mejoraría la forma de llevar a cabo el control de dicho inventario con los que cuenta actualmente y los que se le vayan asignando en un futuro. Entre las ventajas podemos destacar las siguientes:

- Se minimiza el gasto de papelería al llevar el control de manera electrónica.
- El encargado de elaborar el inventario puede realizar consultas desde lugares remotos.
- Se tendrá información más congruente con respecto a lo que se tiene de infraestructura.

OBJETIVO GENERAL

Analizar, diseñar e implementar un sistema que controle la administración del inventario del equipo de cómputo de cada sala de Fundación UNAM, mediante una Base de Datos relacional.

OBJETIVOS PARTICULARES

- Recopilación de información sobre Ingeniería de Software y Bases de Datos.
- Diseñar y desarrollar una Base de Datos de acuerdo a los requerimientos del sistema.
- Implantación del Sistema y de la Base de Datos.

JUSTIFICACION

La propuesta es un sistema clásico, es sistema será capaz de almacenar (Altas) información referente a la infraestructura y hardware existente en la carrera y el que se vaya asignando a la misma; cómo podemos observar básicamente contemplaremos una Base de Datos con diferentes tablas que almacenan la información, las cuales mantendrán una relación, ya que al querer localizar un equipo de cómputo y/o partes del mismo en particular, también se tendrá en cuenta en que área está instalado. El sistema podrá emitir consultas y reportes rápidamente, ya sea por pantalla o papel.

El sistema debe permitir consultar información de las bases de datos para poder hacer modificaciones, ya que pueden existir cambios de orden, es decir, en casos particulares de movimiento de equipo y/o partes del mismo de un espacio físico a otro, se deben de registrar dichas modificaciones; además el sistema permitirá dar de baja información obsoleta, cuando se dé de baja un equipo de computo y/o partes del mismo, es decir, se podrán actualizar las bases de datos.

CONTENIDO

En el presente trabajo se presenta de la siguiente forma:

- Capítulo I. Proporciona los conceptos teóricos fundamentales de Ingeniería de Software y Bases de Datos para el desarrollo del sistema tales como: El Modelo de Cascada, así como aspectos importantes que permitirán esquematizar el diseño de la Base de Datos mediante el Modelo Relacional.
- Capítulo II. Se establece el Análisis del sistema como el primer paso a desarrollar el Modelo de Cascada de Ingeniería de Software empleando el Método CASE.
- Capítulo III. Se detalla el Modelo de Análisis del Sistema, la vista estática del sistema, mostrando sus entidades, Diagrama de Clases, Diagrama de Clases de Diseño y Diagrama de Secuencia.
- Capítulo IV. Se muestra el diseño del sistema, el modelo Entidad-Relación, modelo Flujo de Datos, Diccionario de Datos, Modelo Relacional, Normalización. Se diseña un esquema de base de datos para el Sistema de Inventario de equipo de Cómputo utilizando el Modelo Relacional y la Creación de las tablas del sistema.
- Capítulo V. Se lleva a cabo la implantación del sistema en donde se cumple con todas las funciones descritas en las especificaciones, utilizando la base de datos diseñada en el capítulo IV, además de especificar las herramientas utilizadas para el desarrollo del sistema.

Conclusiones y perspectivas concluye finalmente con el desarrollo del sistema SAIEC (Sistema para Administrar el Inventario de Equipo de Computo).

Bibliografía.

Anexo (Manual de usuario y Manual de Administración).

CAPÍTULO 1

MARCO TEÓRICO

1.1 El software

En los primeros días de la informática, los sistemas basados en computadora se desarrollaban usando técnicas de gestión orientadas al hardware. Los gestores del proyecto se centraban en el hardware, debido a que era el factor principal del presupuesto en el desarrollo del sistema. Para controlar los costos del hardware, los gestores instituyeron controles formales y estándares técnicos. Exigían un análisis y diseño completo antes de que algo se construyera. Medían el proceso para determinar dónde podían hacerse mejoras. Dicho sencillamente, aplicaban los controles, los métodos y las herramientas que reconocemos como ingeniería de hardware. Desgraciadamente el software no era normalmente más que un añadido.

La programación se veía como un arte, existían pocos métodos formales, y pocas personas lo usaban. El programador aprendía normalmente su oficio mediante prueba y error. La jerga y los desafíos de la construcción del software de computadoras crearon una <orden> en la que pocos ejecutivos se preocupan por entrar.

Hoy, la distribución de costos en el desarrollo de sistemas informáticos ha cambiado drásticamente. El software en lugar del hardware es normalmente el elemento principal del costo.

Hoy en día el software tiene un doble papel. Es un producto y, al mismo tiempo, el vehículo para hacer entrega de un producto. Como producto hace entrega de la potencia informática del hardware. Si reside dentro de un teléfono celular u opera dentro de una computadora central, el software es un transformador de información, produciendo, gestionando, adquiriendo, modificando, mostrando o transmitiendo información que puede ser tan simple como un solo bit, o tan complejo como una simulación en multimedia.

Como vehículo utilizado para hacer entrega del producto, el software actúa como la base de control de la computadora (sistemas operativos), la comunicación de información (redes), y la creación y control de otros programas (herramientas del software y entornos).

1.1.1 Características del Software

Cuando se construye hardware, el proceso creativo humano (análisis, diseño, construcción, prueba) se traduce finalmente en una forma física. Si construimos una nueva computadora, nuestro boceto inicial, diagramas formales de diseño y prototipo de prueba, evolucionan hacia un producto físico (pastillas de VLSI, tarjetas de circuitos impresos, fuentes de potencia, etc.).

El Software es un elemento del sistema que es lógico, en lugar de físico. Por lo tanto el software tiene unas características considerablemente distintas a las del hardware.

- El software se desarrolla, no se fabrica en un sentido clásico.
- El software no se estropea.
- La mayoría del software se construye a medida, en vez de ensamblar componentes existentes.

1.1.2 Componentes del Software

A medida que la disciplina del software evoluciona, se crea un grupo de componentes de diseño estándar. Tornillos estándar y circuitos integrados preparados para la venta son solamente los dos mil componentes estándar que utilizan ingenieros mecánicos y eléctricos cuando diseñan sistemas nuevos. Los componentes reutilizables se han creado para que el ingeniero pueda concentrarse en elementos verdaderamente innovadores de un diseño. En el mundo del hardware, la reutilización de componentes es una parte natural del proceso de ingeniería. En el mundo del software, es algo que todavía se tiene que lograr en una escala amplia.

La reutilización.- Es una característica importante para un componente de software de alta calidad. El componente debería diseñarse e implementarse para que pueda volver a ser reutilizado en muchos programas diferentes.

1.1.3 Aplicaciones del Software

El software puede aplicarse en cualquier situación en la que se haya definido previamente un conjunto específico de pasos procedimentales (es decir un algoritmo. Excepciones notables a esta regla son el software de los sistemas expertos y de redes neuronales). El contenido y el determinismo de la información son factores a considerar para determinar la naturaleza de una aplicación del software. El contenido se refiere al significado y a la forma de la información de entrada y de salida.

El determinismo de la información se refiere a la predicción del orden y del tiempo de llegada de los datos. Un programa de ingeniería acepta datos que están en un orden predefinido, ejecuta el algoritmo sin interrupción y produce los datos resultantes en un informe o formato o formato gráfico. Se dice que tales aplicaciones son determinadas. Un sistema operativo multiusuario, por otra parte, acepta entradas que tienen un contenido variado y que se producen en instantes arbitrarios, ejecuta algoritmos que pueden ser interrumpidos por condiciones externas y produce una salida que depende de una función del entorno y del tiempo. Las aplicaciones con estas características se dice que son indeterminadas.

Las siguientes áreas del software indican la amplitud de las aplicaciones potenciales:

Software de sistemas. Es un conjunto de programas que han sido escritos para servir a otros programas. Algunos programas de sistemas (p. ej.: compiladores, editores y utilidades de gestión de archivos) procesan estructuras de información complejas pero determinadas. El área del software de sistemas se caracteriza por una fuerte interacción con el hardware de la computadora; una gran utilización por múltiples usuarios, una operación concurrente que requiere una planificación, una compartición de recursos y una sofisticada gestión de procesos; unas estructuras de datos complejas y múltiples interfaces externas.

Software de tiempo real. El software que mide/analiza/controla sucesos del mundo real conforme ocurren, se denomina de tiempo real. Hay que tener en cuenta que el término "tiempo real" tiene un significado diferente de "interactivo" o "tiempo compartido". Un sistema de tiempo real debe responder dentro de unas ligaduras estrictas de tiempo. El tiempo de respuesta de un

sistema interactivo (o de tiempo compartido) puede ser normalmente sobrepasado sin que se produzca ningún desastre.

Software de gestión. El procesamiento de información comercial constituye la mayor de las áreas de aplicación del software. Los sistemas discretos (por ejemplo: Nóminas, cuentas de haberes/débitos, inventarios, etc.) han evolucionado hacia el software de sistemas de información de gestión (SIG), que accede a una o más bases de datos grandes que contienen información comercial. Las aplicaciones en esta área reestructuran los datos existentes para facilitar las operaciones comerciales o gestionar la toma de decisiones. Además de las tareas convencionales de procedimientos de datos, las aplicaciones del software de gestión también realizan cálculo interactivo (por ejemplo: el procesamiento de transacciones en puntos de ventas).

Software de ingeniería y científico. Está caracterizado por los algoritmos de “manejo de números”.

Software empotrado. Reside en memoria de sólo lectura y se utiliza para controlar productos y sistemas de los mercados industriales y de consumo por ejemplo: el control de las teclas de un horno de microondas.

Software de computadoras personales. El procesamiento de texto, las hojas de cálculo, los gráficos por computadora, multimedia, entretenimientos, gestión de bases de datos, aplicaciones financieras, de negocios y personales, y redes o acceso a bases de datos externas son algunas de los cientos de aplicaciones.

Software de inteligencia artificial. Hace uso de algoritmos no numéricos para resolver problemas complejos para los que no son adecuados el cálculo o el análisis directo.

1.1.4 Ingeniería de Software

Es una disciplina de la ingeniería relativa a las técnicas y métodos, empleados para desarrollar sistemas complejos de software, tales técnicas son realizadas por un equipo en lugar de una sola persona.

1.1.5 Productos del Software

Productos genéricos

- Productos que son producidos por una organización para ser vendidos al mercado.

Productos hechos a medida

- Sistemas que son desarrollados bajo pedido a un desarrollador específico.

La mayor parte del gasto del software es en productos genéricos, pero hay más esfuerzo en el desarrollo de los sistemas hechos a medida.

1.1.6 Características de los Productos del Software

La Ingeniería de Software se enfoca en la producción de sistemas de software de la forma más eficaz posible. Para que un producto de software se diga que está “bien construido” (well –engineered), necesita tener ciertas características:

Mantenible.- Debe ser posible que el software evolucione y que siga cumpliendo con sus especificaciones. Debe estar bien escrito y además documentado.

Confiabilidad.- El software no debe causar daños físicos o económicos en el caso de fallos. Es decir, que el sistema debe desempeñarse tal como el usuario lo solicite y no fallar más que en situaciones permitidas en las especificaciones.

Eficiencia.- El software no debe desperdiciar los recursos del sistema. Debe aprovechar todos los recursos de hardware y software de la forma más eficaz posible.

Utilización adecuada.- El software debe contar con una interfaz de usuario adecuada y su documentación.

Robusto.- El sistema debe comportarse coherentemente en situaciones no previstas en las especificaciones.

Reusable.- Con pocos cambios se pueda usar para desarrollar nuevos sistemas.

Portable.- El sistema pueda ejecutarse en diferentes sistemas de hardware.

Seguro.- Para no entorpecer el funcionamiento de otros sistemas.

1.1.7 Importancia de las Características del Producto

La importancia relativa de las características depende en el tipo de producto y en el ambiente en el que será utilizado. En algunos casos, algunos atributos pueden dominar.

En sistemas de seguridad críticos de tiempo real, los atributos clave pueden ser la confiabilidad y la eficiencia.

Los costos tienden a crecer exponencialmente si son requeridos altos niveles de alguna característica.

1.1.8 Procesos del Software

El software se ha convertido en el elemento clave de la evolución de los sistemas y productos informáticos. En las pasadas cuatro décadas, el software ha pasado de ser una resolución de problemas especializada y una herramienta de análisis de información, a ser una industria por sí misma. Pero la temprana cultura e historia de la programación ha creado un conjunto de problemas que persisten todavía hoy. El software se compone de programas, datos y documentos.

Cada uno de estos elementos componen una configuración que se crea como parte del proceso de la ingeniería del software. El intento de la ingeniería del software es proporcionar un marco de trabajo para construir software con mayor calidad.

Procesos del software. Es el conjunto de actividades técnicas y administrativas necesarias para la fabricación de un producto de software. La madurez de una organización es proporcional a su capacidad para poner en práctica procesos de software que produzcan los resultados previstos.

Estas actividades son:

- Análisis
- Diseño
- Implantación
- Prueba del sistema
- Mantenimiento

Las actividades varían dependiendo de la organización y del tipo de sistema a desarrollarse.

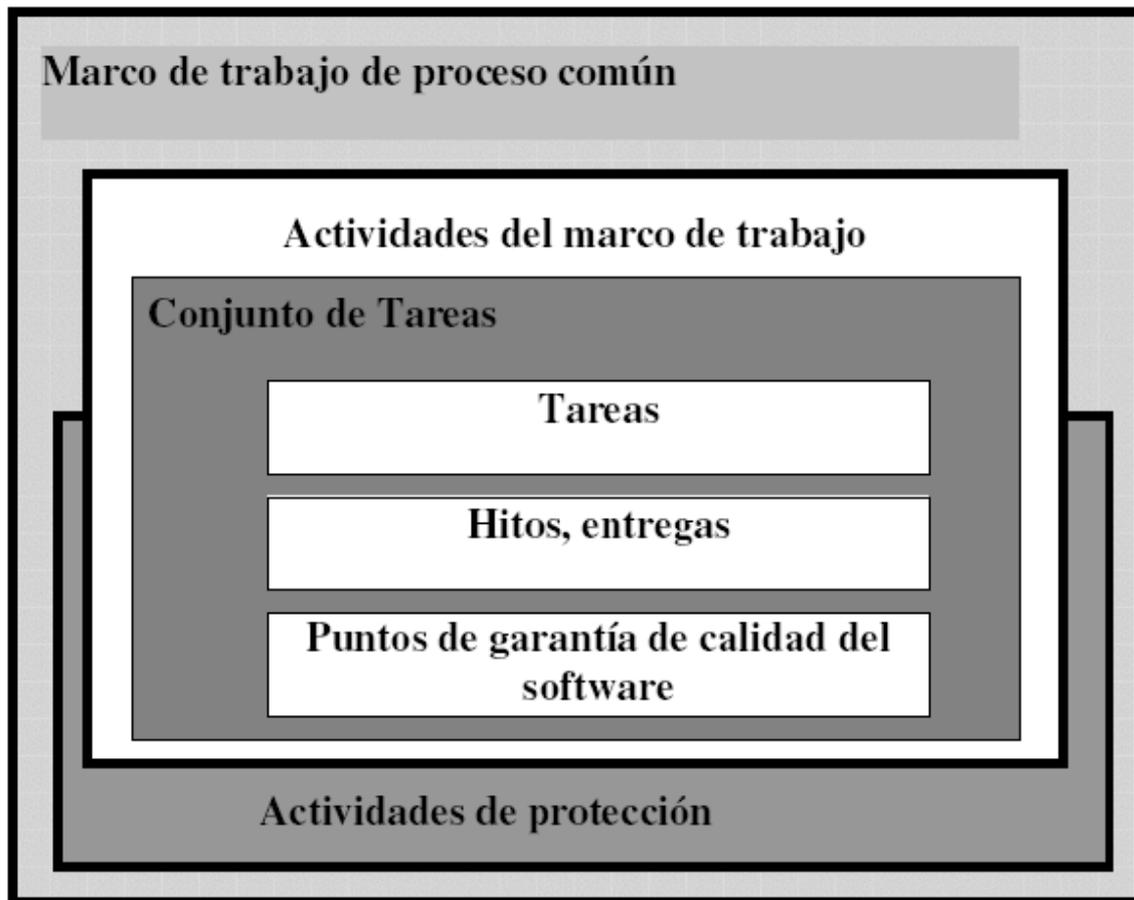


FIG.1.1 PROCESOS DEL SOFTWARE

1.1.9 Paradigmas de Ingeniería de Software

Modelos de proceso o Paradigmas de ingeniería de software.

- Modelo de cascada
 - Separar en distintas fases de especificación y desarrollo.
 - Se caracteriza porque sus fases son de forma secuencial, una después de la otra. (Fig. 1.2)
- Desarrollo evolutivo
 - La especificación y el desarrollo están intercalados.
- Prototipo
 - Un modelo sirve de prototipo para la construcción del sistema final.
- Transformación formal
 - Un modelo matemático del sistema se transforma formalmente en la IMPLANTACIÓN.
- Desarrollo basado en reutilización

- El sistema es ensamblado a partir de componentes existentes.

1.1.9.1 Modelo de Cascada

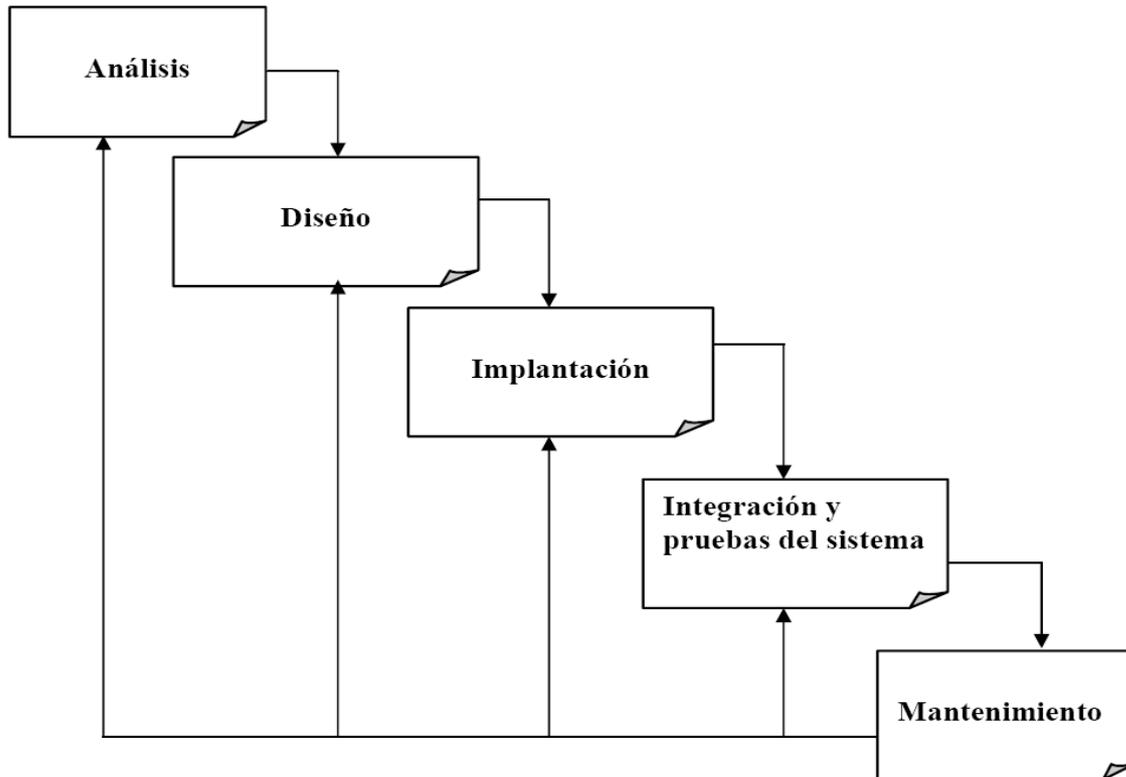


FIG 1.2 MODELO DE CASCADA

1.1.9.1.1 Análisis

Los objetivos de un Análisis del sistema son: Identificar las necesidades del cliente, hacer un análisis técnico y económico, establecer las restricciones de presupuesto y planificación temporal, crear una definición del sistema, que sea la base de desarrollo del sistema y asigne funciones al hardware, software, bases de datos y otros elementos del sistema.

En consecuencia entonces tenemos que el análisis, es un conjunto de hechos, principios y reglas clasificadas y dispuestas de manera ordenada, mostrando un plan lógico en la unión de las partes. Esto se lleva a cabo tomando estos principios:

- Debe presentarse y entenderse el dominio de la información de un problema.
- Definir las funciones que debe realizar el software.
- Representar el comportamiento del software consecuencias de acontecimientos externos.

- Dividir en forma jerárquica los modelos que representan la información, funciones y comportamiento.
- El proceso debe partir desde la información esencial hasta el detalle de la IMPLANTACIÓN.

El modelo de Análisis debe lograr tres objetivos primarios:

- Describir lo que requiere el cliente.
- Establecer una base para la creación de un diseño de software.
- Definir un conjunto de requisitos que se pueden validar una vez que se ha construido el software.

Los servicios del sistema, las restricciones y los objetivos se establecen consultando con los usuarios. Una vez acordados, deben definirse de una manera comprensible, para los usuarios y los desarrolladores del sistema, a través de un documento de requerimientos.

Para poder especificar estos requerimientos se hace uso de estrategias, de tal forma que se entienda lo que desea el cliente, estas pueden ser:

- Revisar la situación actual
- Trabajar en el ámbito del usuario para comprender el contexto, los problemas y las relaciones.
- Entrevistar a los usuarios actuales y potenciales.
- Realizar un video para mostrar cómo podría funcionar el nuevo sistema.

- Investigar los documentos existentes.
- Realizar lluvias de ideas con los usuarios actuales y potenciales.
- Observar las estructuras y los patrones.

1.1.9.1.2 Diseño

Esta etapa es la más importante en el proceso de desarrollo de software, un mal diseño puede probar que funcione el sistema, pero puede ser caro de mantener, difícil de probar y poco confiable.

En esta etapa se especifica cómo va a quedar el programa; tanto en hardware, estructura funcional, pantallas, reportes, procesos, etc.

Algunas herramientas de esta etapa son: especificaciones del sistema, carta estructurada, diagrama de entidad relación, diseño de entradas y salidas diseño de reportes impresos, diseño de la base de datos, diccionario de datos.

Las tareas que se realizan en esta etapa son:

- Seleccionar el hardware y el software necesarios.
- Transformar la especificación funcional obtenida en el análisis a diagramas de diseño.
- Se incorporan medidas de seguridad al sistema.
- Se diseñan formas de entrada y salidas.
- Se diseña la base de datos.
- Los usuarios, administradores y el analista revisan las especificaciones de diseño para mantener exactitud e integridad.

El diseño efectivo se obtiene utilizando una metodología consistente de diseño.

Algunas de estas metodologías las describen Peters (1980), Blank y Krijger (1983). La mayoría de las metodologías se pueden clasificar en:

- Diseño funcional descendente. El sistema se diseña desde un punto de vista funcional, empezando con una visión de alto nivel y refinándola de manera progresiva hasta llegar a un diseño más detallado. Dicha metodología está ejemplificada por el diseño estructurado.
- Diseño orientado a objetos. El sistema se ve más como una colección de objetos que como funciones que pasan mensajes de un objeto a otro.
- Diseño controlado por los datos. Esta metodología, propuesta por Jackson(1975) y Warnier(1977), plantea que la estructura de un sistema de software debe reflejar la estructura de los datos que éste procesa. Por tanto, el diseño del software se obtiene de un análisis de los datos del sistema de entrada y salida.

Los grandes sistemas de software son unas entidades tan complejas, que se pueden utilizar diferentes metodologías en alguna etapa del diseño de distintas partes del sistema. Iian Sommerville define tres notaciones complementarias que pueden utilizarse juntas para describir un diseño de software y estas son:

- Diagramas de flujo de datos. Son diagramas que se utilizan para describir un diseño de sistema de alto nivel; muestran como se transforman los datos al pasar de componente del sistema a otro.
- Diagrama de estructura. Son gráficas de jerarquía que muestran la relación estructural de los componentes de un sistema de software. Esta notación es más útil para describir el diseño de sistemas de alto nivel.

- Un lenguaje para la descripción del diseño. Es una notación con algunos atributos de los lenguajes de programación, adecuada para describir operaciones de control y de diseño detallado. Se puede utilizar en cualquier nivel del sistema.

El diseño de la Base de datos debe hacerse en tres fases:

- Diseño conceptual. Esta fase describe el contenido de la información y no la estructura de almacenamiento.
- Diseño lógico. Es una descripción de la estructura de la Base de Datos que puede ser procesada por un DBMS. Depende de la clase de modelo de datos usado por el DBMS (relacional, red o jerárquico).
- Diseño físico. Depende del DBMS elegido, puesto que se trabaja a nivel de estructuras de almacenamiento. Es una descripción de la implantación de la base de datos en memoria secundaria. Una vez que el diseño físico se ha completado, los esquemas lógicos y físicos se expresan utilizando el lenguaje de definición de datos DBMS elegido, se crea la base de datos, se carga la información y se hacen las pruebas pertinentes.

1.1.9.1.3 Implantación

Es el proceso de instalar el software, como el resultado de un análisis y diseño previo. Antes de implantar un sistema, hay que cerciorarse que cumpla con los requerimientos del análisis y permitir que los usuarios operen sobre él.

En la preparación de la implantación, aunque el sistema esté bien diseñado, su éxito dependerá de su implantación y ejecución, por lo tanto hay que capacitar al usuario que usará el sistema.

Las herramientas que se usan en esta etapa son:

- ✓ Programación estructurada.
- ✓ Procedimiento de pruebas y procedimientos de documentación.

Las tareas a realizar son:

- ✓ Preparar la instalación del equipo necesario.
- ✓ Planear y supervisar la codificación, prueba y corrección del programa.
- ✓ Planear los procedimientos de prueba,
- ✓ El grupo de pruebas verifica los programas.

- ✓ Los usuarios verifican las funciones del sistema.
- ✓ Supervisar la preparación del documento.

1.1.9.1.4 Prueba del Sistema

Durante el proceso de implantación y prueba se deben de implementar todas las estrategias posibles para garantizar que en el uso inicial del sistema éste se encuentre libre de problemas, lo cual se puede descubrir durante este proceso, y llevar a cabo las correcciones de lugar para su buen funcionamiento.

El proceso de pruebas se puede identificar en cinco etapas y son:

- Pruebas de funciones. La prueba de funciones o entidades, es en donde se prueban las funciones de todos los módulos, para garantizar que operan correctamente. Cada función no debe depender de otra, cada función está especificada en el documento de diseño.
- Pruebas de módulos. Un módulo se compone de varias funciones, es necesario revisar las cooperaciones entre ellas, éstas forman un módulo.
- Pruebas de subsistemas. Varios módulos se comunican entre sí, estos forman los subsistemas. La prueba de subsistemas se centra en la prueba de interfaces de aquéllos.
- Prueba del sistema. La prueba del sistema (llamado a veces *PRUEBA DE INTEGRACIÓN*) se lleva a cabo cuando se integran los subsistemas para conformar el sistema completo. En este se encuentran los errores de diseño y la codificación. También se confirma que el sistema cumple con los requisitos planteados inicialmente.
- Prueba de aceptación. En esta etapa, todas las pruebas se realizan mediante el empleo de datos generados por la organización encargada de construir el sistema.

1.1.9.1.5 Mantenimiento

Tiene como objetivo mejorar los productos de software, adaptar productos a nuevos ambientes y corregir problemas.

El mantenimiento, requiere del control administrativo y aspectos técnicos, es decir las herramientas automatizadas para el mantenimiento de software incluyen tanto herramientas técnicas como administrativas. Las herramientas para apoyar los aspectos técnicos se abarcan desde las herramientas de análisis, diseño, implantación, hasta las de depuración y prueba. En el caso de la administración del software tenemos, las formas de solicitudes de cambios, los resúmenes de las solicitudes de cambios, así como una junta de control de cambios que aprueben las modificaciones.

Las herramientas automatizadas más importantes son:

- Editores de texto.
- Ayudas de depuración.
- Generadores de referencia cruzada.
- Editores de enlaces.
- Comparadores.
- Calculadoras métricas de complejidad.
- Sistemas de control de versión.
- Configuración de bases de datos administrativas.

1.2 Teoría del Diseño Conceptual de Bases de Datos

Los sistemas de administración de bases de datos son uno de los productos de software más importantes. Estos proporcionan las herramientas con las cuales los programadores construyen aplicaciones, brindan el ambiente en el cual los paquetes de software se ejecutan, y facilitan la capacidad de administración de datos requerida para los programas de computación.

Un sistema de base de datos puede verse desde dos aspectos, el de los datos y el del programa; desde el punto de vista de los datos cualquier colección de información interrelacionada y agrupada formando una unidad, se denomina Bases de Datos, desde el punto de vista del programa, la Base de Datos es el programa que permite la organización y administración de los datos para su consulta y acceso.

Partes de un sistema de Bases de datos.

Datos.- Son los espacios de almacenamiento que contienen la información de la Base de Datos.

Hardware.- Se compone de volúmenes de almacenamiento secundario, discos, cintas, etc., donde reside la base de datos.

Software.- Se refiere al sistema administrador de base de datos (DBMS) que maneja todas las solicitudes de acceso a la base de datos formuladas por los usuarios.

Usuarios.- Es todo el personal del departamento que requerirá usar el sistema de Base de Datos para realizar consultas o reportes.

1.2.1 Modelos de Bases de Datos

Un modelo de base de datos es una teoría que se usa como base para la definición de datos y para modelar los lenguajes de manipulación de datos.

Requisitos que debe cumplir un modelo de base de datos a nivel conceptual:

- Sencillez
- Tener suficiente poder de expresión para representar situaciones reales (complejas) del mundo.

Existen diferentes modelos:

1. Modelo entidad relación.
2. Modelo de red.
3. Modelo jerárquico
4. Modelo relacional

1.2.1.1 Modelo Entidad-Relación (E/R)

Este es un modelo informal, no es un modelo que se haya usado en los lenguajes de definición de datos. Aunque el modelo entidad relación servirá como teoría inicial para modelar las estructuras de datos y otros conceptos que se usarán en los otros modelos de datos. En particular el modelo entidad relación es un intento adecuado aunque imperfecto de modelar situaciones del mundo real [Ullman, 1999]. Puede pensarse en el modelo entidad relación como un modelo netamente conceptual contrariamente a los otros modelos, los cuáles han servido de base para realizar implementaciones comerciales de sistemas de Bases de Datos.

Entidad

Cosa que existe y es distinguible. Un grupo de entidades similares forman un conjunto de entidades. Las propiedades de las entidades son llamadas **atributos**, los valores para esos atributos están en un **dominio** de valores. Un atributo o conjunto de atributos cuyos valores identifican únicamente cada entidad en un conjunto de entidades es llamada una "llave" o a veces se llama "**clave**" para ese conjunto de entidades.

Los atributos de llave primaria de una entidad no es lo mismo como la llave primaria de una base de datos relacional, esos son conceptos similares pero no hay que confundirlos. La llave primaria de una entidad identifica de manera única a esa entidad, mientras que una llave de una base de datos se refiere a una columna en una tabla que identifica de manera única cada renglón de la tabla. El modelo entidad relación, no es necesariamente lo que se deberá

Implementar como esquema de la base de datos final, aunque de acuerdo, se puede al final elegir el implementar los atributos de la llave primaria como las llaves para la base de datos, pero aún es prematuro en este estado el hacer tal decisión de la IMPLANTACIÓN.

Relación

Una relación entre conjuntos de entidades es simplemente una lista ordenada de conjuntos de entidades, o bien se pueden asociar una relación con una propiedad de las entidades o una acción entre entidades. Un conjunto de entidad puede aparecer más de una vez en la lista. Los tipos de relaciones pueden ser de: uno-uno, muchos-uno, uno-muchos, muchos a muchos.

En la figura 1.3 se bosqueja los tipos de relaciones a considerar en el modelo entidad-relación., las relaciones uno a uno se denotarán por una línea sencilla. La relación uno a muchos, se denota por una línea, en donde un extremo es una sola línea y en otro se hace una bifurcación para denotar que va de uno a muchos. Otra convención, es que la línea que represente una relación puede ser punteada o sólida. Una línea punteada significa que el tipo de relación no es obligatoria esto es que puede o no cumplirse.

Una línea sólida que asocia dos entidades, indica la condición opuesta es decir que es una conexión obligatoria, y por tanto debe siempre cumplirse.

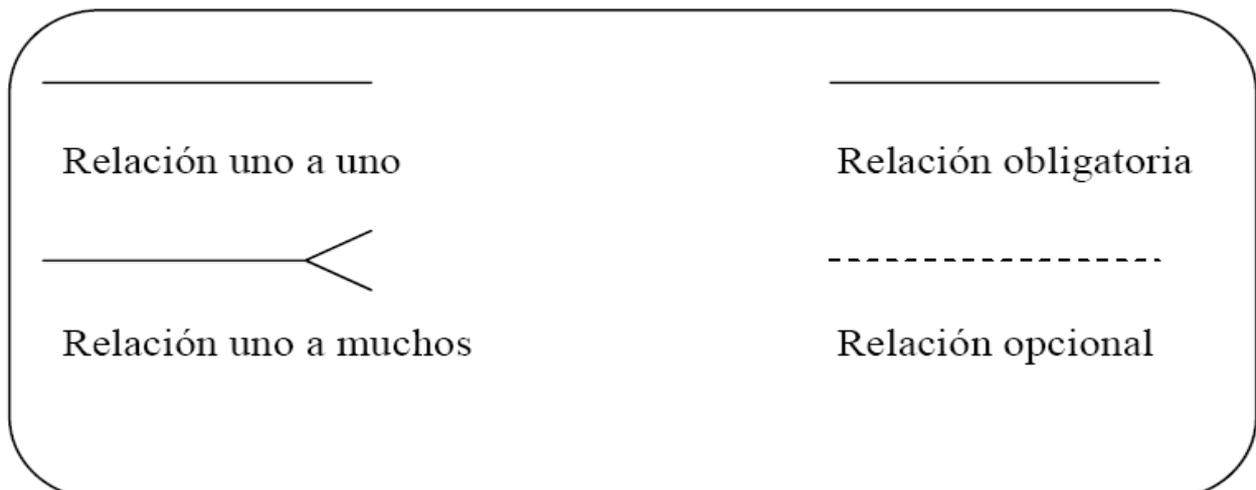


FIG 1.3 TIPOS DE RELACIONES

Diagramas entidad-relación

1. Los rectángulos representan conjunto de entidades.
2. Los círculos representan atributos (ligados a sus conjuntos de entidades mediante flechas sin dirección).
3. Los rombos o rectángulos redondeados representan relaciones ligadas a las entidades por flechas no dirigidas. El orden de las entidades en la lista de la relación puede indicarse por las flechas numeradas. Tal orden es importante cuando la misma entidad aparece más de una vez en la lista. Aunque, en el caso de una relación de muchos a uno se dibuja un arco.
4. Los atributos marcados con una flecha con punta redonda negra, representan una "llave" o "clave".

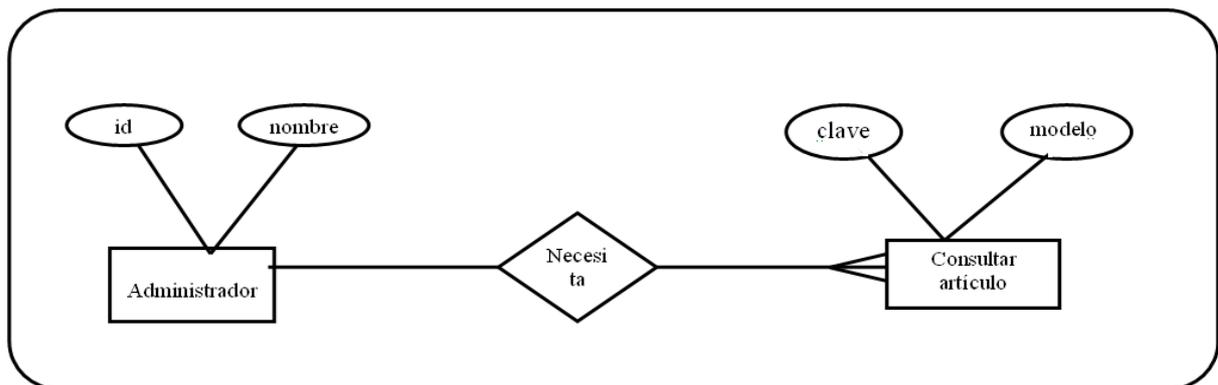


FIG 1.4 EJEMPLO DE DIAGRAMA ENTIDAD-RELACION

En la figura 1.3 observamos como entidades a: Administrador y Consultar artículo, en donde cada entidad se le dibuja algunos de sus atributos asociados y las relación entre estas entidades es vende. La relación entre ambas entidades es uno –muchos.

1.2.1.2 Modelo Relacional

El modelo relacional se representa por medio de una serie de tablas, cada una tiene una o varias columnas con nombres únicos. Cada tabla representa una relación entre un conjunto de entidades, las entidades son representadas por los campos.

Por ejemplo, si se tiene la relación Equipos la cual consiste de los atributos: Clave, Descripción, Modelo y Marca, podemos representar tal relación de la siguiente manera tabular (Fig. 1.5).

TABLA: EQUIPOS

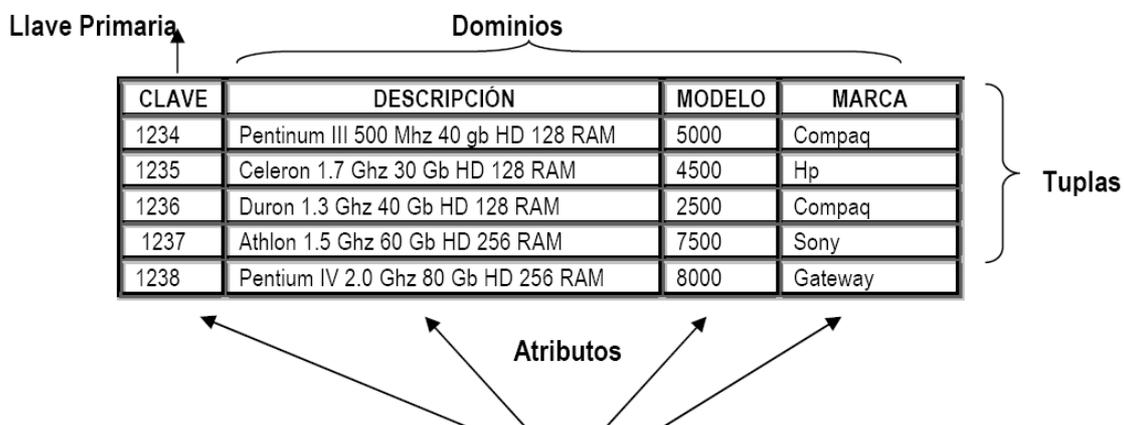


FIG. 1.5 TABLA CON LA INFORMACIÓN GENERAL DE LOS EQUIPOS

En particular una relación corresponde al o que comúnmente se llama tabla.

Ahora escribiremos cada elemento de la relación:

- **Tupla.** Corresponde a una fila de tablas.
- **Cardinalidad.** Corresponde al número de tuplas.
- **Atributo.** Corresponde una columna de una tabla.
- **Llave principal.** Es un identificador único para la tabla; es decir, una columna o combinación de columnas con la siguiente propiedad: nunca existen dos filas de la tabla con el mismo valor en esa columna o combinación de columnas.
- **Dominio.** Es una colección de valores, de los cuales uno o más atributos columna obtienen sus valores reales y se define como un conjunto de valores escalares, todos del mismo tipo.

Ahora veamos otras definiciones:

Un campo es la unidad de datos con nombre más pequeña que se haya almacenado en la Base de Datos y puede verse también como una columna de la tabla. Los campos de la Fig. 1.5 son Clave, Descripción, Modelo y Marca.

Un registro es un conjunto de campos asociados y corresponde a un renglón de la tabla.

Una ocurrencia o instancia de un registro se compone de un grupo de ocurrencias de campos almacenados y relacionados, los cuales se almacenan en el mismo renglón del archivo. Por ejemplo:

1238 Pentium IV 2.0 GHz 80 Gb HD 256 RAM 8000 Gateway

La asociación entre ellos, consiste en que todos representan propiedades de una misma parte específica.

En la mayoría de los sistemas, la ocurrencia de registro almacenado es la unidad de acceso a la base de datos, es decir la unidad en la que el DBMS puede recuperar o almacenar información.

Un archivo almacenado es el conjunto (con nombre) de todas las ocurrencias de un tipo de registro almacenado, y corresponde al nombre de la tabla, la tabla es una relación que se establece entre los campos.

En el modelo relacional las relaciones son representadas a través de archivos que se organizan por llaves, en la organización física de tales archivos es usando el valor de la llave. El hecho de organizar los archivos a través de las llaves permite que las operaciones de inserción y borrado se realicen a través del valor de la llave.

Además el modelo relacional cuenta con bases sólidas en aspectos de las matemáticas, lo cual lo convierte en un vehículo ideal para la enseñanza de los conceptos y principios de los sistemas de bases de datos.

En 1970 Codd [Codd, 1970], propone el uso del modelo relacional para representar a los sistemas de bases de datos. Es un modelo basado en relaciones n-arias y en la normalización de dichas relaciones.

Codd introduce el término de "relación" en su modelo, adoptando el concepto matemático de relación. Así, dados los conjuntos S_1, S_2, \dots, S_n (no necesariamente distintos), R es una relación sobre esos n conjuntos si es un conjunto de n -tuplas; cada una de las cuáles tiene como primer elemento, un elemento de S_1 , como segundo uno de S_2 y así sucesivamente.

En otros términos, se dice que R es un subconjunto del producto cartesiano ($S_1 \times S_2 \times \dots \times S_n$), donde S_j se refiere al j -ésimo dominio de R , y se dice que R es de aridad n . Un dominio es un conjunto de valores posibles. A los elementos del producto Cartesiano se le denomina tuplas, las componentes de las tuplas toman sus valores en el dominio que se definen.

Para fines de notación Codd propone representar una relación por medio de una tabla, la cual consiste de una o varias columnas con nombres únicos. La tabla representa una relación entre un conjunto de entidades y las entidades son representadas por los campos. Entonces podemos ver la propiedad de una tabla que representa a la relación R , de la manera siguiente:

- Cada renglón representa una tupla de R

- El orden de los renglones es intrascendente.
- Todos los renglones son distintos.
- Un registro es un conjunto de campos asociados y corresponde a un renglón de tabla.
- El orden de las columnas es significativo y corresponde al orden S_1, S_2, \dots, S_n de los dominios sobre los que R fue definida.
- El significado de cada columna es convenido por etiquetarlo con un nombre que corresponde a cada dominio.

Para evitarle al usuario la necesidad de conocer las posiciones de los dominios en la definición de cada relación, Codd introduce el concepto de "atributo", el cual se refiere a la etiqueta o identificación de un dominio de la base de datos dentro de la relación R , es decir, si R es de grado 3 y tiene 3 atributos (A_1, A_2, A_3) los cuáles pueden corresponder a los dominios (S_5, S_7, S_{20}). Esto implica que nos referimos a los dominios a través de sus etiquetas, sin necesidad de conocer explícitamente el orden de los dominios. Además, de esta forma cada columna del arreglo puede ser pensada como un atributo y la relación R queda definida sobre un conjunto de atributos.

Una base de datos puede ser vista como una colección finita de relaciones definidas sobre un conjunto finito de atributos. Un esquema relacional se conforma por una o más tablas, las cuáles dinámicamente pueden ser sujetas a inserción de nuevas tuplas, o de la eliminación de algunas de ellas, o bien de la modificación de los componentes de algunas de esas tuplas.

Un esquema de base de datos relacional es un conjunto de esquemas relacionales. A continuación observamos un ejemplo de un esquema en el modelo relacional para el sistema de administración de equipo de cómputo.

Partes (Partes_id, NombrePartes, Descripción)

Los lenguajes de consulta en el modelo relacional, tiene las siguientes propiedades:

- Fácil uso (sin procedimientos)
- Gran poder de expresión

Los lenguajes de consulta para bases de datos relacionales se pueden clasificar por su sintaxis, en:

1. Álgebra relacional
2. Cálculo relacional de dominios
3. Cálculo relacional de tuplas

1.2.1.2.1 Reglas de Integridad

El modelo relacional incluye dos reglas generales de integridad, las cuales se refieren respectivamente a las **llaves primarias (primary key)** y a las **llaves ajenas (foreign key)**.

Llave Primaria (Primary Key)

En términos informales, la llave primaria de una relación es sólo un identificador único para esa relación.

Es posible tener una relación con más de un identificador único. En un caso así, se dirá que la relación tiene varias *llaves candidatas*, nosotros escogeríamos entonces una de esas llaves candidatas como llave primaria y las demás las llamaremos entonces *llaves alternativas*.

De una forma más precisa se define el término “llave candidata” de la siguiente manera:

El atributo K (posiblemente compuesto) de la relación R es una llave candidata de R si y solo si satisface las siguientes dos propiedades, independientes del tiempo:

1. Unicidad.

En cualquier momento dado, no existen dos tuplas en R con el mismo valor de K.

2. Minimalidad.

Si K es compuesto, no será posible eliminar ningún componente de K sin destruir la propiedad de unicidad.

Del conjunto de llaves candidatas de una relación dada, se elige una y sola una como **llave primaria** de esa relación; las demás si existen, se llamarán llaves alternativas. Así una llave alterna es una llave candidata que no es la llave primaria.

Llave Foránea (Foreign Key)

De modo informal se puede definir que una llave ajena es un atributo (quizá compuesto) de una relación R2 cuyos valores deben concordar con los de la llave primaria de alguna relación R1 (donde R1 y R2 no necesariamente son distintos, como veremos más adelante).

Un valor de llave ajena representa una Referencia a la tupla donde se encuentra el valor correspondiente de la llave primaria (la tupla referida o tupla objetivo). Por tanto, el problema de garantizar que la B.D. no incluya valores no válidos de una llave ajena se conoce como el problema de la integridad referencial. La restricción según la cuál los valores de una llave ajena deben concordar con los valores de la llave primaria correspondiente se conoce como restricción referencial. La relación que contiene a la llave ajena se conoce como relación referencial y la relación que contiene a la llave primaria correspondiente se denomina relación referida o relación objetivo.

Se puede definir el término llave ajena de una forma más precisa:

El atributo LF (quizá compuesto) de la relación base R2 es una **llave foránea** si y solo si satisface las siguientes dos propiedades independientes del tiempo.

1. Cada valor de LF es nulo del todo o bien no nulo del todo (con “nulo del todo” o “no nulo del todo” queremos decir que, si LF es compuesto, todos sus componentes son nulos o bien todos sus componentes son no nulos, no una combinación).

2. Existe una relación base R1 con llave primaria LP tal que cada valor no nulo de LF es idéntico al valor de LP en alguna tupla de R1.

I. Regla de Integridad de las Entidades

“Ningún componente de la llave primaria de una relación base puede aceptar valores nulos”

II. Regla de integridad Referencial

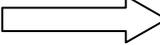
“La base de datos no debe contener valores de llave ajena sin concordancia”

1.2.1.2.2 El proceso de Normalización

La normalización es el proceso de transformación de representaciones de datos de usuarios en conjuntos estables de estructuras de datos de menor tamaño. Además de ser más sencillas, tales estructuras son más estables. Las estructuras de datos normalizadas son más fáciles de mantener.

INTEGRIDAD

Propiedades de las relaciones.
No existen tuplas duplicadas
Las tuplas no tienen orden
Los atributos tampoco tienen orden
Los valores de los atributos son atómicos

Si R satisface  Esta Normalizada

FORMAS NORMALES

El objetivo de las formas normales es que dado el modelo original nos proporcionan un modelo entidad-vínculo que respete la integridad de la B.D.

1N Elimina atributos repetidos
2N Eliminamos redundantes
3N Elimina datos no dependientes de la clave
FN BC 3.%N
4N (Óbice-Codd)
5N

Formas Normales para resolver problemas de integridad en las B.D. Relacionales

Si permitimos que los valores de los atributos sean no atómicos entonces en el diseño de la B.D. se tiene que llegar hasta 4N.

La normalización minimiza la redundancia de datos (es decir la repetición del mismo valor en un atributo), ya que la redundancia de datos causa problemas de integridad: Las transacciones de actualización y borrado pueden no ser consistentes en todas las copias de los datos, causando inconsistencia de la base de datos. Además la normalización ayuda a identificar entidades, relaciones y tablas faltantes.

Los Tres Pasos de la Normalización

A. La primera etapa del proceso incluye la eliminación de grupos repetidos y la identificación de la llave primaria. Con el fin de hacer esto, la relación necesita desglosarse en dos o más relaciones.

B. El segundo paso asegura que todos los atributos no-llave, sean completamente dependientes de la llave primaria. Todas las dependencias normales se eliminan y se colocan en otra relación.

C. El tercer paso elimina cualquier dependencia transitoria. Una dependencia transitoria es aquella en la cual sus atributos no-llave son dependientes de otros atributos no-llave.

Forma Normal de Boyce – Codd (4FN)

En un estudio posterior [Codd, 1974], Codd introduce una nueva forma normal que comúnmente se ha denominado la forma normal de Boyce-Codd, y menciona que una relación R está en forma normal de Boyce-Codd si y solo si, primero se encuentra en tercera forma normal y todas las dependencias no triviales tienen un lado izquierdo que contiene una llave de R.

1.2.3 El Sistema Administrador de Base de Datos (DBMS)

La administración de Base de datos se puede aplicar a las distintas formas en que la información es clasificada, almacenada y usada.

El software que permite a una o más personas el usar y/o modificar los datos de una base de datos se denomina Sistema Administrador de Base de Datos (DBMS). Además permite al usuario el tratar con los datos en términos abstractos, más que como datos almacenados en una computadora.

Conceptualmente lo que sucede es lo siguiente:

1. Un usuario solicita acceso, empleando algún sublenguaje de datos determinado (por ejemplo, SQL).
2. El DBMS interpreta esa solicitud y la analiza.

3. El DBMS inspecciona, en orden, el esquema externo de ese usuario, la correspondencia externa/conceptual asociada, el esquema conceptual, la correspondencia conceptual/interna, y la definición de la estructura de almacenamiento.

4. El DBMS ejecuta las operaciones necesarias sobre la base de datos almacenada.

Entre las funciones que realiza un DBMS están:

- Definición de datos

Debe ser capaz de aceptar definiciones de datos (esquemas externos, el esquema conceptual, el esquema interno y todas las correspondencias asociadas).

- Manipulación de datos

Debe ser capaz de atender las solicitudes del usuario para extraer, y quizá poner al día, datos que ya existen en la base de datos, o para agregar en ella datos nuevos.

Es decir, debe incluir un componente procesador de DML.

- Seguridad e Integridad

Debe supervisar las solicitudes de los usuarios y rechazar los intentos de violar las medidas de seguridad e integridad definidas por el DBA.

- Recuperación y Concurrencia de Datos

Debe cuidar del cumplimiento de ciertos controles de recuperación y concurrencia.

- Diccionario de Datos

Debe incluir una función para el registro y mantenimiento de un Diccionario de Datos.

Un Diccionario de datos es una BD, contiene datos (“metadatos”) acerca de los datos de la BD es decir, definiciones de otros objetos del sistema y no solo los datos en sí.

En particular en un Diccionario de Datos se almacenarán:

- ✓ Los diversos esquemas y correspondencias (externos, conceptuales, etc.).
- ✓ Referencias para determinar cuáles programas usan cuáles partes de la BD.
- ✓ Referencias para determinar cuáles usuarios requieren cuáles informes.
- ✓ Referencias para determinar cuáles terminales están conectadas al sistema.
- ✓ Desempeño.

El DBMS deberá ejecutar todas sus funciones de la forma más eficiente posible.

En síntesis un DBMS constituye una interfaz entre el usuario y el sistema de base de datos.

1.2.3.1 Arquitectura para un DBMS

La arquitectura para un DBMS proporciona un marco de referencia para describir los conceptos generales de bases de datos y explicar la estructura de sistemas específicos. La

propuesta de Date [Date 2001], que en general concuerda con la arquitectura de ANSI/SPARC [6], figura 1.6 consta de tres niveles:

El nivel interno.

Se ocupa de la forma en cómo se almacenan físicamente los datos.

El nivel externo.

Se ocupa de la forma en cómo los usuario perciben los datos.

El nivel conceptual.

Es un nivel de mediación entre los niveles anteriores.

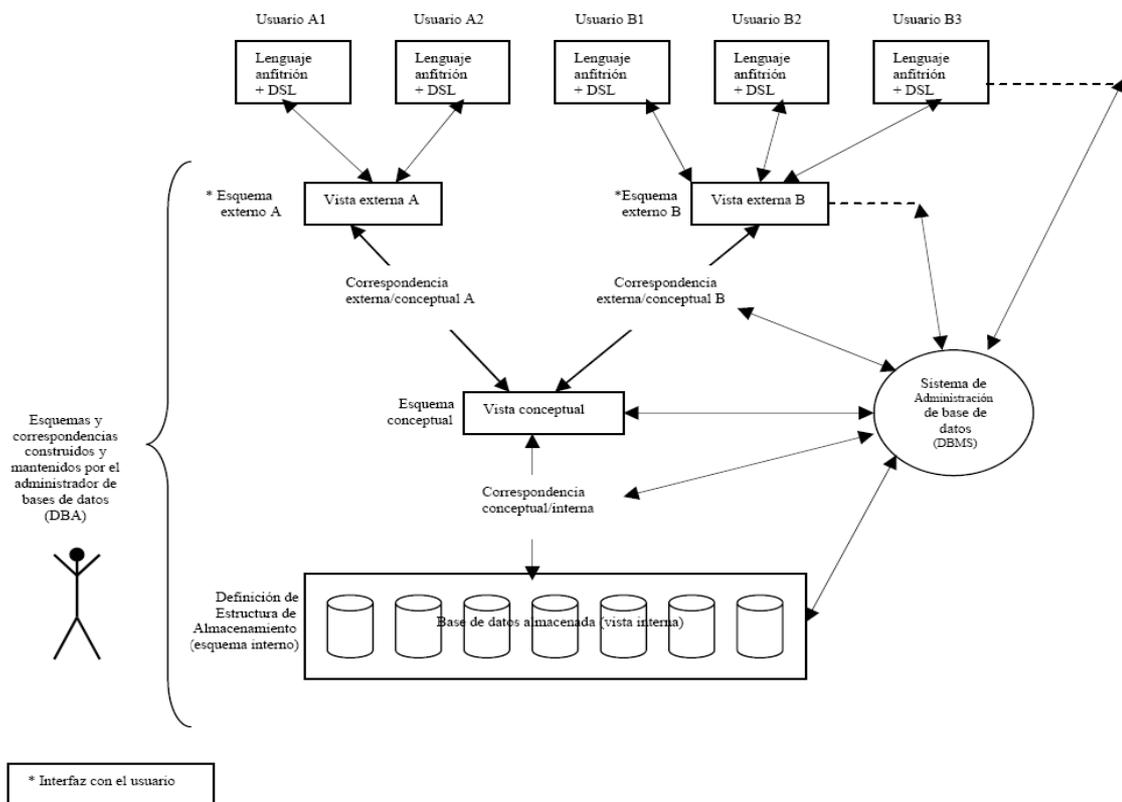


FIG. 1.6 ARQUITECTURA DE UNA DBMS

Etapas del diseño de una Base de Datos Relacional

1. Modelo conceptual de datos. En la que se produce el diseño. Aquí se recogen los aspectos relevantes de los datos de una parcela del mundo real, relacionada con la aplicación que se intenta desarrollar por lo general se emplean para esta descripción los conceptos del modelo de entidad-relación.

2. Diseño lógico de la Base de Datos. Es la segunda etapa del diseño, y se parte del modelo conceptual. En el que se describen las tablas y columnas que formaran la Base de datos.

3. Diseño físico de la Base de datos. Se produce en la tercera parte del diseño, a partir del diseño lógico. En el que se detallan las características físicas de los archivos donde se van a almacenar las tablas.

1.2.4 Modelo Funcional

El objetivo de un modelo funcional es el definir en los términos más simples, lo que el negocio actualmente hace. El modelo funcional nos deberá llevar a un análisis más detallado del sistema, y debe servir para hacer una revisión de las precisiones de los subsiguientes pasos en el análisis, que a saber son: los diagramas entidad-relación y el modelo de flujo de datos.

Se deberá iniciar con el componente más fundamental, un modelo que ilustre las funciones principales del negocio que se analiza. Deben presentarse todas las funciones peculiares del mismo.

El modelo funcional es jerárquico, se inicia con una simple proposición estratégica del negocio, una descripción escueta del negocio entero y se divide típicamente entre siete y ocho grandes funciones, y entonces se revisa cada una de estas grandes funciones para pasar a una descripción más detallada que cubra el alcance total del negocio, el proceso de construir este modelo es apropiadamente nombrado descomposición funcional.

1.2.5 Modelo de Flujo de Datos

El analista se verá ante una gran variedad de herramientas de modelado en proyectos del mundo real. Aunque los detalles (y formas) de estas herramientas varían mucho, siguen los principios básicos.

Los Diagramas de Flujos de Datos es una herramienta que permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por “conductos” y “depósitos de almacenamiento” de datos. En otras palabras, un “Diagrama de Flujo de

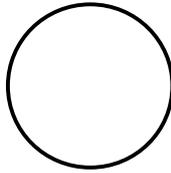
datos es una técnica gráfica que representa el flujo de información y las transformaciones que se aplican a los datos al moverse desde la entrada a la salida”.

También es una de las herramientas más comúnmente usadas, sobre todo por sistemas operacionales en los cuales las funciones del sistema son de gran importancia y son más complejas que los datos que éste maneja. Los DFD se utilizaron por primera vez en la Ingeniería de software como notación para el estudio del diseño de sistemas.

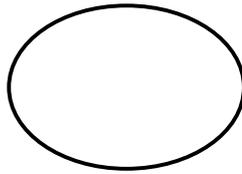
Componentes de un Diagrama de Flujo de Datos

Los componentes de un diagrama típico de flujo de datos: Procesos, flujos, almacén y terminador.

El proceso. El primer componente del DFD se conoce como proceso. Los sinónimos comunes son burbuja, función o transformación. El proceso muestra una parte del sistema que transforma entradas en salidas. El proceso se representa gráficamente como un círculo.



Algunos analistas prefieren usar un óvalo o un rectángulo con esquinas redondeadas.



Las diferencias entre estas tres formas son puramente estéticas, aunque obviamente es importante usar la misma forma de manera consistente para representar todas las funciones de un sistema.

El Flujo. Un flujo se representa gráficamente por medio de una flecha que entra o sale de un proceso. El flujo se usa para describir el movimiento de bloques o paquetes de información de una parte del sistema a otra. Por ello, los flujos representan datos en movimiento, mientras que los almacenes representan datos en reposo.



Los flujos muestran la dirección: una cabeza de flecha en cualquier extremo (o posiblemente ambos) del flujo indica si los datos (o el material) se está moviendo hacia fuera de un proceso (o ambas cosas).

El Almacén. El almacén representa la información almacenada que puede ser utilizada por el software. Se utiliza para modelar una colección de paquetes de datos en reposo. Se denota por dos líneas paralelas.



La base de datos existe como un área de almacenamiento diferida en el tiempo, necesaria entre dos procesos que ocurren en estos momentos diferentes.

El Terminador. El siguiente componente del DFD es un terminador; gráficamente se representa como un rectángulo.



Los terminadores representan entidades externas con las cuales el sistema se comunica. Es una persona o un grupo, que está dentro de la misma compañía u organización, pero fuera de control del sistema que se está modelando.

Los terminadores son externos al sistema que se está modelando; los flujos que los conectan a diversos procesos en el sistema representan la interfaz entre él y el mundo exterior.

1.2.6 Metodología para desarrollar el Análisis de Sistemas de Bases de Datos

Un método de desarrollo sano es lo que se necesita, que sea completo, que sea un método de autodocumentación y de entendimiento inmediato tanto para usuarios finales como para profesionales del procesamiento de datos. También los métodos de desarrollo se convierten en el marco para controlar todas las fases del ciclo de vida de la aplicación y aún este deberá de ser flexible y de respuesta a cambios tales como nuevas demandas de los usuarios finales o nuevas tendencias en la tecnología.

Obviamente existen diversas metodologías que nos auxilian durante todo el ciclo de vida del software, cada metodología se basa en un conjunto de técnicas, las cuales generalmente son compartidas en diferentes métodos. Una de las metodologías que

últimamente ha sido ampliamente utilizada por las compañías de desarrollo de Bases de Datos, es la denominada **CASE**.

El análisis es el estudio de un problema, antes de tomar una acción en computación, generalmente la acción consiste en la implantación de un nuevo sistema (que puede incluir procesos manuales). El análisis del sistema es el primer paso crítico de cualquier método de desarrollo de aplicación sano. El diseñador de aplicaciones deberá analizar el negocio y sus problemas de manera significativa, antes de que se espere puedan ofrecer una solución real para el sistema. **El método CASE** proporciona varias herramientas para modelar y para asistir al diseñador durante este ciclo de desarrollo.

Los pasos a seguir, que recomienda el método CASE al realizar el análisis de un sistema son:

1. Entrevistas con los expertos del negocio.
2. Definir el modelo funcional del negocio.
3. Definir el modelo entidad-relación.
4. Construir el modelo de flujo de datos.

CAPÍTULO 2

ANÁLISIS DEL SISTEMA

2.1 Introducción

El análisis, es un conjunto de hechos, principios y reglas clasificadas y dispuestas de manera ordenada, mostrando un plan lógico en la unión de las partes.

2.2 Definición y planteamiento del problema

El control de inventario de las salas de cómputo de Fundación UNAM se lleva a cabo de forma manual, es decir, se ocupa formatos donde se escriben datos tales como material existente o el que se va asignando, este material consiste en: lugares físicos, mobiliario y hardware. En muchas ocasiones existe mucha incongruencia en el reporte de inventario ya que pueden extraviarse algunos formatos o se mueve mobiliario de un lugar a otro y no se actualiza en los formatos.

Una solución a este problema, es crear un sistema que automatice el proceso de almacenamiento del material (infraestructura) que existe actualmente en las diferentes salas de cómputo y el que se le vaya asignando, para ello es necesario contar con un catálogo de artículos para que los departamentos de la carrera puedan hacer solicitudes de material, mediante un proceso de adquisiciones se ubique de manera automática el material solicitado y finalmente mediante la creación de reportes se pueda consultar cuanto material existe en las salas de cómputo y donde se encuentra ubicado.

2.3 Entrevista

La fase de recolección de información es quizás la más crítica de todas. Después de entrevistar a los encargados de las salas de cómputo se deberá conocer tanto acerca de la organización, como para poder hacer un primer bosquejo de su situación general, la entrevista con los encargados puede ser secundaria para uno, pero realmente no lo es. Hay algunas reglas básicas especiales para el proceso de entrevistas direccionadas por el método CASE.

Como en cualquier sesión de entrevistas es esencial establecer confianza con el entrevistado y hacer el mejor uso del tiempo que ellos gastan con uno, pero igualmente es muy crítico el hacer preguntas adecuadas. Cada uno de los participantes tiene su propia percepción acerca de lo que es importante en la organización. Uno debe de poder extraer información precisa de cada entrevistado y dejar fuera la información tangencial que no servirá para el desarrollo del sistema.

2.4 Narrativa

En la mayoría de las instituciones educativas que cuentan con infraestructura de procesamiento de información, (si no es que en todas), llevan a cabo el proceso de inventario por medio de formatos en papel, dicho proceso en muchas ocasiones confuso, tedioso y en el peor de los casos “incongruente”, actualmente se requiere aprovechar la tecnología que existe para automatizar el procesamiento de dicha información.

Actualmente en esta institución (Fundación UNAM de la FES Aragón) se lleva a cabo el proceso de inventario de equipo de cómputo en formatos de manera tradicional para después archivarlos en pequeñas y sencillas estructuras de almacenamiento, lo cual lo hace de una manera poco eficiente.

Por lo que la institución quiere actualizar este sistema a una base de datos con la que se mejoraría la forma de llevar a cabo el control de inventario con los que cuenta actualmente, entre las ventajas podemos destacar las siguientes:

- ✓ Se minimiza el gasto de papelería al llevar el control de manera electrónica
- ✓ El encargado de elaborar el inventario puede realizar consultas desde otras salas de cómputo (lugares remotos)
- ✓ Se tendrá información más congruente con respecto a lo que se tiene de infraestructura

2.5 Análisis de requerimientos

Se requiere el desarrollo de un sistema que permita el manejo de forma automatizada de la información administrativa con que trabaja la Facultad de Estudios Superiores Aragón en la organización de Fundación UNAM, en el módulo de infraestructura almacenará la información referente al hardware que están asignados a cada sala de cómputo. El sistema permitirá dar de alta a la infraestructura hardware que llegue a las salas de cómputo, así como almacenar el ya existente a través de una base de datos con tablas relacionadas, ya que también permitirá agilizar la presentación de reportes a usuarios del sistema a través de la intranet.

2.5.1 Requerimientos No Funcionales

- Se propone que el sistema se desarrolle bajo el ambiente Windows.
- Se propone un gestor de Base de Datos Oracle.
- Los requerimientos de hardware para trabajar con el sistema son:
 - Pentium III en adelante
 - 128 Mb en RAM
 - Disco Duro de 10 GB
 - CD-Writer (recomendado)
 - Puertos USB (recomendado)

2.5.2 Requerimientos Funcionales

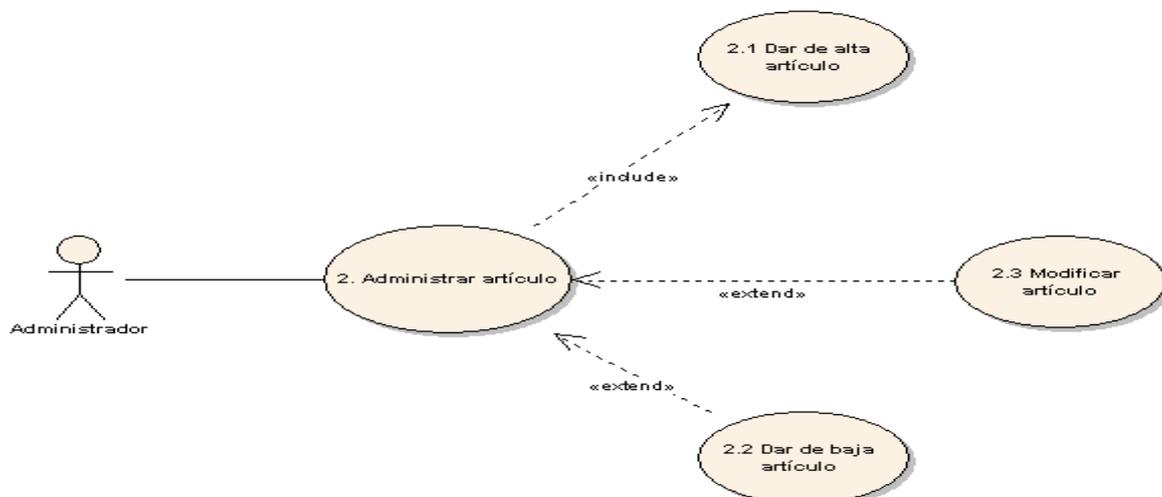
- Llevar un control de altas de equipo de cómputo que se utiliza en las salas de cómputo de Fundación UNAM.
- Llevar un control de consulta o búsqueda de algún equipo(s) o parte(s) del equipo de cómputo (ratón, teclado, monitor, etc.) en particular por pantalla, para su pronta localización.
- Llevar un control de reportes generales de hardware por pantalla y papel y comparar lo que se tiene físicamente con lo que se reporta de la base de datos.
- Llevar un control de bajas de hardware por pantalla y papel, también de forma general controlar lo que existe como baja en la base de datos con lo que se dio de baja en algún espacio físico.
- Controlar los reportes emitidos ya sea por papel o por pantalla en la Intranet.
- Validar con una contraseña el acceso para que se lleven a cabo los requerimientos funcionales.

2.6 Narrativas de Caso de Uso en Formato Expandido

2.6.1 Administrar Artículos

Caso de uso: Administrar artículos.

Actor: Administrador



Descripción: El administrador podrá administrar el (los) artículos(s).

Precondiciones:

- El administrador debe haber ingresado al sistema.

Flujo:

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El administrador ingresa a "Administración de artículos".	2	El sistema despliega la pantalla con las siguientes opciones: "Dar de alta artículo", "Dar de baja artículo", "Modificar artículo".	

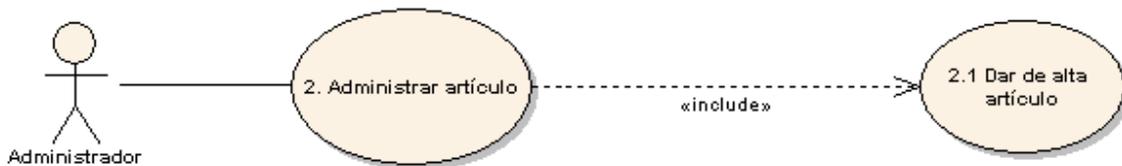
Poscondiciones:

- El administrador podrá administrar un o varios artículos.

2.6.2 Dar de Alta Artículo

Caso de uso: Dar de alta artículo.

Actor: Administrador



Descripción: El administrador podrá dar de alta uno o varios artículos.

Precondiciones:

- El administrador debe haber ingresado al sistema.

Flujo:

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El administrador selecciona "Dar de alta artículo".	2	El sistema pide Tipo de artículo, Nombre del artículo, Fecha.	
3	El administrador captura El Tipo de artículo, Nombre del artículo y Fecha.	4	El sistema guarda la información.	E1, E2
		5	El sistema informa al administrador que el artículo a ha sido dado de alta.	

Excepciones:

Id	Nombre	Acción
E1	Campo vacío.	El sistema informa al cliente que un campo está vacío.
E2	Numero de inventario repetido.	El sistema informa al administrador que el número de inventario ya existe, tendrá oportunidad de escoger uno nuevo.

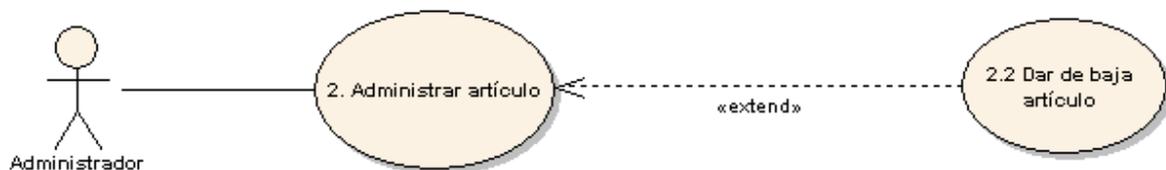
Poscondiciones:

- El administrador a dado de alta uno o varios artículos.

2.6.3 Dar de Baja Artículo

Caso de uso: Dar de baja artículo.

Actor: Administrador



Descripción: El administrador podrá dar de baja el (los) artículo(s).

Precondiciones:

- El administrador debe haber ingresado al sistema.

Flujo:

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El administrador selecciona "Dar de baja artículo".	2	El sistema pide Tipo de artículo, Nombre del artículo, Fecha.	
3	El administrador material El Tipo de artículo, Nombre del artículo y Fecha.	4	El sistema guarda la información.	E1, E2
		5	El sistema informa al Administrador que el artículo ha sido dado de baja.	

Excepciones:

Id	Nombre	Acción
E1	Campo vacío.	El sistema informa al administrador que un campo está vacío.
E2	Información errónea	El sistema informa al administrador que los datos son incorrectos.

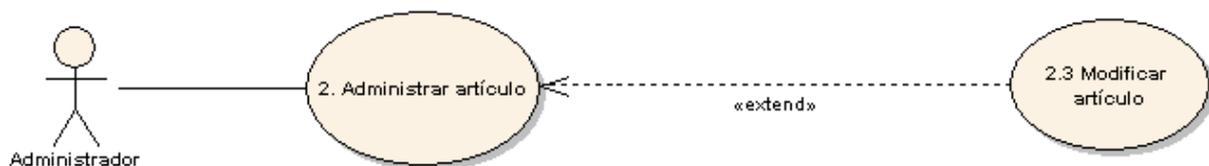
Poscondiciones:

- El administrador ha dado de baja el (los) artículo(s).

2.6.4 Modificar Artículo

Caso de uso: Modificar artículo.

Actor: Administrador



Descripción: El administrador podrá modificar información de uno o varios artículos.

Precondiciones:

- El administrador debe haber ingresado al sistema.

Flujo:

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El administrador selecciona "Modificar artículo".	2	El sistema pide Tipo de artículo, Nombre del artículo, Fecha.	
3	El administrador captura Tipo de artículo, Nombre del artículo, Fecha.	4	El sistema despliega la pantalla con la información del artículo.	E1, E2
5	El administrador modifica la información del artículo.	6	El sistema guarda la información.	
		7	El sistema informa al administrador que el artículo ha sido modificado.	

Excepciones:

Id	Nombre	Acción
E1	Campo vacío.	El sistema informa al administrador que un campo está vacío.
E2	Información errónea.	El sistema informa al administrador que los datos son incorrectos.

Poscondiciones:

- El administrador ha modificado uno o varios artículos.

2.7 Análisis Costo/Beneficio

El Análisis de Costo/Beneficio es una técnica usada para evaluar programas o proyectos de inversión, que consiste en comparar costos con los beneficios asociados a la realización del proyecto. Un proyecto estará bien aspectado si los beneficios superan los costos. Los beneficios pueden ser de tipo monetario o social, directo o indirecto. La diferencia esencial entre el análisis de costo/beneficio y los métodos ordinarios de evaluación de inversiones, es el énfasis en los costos y beneficios sociales. El objetivo consiste en identificar y medir las pérdidas y las ganancias en el bienestar económico que recibe la sociedad en su conjunto.

Para el caso del Sistema para Administrar el Inventario de Equipo de Cómputo (Fundación UNAM), será desarrollado en un entorno Web, para lo cual la Facultad de Estudios Superiores Aragón ya dispone de un servidor que lo soporte.

Los costos de para el sistema se distribuyen de la siguiente manera:

- Equipo (12 computadoras) con las siguientes características mínimas:

CARACTERISTICAS		
Procesador	Procesador AMD® Opteron™ doble núcleo; 1210; 1.8GHz,2X1MB Cache	\$96.00
Memoria RAM	Memoria DDR2 de 1GB, DDR2, 800MHz, 1x1G, Dual Ranked DIMM	\$ 31.00
Disco Duro	Disco duro SATA 80GB 7.2K RPM 3Gbps 3.5-in Cabled	\$42.00
Tarjeta de Red	Adaptador de red integrado de un solo puerto Gigabit	\$18.00
DVD RW	LG 20x Negro OEM H55N	\$30.00
TOTAL		\$ 217.00

Nota: el costo total es por cada equipo, los precios se manejan en dólares

- Incorporación de una UPS (unidad de alimentación ininterrumpida, también conocida como SAI, que sirve para proteger nuestros datos en caso de que se vaya la luz), como medio de protección para el sistema.

UPS	Trip Lite UPS 1000VA OMNIVSINT1000
TOTAL	\$ 155.00

- No se incorpora monitor ya que la Facultad de Estudios Superiores Aragón cuenta con uno acorde al servidor.
- El software para el desarrollo del sistema es de licencia libre, lo que no representa costo para la Facultad de Estudios Superiores Aragón.

Software	Cantidad	Costo	Total
	1	\$ 00.00	\$00.00

- Programación

Programación	Cantidad	Costo	Total
	1	\$ 00.00	\$00.00

- Adiestramiento

Adiestramiento	Horas
TOTAL	60 hrs / 6 empleados

- Actualización Software de computadoras existentes, Windows Update.

Actualización Software	Horas / Empleados	Costo	Total
	180hrs / 6 empleados	\$ 12.00	\$96.00

- Actualización Hardware de computadoras ya existentes.

Actualización Hardware	Memorias RAM	Costo	Total
	8	\$ 31.00	\$296.00

Costo de:	Valor
Desarrollo	\$ 00.00
Actualización	\$ 296.00
Equipo	\$3055.00
Horas Hombre	\$ 00.00

Costo del Sistema = \$ 3351.00

Beneficios

El sistema proveerá en tiempo real una manera eficiente la administración del equipo de cómputo para cada sala de cómputo de Fundación UNAM.

Beneficios Tangibles:

- Información exacta y precisa en cualquier momento y lugar durante el proceso de inventariado: Información en tiempo real.
- Ahorro y reducción de tiempo y costos, durante el proceso de inventariado.
- Mejora en la productividad de los empleados: Ofreciéndoles funcionalidades fáciles de utilizar.

- Eficiencia, reduciendo los errores durante el proceso, ya sea por datos erróneos o datos perdidos.

Beneficios Intangibles:

- Los empleados puedan hacer el inventariado en cualquier parte vía WEB.
- Integración de otras salas cómputo externas a Fundación UNAM, facilitando la integración al sistema.

CAPÍTULO 3

DISEÑO DEL SISTEMA

3.1 Introducción

El desarrollo de sistemas se entiende como un ciclo de vida de software que se repite una y otra vez como consecuencia de que el desarrollo de sistemas siempre está sujeto a mejora continua, ya que nunca se puede decir que un sistema está desarrollado o que trabaja al 100% porque éste estará sujeto a cambios imprevistos o actualización continua.

3.2 Modelo de Análisis del Sistema

La finalidad de los diagramas es presentar diversas perspectivas de un sistema, similar a las cuales se les conoce como *modelo*.

Estos son algunos puntos importantes para comenzar hacer el análisis del sistema:

- Los documentos generados deben explicar de manera clara y sin ambigüedades, las reglas de negocio del software.
- Se modela el sistema con alto nivel de abstracción.
- Los documentos de análisis serán la guía para la lógica de negocio

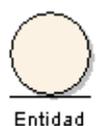
En el momento de hacer el primer diagrama de clases ya se tiene una lista de clases con algunos de sus atributos y operaciones. Sin embargo es necesario reflexionar y abstraer sobre la organización de esas clases estudiando las relaciones de herencia, agregación, etc.

El modelo de análisis sugiere la siguiente agrupación de clases:

- Clases de interacción con el usuario (Boundary):



- Clases de persistencia (entity)



- Clases de regla de negocio (control)



Clases “Boundary”

En un desarrollo de tipo WEB, cada HTML y JSP, representará una clase “Boundary”, en donde los métodos estarán determinados por las ligas o botones incluidos.

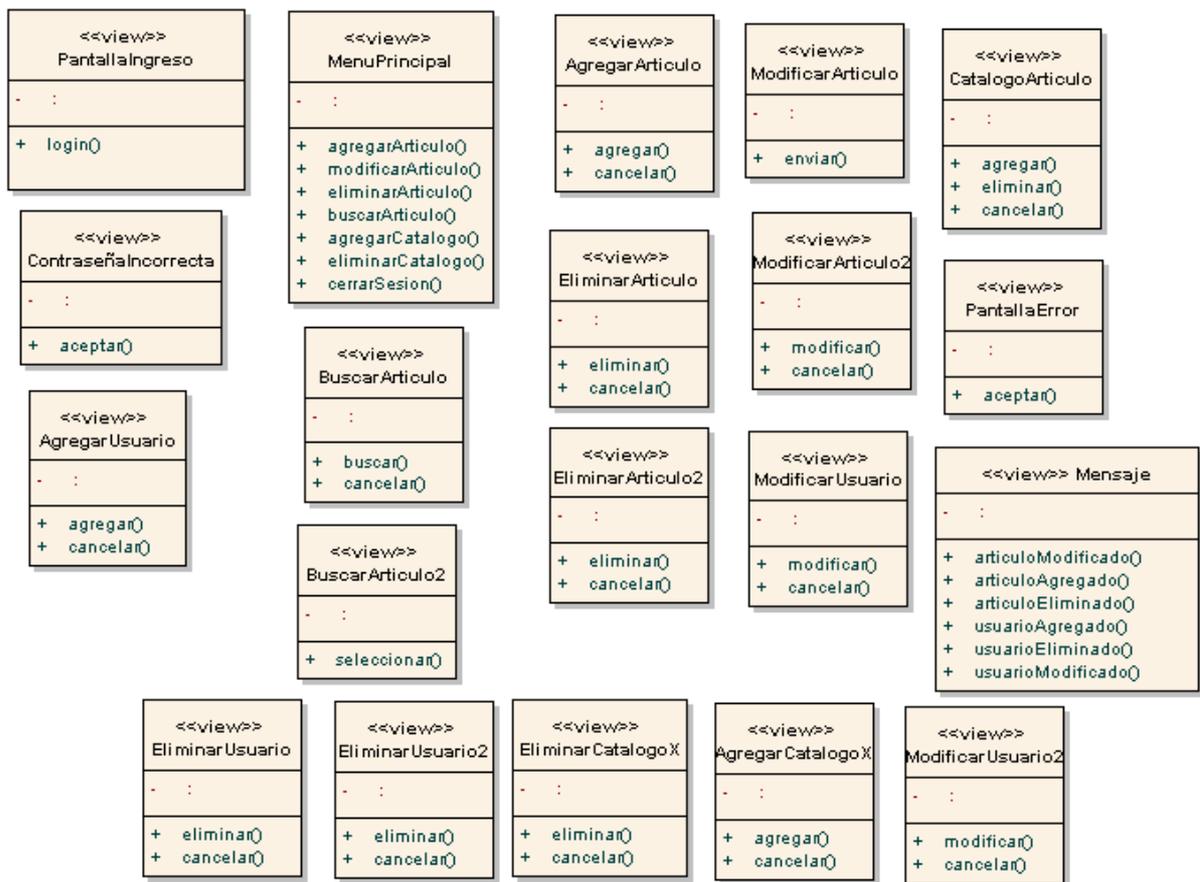


FIG. 3.1 DIAGRAMA DE CLASES DE INTERACCIÓN CON EL USUARIO PARA FUNDACION UNAM (BOUNDARY)

Clases de “Regla de negocio”

Las clases de tipo “Regla de negocio” o “de control” son clases que involucran la operación del resto del sistema. Están incluidas aquellas entidades que definen la lógica y aquellas que realizan cálculos u operaciones que involucran a las clases “Entity”.

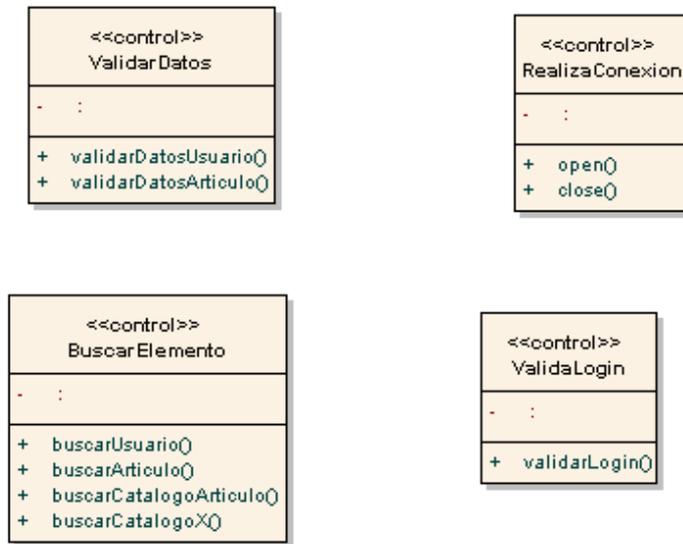


FIG. 3.2 CLASES DE CONTROL PARA FUNDACION UNAM

Clases “Entity”

Las clases de tipo “entity” son los candidatos a tener una larga vida durante la ejecución del sistema; es decir, serán clases cuyo estado debe ser almacenado en disco.

Generalmente, son clases que son identificadas de manera inmediata, durante el flujo de análisis del desarrollo.

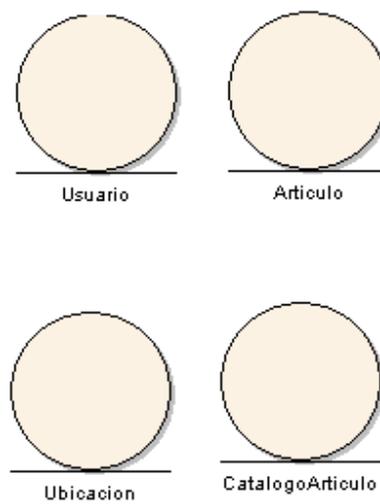


FIG. 3.3 CLASES DE ENTIDAD (ENTITY)

3.3 Diagrama de Clases

Diagrama de clase: Describe gráficamente las características de las clases de software en una aplicación.

Clase: Es la unidad básica que encapsula información.

Estos diagramas son los más importantes del diseño orientado a objetos, son la “piedra angular” de nuestro diseño. En ellos contiene toda la información de todas las clases y sus relaciones con otras clases, aunque son los más importantes no se llegan a ellos directamente dado que tienen un gran nivel de abstracción dado que contemplan el modelo globalmente sin particularizarse en ningún escenario concreto.

3.4 Diagrama de Clase De Diseño

La representación de atributos y operaciones tiene asociada también una notación específica:

- Atributos y métodos públicos: +
- Atributos y métodos privados: -
- Atributos y métodos protegidos: #
- Atributos y métodos estáticos: _____

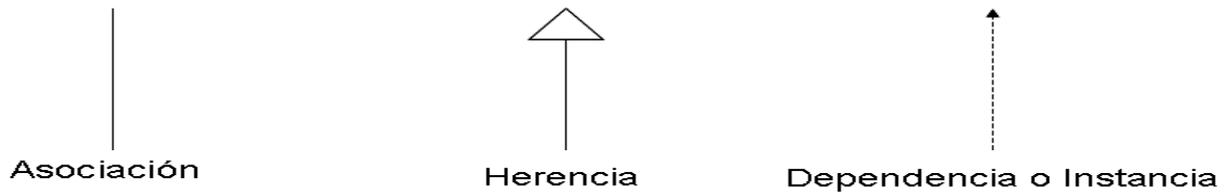


FIG. 3.4 EJEMPLO DE CLASE DE DISEÑO

Asociación:

La asociación permite relacionar clases que colaboran entre sí.

Es una relación entre dos clases que indica alguna conexión entre ellas.

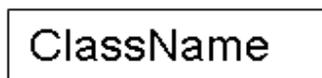


Se representa como una línea entre clases además permite asociar clases que colaboran entre sí.

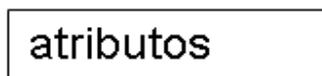


Además de la notación definida, será posible representar a una clase como un rectángulo dividido en tres secciones:

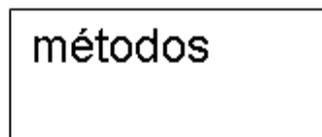
Nombre de la Clase.



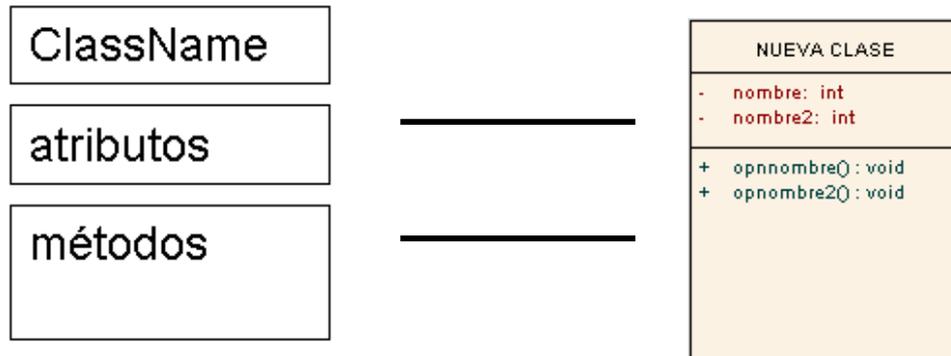
Atributos que caracterizan a la clase



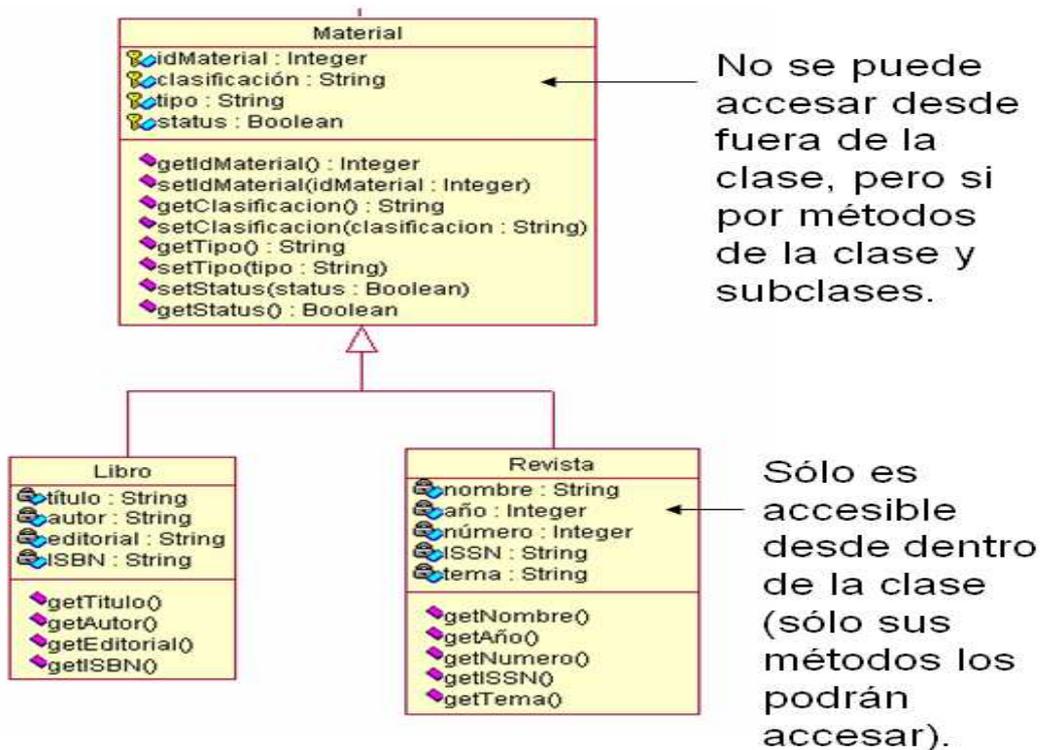
Métodos u operaciones, son la forma como interactúa el objeto con su entorno.



Juntando las 3 secciones se conformaría el siguiente ejemplo:



Herencia: indica que una subclase hereda los métodos y atributos especificados por una superclase, por ende la Subclase además de poseer sus propios métodos y atributos, poseerá las características y atributos visibles de la Super Clase (public y protected).



Dependencia o Instancia: representa una clase que es instanciada (dependiente de otra clase).



Una aplicación que instancia una ventana (la creación del objeto Ventana está condicionado a la instanciación proveniente desde el objeto Aplicación). El objeto Ventana no se almacena dentro de Aplicación.

Visto lo anterior se presenta el diagrama de clases de diseño para el Sistema de Inventario para Fundación UNAM.

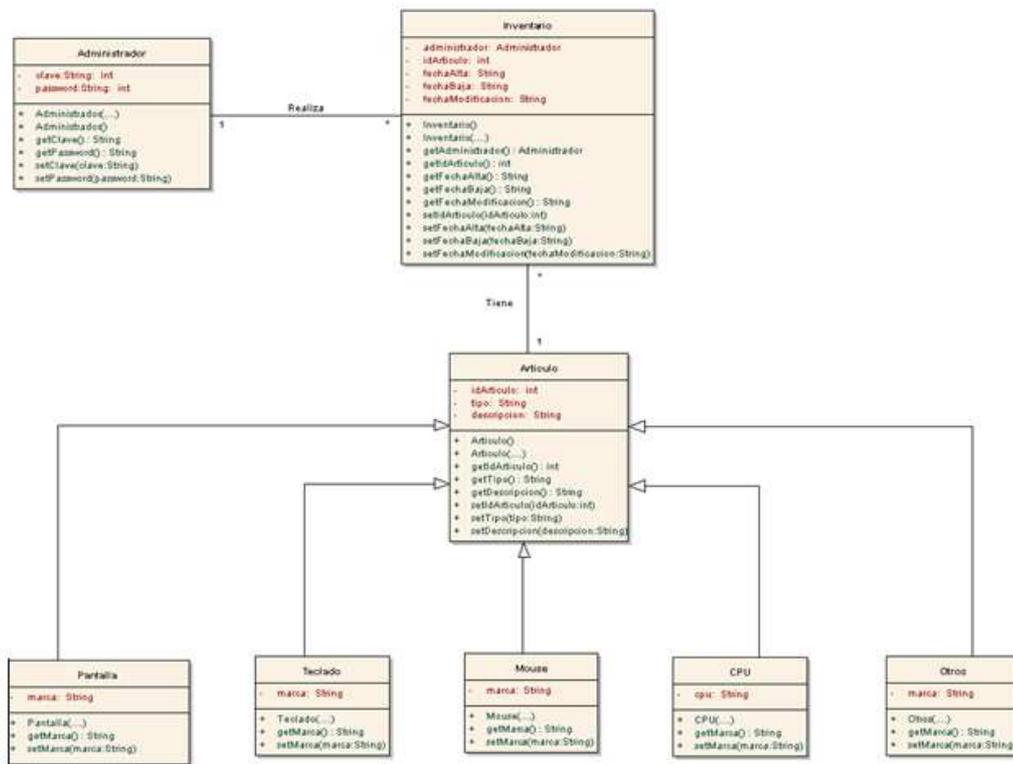


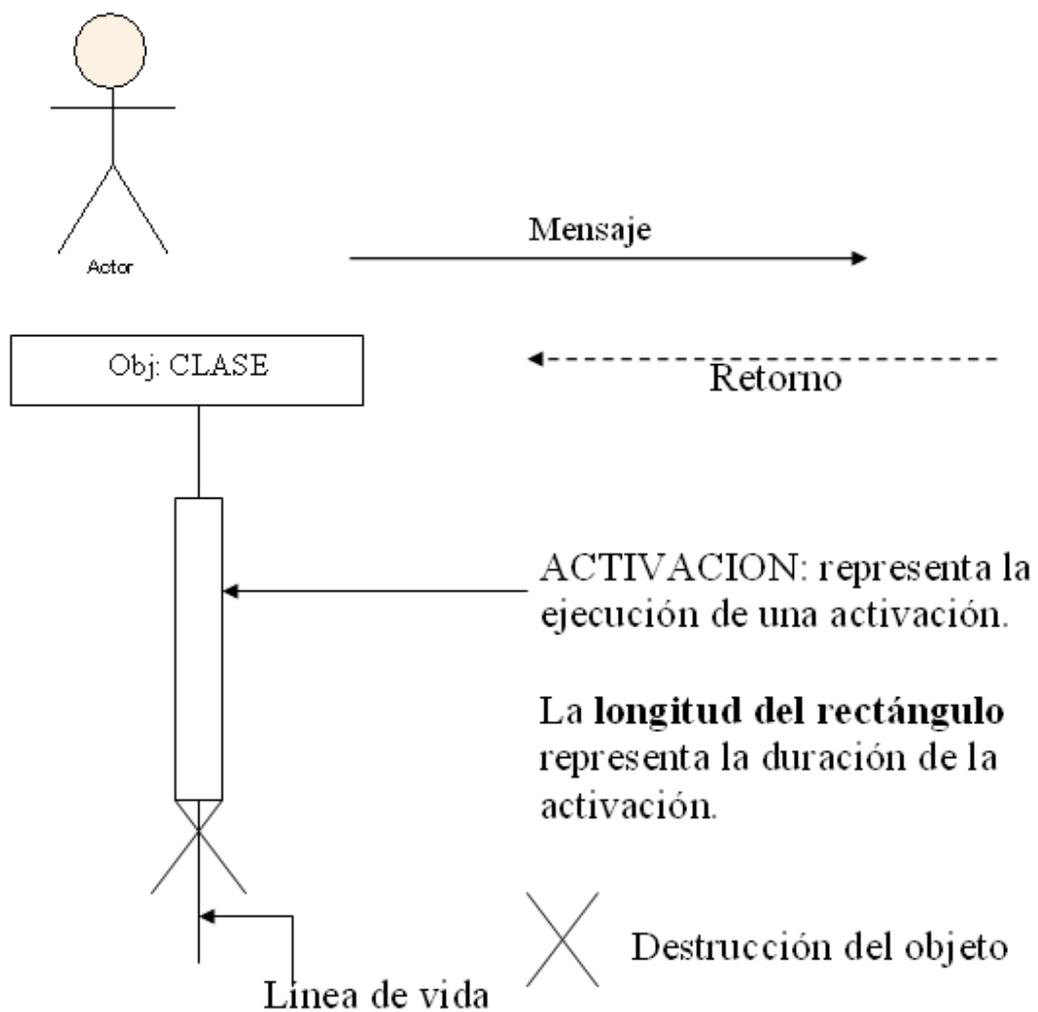
FIG.3.5 DIAGRAMA DE CLASE DE DISEÑO PARA EL SISTEMA DE INVENTARIO FUNDACION UNAM

3.5 Diagrama de Secuencia

Muestra gráficamente los eventos que fluyen de los actores al sistema, además representa la forma en cómo un actor o cliente y las clases, se comunican entre sí en petición a un evento.

Muestra el comportamiento dinámico del sistema orientado al tiempo. El comportamiento del sistema es una descripción de lo que hace, sin explicar la manera en que lo hace.

Los elementos que constituyen a un diagrama de secuencia son los siguientes:



Un diagrama de secuencia destaca la ordenación temporal de los mensajes. Se realizan en una secuencia establecida y ésta toma su tiempo en ir de principio a fin.

Se forma colocando en primer lugar, los objetos o clases que participan en la interacción, en la parte superior del diagrama. A continuación se colocan los mensajes que estos objetos envían y reciben, en orden temporal de arriba hacia abajo. En el diagrama, el tiempo avanza hacia abajo.

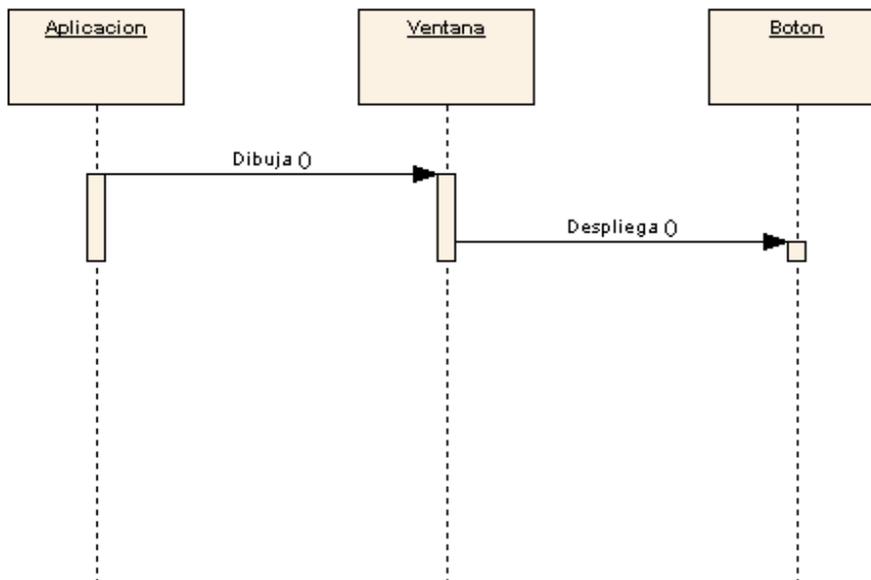
Dimensiones

El Diagrama de secuencia tiene 2 dimensiones:

- Dimensión Horizontal es la disposición de objetos y clases.
- Dimensión Vertical muestra el paso del tiempo.

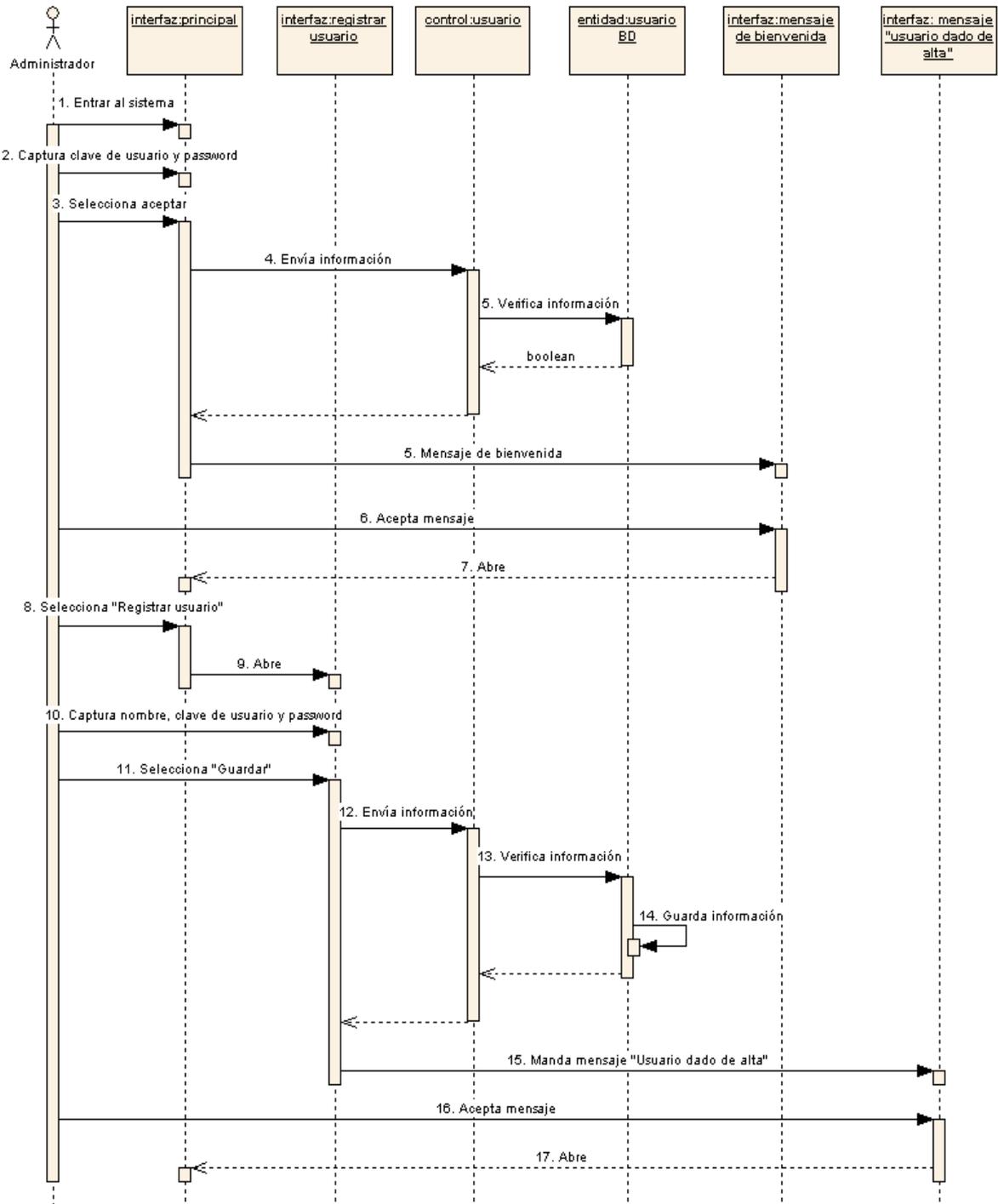
Ejemplo:

Una aplicación posee una ventana y ésta a su vez tiene internamente un botón.

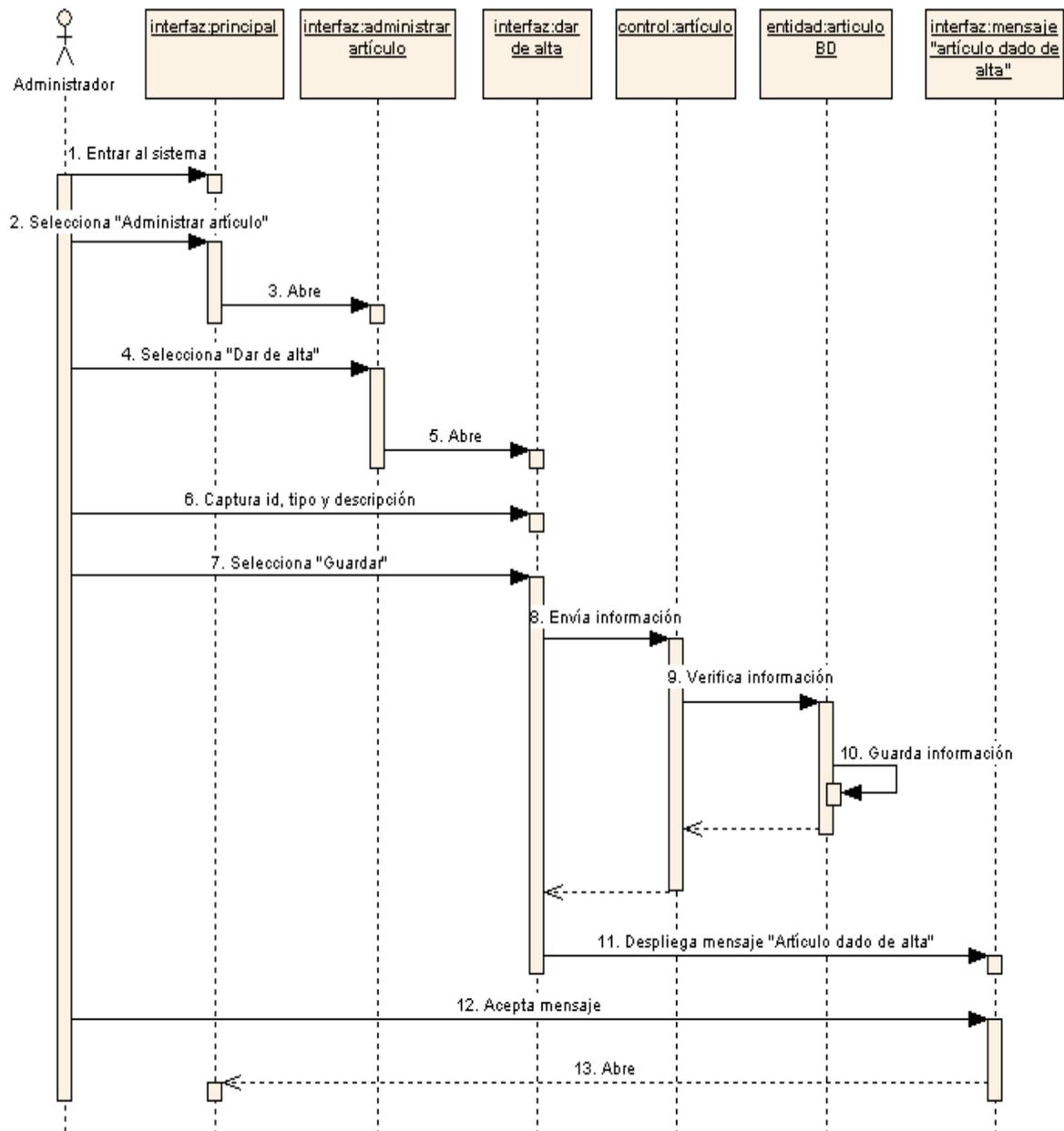


A continuación se presentan los diagramas de secuencia para el sistema de inventario para Fundación UNAM:

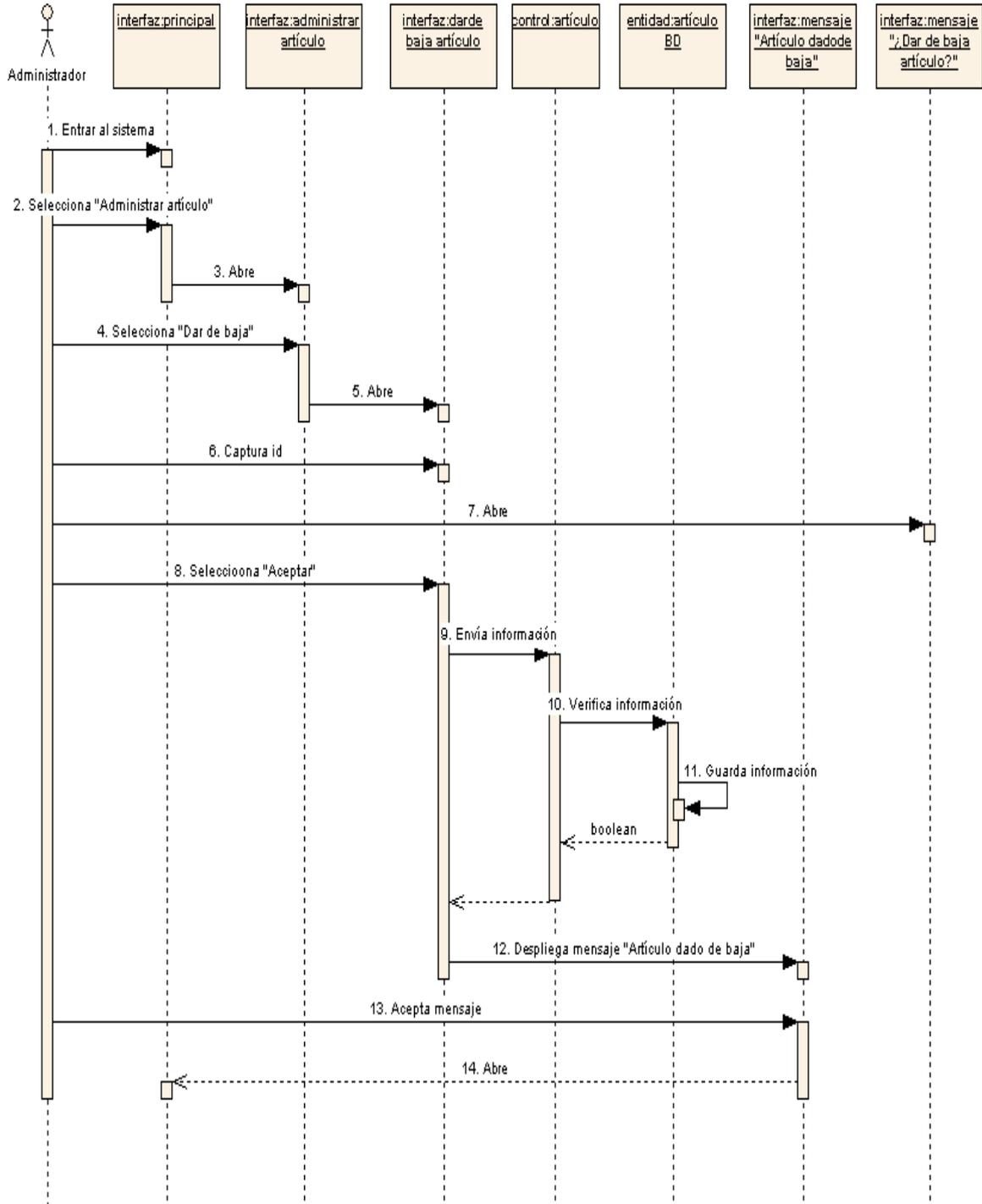
REGISTRAR USUARIO (ADMINISTRADOR):



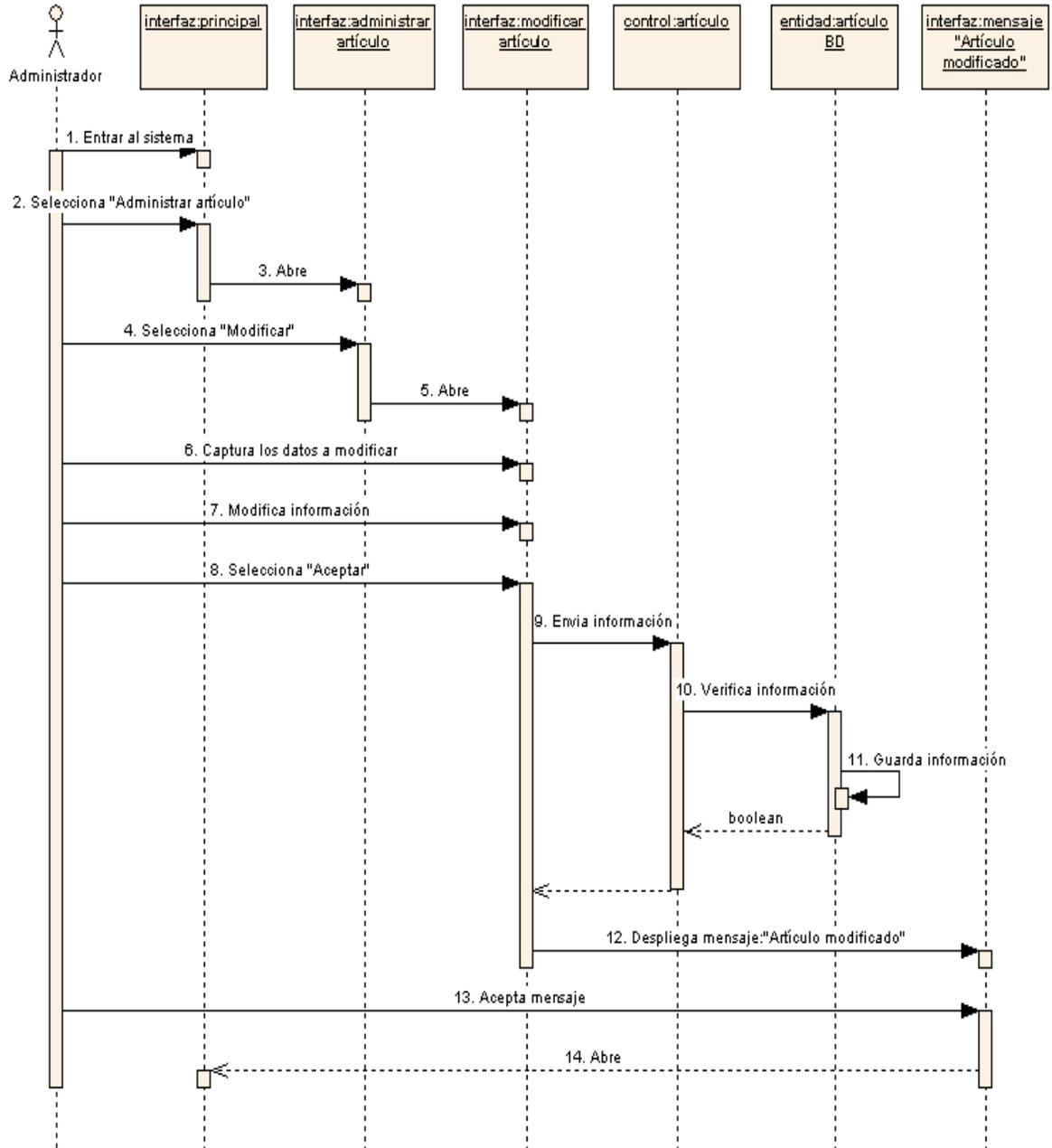
ADMINISTRADOR-DAR DE ALTA ARTÍCULO



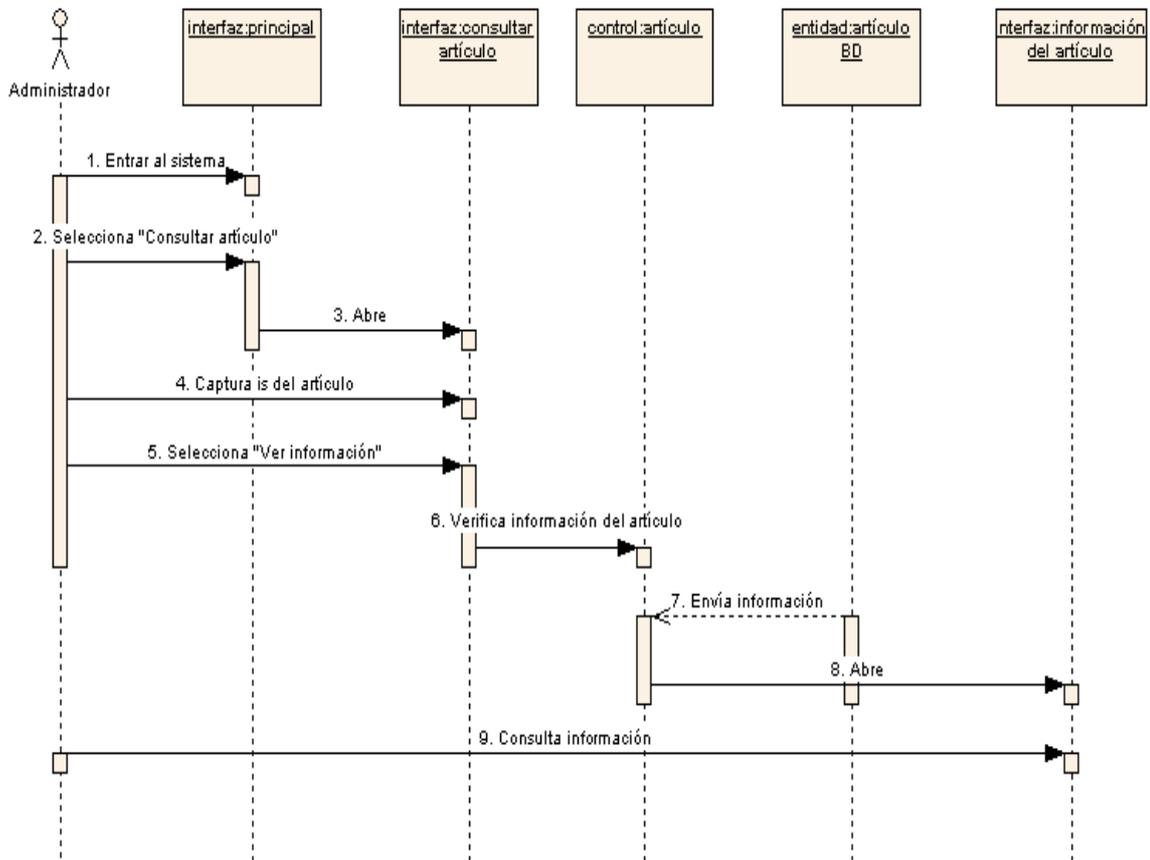
ADMINISTRADOR-DAR DE BAJA ARTÍCULO



ADMINISTRADOR-MODIFICAR ARTÍCULO



ADMINISTRADOR-CONSULTAR ARTÍCULO



CAPÍTULO 4

ANÁLISIS Y DISEÑO DE LA BASE DE DATOS

4.1 Introducción

El análisis, es un conjunto de hechos, principios y reglas clasificadas y dispuestas de manera ordenada, mostrando un plan lógico en la unión de las partes.

La idea central del diseño de bases de datos, es el concepto de dependencia funcional de los datos, esta será una restricción que reduce el número de posibilidades de esquemas relaciones que se pueden utilizar para obtener un adecuado esquema relacional.

4.2 Modelo Funcional

El objetivo de un modelo funcional es el de definir en los términos más simples, lo que el negocio actualmente hace. El modelo funcional nos deberá llevar a un análisis más detallado del sistema, y debe servir para hacer una revisión de las precisiones de los subsiguientes pasos en el análisis, que a saber son: los diagramas entidad-relación y el modelo de flujo de datos.

4.2.1 Arquitectura del Sistema

A continuación se representa el nivel superior de la jerarquía funcional con el propósito fundamental del sistema.

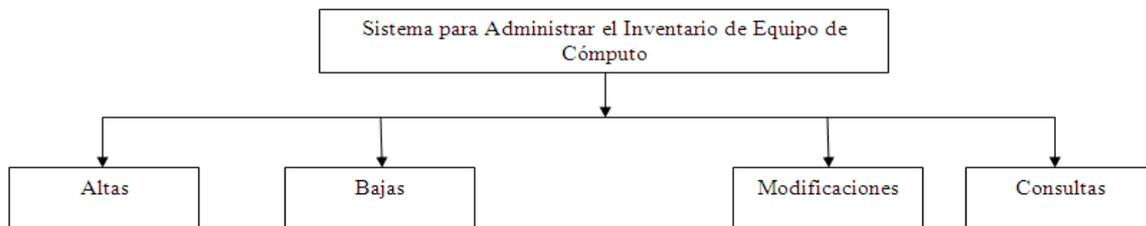


FIG.4.1 SISTEMA PARA ADMINISTRAR EL INVENTARIO DE EQUIPO DE CÓMPUTO EN SU NIVEL MAS ALTO (JERARQUÍA FUNCIONAL)

En las siguientes figuras se han aplicado las técnicas de descomposición funcional del sistema. Es decir, se presentan algunos elementos relevantes de la descomposición funcional para nuestro SISTEMA PARA ADMINISTRAR EL INVENTARIO DE EQUIPO DE CÓMPUTO.

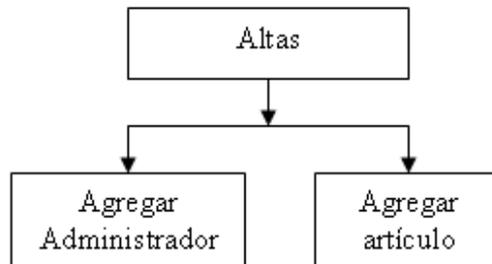


FIG. 4.2 PARTICIÓN DE LA FUNCIÓN ALTAS DEL SISTEMA

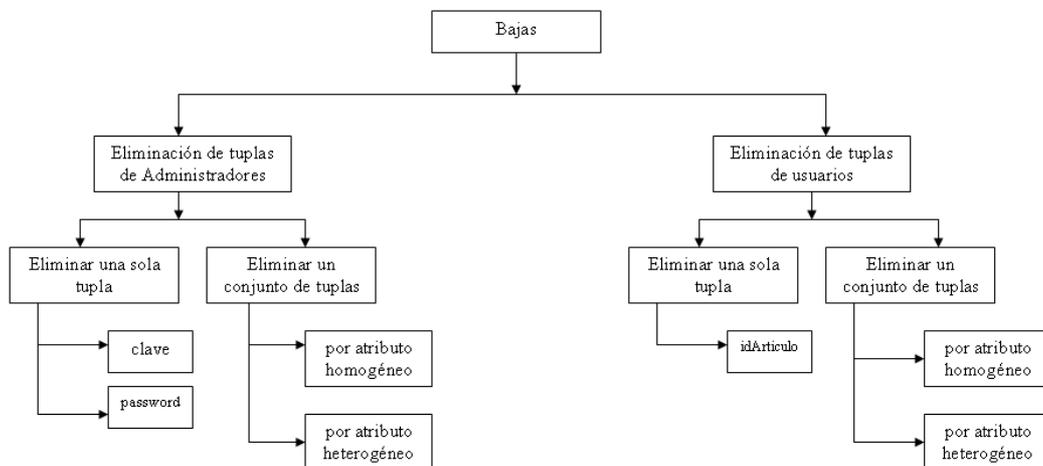


FIG. 4.3 PARTICIÓN DE LA FUNCIÓN BAJAS DEL SISTEMA

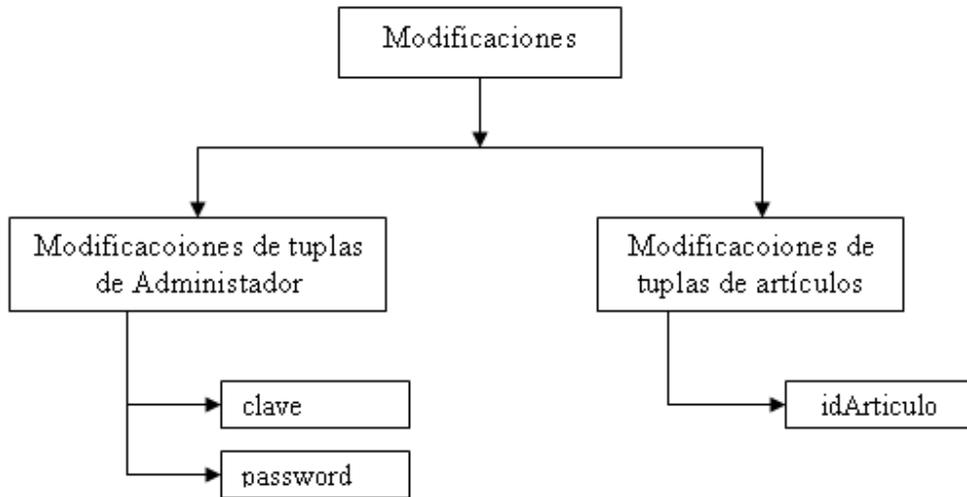


FIG. 4.4 PARTICIÓN DE LA FUNCIÓN MODIFICACIONES DEL SISTEMA

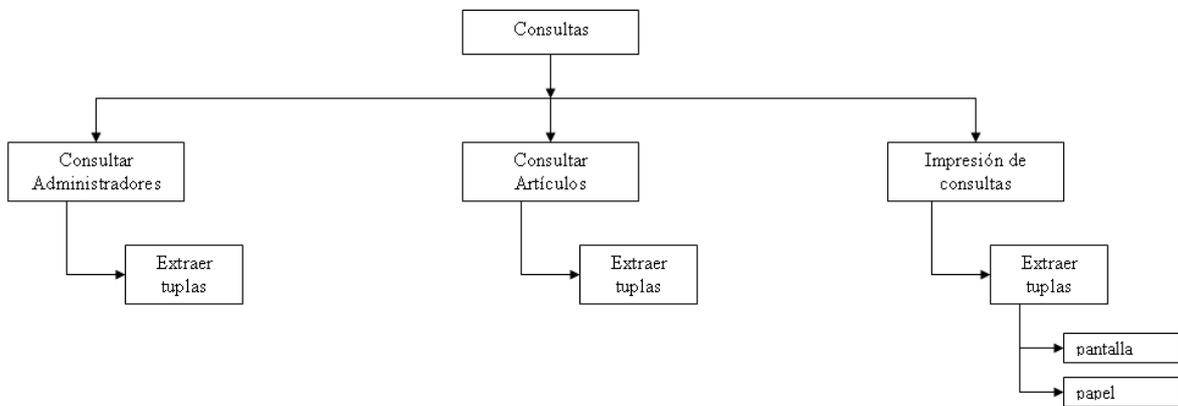


FIG. 4.5 PARTICIÓN DE LA FUNCIÓN CONSULTAS DEL SISTEMA

4.3 Modelo Entidad-Relación (E/R)

Si el modelo funcional es una mitad de la ecuación, el modelo entidad-relación es la otra mitad. El modelo entidad relación es un modelo netamente conceptual. Nos permitirá definir y entender las cosas significativas acerca de cuál información necesita ser conocida o manejada y también las relaciones entre estas cosas, además de, describir, modelar las estructuras de datos y otros conceptos a través de un diagrama como se muestra en la figura 4.1.

Se listan las entidades, las relaciones y sus atributos siguientes de acuerdo al análisis del problema.

Entidades

Una entidad es algo que existe y es distinguible.

- Administrador (Administrador de artículos)
- Inventario (Parte en la cual se da de alta, baja o modifica un artículo)
- Artículo (Son todos los artículos que existen dentro de una Sala de Cómputo)
- Monitor (Artículo de inventario)
- Teclado (Artículo de inventario)
- Mouse (Artículo de inventario)
- CPU (Artículo de inventario)
- Otros (Artículo de inventario)

Relaciones

Una relación entre conjuntos de entidades es simplemente una lista ordenada de conjuntos de entidades, o bien se pueden asociar una relación con una propiedad de las entidades o una acción entre entidades.

INVENTARIO **CONSTA DE** ARTICULOS

ADMINISTRADOR **REALIZA** INVENTARIO

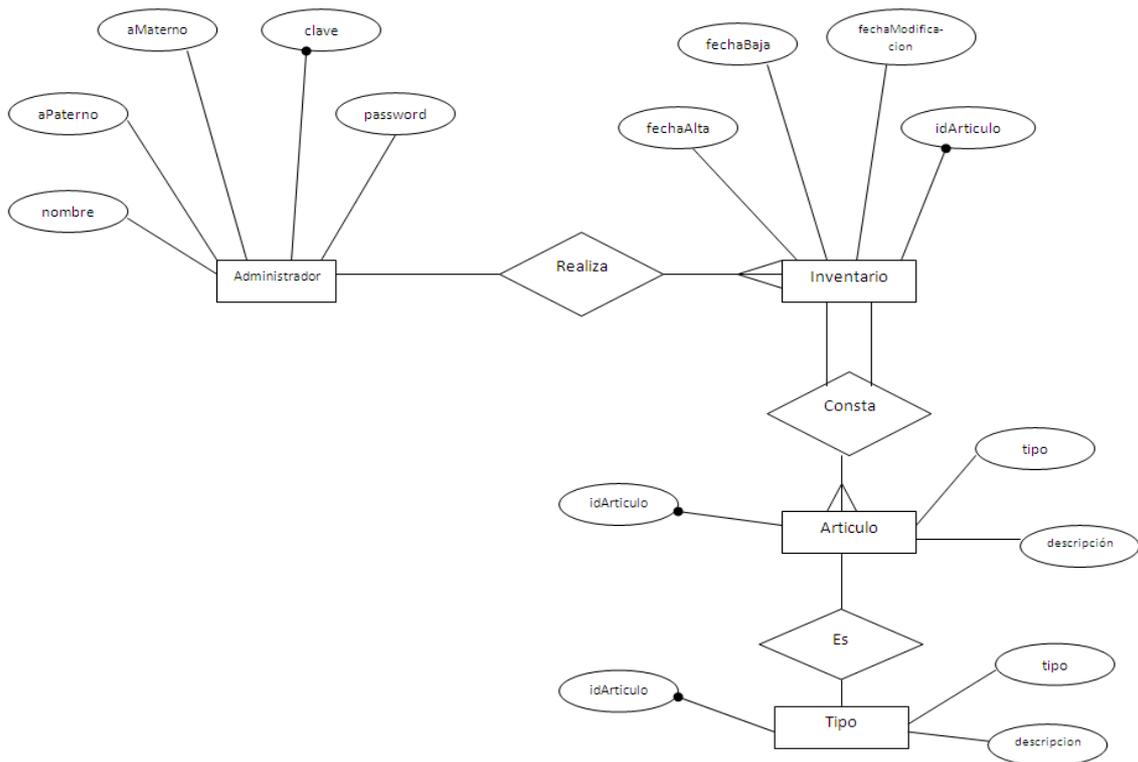
Atributos de las entidades

Las propiedades de las entidades son llamadas **atributos**.

1. ADMINISTRADOR (clave, password, nombre, aPaterno, aMaterno)
2. INVENTARIO (idArtículo, fechaAlta, fechaBaja, fechaModificación)
3. ARTICULO (idArtículo, tipo, descripción)
4. TIPO (idTipo, descripción, marca)

Diagrama E/R

De acuerdo con los datos adscritos anteriormente se bosqueja el siguiente diagrama entidad-relación para el Sistema de Administración de Inventario de Equipo de Cómputo , donde se puede ver las entidades, sus atributos y relaciones, la llave ó clave de cada entidad está marcado con ● ————— .



4.6 DIAGRAMA GENERAL E/R

4.4 Modelo de Flujo de Datos

Una vez que el modelo funcional y el modelo entidad relación han sido determinados, se está preparando entonces a la fase analítica final. Construir el modelo de flujo de datos. Este último encuaderna las funciones del negocio y las titula, permite una representación gráfica de sus interrelaciones, muestra el flujo de entradas y salidas de datos a través del negocio total.

De acuerdo al análisis realizado anteriormente tenemos las siguientes funciones:

- Altas
- Bajas
- Modificaciones
- Consultas

El modelo de flujo de datos inicia con una función principal (Sistema para Administrar el Inventario de Equipo de Cómputo), esta última la encapsula en un grafo que titula con esa actividad principal, y de ahí parten flechas de entrada y salida que representan el flujo de los datos, tal y como se muestra en la figura 4.2.

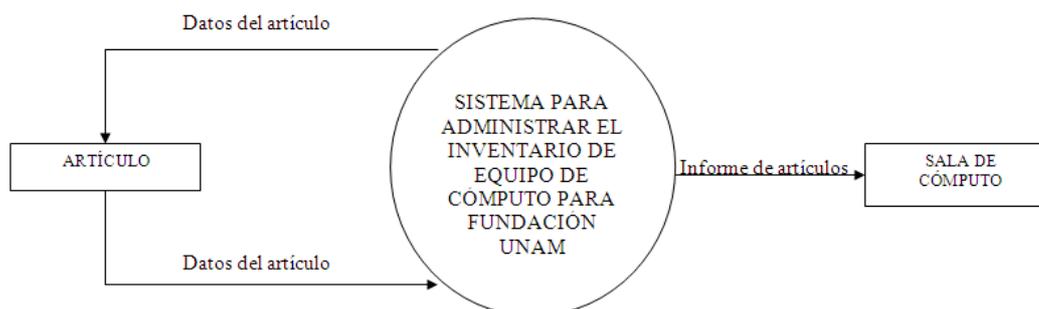


FIG. 4.7 D.F.D NIVEL CERO

Una vez que se tiene el modelo funcional inicial, puede irse refinando tal modelo, pasando a nuevos niveles que proporcionen mayor detalle del sistema D.F.D nivel 1 (figura 4.8).

Como se ve en la figura 4.8, se contarán con módulos que permitan automatizar el almacenamiento de información en la sala de cómputo, esta información consistirá en material (otros) y hardware que se lo otorgue a dicha sala de cómputo, para que finalmente mediante la generación de reportes pueda llevarse a un control de inventario.

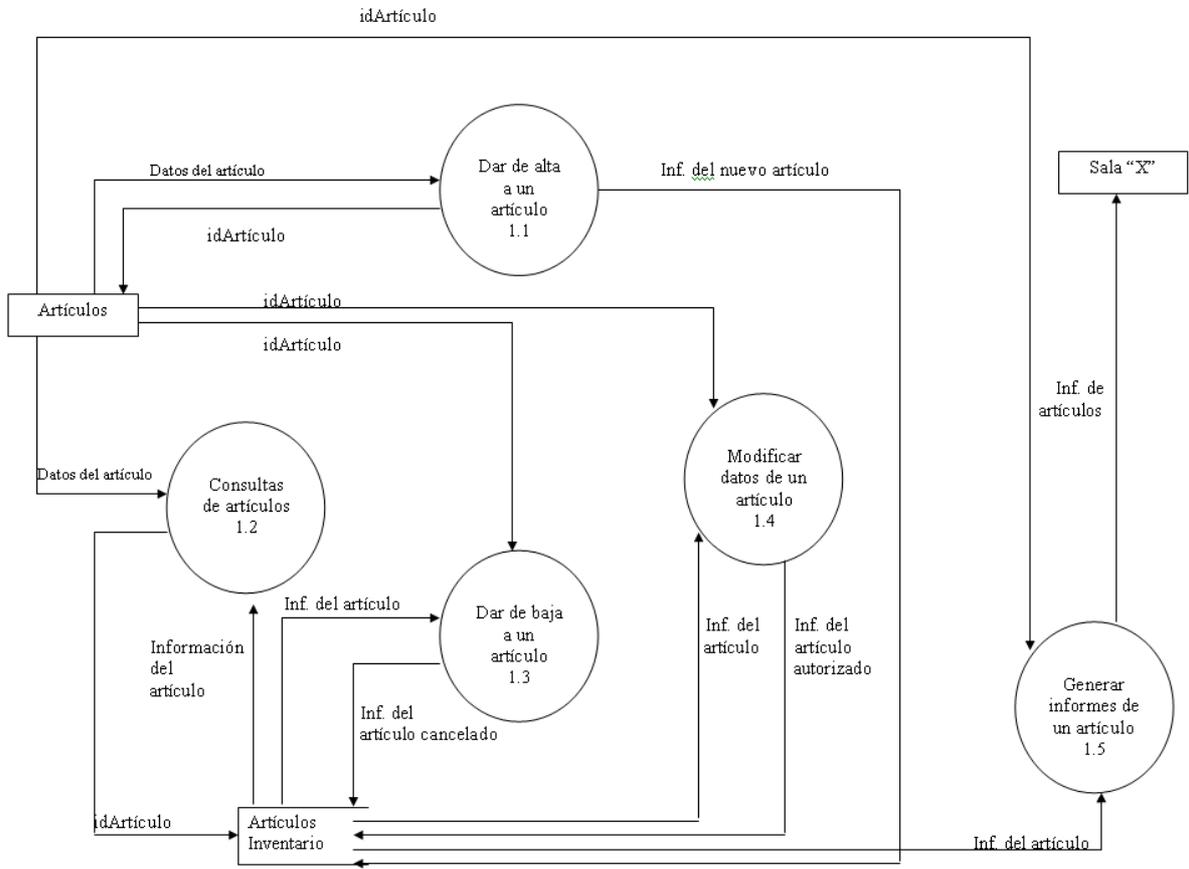


FIG. 4.8 D.F.D DE NIVEL 1

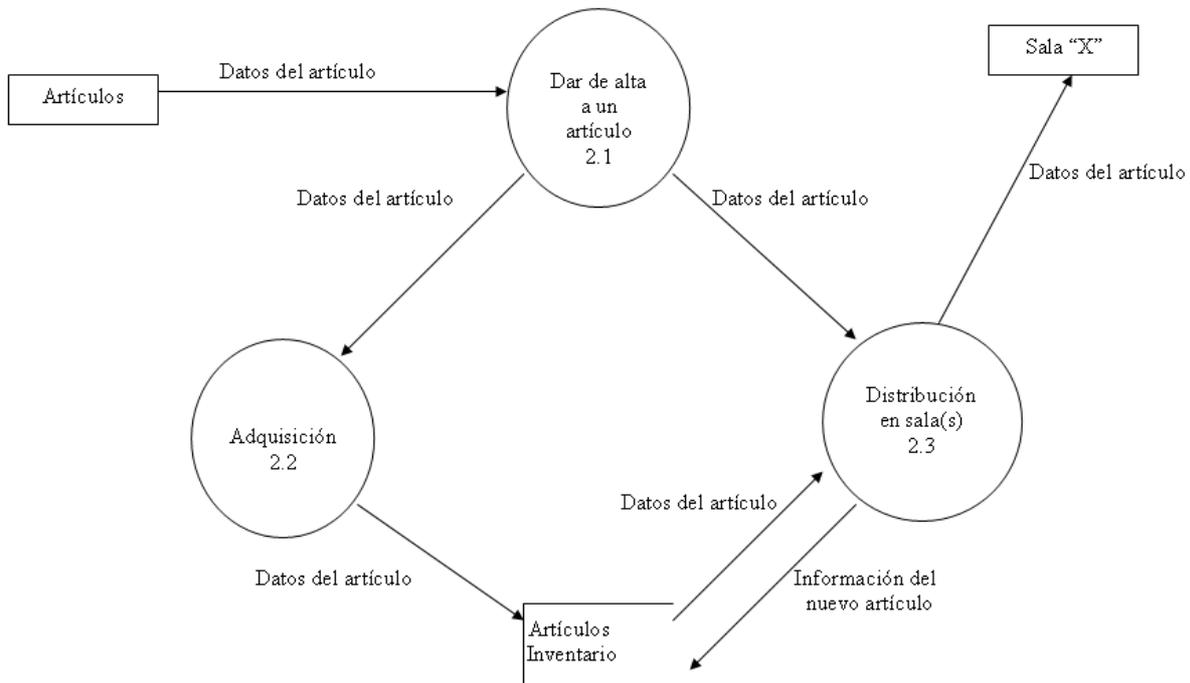


FIG. 4.4 EJEMPLO DFD DE NIVEL 2 PARA EL CASO “DAR DE ALTA A UN ARTÍCULO”

El control de dicho inventario debe de llevar una secuencia lógica, desde las solicitudes emitidas por cada sala de cómputo, el control de las adquisiciones y finalmente mediante reportes confirmar la existencia de dicho material en los lugares de ubicación que se les asigno.

4.5 Diccionario de Datos

La segunda herramienta importante de modelado en proyectos es el diccionario de datos; aunque no tiene la presencia y el atractivo gráfico de los DFD, es crucial. Sin los diccionarios de datos, el modelo de los requerimientos del usuario no puede considerarse completo; todo lo que se tendría es un tipo borrador.

Sin el diccionario de datos el análisis de sistemas se extraviará y el usuario no podrá estar seguro de que entendió los detalles de la aplicación.

El diccionario de datos es un listado organizado de todos los datos pertinentes al sistema, con definiciones precisas y rigurosas para que tanto el usuario como el analista tengan un entendimiento común de todas las altas, eliminaciones o modificaciones de los artículos.

El diccionario de datos define los datos de la siguiente manera:

- Describe el significado de los flujos y almacenes que se muestran en los DFD.
- Describe la composición de agregados de paquetes de datos que se mueven a lo largo de los flujos, por ejemplo, como domicilio de un usuario, puede descomponerse en unidades como ciudad, estado, código postal, etc.
- Describen la composición de los paquetes de datos en los almacenes.
- Especifica los valores y unidades relevantes de piezas elementales de información en los flujos de datos y en los almacenes de datos.
- Describe los detalles de las relaciones entre almacenes que se enfatizan en un diagrama de entidad-relación.

El diccionario de datos lo crea el analista durante el desarrollo del modelo del sistema, pero el usuario debe ser capaz de leerlo y entenderlo para poder verificar el modelo.

Es difícil que el usuario lea con detalle todo el diccionario de datos para asegurarse de que está correcto, pero si puede verificar que es correcto en conjunto con el DFD o la especificación del proceso que se esté leyendo.

DICCIONARIO DE DATOS

NOMBRE DE LA ENTIDAD	NOMBRE DEL ATRIBUTO	DESCRIPCIÓN	TAMAÑO
Administrador	clave	*Alfanumerico* [A-Z a-z 0 1 2 3 ...9]	15
	password	*Alfanumerico* [A-Z a-z 0 1 2 3 ...9]	15
	nombre	*Cadena de Caracteres* [A-Z a-z]	15
	aPaterno	*Cadena de Caracteres* [A-Z a-z]	20
	aMaterno	*Cadena de Caracteres* [A-Z a-z]	20
Inventario	idArtículo	*Digitos* [0 1 2 3 ...9]	5
	fechaAlta	*Alfanumerico* [A-Z a-z 0 1 2 3 ...9]	14
	fechaBaja	*Alfanumerico* [A-Z a-z 0 1 2 3 ...9]	14
	fechaModificación	*Alfanumerico* [A-Z a-z 0 1 2 3 ...9]	14
Articulo	idArtículo	*Digitos* [0 1 2 3 ...9]	5
	tipo	*Cadena de Caracteres* [A-Z a-z]	15
	descripción	*Cadena de Caracteres* [A-Z a-z]	1000
Tipo	idTipo	**Digitos* [0 1 2 3 ...9]	5
	descripción	*Cadena de Caracteres* [A-Z a-z]	1000
	marca	*Cadena de Caracteres* [A-Z a-z]	20

4.6 Modelo Relacional

Aplicando los términos que definen la estructura de datos relacionales, en los siguientes apartados se muestra cómo quedará cada una de las tablas.

Tabla Administrador

Administrador (**clave**, **password**, nombre, aPaterno, aMaterno)

TABLA ADMINISTRADOR				
clave	password	nombre	aPaterno	aMaterno

Donde:

clave es la llave primaria

Tabla Inventario

Inventario (**idArtículo**, fechaAlta, fechaBaja, fechaModificación)

TABLA INVENTARIO			
idArtículo	fechaAlta	fechaBaja	fechaModificación

Donde:

idArtículo es la llave primaria

Tabla Artículo

Artículo (**idArtículo**, tipo, descripción, marca)

TABLA ARTICULO		
idArtículo	tipo	descripción

Donde:

idArtículo es la llave primaria

Tabla Tipo

Tipo (**idTipo**, descripción, marca)

TABLA TIPO		
idTipo	descripción	marca

Donde:

idTipo es la llave primaria

4.7 Normalización

De lo anterior, el modelo relacional quedaría de esta manera:

Administradores (**clave**, password, nombre, aPaterno, aMaterno)

Inventario (**idArticulo**, tipo, descripción)

Articulo (**idArticulo**, tipo, descripción, marca)

Tipo (**idTipo**, descripción, marca)

Administrador (**clave**, password, nombre, aPaterno, aMaterno)

1FN: Sus atributos son atómicos

2FN: Todos sus atributos “No llave” dependen completamente de la llave

3FN: Se encuentra en 2FN y los atributos no llave son mutuamente independientes

Inventario (**idArtículo**, fechaAlta, fechaBaja, fechaModificación)

1FN: Sus atributos son atómicos

2FN: Todos sus atributos “No llave” dependen completamente de la llave

3FN: Se encuentra en 2FN y los atributos no llave son mutuamente independientes

Articulo (**idArticulo**, tipo, descripción)

1FN: Sus atributos son atómicos

2FN: Todos sus atributos “No llave” dependen completamente de la llave

3FN: Se encuentra en 2FN y los atributos no llave son mutuamente independientes

Articulo (**idTipo**, descripción, marca)

1FN: Sus atributos son atómicos

2FN: Todos sus atributos “No llave” dependen completamente de la llave

3FN: Se encuentra en 2FN y los atributos no llave son mutuamente independientes

4.8 Diseño de la Base de Datos

A la hora de seleccionar la base de datos más adecuada es muy importante realizar un estudio que nos permita tener una idea aproximada del volumen que va alcanzar no solo cuando se implemente, sino a lo largo de por lo menos los cinco años siguientes. Otro de los puntos importantes a tener en cuenta al seleccionar una base de datos es el rendimiento, obviamente para que el rendimiento sea aceptable no basta sólo con seleccionar el mejor y más rápido de los SGBDR, sino que también es necesario que todas las infraestructuras que haya alrededor estén a la altura.

En el caso de que el sistema este en Red, el servidor debe estar bien dimensionado para realizar esta tarea, siendo recomendable en todos los paquetes analizados que tengan una buena cantidad de memoria RAM y si es posible dos procesadores.

La elección del motor de Bases de Datos a utilizar no es trivial; es una de las primeras decisiones más importantes que se toman. Todos tienen sus ventajas y sus desventajas; por ello se debe hallar un punto medio entre las posibilidades que nos brindan y su facilidad de implementación. En este caso el utilizaremos el motor de Bases de Datos Data Base Oracle 10g y como cliente SQL Developer.

¿Por qué Data Base Oracle 10g? Data Base Oracle 10g es muy rápido, confiable, robusto y fácil de usar tanto para volúmenes de datos grandes como pequeños. Además tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios. Sin embargo bajo constante desarrollo, MySQL hoy en día ofrece un rico y muy útil conjunto de funciones. La conectividad, velocidad y seguridad hace de MySQL altamente conveniente para acceder a bases de datos en Internet.

¿Por qué el cliente SQL Developer? SQL Developer es una interesante aplicación gráfica programada en Java que permite visualizar/manejar fácilmente la estructura de una base de datos JDBC. Puede ejecutar comandos sql, ver la estructura de la base de datos en formato árbol, etc.

4.9 Crear Tablas

En el texto siguiente, se muestra como se crean las tablas del sistema con sus respectivas llaves primarias con lenguaje SQL, tomadas del archivo System.tab.

```
/*=====*\n|| Archivo: System.tab ||\n|| Descripción: Generación de las tablas del Sistema para Administrar el equipo de Cómputo ||\n|| Autor: Sergio Andres Hernández González ||\n\\*-----*/\n\nCREATE TABLE ADMINISTRADOR\n(CLAVE VARCHAR (15) NOT NULL,\nPASSWORD VARCHAR (15),\nNOMBRE VARCHAR(15),\nAPATERNO VARCHAR (20),\nAMATERNO VARCHAR (20),\nCONSTRAINT ADMINISTRADOR_PK PRIMARY KEY (CLAVE)\n);
```

FIG.4.5 CREA LA TABLA DE ADMINISTRADOR

```

/*=====*\
|| Archivo: System.tab ||
|| Descripción: Generación de las tablas del Sistema para Administrar el equipo de Cómputo ||
|| Autor: Sergio Andres Hernández González ||
\*=====*/

CREATE TABLE INVENTARIO
(IDARTICULO INT (5) NOT NULL,
FECHAALTA VARCHAR (15),
FECHABAJA VARCHAR (15),
FECHAMODIFICACION VARCHAR (20),
CONSTRAINT INVENTARIO_PK PRIMARY KEY (IDARTICULO)
);

```

FIG.4.6 CREA LA TABLA INVENTARIO

```

/*=====*\
|| Archivo: System.tab ||
|| Descripción: Generación de las tablas del Sistema para Administrar el equipo de Cómputo ||
|| Autor: Sergio Andres Hernández González ||
\*=====*/

CREATE TABLE ARTICULO
(IDARTICULO INT (5) NOT NULL,
TIPO VARCHAR (15),
DESCRIPCION VARCHAR (15),
CONSTRAINT ARTICULO_PK PRIMARY KEY (IDARTICULO)
);

```

FIG.4.7 CREA LA TABLA ARTÍCULO

```

/*=====*\
|| Archivo: System.tab ||
|| Descripción: Generación de las tablas del Sistema para Administrar el equipo de Cómputo ||
|| Autor: Sergio Andres Hernández González ||
\*=====*/

CREATE TABLE TIPO
(IDTIPO INT (5) NOT NULL,
DESCRIPCION VARCHAR (15),
MARCA VARCHAR (15),
CONSTRAINT TIPO_PK PRIMARY KEY (IDTIPO)
);

```

FIG.4.8 CREA TABLA TIPO

El diseño de la Base de Datos fue una de parte importante en el desarrollo del sistema, por que se contempla el comportamiento correcto o incorrecto del sistema.

La generación de los scripts de la Base de Datos es muy importante ya que gracias a ellos se podrán guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente.

4.10 Etapas y Tiempo para el desarrollo del Sistema (Diagramas de Gantt y Pert).

Para realizar un mejor desarrollo del sistema se utilizan herramientas gráficas para mostrar el tiempo de dedicación previsto para las diferentes tareas o actividades a realizar, las graficas a utilizar son las siguientes:

DIAGRAMA DE GANTT

El Diagrama de Gantt de Henry Laurence Gantt proporciona una descripción gráfica y la programación temporal de todas las actividades, elementos y dependencias de un proyecto o programa. Es un tipo especial de diagrama de barra horizontal que es muy común que se utilice en la gerencia de proyectos para representar las fases y las actividades de una estructura no interrumpida del trabajo de proyecto.

DIAGRAMA DE PERT

El Diagrama de Pert es la técnica de grafos, mediante ella se esquematiza la relación del proyecto en diferentes situaciones y actividades. En el grafo se representan las actividades con flechas y las situaciones en nudos; a cada actividad se le asigna su tiempo de duración y se establecen las relaciones entre actividades y situaciones mediante una secuenciación.

Para la realización del grafo es importante recordar lo siguiente:

- El nodo 1 representa la situación en la que aún no se ha llevado a cabo ninguna actividad.
- Los nodos (representados mediante círculos) se enumeran de forma correlativa conforme se van realizando las actividades.
- Las actividades (representadas mediante flechas) se identifican con letras mayúsculas, y bajo ellas se anotan los días de duración.

A continuación se presentan los diagramas de tiempo, diagrama de Gantt y diagrama de Pert, para desarrollar el sistema de inventario:

ID	TAREA	PRECEDENCIA	FECHA INICIO	FECHA FINAL	DURACION
A	Entrevista	...	10-Jul-08	16-Jul-08	6 días
B	Identificación del problema	...	17-Jul-08	31-Jul-08	16 días
C	Determinación de los requisitos	A	01-Ago-08	12-Ago-08	12 días
D	Analizar las necesidades del sistema	B	13-Ago-08	22-Ago-08	9 días
E	Preparación y propuestas	C	23-Ago-08	05-Sep-08	13 días
F	Diseño del sistema	D	06-Sep-08	02-Oct-08	26 días
G	Desarrollo del sistema	E	03-Oct-08	07-Dic-08	65 días
H	Desarrollo de pruebas y correcciones al sistema	F	08-Dic-08	28-Ene-09	51 días
I	Implantación del sistema	G,H	28-Ene-09	03-Feb-09	6 días

Fig. 4.9 Diagrama de tiempo

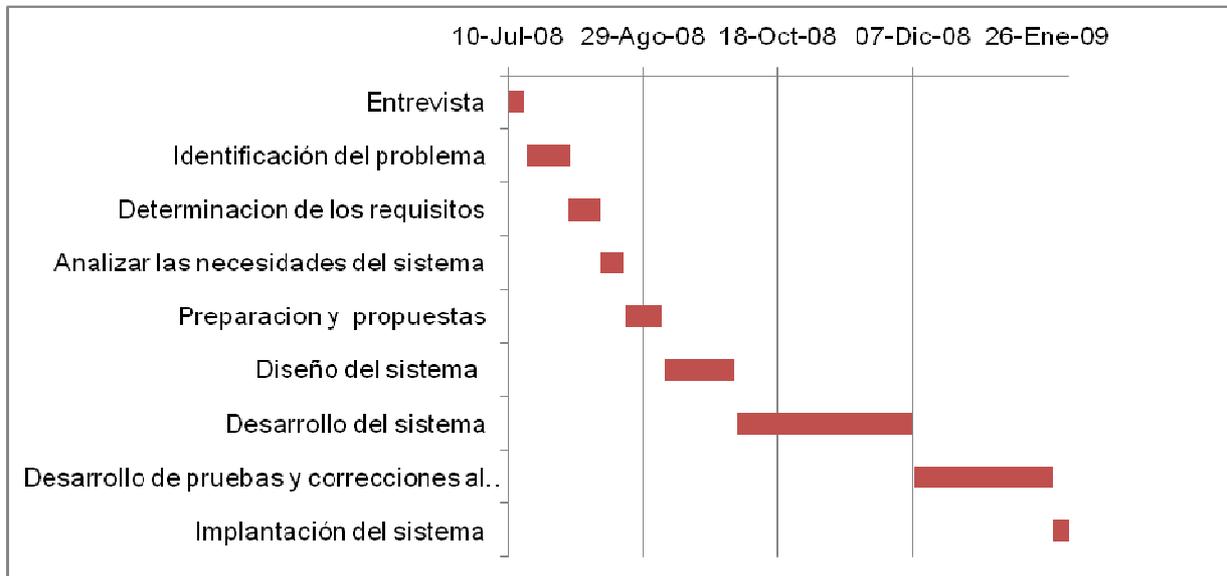


Fig. 5.0 Diagrama de Gantt

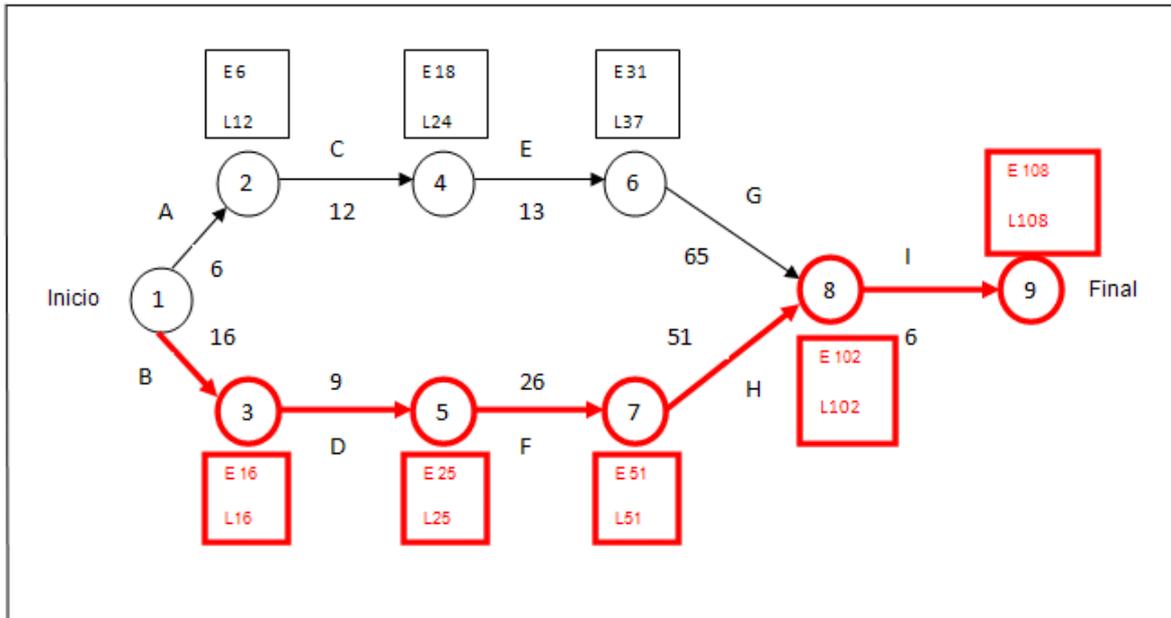


Fig.5.1 Diagrama Pert (camino crítico)

El camino crítico es el que tiene una mayor duración entre los nodos inicial y final, coincide con la duración mínima del proyecto. Está formado por las situaciones en la que el tiempo "early" y "last" son iguales (situaciones críticas), las actividades que unen estos nodos son las actividades críticas, aquellas que no admiten retraso en su ejecución ya que esto implicaría un retraso en su ejecución ya que esto implicaría un retraso del proyecto en general. Por este motivo, el camino crítico muestra parte del proyecto que hay que vigilar con mayor atención, puesto que es esta parte del proyecto donde pueden aparecer problemas de retraso en la realización del proyecto planificado.

CAPÍTULO 5

IMPLANTACIÓN DEL SISTEMA

5.1 Introducción

Como resultado de un análisis y diseño previo procederemos a desarrollar la etapa de IMPLANTACIÓN descrita en este capítulo.

El objetivo de este capítulo es:

- Breve explicación acerca de java así como el framework Struts para diseñar dicho sistema.
- Implantar un sistema que cumpla con todas las funciones descritas en las especificaciones.
- Utilizar la Base de Datos diseñada en el (capítulo 4), en la IMPLANTACIÓN del sistema.
- Especificar las herramientas utilizadas para el desarrollo del sistema.

Para la creación de la Base de Datos del Sistema se utilizó el gestor de bases de datos MySQL y para su aplicación, acceso y manipulación Java SUN (JDK 1.5) y el uso del framework Struts.

5.2 Breve explicación

JAVA

Java se creó como parte de un proyecto de investigación para el desarrollo de software avanzado para una amplia variedad de dispositivos de red y sistemas embebidos. La meta era diseñar una plataforma operativa sencilla, fiable, portable, distribuida y de tiempo real. Cuando se inició el proyecto, C++ era el lenguaje del momento. Pero a lo largo del tiempo, las dificultades encontradas con C++ crecieron hasta el punto en que se pensó que los problemas podrían resolverse mejor creando una plataforma de lenguaje completamente nueva. Se extrajeron decisiones de diseño y arquitectura de una amplia variedad de lenguajes como Eiffel, SmallTalk, Objective C y Cedar/Mesa. El resultado es un lenguaje que se ha mostrado ideal para desarrollar aplicaciones de usuario final seguras, distribuidas y basadas en red en un amplio rango de entornos desde los dispositivos de red embebidos hasta los sistemas de sobremesa e Internet.

Java fue diseñado para ser:

- **Sencillo, orientado a objetos y familiar:** Sencillo, para que no requiera grandes esfuerzos de entrenamiento para los desarrolladores. Orientado a objetos, porque la tecnología de objetos se considera madura y es el enfoque más adecuado

para las necesidades de los sistemas distribuidos y/o cliente/servidor. Familiar, porque aunque se rechazó C++, se mantuvo Java lo más parecido posible a C++, eliminando sus complejidades innecesarias, para facilitar la migración al nuevo lenguaje.

- **Robusto y seguro:** Robusto, simplificando la gestión de memoria y eliminando las

complejidades de la gestión explícita de punteros y aritmética de punteros del C. Seguro para que pueda operar en un entorno de red.

- **Independiente de la arquitectura y portable:** Java está diseñado para soportar aplicaciones que serán instaladas en un entorno de red heterogéneo, con hardware y sistemas operativos diversos. Para hacer esto posible el compilador Java genera 'bytecodes', un formato de código independiente de la plataforma diseñado para transportar código eficientemente a través de múltiples plataformas de hardware y software. Es además portable en el sentido de que es rigurosamente el mismo lenguaje en todas las plataformas. El 'bytecode' es traducido a código máquina y ejecutado por la Java Virtual Machine, que es la implementación Java para cada plataforma hardware-software concreta.

- **Alto rendimiento:** A pesar de ser interpretado, Java tiene en cuenta el rendimiento, y particularmente en las últimas versiones dispone de diversas herramientas para su optimización. Cuando se necesitan capacidades de proceso intensivas, pueden usarse llamadas a código nativo.

- **Interpretado, multi-hilo y dinámico:** El intérprete Java puede ejecutar bytecode en cualquier máquina que disponga de una Máquina Virtual Java (JVM). Además Java incorpora capacidades avanzadas de ejecución multi-hilo (ejecución simultánea de más de un flujo de programa) y proporciona mecanismos de carga dinámica de clases en tiempo de ejecución.

STRUTS

Struts Framework originalmente fue creado por Craig R. McClanahan y donado al proyecto Jakarta de Apache Software Foundation (ASF) en el año 2000. En Junio de 2001 se libera la primera versión la cual es la 1.0, desde entonces muchos desarrolladores han participado en la mejora continua de este Framework.

¿Para qué sirve Struts Framework?

Apache Struts es un Framework Open-Source que sirve para la creación de aplicaciones Web.

Algunas de las características más importantes de Struts Framework son las siguientes:

- Una arquitectura general basada en los principios de diseño de

Modelo-Vista-Controlador (MVC) en el que todas las peticiones son procesadas por el controlador que realiza todo el control y despacha las peticiones a los componentes de la aplicación apropiados, basándose en los identificadores lógicos que reducen el acoplamiento entre capas.

- Capacidades de manejo de formularios, como el JavaBean ActionForm que representa el estado del lado del servidor de los campos de entrada de un formulario, y un marco de trabajo de validación que externaliza la configuración de un conjunto de chequeos de exactitud que se aplican a los valores de los campos de entrada, además implementa estos chequeos

tanto en el lado del cliente como en el lado del servidor

- El marco de trabajo Tiles para el control de la distribución de componentes, que soporta la creación de sofisticadas plantillas que se pueden reutilizar entre varias páginas, esto permite una fácil modificación del aspecto y comportamiento general de toda la aplicación.
- Los Taglibs son un conjunto de etiquetas JSP personalizadas que simplifican el proceso de crear las etiquetas HTML de la aplicación para la capa "Vista", y que trabaja conjuntamente con las capacidades de manejo de formularios y la arquitectura general del controlador.

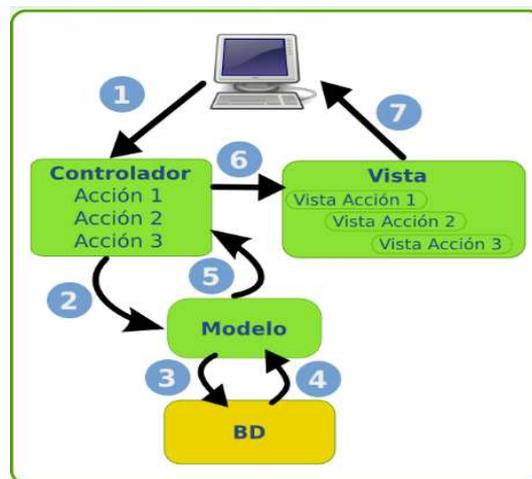


FIG. 5.1 Modelo-vista-Controlador (MVC)

5.3 Descripción del Sistema

- El Sistema creará una aplicación con interfaz gráfica que permita la Administración del Inventario de Equipo de Cómputo.
- Dicha aplicación facilitará el acceso a la información de la base de datos permitiendo visualizar, editar y actualizar a través de la misma.
- Utilizará la información de la base de datos del sistema System.tab creada por MySQL.

5.4 Diseño de las Interfaces

En esta parte de la tesina solo se presentaran los códigos y pantallas más importantes del sistema de inventario.

5.4.1 Interfaz Inicio del Sistema

Para entrar al sistema es necesario tener un nombre de usuario y una contraseña.

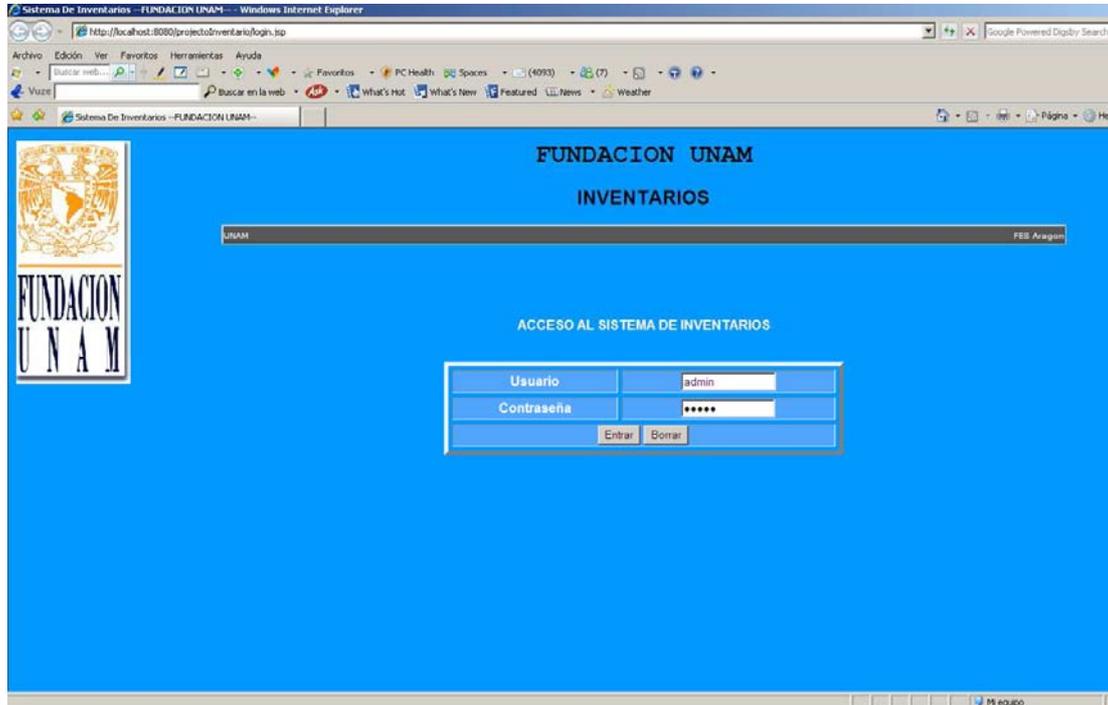


FIG. 5.2 Pantalla de acceso

El código JSP para la pantalla de acceso es el siguiente:

```
<!-- taglib uri="http://jakarta.apache.org/struts/tags-bean" prefix="bean" -->
<!-- taglib uri="http://jakarta.apache.org/struts/tags-html" prefix="html" -->
<!-- taglib uri="http://jakarta.apache.org/struts/tags-logic" prefix="logic" -->

<html:html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>: SISTEMA DE INVENTARIO : FUNDACION UNAM</title>
</head>
<body style="background-color: 0099FF">

<center><h1>FUNDACION UNAM</center></h1>
<center><h2> INVENTARIOS</center></h2>
<table class="tdborder" width="740" border="1" bordercolor="white" cellpadding="0" cellspacing="0">
<tr>
<td align="right" bgcolor="#857575" bordercolorlight="#333333" bordercolordark="#666666">
<table align="right">
<tr>
<td align="left">
<table align="left">
<tr>
<td align="right" colspan="2" style="border: none; padding: 0 10px 0 0;">
<strong>ACCESO AL SISTEMA DE INVENTARIOS</strong></td>
</tr>
<tr>
<td colspan="2" style="border: none; padding: 0 10px 0 0;">
<form action="" method="post" name="Form1">
<table border="5" align="center" cellpadding="6" cellspacing="0" style="width: 364px">
<tr>
<td align="center" colspan="2" style="border: none; padding: 0 10px 0 0;">
<table align="center">
<tr>
<td align="center" colspan="2" style="border: none; padding: 0 10px 0 0;">
<input type="text" maxlength="16" size="16" name="Usuario"/>
</td>
</tr>
<tr>
<td align="center" colspan="2" style="border: none; padding: 0 10px 0 0;">
<input type="password" maxlength="16" size="16" name="Contraseña"/>
</td>
</tr>
</table>
</tr>
<tr>
<td align="center" colspan="2" style="border: none; padding: 0 10px 0 0;">
<input type="button" value="Entrar" />
<input type="button" value="Borrar" />
</td>
</tr>
</table>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>
```

5.4.2 Interfaz Menú Principal del Sistema

Una vez que el usuario este dentro del Sistema, este muestra un menú principal, donde muestra los módulos principales para facilitar con ello la elección de cada una de ellos. Estos módulos son los siguientes:

- Objetivo
- Alta Artículo
- Editar Artículo
- Eliminar Artículo
- Consultar Artículo
- Consultar todos los Artículos

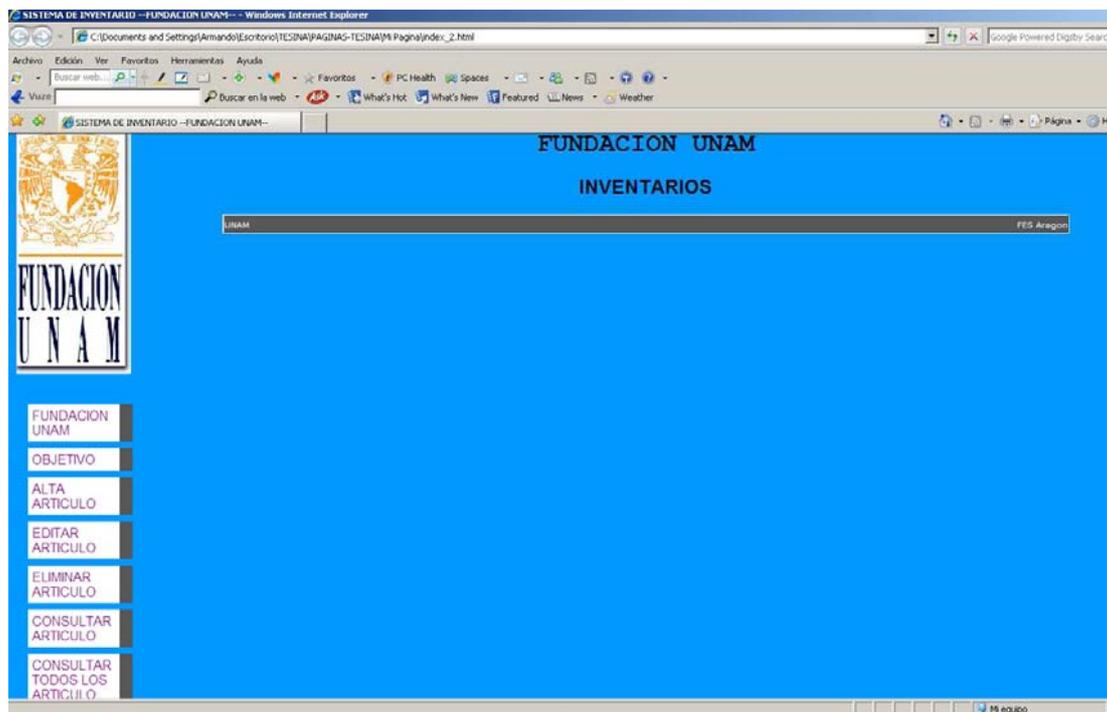


FIG. 5.3 Pantalla de Menú Principal

La siguiente pantalla muestra el objetivo de este Sistema de Inventario:

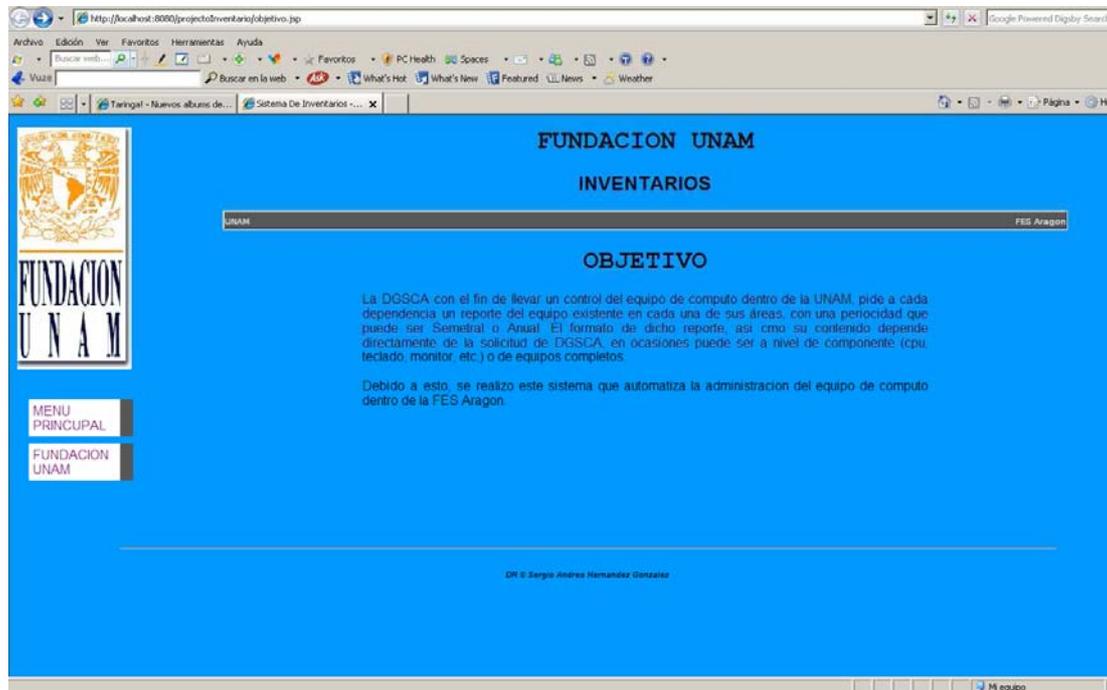


FIG. 5.4 Pantalla Objetivo del Sistema

5.4.3 Interfaz Alta de Artículo

La pantalla para dar de alta un artículo es la siguiente:

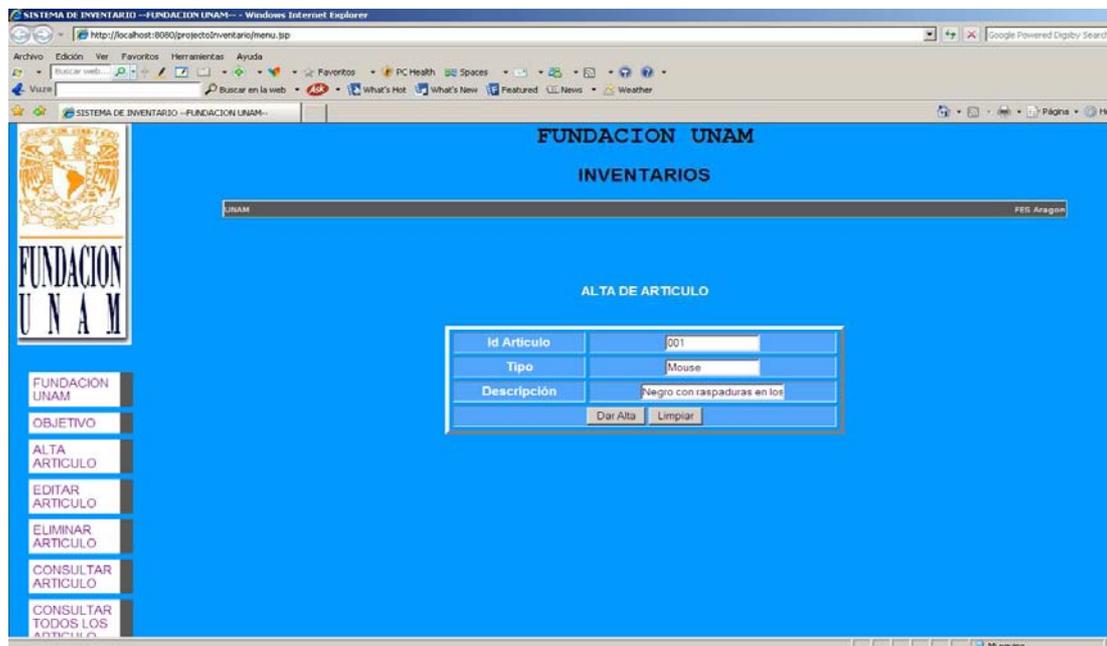


FIG. 5.5 Pantalla Alta de Artículo

El código para dar de alta un artículo es el siguiente:

```
public class ServicioInventario extends Servicio {
    // Se hace query para la inserción de datos
    private static final String insertar = "INSERT INTO INVENTARIO2 (IDARTICULO, TIPO, FECHAALTA, DESCRIPCION) VALUES (?, ?, ?, ?)";
    // Código para la inserción de un artículo
    public boolean insertarInventario(Inventario inventario) {
        // PreparedStatement contienen órdenes SQL precompiladas que pueden
        // ejecutarse muchas veces
        PreparedStatement pst = null;
        try {
            // Por medio de "super" hacemos llamada a la superclase, el cual se
            // llama "Servicio",
            // dicha clase contiene el metodo "conectarse".
            super.conectarse();
            conexion.setAutoCommit(false);
            pst = conexion.prepareStatement(insertar);
            pst.setInt(1, inventario.getIdArticulo());
            pst.setString(2, inventario.getTipo());
            pst.setString(3, inventario.getFechaAlta());
            pst.setString(4, inventario.getDescripcion());
            // Ejecuta la actualización que haces en la tabla
            pst.executeUpdate();
            conexion.commit();
            System.out.println("SE INSERTO EL ARTICULO");
            // Genera un a excepción de tipo SQLException si no se inserto un
            // artículo
        } catch (SQLException ex) {
            try {
                conexion.rollback();
                System.out.println("ERROR.....NO SE GUARDO EL ARTICULO");
                return false;
            } catch (SQLException ex1) {
                System.out.println("NO SE PUDO REGRESAR A SU ESTADO ORIGINAL");
            }
        } finally {
            if (pst != null) {
                try {
                    pst.close();
                } catch (SQLException ex) {
                }
            }
        }
        super.desconectarse();
    }
}
```

Código JSP para dar de alta un artículo:

```
<center><h1>FUNDACION UNAM</center></h1>
<center><h4> INVENTARIOS</h4></center>
<center>
<table class="tblborder" width="740" border="1" bordercolor="white" cellspacing="0" cellpadding="0" ></center>
<tr>
<td height="20%" bgcolor="#575757" bordercolorlight="#333333" bordercolordark="#000000"
<table align="right"><font size="1" face="Verdana" color="#ffffff">UNAM</font>
</table>
<table align="left">
<font size="1" face="Verdana" color="#ffffff">FES ARAGON</font>
</table>
</td>
</tr>
</table>
<br>
<center> <font face="arial" color="white"><br><br><strong>ALTA DE ARTICULO</strong></b></font></center>
<form action="" method="post" name="Form1">
<br>
<table border="5" align="center" cellspacing="5" width="35%">
<tr>
<td class="yo" bgcolor="#0099FF" align="center"><font face="arial" color="#ffffff"><strong>Id Articulo</strong></td></font>
<td class="yo" bgcolor="#0099FF"><center><input type="text" maxlength="15" size="15" name="idArticulo"></td></center>
</tr>
<tr>
<td class="yo" bgcolor="#0099FF" align="center"><font face="arial" color="#ffffff"><strong>Tipo</strong></td></font>
<td class="yo" bgcolor="#0099FF"><center><input type="text" maxlength="15" size="15" name="tipo" ></td></center>
</tr>
<tr>
<td class="yo" bgcolor="#0099FF" align="center"><font face="arial" color="#ffffff"><strong>Descripcion</strong></td></font>
<td class="yo" bgcolor="#0099FF"><center><input type="text" maxlength="1000" size="25" name="descripcion" ></td></center>
</tr>
<tr>
<td class="yo" bgcolor="#0099FF" colspan="2" align="center">
<input type="submit" name="enviados" value="Dar Alta">
<input type="reset" name="limpia" value="Limpiar">
</td>
</tr>
</table>
```

5.4.3 Interfaz Editar Artículo

La pantalla para editar un artículo es la siguiente, la cual pide el Id del Artículo a modificar:

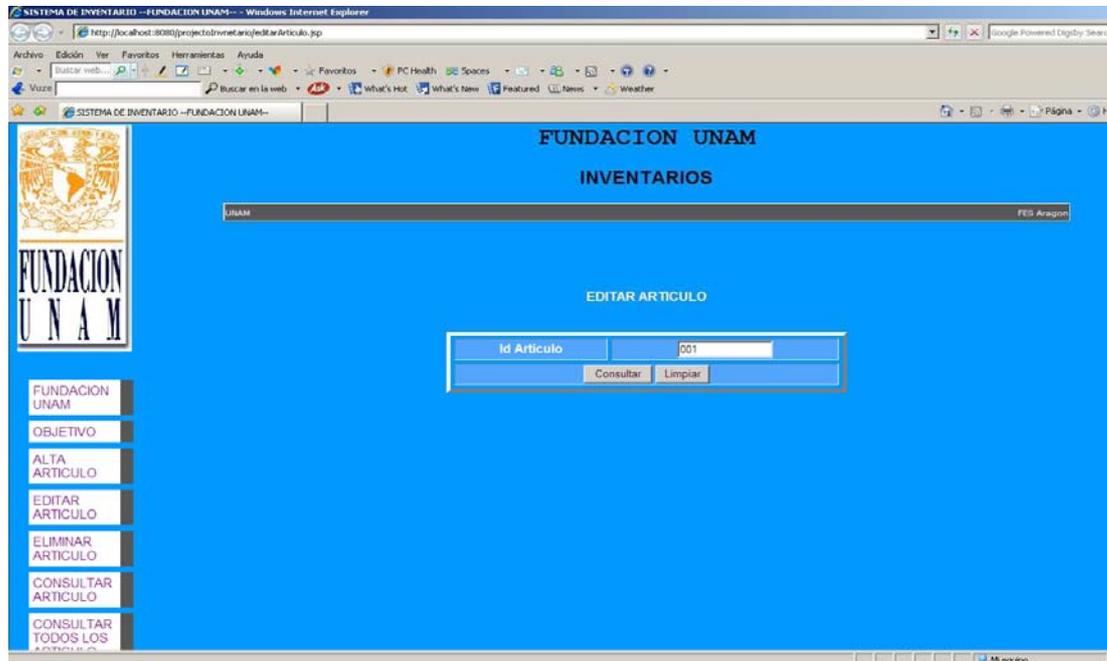


FIG. 5.6 Pantalla que pide el ID del Artículo para su modificación

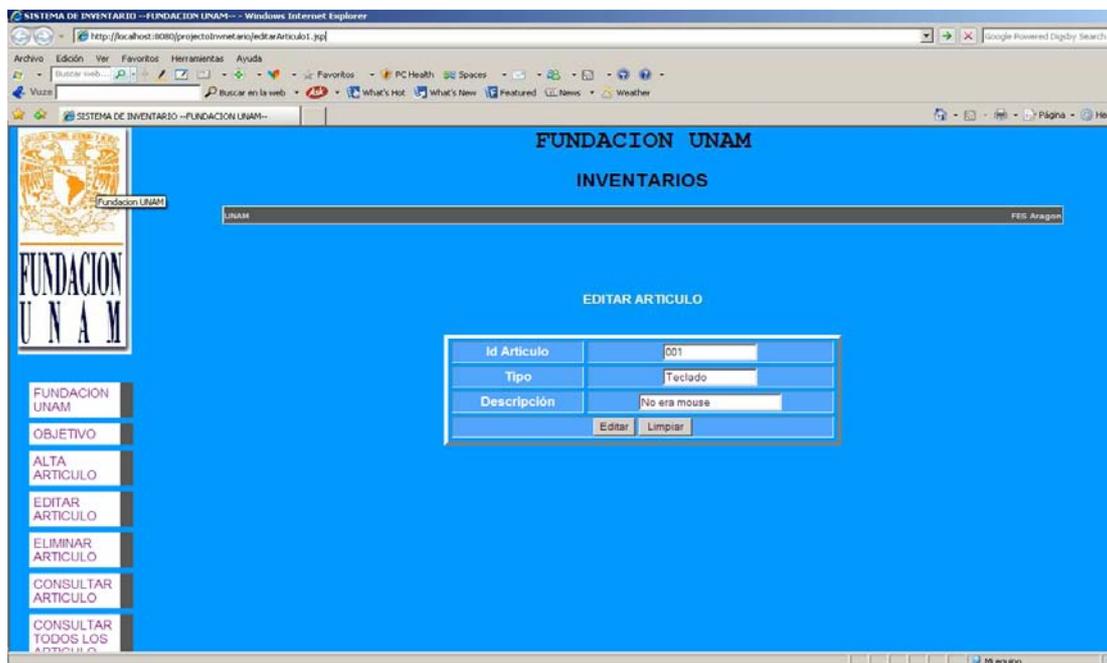


FIG. 5.7 Pantalla para la edición del artículo

El código para editar un artículo es el siguiente:

```
public class ServicioInventario extends Servicio {
    //Se hace query para la modificación de artículos
    private final static String modificar = "UPDATE INVENTARIO2 SET TIPO=?, FECHAALTA=?, DESCRIPCION=? WHERE IDARTICULO=?";
    //Codigo para la modificación de un artículo
    public boolean modificarInventario(Inventario inventario) {
        // PreparedStatement contienen órdenes SQL precompiladas que pueden
        // ejecutarse muchas veces
        PreparedStatement pstat = null;
        try {
            //Por medio de "super" hacemos llamada a la superclase, el cual se llama "Servicio",
            //dicha clase contiene el metodo "conectarse".
            super.conectarse();
            conexion.setAutoCommit(false);
            pstat = conexion.prepareStatement(modificar);
            pstat.setInt(1, inventario.getIdarticulo());
            pstat.setString(2, inventario.getTipo());
            pstat.setString(3, inventario.getFechaalta());
            pstat.setString(4, inventario.getDescripcion());
            pstat.executeUpdate();
            conexion.commit();
        } catch (SQLException ex) {
            try {
                conexion.rollback();
                System.out.println("ERROR....NO SE MODIFICO");
                return false;
            } catch (SQLException ex1) {
                System.out.println("NO SE PUDO REGRESAR A SU ESTADO ORIGINAL");
            }
        } finally {
            if (pstat != null) {
                try {
                    pstat.close();
                } catch (SQLException ex) {
                }
            }
            super.desconectarse();
        }
    }
}
```

Código JSP para la edición del artículo:

```
<center><h4> INVENTARIOS</h4></center>
<table border="1" width="740" border="1" bordercolor="white" cellspacing="0" cellpadding="0" ></center>
<tr>
<td height="20%" bgcolor="#575757" bordercolorlight="#333333" bordercolordark="#cccccc">
<table align="right"><font size="1" face="Verdana" color="#ffffff">UNAM</font>
</table>
<table align="left">
<font size="1" face="Verdana" color="#ffffff">FES ARAGON</font>
</table>
</td>
</tr>
</table>
<br>
<center> <font face="arial" color="white"><br><br><strong>EDITAR ARTICULO</strong></font></center>
<form action="" method="post" name="Form1">
<br>
<table border="5" align="center" cellspacing="5" width="35%">
<tr>
<td class="yo" bgcolor="#0099FF" align="center"><font face="arial" color="#ffffff"><strong>Id Artículo</strong></td></font>
<td class="yo" bgcolor="#0099FF"><input type="text" maxlength="15" size="15" name="idArticulo"></td></center>
</tr>
<tr>
<td class="yo" bgcolor="#0099FF" align="center"><font face="arial" color="#ffffff"><strong>Tipo</strong></td></font>
<td class="yo" bgcolor="#0099FF"><input type="text" maxlength="15" size="15" name="tipo" ></td></center>
</tr>
<tr>
<td class="yo" bgcolor="#0099FF" align="center"><font face="arial" color="#ffffff"><strong>Descripción</strong></td></font>
<td class="yo" bgcolor="#0099FF"><input type="text" maxlength="1000" size="25" name="descripcion" ></td></center>
</tr>
<tr>
<td colspan="2">
<!--BOTONES ENTRAR Y ENVIAR-->
<td class="yo" bgcolor="#0099FF" colspan="2" align="center">
<input type="submit" name="enviaDatos" value="Editar">
<input type="reset" name="limpia" value="Limpiar">
</td>
</tr>
</table>
```

5.4.5 Interfaz Baja Artículo

La pantalla para dar de baja un artículo es la siguiente:

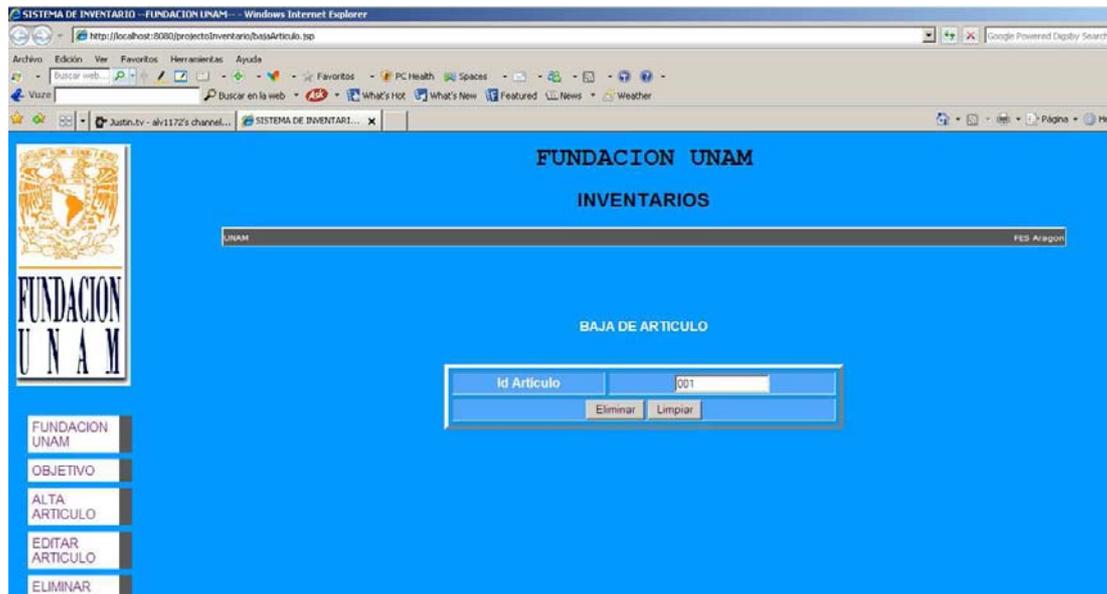


FIG. 5.8 Pantalla Baja de Artículo

El código para dar de baja un artículo es el siguiente:

```
package sergio.tesina.inventario;

import java.sql.PreparedStatement;

public class ServicioInventario extends Servicio {
    //Se hace query para la eliminación de artículo
    private final static String eliminar = "DELETE FROM INVENTARIO2 WHERE ID=?";

    //Codigo para eliminación de un artículo
    public boolean eliminarInventario(Inventario inventario) {

        // PreparedStatement contienen órdenes SQL precompiladas que pueden
        // ejecutarse muchas veces
        PreparedStatement pstmt=null;
        try{
            //Por medio de "super" hacemos llamada a la superclase, el cual se llama "Servicio",
            //dicha clase contiene el metodo "conectarse".
            super.conectarse();
            //conexion.setAutoCommit(false);
            pstmt=conexion.prepareStatement(eliminar);
            pstmt.setInt(1, inventario.getIdArticulo());
            pstmt.executeUpdate();
            return true;
        }catch(SQLException ex){
            ex.printStackTrace();
            return false;
        }
    }
}
```

Código JSP para dar de baja un artículo:

```

<html:form action="/inventario">
  <body style="background-color: 0099FF">
<image alt="Fundación UNAM" src="Copia de funam.PNG" width="100" height="200" align="left">
<center><h1>FUNDACION UNAM</h1></center>
<center><h4> INVENTARIOS</h4></center>
<center>
  <table class ="tdborder" width="740" border="1" bordercolor="white" cellspacing="0" cellpadding="0" ></center>
  <tr>
    <td height="204" bgcolor="#575757" bordercolorlight="#333333" bordercolordark="#cccccc">
      <table align="right"><font size="1" face="Verdana" color="#ffffff">UNAM</font>
      </table>
      <table align="left">
        <font size="1" face="Verdana" color="#ffffff">FES ARAGON</font>
      </table>
    </td>
  </tr>
</table>
<br>
<br>
<center> <font face="arial" color="white"><br><strong>BAJA DE ARTICULO</strong></b></font></center>
<form action="" method="post" name="Form1">
  <br>
  <table border="5" align="center" cellspacing="5" width="35%">
    <tr>
      <td class="yo" bgcolor="#0099FF" align="center"><font face="arial" color="#ffffff"><strong>Id Articulo</strong></td></tr>
      <td class="yo" bgcolor="#0099FF"><center><input type="text" maxlength="15" size="15" name="IdArticulo"></td></center>
    </tr>
    <tr>
      <td colspan="2">
        <!--BOTONES ENTRAR Y ENVIAR-->
        <table border="0" align="center">
          <tr>
            <td colspan="2">
              <input type="submit" name="enviaDatos" value="Eliminar">
              <input type="reset" name="limpia" value="Limpiar">
            </td>
          </tr>
        </table>
      </td>
    </tr>
  </table>
</form>
<ul class="navbar"><br><br><br>
<li><a href="fundacionunam.html">FUNDACION UNAM</a>
<li><a href="objetivo.html">OBJETIVO</a>
<li><a href="indice.html">ALTA ARTICULO</a>

```

5.4.6 Interfaz Consultar Artículo

La pantalla para consultar un artículo es la siguiente, la cual pide el Id del Artículo a consultar:

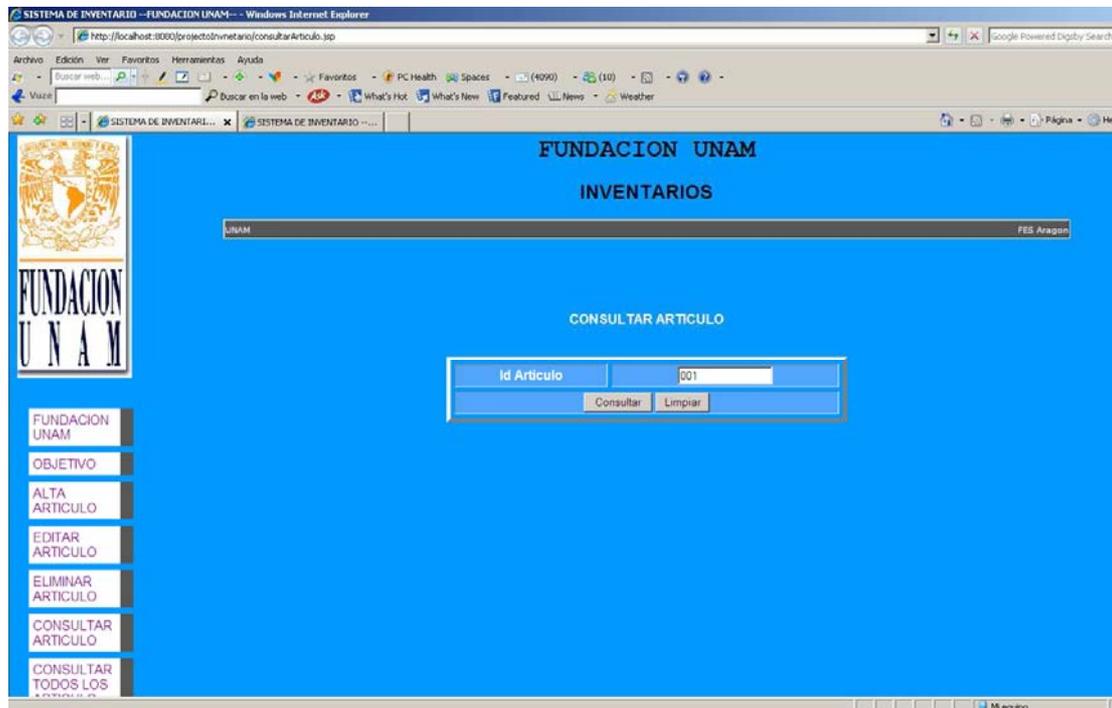


FIG. 5.9 Pantalla que pide el Id del Artículo para su consulta

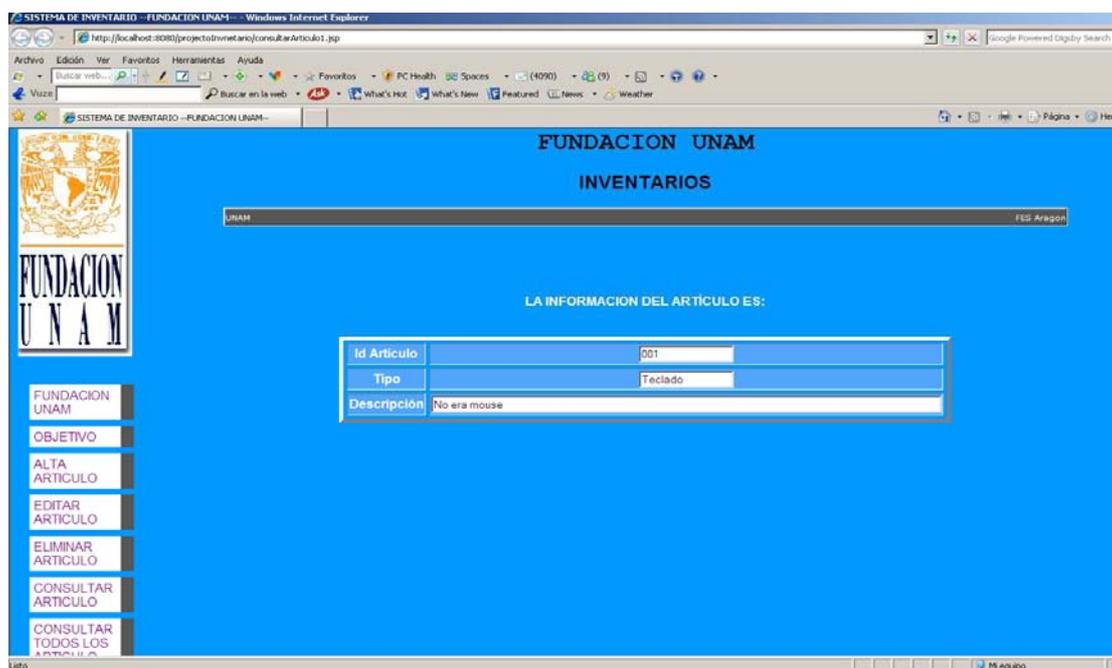


FIG. 5.10 Pantalla que muestra la información del artículo consultado

El código para consultar un artículo es el siguiente:

```
//Se hace query para la consulta de articulos
private final static String consultar = "SELECT IDARTICULO, TIPO, FECHAALTA, DESCRIPCION FROM INVENTARIO2 WHERE IDARTICULO=?";
//Codigo para la consulta de un articulo
public Inventario consultarInventario(Inventario inventario) {
    // PreparedStatement contienen órdenes SQL precompiladas que pueden
    // ejecutarse muchas veces
    PreparedStatement buscar = null;
    //Devuevle el resultado en un objeto de la clase ResultSet
    ResultSet rset = null;
    try {
        //Por medio de "super" hacemos llamada a la superclase, el cual se llama "Servicio",
        //dicha clase contiene el metodo "conectarse".
        super.conectarse();
        conexion.setAutoCommit(false);
        buscar = conexion.prepareStatement(consultar);
        buscar.setInt(1, inventario.getIdArticulo());
        rset = buscar.executeQuery();
        if (rset.next()) {
            inventario = new Inventario();
            inventario.setIdArticulo(rset.getInt(1));
            inventario.setTipo(rset.getString(2));
            inventario.setFechaAlta(rset.getString(3));
            inventario.setDescripcion(rset.getString(4));
        } else {
            inventario = new Inventario(0, "", "", "");
        }
        conexion.commit();
    } catch (SQLException ex) {
        try {
            conexion.rollback();
            System.out.println("ERROR...NO SE ENCONTRO");
        } catch (SQLException ex1) {
            System.out.println("NO SE PUDO REGRESAR A SU ESTADO ORIGINAL");
        }
    } finally {
        try {
            if (rset != null) {
```

Código JSP para consultar un artículo es:

```
<center>
<table class="tbodyborder" width="740" border="1" bordercolor="white" cellspacing="0" cellpadding="0"></center>
<tr>
<td height="20" bgcolor="#575757" bordercolorlight="#333333" bordercolordark="#cccccc"
<table align="right"><font size="1" face="Verdana" color="#ffffff">UNAM</font>
</table>
<table align="left">
<font size="1" face="Verdana" color="#ffffff">FES AR&GON</font>
</table>
</td>
</tr>
</table>
<br>
<br>
<center> <font face="arial" color="white"><br><br><strong>LA INFORMACION DEL ARTÍCULO ES:</strong></b></font></center>
<form action="" method="post" name="form1">
<br>
<table border="5" align="center" cellspacing="5" width="354">
<tr>
<td class="yo" bgcolor="#0099FF" align="center"><font face="arial" color="#ffffff"><strong>Id Articulo</strong></td></font>
<td class="yo" bgcolor="#0099FF"><center><input type="text" maxlength="15" size="15" name="idArticulo"></td></center>
</tr>
<tr>
<td class="yo" bgcolor="#0099FF" align="center"><font face="arial" color="#ffffff"><strong>Tipo</strong></td></font>
<td class="yo" bgcolor="#0099FF"><center><input type="text" maxlength="15" size="15" name="tipo" ></td></center>
</tr>
<tr>
<td class="yo" bgcolor="#0099FF" align="center"><font face="arial" color="#ffffff"><strong>Descripción</strong></td></font>
<td class="yo" bgcolor="#0099FF"><center><input type="text" maxlength="1000" size="100" name="descripcion" ></td></center>
</tr>
</table>
</td>
</tr>
</table>
<br>
<br>
<br>
<ul class="navbar"><br><br><br><br>
<li><a href="fundacionunam.html">FUNDACION UNAM</a>
<li><a href="objetivo.html">OBJETIVO</a>
<li><a href="indice.html">ALTA ARTICULO</a>
<li><a href="indice.html">EDITAR ARTICULO</a>
<li><a href="indice.html">ELIMINAR ARTICULO</a>
<li><a href="indice.html">CONSULTAR ARTICULO</a>
<li><a href="indice.html">CONSULTAR TODOS LOS ARTICULO</a>
<li><a href="f6.html">REPORTES</a>
</ul>
```

5.5 Plan de Implantación

Descripción:

Este Plan de Instalación del Software es un plan para instalar el software en el sitio del usuario, incluyendo las preparaciones, capacitación al usuario y conversiones de sistemas existentes. El SIP esta desarrollado cuando el producto está listo para su instalación en los sitios del usuario.

Descripción del sistema:

El Sistema para Administrar el Equipo de Cómputo (Fundación UNAM) es un software de tipo administrativo en el cual se quedará registrada la información del equipo de cómputo existente en para cada sala de computo de Fundación UNAM. El sistema será operado en cada sala de cómputo, ubicadas en la FES Aragón. La propuesta de solución es crear un sitio WEB que contendrá la aplicación con las pantallas de captura necesarias para el registro del equipo de cómputo.

Descripción de la Instalación:

La instalación del aplicativo debe de ser de forma centralizada, ya que es una solución web y el acceso a esta será de forma remota a través de internet. El proceso de instalación comprende una descripción de todos y cada uno de los pasos que se deben llevar a cabo para instalar el aplicativo, desde la recuperación de la versión de código del repositorio, los scripts de base de datos, hasta su compilación y despliegue dentro del servidor de aplicaciones de Oracle (OAS).

Los pasos descritos como parte del proceso de instalación son los siguientes:

- Instalación de Herramientas de Software
 - OAS (Oracle Application Server)
 - JDK 1.5
 - Data Base Oracle 10g
 - Ant (Herramienta de compilación)
 - IDE Eclipse 3.4 (Ganymede)
- Código Fuente
 - Recuperación de la versión de código del repositorio de SVN.
 - Compilación del código fuente y generación de la aplicación Web. (War)
- Base de datos

- Recuperación los scripts para la generación de la base de datos (Solo en caso de ser una instalación inicial)
- Recuperación de los respaldos del esquema de base de datos del sistema.
- Ejecución de los scripts para generación de la base de datos y carga de catálogos.
- Servidor de aplicaciones
 - Configuración del servidor de aplicaciones de Oracle (OAS)
 - Despliegue del aplicativo dentro del servidor de aplicaciones Oracle.

5.6 Capacitación

La capacitación (talleres) se llevara a cabo dentro de las salas de cómputo de Fundación UNAM con una duración de de 3 a 4 horas por sesión con el siguiente temario:

- Introducción a las características principales del sistema
- Acceso al sistema
- Alta de artículos
- Editar artículo
- Baja de artículo
- Consultar artículo
- Asesoramiento sobre uso del sistema (Un capacitador para las personas a capacitar)
- Sesión de preguntas, dudas y respuestas

Como parte del proceso de capacitación se contempla la generación del siguiente material de apoyo:

- Manual de usuario del sistema

5.7 Mantenimiento del sistema

Para darle el mantenimiento prudente al Sistema Para Administrar el Inventario de Equipo de Cómputo (Fundación UNAM), es necesario tener en cuenta lo siguiente:

- Verificar que las necesidades del usuario se alcancen con el sistema implementado, con base en la información exacta, precisa y oportuna, así como la optimización de costos, recursos y la minimización de tiempo para obtener los resultados.
- Llevar registro anual detallado mes a mes de los movimientos, resultados y problemas.
- Actualizar la viabilidad y la factibilidad de integrar el sistema a otros.
- Recolectar información sobre comentarios y sugerencias con base en observaciones y análisis por parte del personal involucrado, y que sirva de retroalimentación.
- Implementar las sugerencias de mayor prioridad y aplicarlas en el sistema en un rango de fechas determinado.

CONCLUSIONES

CONCLUSIONES

Gracias al Diplomado “Desarrollo de Sistemas en Web” adquirí los conocimientos y las herramientas necesarias que me permitieron desarrollar e implementar este sistema de inventarios para Fundación UNAM así como también de conseguir un trabajo estable y seguro.

Dado a los avances y cambios que surgen dentro de nuestro nivel de estudio, es muy significativo adquirir actualizaciones y/o capacitaciones, ya que así estaremos al nivel para poder realizar un trabajo más eficiente y eficaz para un futuro desempeño.

En cuanto a la tesina, se cumplió con el objetivo el cual fue analizar, diseñar e implementar un sistema que controla la Administración de Equipo de Cómputo para las salas de Fundación UNAM. Se recopiló la información necesaria de Ingeniería de Software y Bases de Datos para poder diseñar y desarrollar la base de datos que cumple con los requerimientos del sistema. De acuerdo con los datos recopilados para el Análisis, se analizaron las necesidades del encargado de la sala de cómputo, desarrollando así un Sistema que permite contar con la información tan precisa y actualizada como lo es de un **Inventario Sistematizado**. Controlando todos los movimientos que se generan en las salas de cómputo.

Después de analizar el trabajo realizado en el desarrollo de la aplicación, es importante mencionar que Struts facilitó el proceso de implementación del sistema, ya que proporcionó funcionalidades fundamentales a la aplicación para la Administración de Inventario para Fundación UNAM, entre los más importantes podemos mencionar los identificadores utilizados en las vistas que disminuyeron el tiempo de desarrollo, el manejo de las formas JSP y su mapeo a clases Java de manera automática, así como la configuración mediante un archivo específico de la navegación del sistema.

GLOSARIO

GLOSARIO

Algoritmo: Conjunto de instrucciones que permite la resolución de un problema pasó a paso

Análisis: Investigación de las propiedades del sistema

Aplicación: Una aplicación es un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar un o diversos tipos de trabajo.

Arquitectura de software: Estructura de ese sistema, que incluye componentes de software, las propiedades visibles externas de esos componentes, y las relaciones entre estos.

Atributo: Son las propiedades de los objetos

Base de datos relacional: Sistema de almacenamiento de datos basado en un conjunto de tablas unidas mediante relaciones.

Clase: Las clases son declaraciones o abstracciones de objetos, lo que significa, que una clase es la definición de un objeto. Cuando se programa un objeto y se definen sus características y funcionalidades, realmente se programa una clase.

Caso de uso: Técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. Normalmente, en los casos de usos se evita el empleo de jergas técnicas, prefiriendo en su lugar un lenguaje más cercano al usuario final.

Diccionario de datos: De metadatos que contiene las características lógicas de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.

Hardware: Todos los componentes físicos de la computadora y sus periféricos.

Implantación: Desarrollar o llevar a cabo un programa o actividad mediante un procedimiento o plan definido previamente.

Inventario: Listado de todos los bienes muebles (computadores, escritorios, etc.) o inmuebles (maquinaria, equipo, lotes, edificios, etc.) que la tiene la entidad en un momento determinado.

Interfaz: Clase en la que todos sus métodos son abstractos. En ellas se especifica qué se debe hacer pero no su implementación. Serán las clases que implementen estas interfaces las que implementen el comportamiento de los métodos.

Java: Lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Jsp: Tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

Llave foránea: Conjunto de atributos común a dos entidades que sirve como relación entre las dos entidades.

Llave primaria: Conjunto de atributos que distingue cada ocurrencia de una entidad de forma inequívoca a las demás.

Modelo entidad-relación: Herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para un sistema de información, sus inter-relaciones y propiedades.

Método: Porción de código asociada exclusivamente a una clase (llamados métodos de clase o métodos estáticos) o a un objeto (llamados métodos de instancia). Análogamente a los procedimientos en los lenguajes imperativos, un método consiste generalmente de una serie de sentencias para llevar a cabo una acción, un juego de parámetros de entrada que regularán dicha acción y, posiblemente, un valor de salida (o valor de retorno) de algún tipo.

Metadato: Datos que describen otros datos. En general, un grupo de metadatos se refiere a un grupo de datos, llamado recurso.

Modelo relacional: En este modelo se representan los datos y las relaciones entre estos, a través de una colección de tablas, en las cuales los renglones (tuplas) equivalen a los cada uno de los registros que contendrá la base de datos y las columnas corresponden a las características (atributos) de cada registro localizado en la tupla.

Proceso de software: Marco de trabajo para las tareas que se requieren en la construcción de software de alta calidad.

Requerimientos: necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio

SSH: (Secure SHell). Protocolo seguro y un conjunto de herramientas para reemplazar otras más comunes (inseguras). Fue diseñado desde el principio para ofrecer un máximo de seguridad y permitir el acceso remoto a servidores de forma segura.

Software: Conjunto de instrucciones codificadas para ser leídas e interpretadas por una computadora

Struts: Herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma J2EE (Java 2, Enterprise Edition). Struts se desarrollaba como parte del proyecto Jakarta de la Apache Software Foundation, pero actualmente es un proyecto independiente conocido como Apache.

SVN: Subversión es una aplicación para el control de versiones que nos permite gestionar los cambios y versiones que realizamos en nuestros desarrollos de una forma sencilla.

SFTP: Protocolo de red que proporciona la funcionalidad necesaria para la transferencia y manipulación de archivos sobre un flujo de datos fiable. Se utiliza comúnmente con SSH para proporcionar la seguridad a los datos, aunque permite ser usado con otros protocolos de seguridad.

BIBLIOGRAFIA

1. Ian Sommerville 1995. Ingeniería de Software, 5ª. Edición. Capitulo 1. Addison Wesley.
2. Roger s. Pressman 1997. Ingeniería del Software, Un enfoque práctico. 4ª Edición. McGraw HILL.
3. C.J. Date 2001. Introducción a los sistemas de Bases de Datos. 5a. Edición. Prentice Hall.
4. T. Hawryskiewicz 2002. Análisis y diseño de base de datos. 1ª Edición.
5. Kimmel Paul 2006 Manual de UML. 1ª Edición. MCGRAW-HILL / INTERAMERICANA DE MEXICO
6. Jesús Sánchez Allende, Gabriel Huecas Fernández-Toribio, Baltasar Fernández Manjón. Programación en JAVA 2 2005. Mc Graw Hill.
7. Rogers Cadenhead, Laura Lemay. Sams Teach Yourself Java 5 in 21 days 2007. Sams Publishing

ANEXO
(MANUAL DE USUARIO Y
MANUAL DE
ADMINISTRACIÓN)

SISTEMA PARA ADMINSTRAR EL INVENTARIO DE EQUIPO DE CÓMPUTO.

Fundación UNAM

MANUAL DE USUARIO

Elaborado por:

Sergio Andrés Hernández González

Mayo 2009

Contenido

Antecedentes	2
Introducción al sistema	2
Modulo de Consulta Artículo	3
Modulo de Alta Artículo	5
Modulo de Eliminar Artículo	6
Modulo de Consultar Artículo	7
Requerimientos para el uso del Sistema	7

ANTECEDENTES

El Sistema para Administrar el Inventario de Equipo de Cómputo (Fundación UNAM) es un software de tipo administrativo en el cual se verá registrada la información del equipo de cómputo existente para cada sala de cómputo de Fundación UNAM.

La aplicación cuenta con la funcionalidad de captura necesaria para el registro del equipo de cómputo. El sistema es una herramienta de software disponible en un entorno Web (Internet).

INTRODUCCIÓN AL SISTEMA

En el presente manual se explica de forma detallada las operaciones que tiene el Sistema para Administrar el Inventario de Equipo de Cómputo (Fundación UNAM).

El Sistema para Administrar el Inventario de Equipo de Cómputo (Fundación UNAM) se compone por los siguientes módulos:

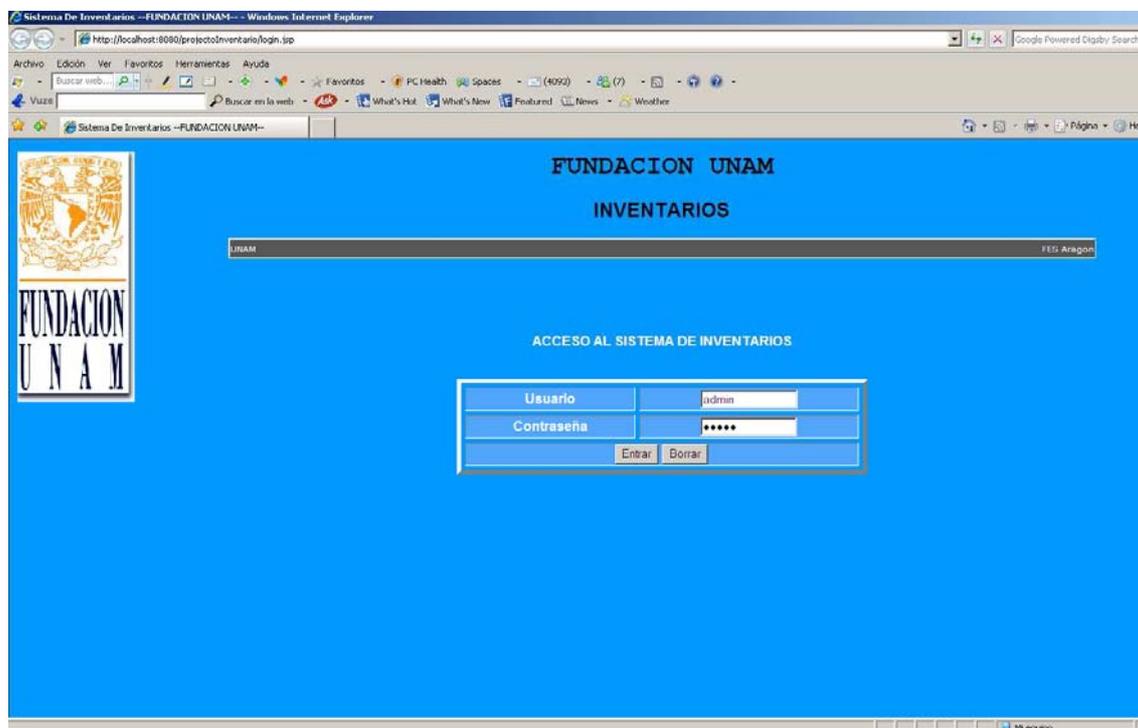
- Módulo de objetivo
- Módulo alta artículo
- Módulo editar artículo
- Módulo eliminar artículo
- Módulo modificar artículo

Propósito. El propósito del manual es explicar la operación de los módulos del sistema:

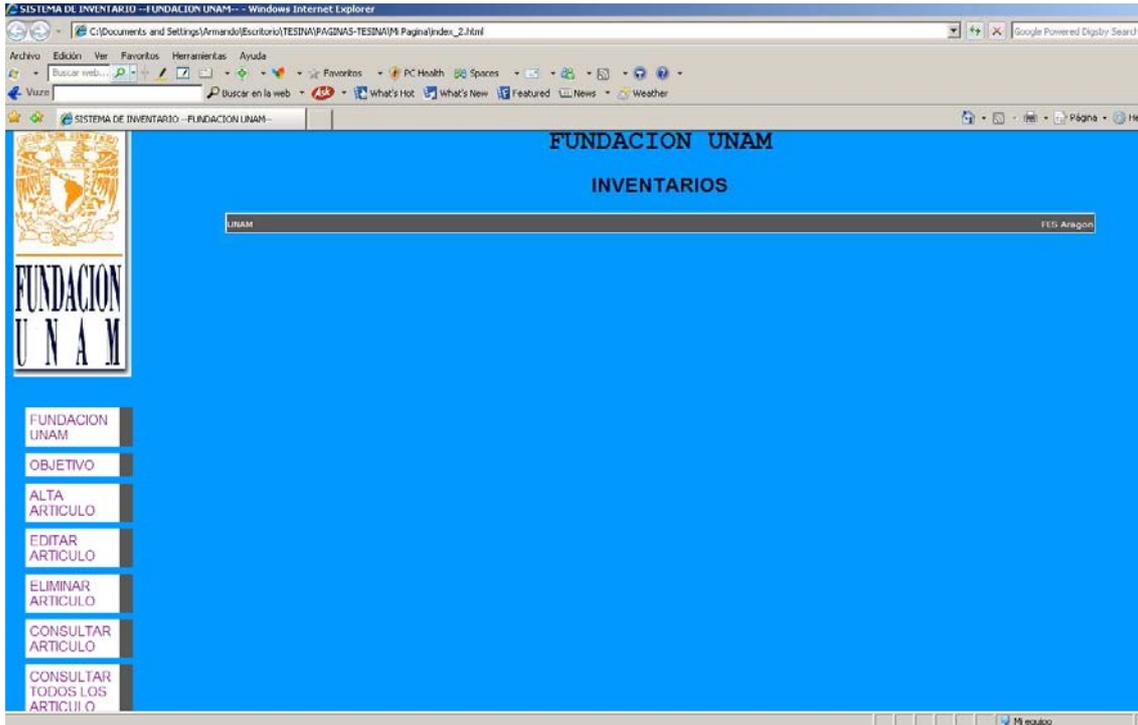
- Objetivo
- Alta Artículo
- Editar Artículo
- Eliminar Artículo
- Modificar Artículo

MODULO DE CONSULTA ARTÍCULO

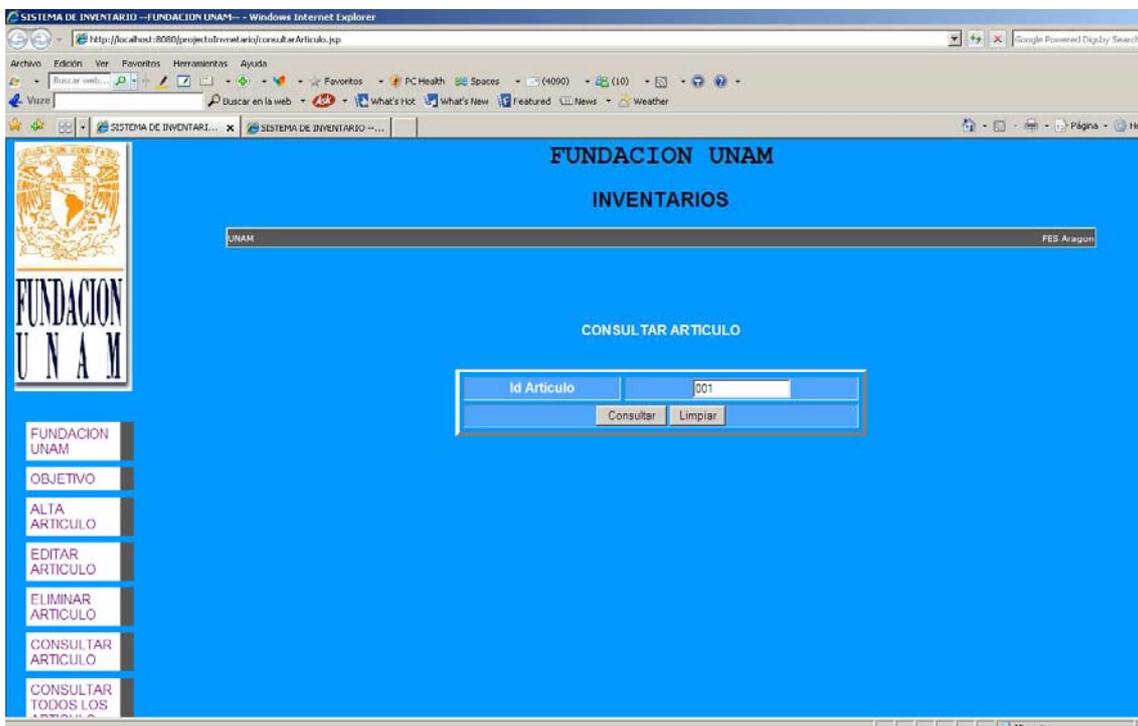
Al iniciar la sesión de trabajo en el sistema, este despliega la siguiente pantalla, solicitando su nombre y contraseña de acceso:



Una vez iniciada la sesión en el sistema, se muestra la ventana de bienvenida, con las siguientes opciones en el menú de operación: objetivo, alta artículo, editar artículo, eliminar artículo, modificar artículo.



Para realizar la consulta de un artículo se introduce los datos en el campo de captura presentado. El criterio de búsqueda disponible es "id Artículo"; mostrando el resultado de búsqueda en otra pantalla con los datos del artículo.



MODULO DE ALTA ARTÍCULO

Al seleccionar “Alta Artículo”, el sistema presenta la pantalla de captura para agregar un artículo. La pantalla de captura presenta los campos: id Artículo, Tipo, Descripción, como se muestra a continuación:

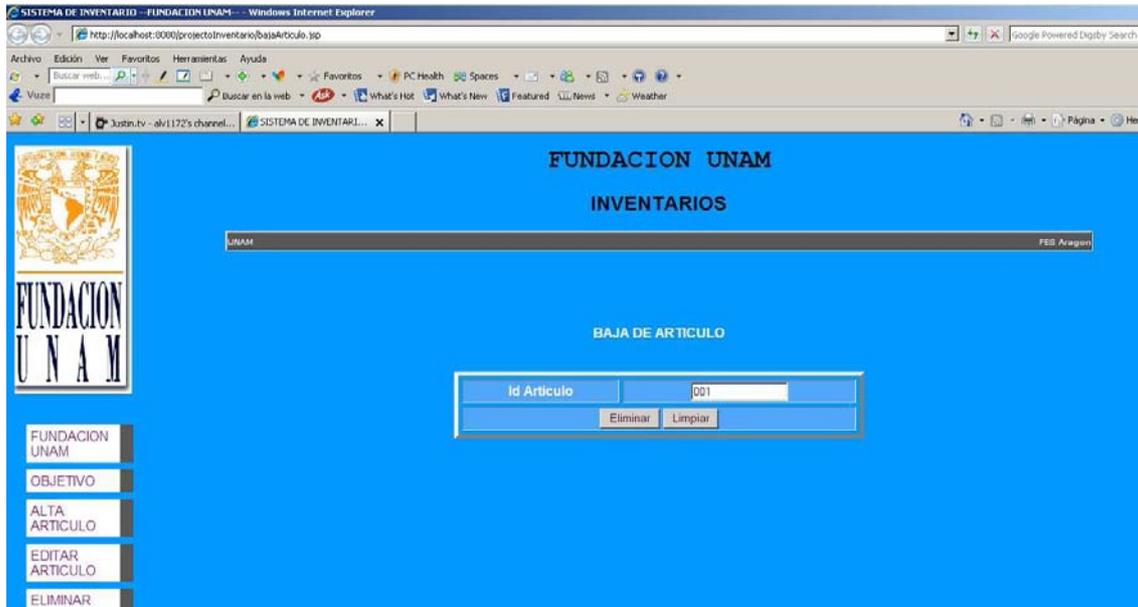
The screenshot shows a web browser window titled "SISTEMA DE INVENTARIO --FUNDACION UNAM--". The address bar shows "http://localhost:8080/proyecto/inventario/menu.jsp". The page content includes the UNAM logo, the text "FUNDACION UNAM INVENTARIOS", and a navigation menu on the left with items like "FUNDACION UNAM", "OBJETIVO", "ALTA ARTICULO", "EDITAR ARTICULO", "ELIMINAR ARTICULO", "CONSULTAR ARTICULO", and "CONSULTAR TODOS LOS ARTICULO". The main content area is titled "ALTA DE ARTICULO" and contains a form with the following fields:

Id Artículo	<input type="text" value="001"/>
Tipo	<input type="text" value="Mouse"/>
Descripción	<input type="text" value="Negro con raspaduras en los"/>

Below the form are two buttons: "Dar Alta" and "Limpiar".

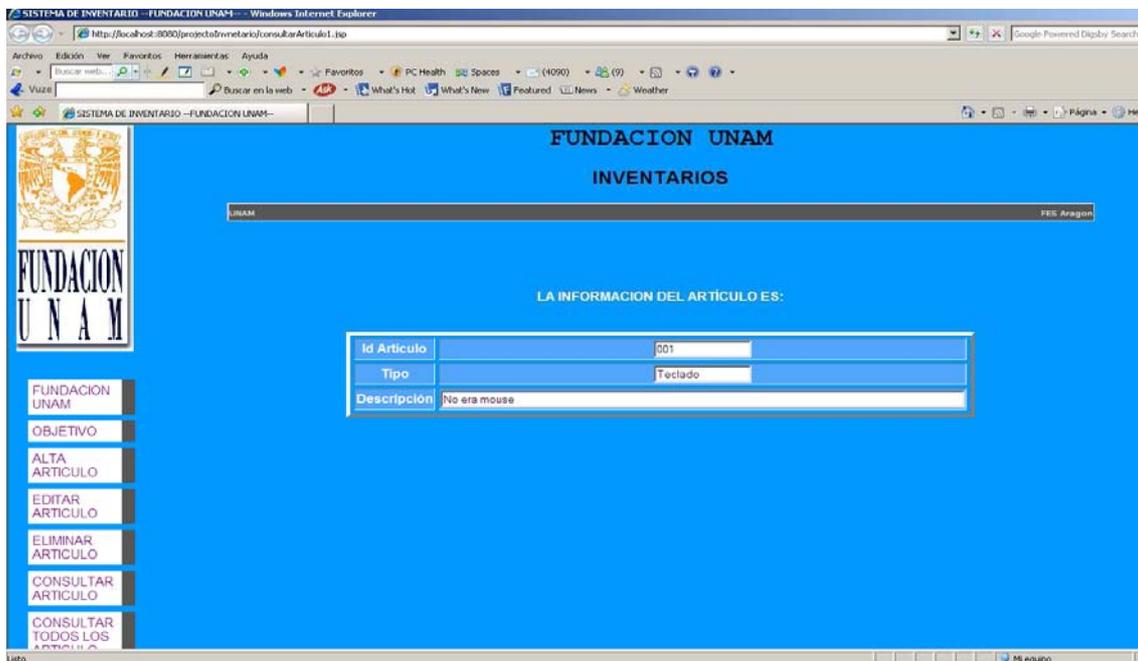
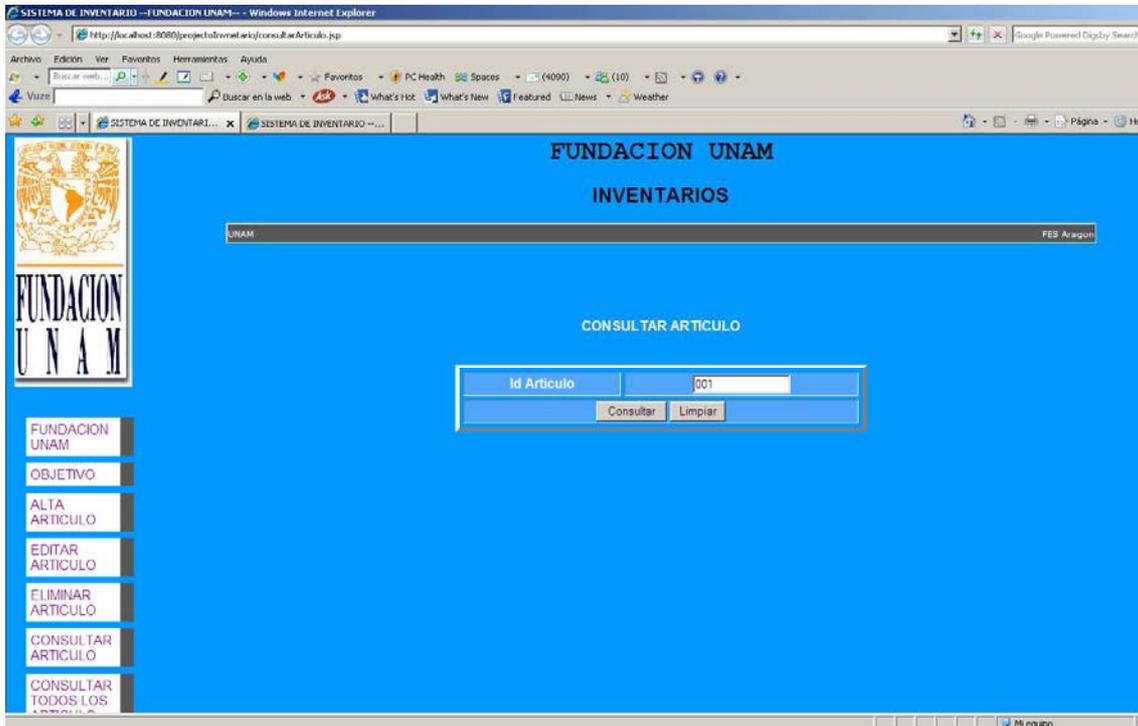
MODULO DE ELIMINAR ARTÍCULO

Al seleccionar la opción “Eliminar Artículo”, el sistema presenta el campo de captura, para eliminar el artículo, tal y como se muestra a continuación:



MODULO DE CONSULTAR ARTÍCULO

Al seleccionar “Consultar Artículo” el sistema presenta un campo de captura para consultar el artículo, una vez que el usuario haya puesto u escrito en los campos de captura los criterios de búsqueda y haber seleccionado el botón consultar, el sistema despliega una nueva ventana con la información del artículo.



Requerimientos para el uso del Sistema

SISTEMA OPERATIVO

Para un mejor funcionamiento del sistema para el Seguro Médico para una Nueva Generación debe tener como mínimo las siguientes características:

- Internet Explorer 5.0 o superior, ó Internet Explorer 5.1 y/o Firefox 3.0 o superior. Sin embargo, para una mejor visualización le sugerimos utilizar Internet Explorer 6.0 o superior para PC.
- La resolución de su video de la PC sea 800 x 600 pixeles o superior

SISTEMA PARA ADMINSTRAR EL INVENTARIO DE EQUIPO DE CÓMPUTO.

Fundación UNAM

MANUAL DE ADMINISTRADOR

Elaborado por:

Sergio Andrés Hernández González

Mayo 2009

CONTENIDO

RESUMEN DEL SISTEMA.....	3
TAREAS ADMINISTRATIVAS.....	3
INVENTARIO DE SOFTWARE.....	3
INSTALACION.....	4
PROCEDIMIENTOS DE INSTALACIÓN	4
APENDICE.....	6
INSTALACION OAS.....	6
APENDICE.....	15
INSTALACIÓN DEL SISTEMA PARA ADMINISTRAR EL INVENTARIO DE EQUIPO DE CÓMPUTO (FUNDACIÓN UNAM).....	15
INTRODUCCIÓN.....	16
PRERREQUISITOS DE INSTALACIÓN.....	17
GENERACIÓN DE APLICACIÓN WEB.....	19
CONFIGURACIÓN DEL OAS.....	23
INSTALACIÓN DE LA APLICACIÓN (DESPLIEGUE).....	35
RE INSTALACIÓN DE LA APLICACIÓN (REDEPLOY).....	47

Resumen del Sistema.

El Sistema para Administrar el Inventario de Equipo de Cómputo (Fundación UNAM) es un software de tipo administrativo en el cual se verá registrada la información del equipo de cómputo existente para cada sala de cómputo de Fundación UNAM.

La aplicación cuenta con la funcionalidad de captura necesaria para el registro del equipo de cómputo. El sistema es una herramienta de software disponible en un entorno Web (Internet).

Tareas Administrativas

Las tareas a ejecutar como parte del proceso de instalación del sistema son las siguientes:

Actividad	Secuencia de ejecución
Instalación de JDK 1.5	1
Instalación de OAS (Oracle Application Server)	2
Instalación de Data Base Oracle 10g	3
Configuración del servidor de aplicaciones de Oracle (OAS)	4
Generación del esquema de Base de datos y carga de catálogos aplicativos.	5
Ant (Herramienta de compilación)	6
Compilación y generación de la aplicación	7
Despliegue de la aplicación	8

La instalación de la infraestructura de software necesaria para el despliegue de la aplicación, debe de llevarse a cabo en un periodo máximo de dos a cinco semanas, con la restricción que involucre el no contar con los privilegios de acceso al equipo físico.

Inventario de Software

El software necesario para apoyar el proceso de instalación es:

- Cliente SSH(Secure Shell)
- IDE Eclipse 3.4 ganymede
- PlugIn Subclipse (Cliente SVN (subversion))
- Cliente SFTP

Instalaciones

Las instalaciones físicas a utilizar durante el proceso de instalación, serán en las instalaciones de la Facultad de Estudios Superiores Aragón, ya que estas cuentan con el acceso de red necesario hacia el servidor.

Procedimientos de Instalación

Los procedimientos para la instalación del aplicativo se describen a continuación:

- Instalación de JDK 1.5
 - Descargar la distribución del JDK 1.5 (Java SE Development Kit), del sitio de Sun (<http://java.sun.com/>), de la plataforma adecuada.
 - Instalar JDK 1.5
- Instalación de OAS (Oracle Application Server)(ver apéndice Instalación OAS)
- Instalación de Data Base Oracle 10g
 - Ver guías de instalación del producto <http://www.oracle.com/pls/db10g/>
- Configuración del servidor de aplicaciones de Oracle (OAS) (ver apéndice de compilación y deploy).
- Generación del esquema de Base de datos y carga de catálogos aplicativos.
- Ant (Herramienta de compilación).
 - Descarga de la herramienta ANT (apache-ant-1.7.1-bin), como apoyo para la compilación y generación de la aplicación WEB. (<http://ant.apache.org/bindownload.cgi>)
- Compilación y generación de la aplicación.
 - Para realizar este paso se debe de instalar un cliente de SVN que permita la descarga de código fuente. Se recomienda el uso del IDE Eclipse 3.4, como cliente para este repositorio.
 - Ejecutar el comando de compilación y generación de la aplicación WEB. (ver apéndice de compilación y deploy).

- Despliegue de la aplicación. (ver apéndice de compilación y deploy).

APENDICE

INSTALACION OAS

Prerrequisitos

Hardware

- Sistema Operativo: AIX 5L Version 5.3
- IP: La dirección IP de ser estática
- Nombre de anfitrión (hostname): No debe exceder 255 caracteres
- Tipo de Procesador: All AIX Common Hardware Reference Platform processors (64-bit)
- Memoria: 512 MB
- Espacio en Disco: Instalación básica de 2.45 GB
- Espacio en el directorio /tmp de 400 MB
- Espacio en Swap de 1.5 GB

Software

- Nivel de Mantenimiento: 03 o superior
- Paquetes: bos.adt.base
bos.adt.lib
bos.adt.libm
bos.perf.libperfstat
bos.perf.perfstat
bos.perf.proctools
- Parches: IY76140

Comprobación de requisitos

Nivel de mantenimiento:

```
/opt/oracle> oslevel -r  
5300-05
```

Paquetes:

```
/opt/oracle> lspp -l bos.adt.base bos.adt.lib bos.adt.libm bos.perf.perfstat bos.perf.libperfstat
```

Fileset	Level	State	Description
---------	-------	-------	-------------

Path: /usr/lib/objrepos

bos.adt.base	5.3.7.0	COMMITTED	Base Application Development Toolkit
bos.adt.lib	5.3.0.60	COMMITTED	Base Application Development Libraries
bos.adt.libm	5.3.0.40	COMMITTED	Base Application Development Math Library
bos.perf.libperfstat	5.3.7.0	COMMITTED	Performance Statistics Library Interface
bos.perf.perfstat	5.3.7.0	COMMITTED	Performance Statistics Interface

Path: /etc/objrepos

bos.perf.libperfstat	5.3.0.50	COMMITTED	Performance Statistics Library Interface
bos.perf.perfstat	5.3.7.0	COMMITTED	Performance Statistics Interface

Parches:

```
/usr/sbin/instfix -i -k "IY66513 IY70159 IY66513 IY60930 IY59386"
```

All filesets for IY66513 were found.

All filesets for IY70159 were found.

All filesets for IY66513 were found.

All filesets for IY60930 were found.

All filesets for IY59386 were found.

Antes de iniciar con la instalación es necesario ejecutar con el usuario root el script:

```
/opt/oracle/download/oas/Disk1/rootpre/rootpre.sh
```

Instalación OAS

El directorio donde se encuentran los instaladores es

```
/opt/oracle/download
```

Los archivos que contienen el instalador son

```
soa_ibm_aix_101310_disk1.cpio
```

```
soa_ibm_aix_101310_disk2.cpio
```

Extracción:

```
cd /opt/oracle/download/oas
```

```
cpio -idcmv < ../soa_ibm_aix_101310_disk1.cpio
```

```
cpio -idcmv < ../soa_ibm_aix_101310_disk2.cpio
```

Para iniciar con el proceso de instalación se debe ejecutar:

```
/opt/oracle/download/oas/Disk1/runInstaller
```

A continuación, se debe especificar que rootpre.sh ya ha sido ejecutado.

```
Has 'rootpre.sh' been run by root? [y/n] (n) y
```

El directorio donde va a residir la instalación de Oracle Application Server es

```
/opt/oracle/midtier
```

Comienza el proceso de instalación como se muestra en las siguientes pantallas.



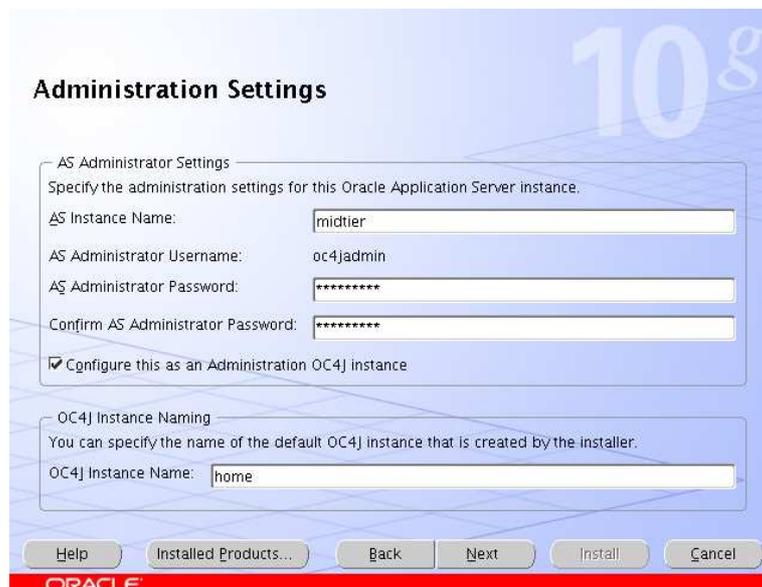
Selecciona el tipo de instalación, en este caso se selecciona la segunda opción y se da siguiente.



La configuración de puertos es manual.



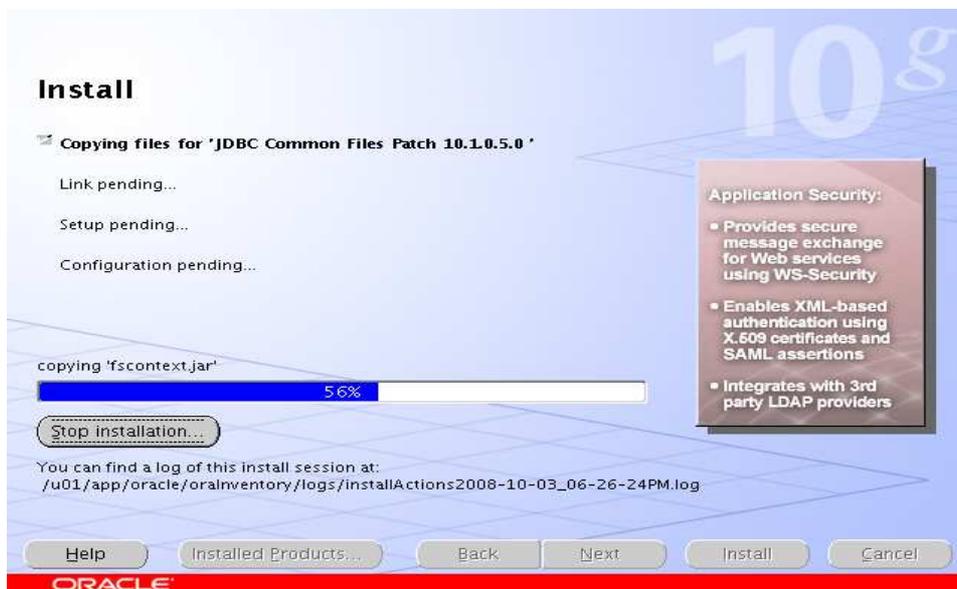
Se especifica el nombre de la instancia (midtier) y la contraseña del usuario administrador (oc4jadmin).

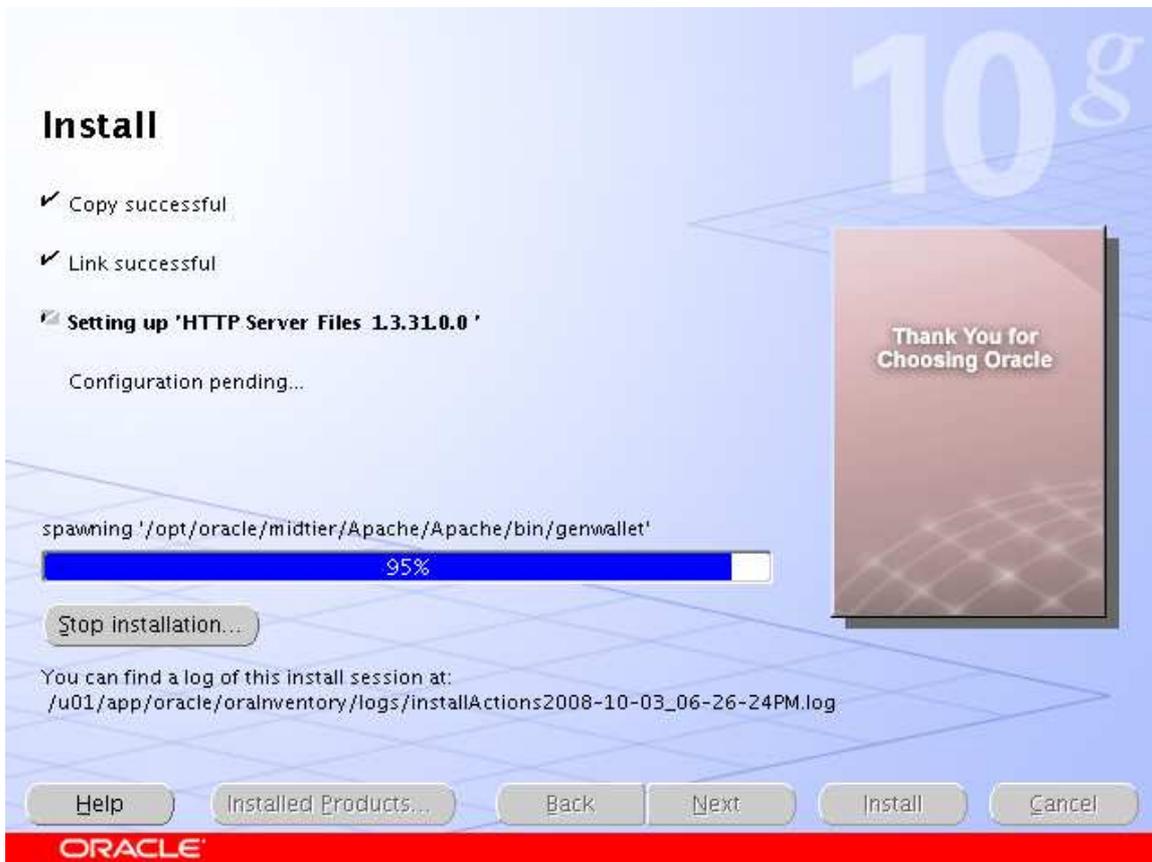


El resumen de instalación es mostrado a continuación:

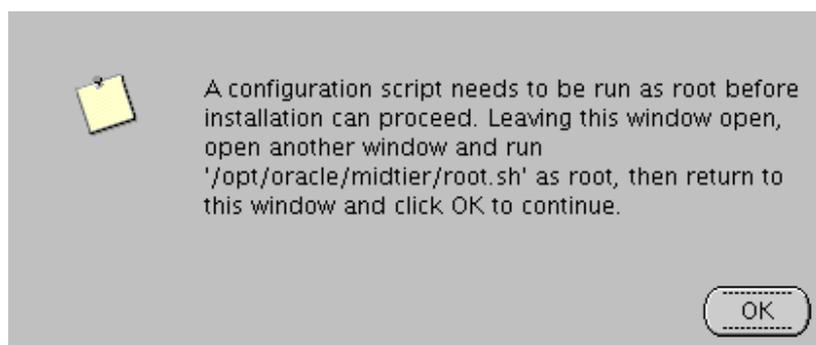


Comienza el proceso:





Es necesario ejecutar un nuevo script con el usuario root:



```
desarrollo:/opt/oracle/midtier > ./root.sh
```

Running Oracle10 root.sh script...

The following environment variables are set as:

ORACLE_OWNER= oracleas

ORACLE_HOME= /opt/oracle/midtier

Enter the full pathname of the local bin directory: [/usr/local/bin]:

The file "dbhome" already exists in /usr/local/bin. Overwrite it? (y/n) [n]:

The file "oraenv" already exists in /usr/local/bin. Overwrite it? (y/n) [n]:

The file "coraenv" already exists in /usr/local/bin. Overwrite it? (y/n) [n]:

Adding entry to /etc/oratab file...

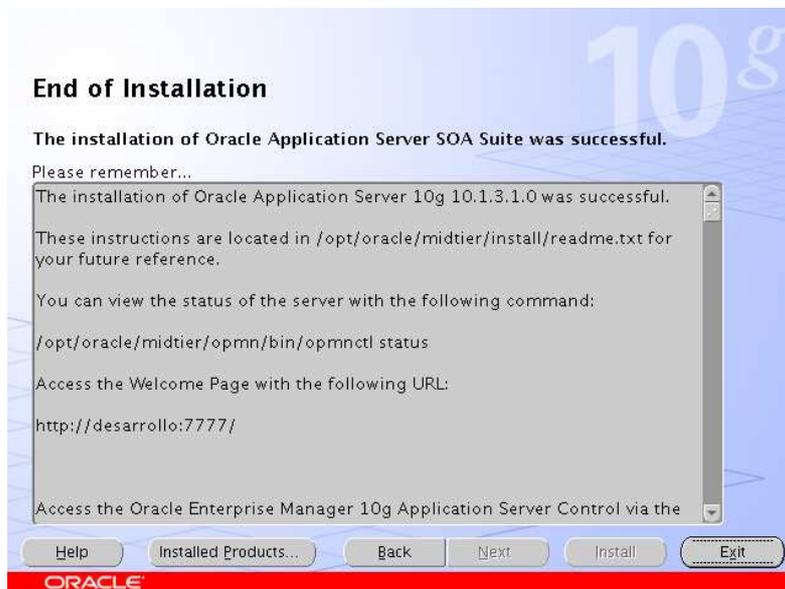
Entries will be added to the /etc/oratab file as needed by

Database Configuration Assistant when a database is created

Finished running generic part of root.sh script.

Now product-specific root actions will be performed.

Por ultimo, se muestra la pantalla de fin del proceso de instalación.



APENDICE

Instalación del Sistema para Administrar el Inventario de Equipo de Cómputo (Fundación UNAM)

Instalación del Sistema para Administrar el Inventario de Equipo de Cómputo (Fundación UNAM)

Introducción

Propósito

Definir los procedimientos de instalación del Sistema para Administrar el Inventario de Equipo de Cómputo (Fundación UNAM).

Alcance

Proporcionar información necesaria para la correcta instalación del Sistema para Administrar el Inventario de Equipo de Cómputo (Fundación UNAM), que comprende los procedimientos de compilación y despliegue de la aplicación.

Prerrequisitos de Instalación

Para poder llevar a cabo la instalación del aplicativo se deben de seguir los siguientes pasos en el orden cronológico que se presentan:

- Generación del Esquema de Base de datos y carga de catálogos.
- Generación de WEB Application
- Configuración del OAS
- Despliegue del Aplicativo sobre el OAS

Los procedimientos descritos a continuación se ejemplifican en un sistema operativo Linux, para su mejor comprensión.

Los prerrequisitos necesarios para la instalación del aplicativo se describen a continuación:

- Contar con el esquema de base de datos. Antes de realizar la instalación del aplicativo es necesario contar con el esquema de base de datos para el mismo, el nombre de este esquema debe ser "saiec".
- Para el caso de un ambiente nuevo, es necesario cargar la información de los catálogos e inicializar las secuencias correspondientes.
- Si se trata de un proceso de recuperación de un ambiente productivo, es necesario tomar el último respaldo de la base de datos y cargarlo sobre el esquema "saiec", se debe verificar que las secuencias del aplicativo correspondan con las tablas operativas, en el caso de no ser así, estas se deben ajustarse para quedar sincronizadas.
- Contar con una instancia del servidor de aplicaciones OAS, con los siguientes parámetros de memoria. (Max 768MB, min 2MB) como se muestra en la siguiente imagen:

Page Refreshed Mar 9, 2009 1:53:59 PM CDT

Multiple VM Configuration

Number of VM Processes

Ports

TIP Be sure that the port ranges specified below are large enough to accommodate the total number of processes

RMI Port RMIS Port

JMS Port

Web Sites

Name	Port	Protocol
default-web-site	<input type="text" value="12501-12600"/>	<input type="text" value="ajp"/> <input type="button" value="C"/>

Command Line Options

Start-parameters: Java Options

Server VM Enable J2SE 5.0 Platform MBeans Only applicable to Java HotSpot VM

Verbose Verbose:gc

Maximum heap size Initial heap size

Options	Delete
<input type="text" value="-Djava.security.policy=\$ORACLE_HOME/j2ee/smg/conf/gjjava2.policy"/>	<input type="button" value="Delete"/>
<input type="text" value="-Djava.awt.headless=true"/>	<input type="button" value="Delete"/>
<input type="text" value="-Dhttp.wabdir.enable=false"/>	<input type="button" value="Delete"/>
<input type="button" value="Add Another Row"/>	

Start-parameters: OC4J Options

Options	Delete
	<input type="button" value="Delete"/>

Generación de Aplicación WEB

Instalación JDK y ANT

El procedimiento para compilar el aplicativo es el siguiente:

Instalar el JDK 1.5., Descargar y descomprimir la versión en su carpeta respectiva. Por ejemplo: shernandez@shernandez:~/softwareinstalar/carpeta sin título\$./jdk-6u7-linux-i586.bin

Nota. De acuerdo a la versión a instalar de Java, el nombre de la carpeta que se genere al descomprimir el archivo, corresponderá con la versión y está podrá variar dependiendo de la versión descargada. El JDK se puede disponer de la siguiente dirección <http://java.sun.com/javase/downloads/index.jsp>. Para el caso del Sistema para Administrar el Inventario de Equipo de Cómputo (FundaciónUNAM), la versión recomendada es JDK 1.5, aun que la aplicación esta preparada para soportar versiones superiores.



```
shernandez@shernandez: ~/softwareinstalar/carpeta
Archivo Editar Ver Terminal Solapas Ayuda
Sun Microsystems, Inc. Binary Code License Agreement
for the JAVA SE DEVELOPMENT KIT (JDK), VERSION 6

SUN MICROSYSTEMS, INC. ("SUN") IS WILLING TO LICENSE THE
SOFTWARE IDENTIFIED BELOW TO YOU ONLY UPON THE CONDITION
THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS BINARY
CODE LICENSE AGREEMENT AND SUPPLEMENTAL LICENSE TERMS
COLLECTIVELY "AGREEMENT". PLEASE READ THE AGREEMENT
CAREFULLY. BY DOWNLOADING OR INSTALLING THIS SOFTWARE, YOU
ACCEPT THE TERMS OF THE AGREEMENT. INDICATE ACCEPTANCE BY
SELECTING THE "ACCEPT" BUTTON AT THE BOTTOM OF THE
AGREEMENT. IF YOU ARE NOT WILLING TO BE BOUND BY ALL THE
TERMS, SELECT THE "DECLINE" BUTTON AT THE BOTTOM OF THE
AGREEMENT AND THE DOWNLOAD OR INSTALL PROCESS WILL NOT
CONTINUE.

1. DEFINITIONS. "Software" means the identified above in
binary form, any other machine readable materials
including, but not limited to, libraries, source files,
header files, and data files), any updates or error
corrections provided by Sun, and any user manuals,
programming guides and other documentation provided to you
-Más -
```

Instalar el Apache ANT 1.7.1, descargar y descomprimir la versión en su carpeta respectiva, por ejemplo: /software/apache-ant-1.7.1

La herramienta de apoyo para el armado de la aplicación WEB se puede descargar de la url: <http://ant.apache.org/bindownload.cgi>

Compilación y armado de la aplicación WEB

Una vez instalado el JDK y ANT, se deben de colocar las siguientes variables de ambiente para realizar el compilado:

```
export JAVA_HOME=<Ruta donde esta instalado el JDK>
export ANT_HOME=<Ruta donde esta instalado el ANT>
export PATH=$ANT_HOME/bin:$JAVA_HOME/bin:$PATH
```

Ejemplo:

```
export JAVA_HOME=/home/rsanchezs/software/jdk1.5.0_16
export ANT_HOME=/home/rsanchezs/software/apache-ant-1.7.1
export PATH=$ANT_HOME/bin:$JAVA_HOME/bin:$PATH
```

Para esto se debe abrir una consola de línea de comando y ejecutar lo comandos antes mencionados.

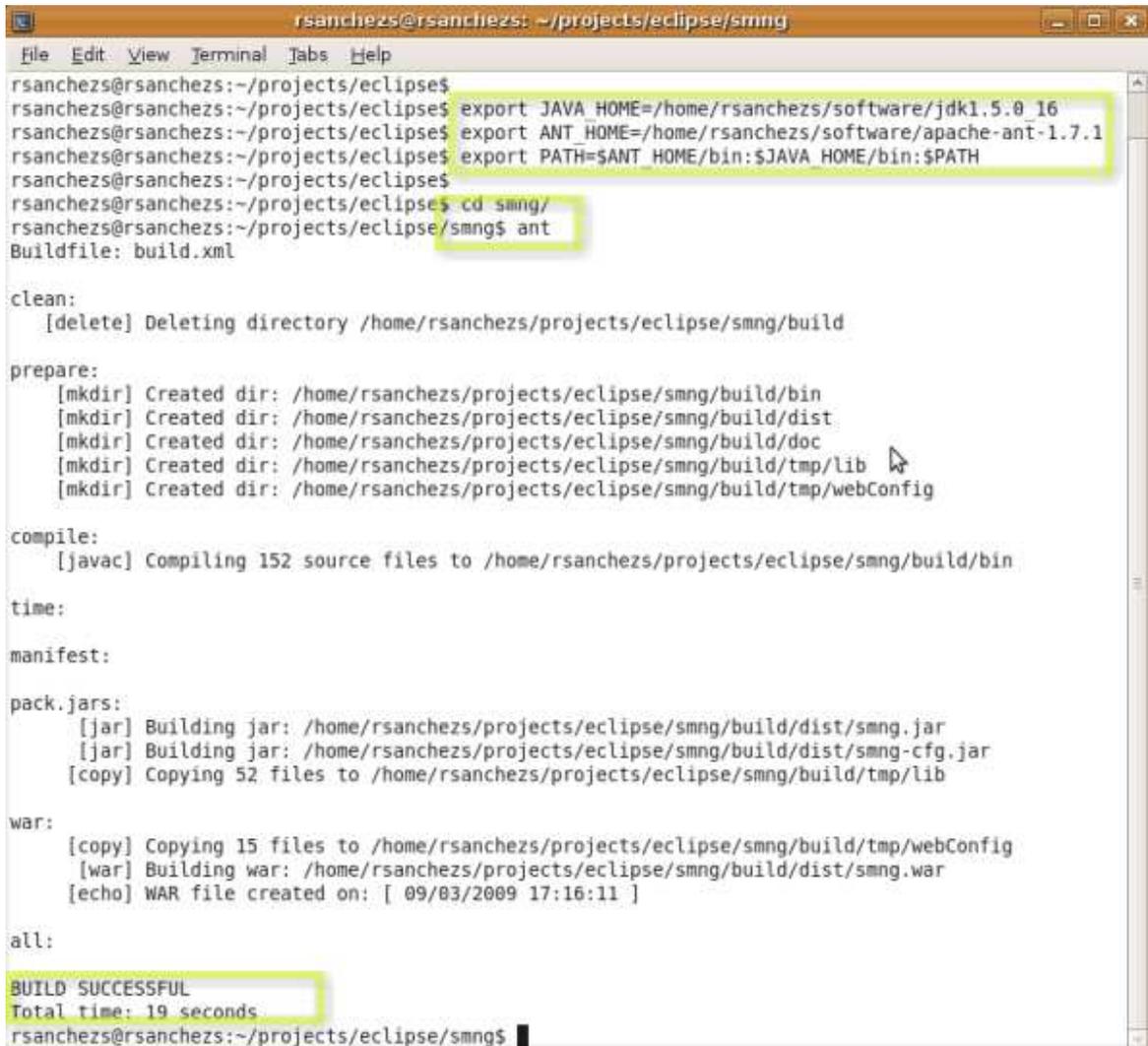


```
File Edit View Terminal Tabs Help
rsanchezs@rsanchezs:~$
rsanchezs@rsanchezs:~$
rsanchezs@rsanchezs:~$
rsanchezs@rsanchezs:~$ export JAVA_HOME=/home/rsanchezs/software/jdk1.5.0_16
rsanchezs@rsanchezs:~$ export ANT_HOME=/home/rsanchezs/software/apache-ant-1.7.1
rsanchezs@rsanchezs:~$ export PATH=$ANT_HOME/bin:$JAVA_HOME/bin:$PATH
rsanchezs@rsanchezs:~$
```

Una vez exportadas las variables de ambiente se debe de ingresar a la carpeta donde se encuentra el código fuente del proyecto.

Ejemplo: **shernandez@shernandez:~/projects/eclipse\$ cd saiec**

Y por ultimo, se debe ejecutar el comando “ant” sobre la línea de comando para iniciar la compilación y generar la aplicación WEB.



```
rsanchezs@rsanchezs: ~/projects/eclipse/smng
File Edit View Terminal Tabs Help
rsanchezs@rsanchezs:~/projects/eclipse$ export JAVA_HOME=/home/rsanchezs/software/jdk1.5.0_16
rsanchezs@rsanchezs:~/projects/eclipse$ export ANT_HOME=/home/rsanchezs/software/apache-ant-1.7.1
rsanchezs@rsanchezs:~/projects/eclipse$ export PATH=$ANT_HOME/bin:$JAVA_HOME/bin:$PATH
rsanchezs@rsanchezs:~/projects/eclipse$ cd smng/
rsanchezs@rsanchezs:~/projects/eclipse/smng$ ant
Buildfile: build.xml

clean:
  [delete] Deleting directory /home/rsanchezs/projects/eclipse/smng/build

prepare:
  [mkdir] Created dir: /home/rsanchezs/projects/eclipse/smng/build/bin
  [mkdir] Created dir: /home/rsanchezs/projects/eclipse/smng/build/dist
  [mkdir] Created dir: /home/rsanchezs/projects/eclipse/smng/build/doc
  [mkdir] Created dir: /home/rsanchezs/projects/eclipse/smng/build/tmp/lib
  [mkdir] Created dir: /home/rsanchezs/projects/eclipse/smng/build/tmp/webConfig

compile:
  [javac] Compiling 152 source files to /home/rsanchezs/projects/eclipse/smng/build/bin

time:

manifest:

pack.jars:
  [jar] Building jar: /home/rsanchezs/projects/eclipse/smng/build/dist/smng.jar
  [jar] Building jar: /home/rsanchezs/projects/eclipse/smng/build/dist/smng-cfg.jar
  [copy] Copying 52 files to /home/rsanchezs/projects/eclipse/smng/build/tmp/lib

war:
  [copy] Copying 15 files to /home/rsanchezs/projects/eclipse/smng/build/tmp/webConfig
  [war] Building war: /home/rsanchezs/projects/eclipse/smng/build/dist/smng.war
  [echo] WAR file created on: [ 09/03/2009 17:16:11 ]

all:
BUILD SUCCESSFUL
Total time: 19 seconds
rsanchezs@rsanchezs:~/projects/eclipse/smng$
```

La aplicación WEB se va a generar en la carpeta `/build/dist/`, ubicada dentro del directorio del código fuente. El nombre del archivo es **“saiec.war”**, el cual se va a desplegar posteriormente dentro del servidor de aplicaciones (OAS).

Configuración del OAS

Pool de conexiones

Ingresar a la consola del servidor de aplicaciones de Oracle en la siguiente url:
<http://<Servidor donde esta instalado el OAS>:8888/em>. Por ejemplo:
<http://localhost:8888/em>.

Oracle Enterprise Manager (oc4jadmin) - OC4J: home - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://192.168.1.201:8888/em/console/ias/oc4j/home

ORACLE Enterprise Manager 10g
Application Server Control

OC4J: home

Página Refrescada 02-mar-2009 10H00' CST • Ver Datos [Refrescamiento Manual]

Inicio Aplicaciones Servicios Web Rendimiento Administración

General

Estado **Arriba** [Parar] [Reiniciar]

Hora de Inicio **02-mar-2009 09H56' CST**

Versión **10.1.3.4.0**

Directorio Raíz de Oracle **/home/jesumca/software/oc4j**

Host **jesusmca-laptop**

Máquinas Virtuales **1**

Notificaciones **0**

Respuesta y Carga

0,26
0,24
0,22
0,20
0,18
0,16
0,14
0,12
0,10
0,08
0,06
0,04
0,02
0

9:58:51 9:59:20 9:59:40 10:00:00
02-mar-2009

■ Tiempo de Procesamiento de Solicitud (segundos)
■ Solicitudes por Segundo

Inicio Aplicaciones Servicios Web Rendimiento Administración

Configurar | Logs | Ayuda | Desconexión

Copyright © 1996, 2008, Oracle. Todos los Derechos Reservados.
Oracle, JD Edwards, PeopleSoft y Siebel son marcas comerciales registradas de Oracle Corporation y/o sus subsidiarias. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.
Agencia de Oracle Enterprise Manager 10g Application Server Control

Listo

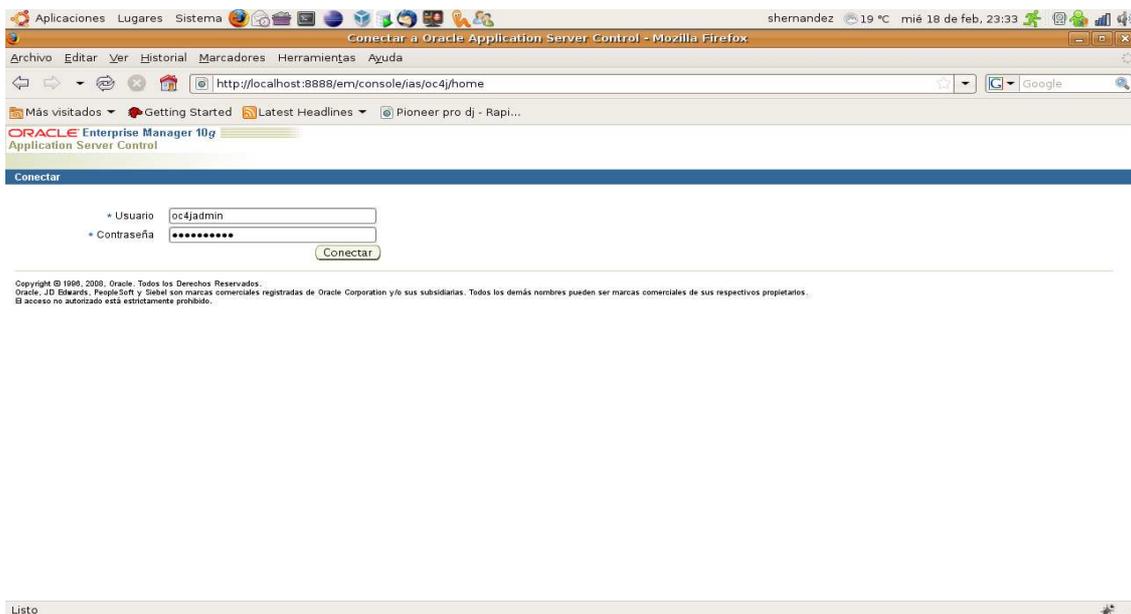
Sistema de Administración de Equipo de Cómputo

Fundación UNAM

Proporcionar el usuario y contraseña de acceso para ingresar. Por ejemplo:

Usuario= oc4jadmin

Password= password definido al momento de configurar el OC4J



Sistema de Administración de Equipo de Cómputo

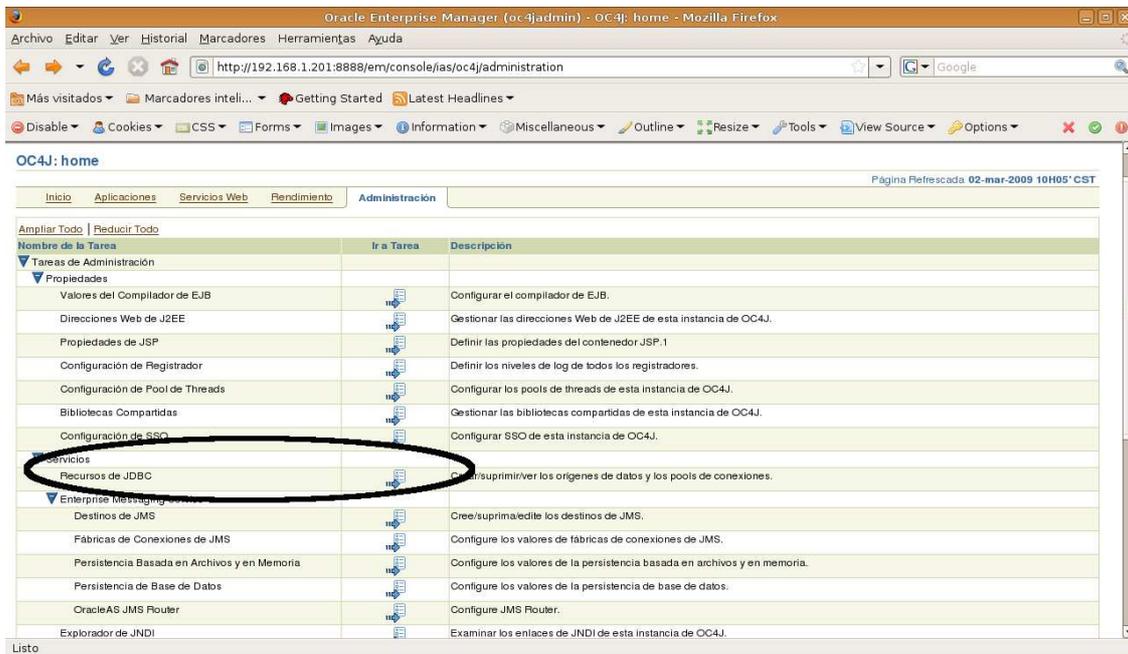
Fundación UNAM

The screenshot shows the Oracle Enterprise Manager 10g Application Server Control interface. The browser address bar displays `http://localhost:8888/em/console/ias/oc4j/home`. The page title is "OC4J: home". The navigation menu includes "Inicio", "Aplicaciones", "Servicios Web", "Rendimiento", and "Administración". The "Administración" tab is selected. The "General" section shows the server status as "Arriba" (Up) with a green checkmark icon. The "Hora de Inicio" is "18-feb-2009 23H27' CST". The "Versión" is "10.1.3.4.0". The "Directorio Raíz de Oracle" is "/home/shernandez/software/oc4j_2". The "Host" is "localhost". The "Máquinas Virtuales" count is "1" and "Notificaciones" is "0". The "Respuesta y Carga" section contains a line graph showing "Tiempo de Procesamiento de Solicitud (segundos)" and "Solicitudes por Segundo" for the date "18-feb-2009". The graph shows a peak in request rate around 21:34:00. The "Listo" status bar is visible at the bottom.

En la página home del OAS, ingresar a la opción identificada como administración.

This screenshot is similar to the first one but shows a different time: "02-mar-2009 10H00' CST". The "Administración" tab in the navigation menu is circled in black. The "General" section shows the server status as "Arriba" (Up) with a green checkmark icon. The "Hora de Inicio" is "02-mar-2009 09H56' CST". The "Versión" is "10.1.3.4.0". The "Directorio Raíz de Oracle" is "/home/jesusmca/software/oc4j". The "Host" is "jesusmca-laptop". The "Máquinas Virtuales" count is "1" and "Notificaciones" is "0". The "Respuesta y Carga" section contains a line graph showing "Tiempo de Procesamiento de Solicitud (segundos)" and "Solicitudes por Segundo" for the date "02-mar-2009". The graph shows a peak in request rate around 9:58:51. The "Listo" status bar is visible at the bottom.

Dentro del módulo de Administración seleccionar la opción identificada como “Recursos JDBC” y seleccionar “Crear/suprimir/ver los orígenes de datos y los pools de conexiones.”



Dentro del módulo de Recursos de JDBC, “Pool de Conexiones” seleccionar “Crear”.

Oracle Enterprise Manager 10g
Application Server Control

OC4J home >
Recursos de JDBC

Página Refrescada 02-mar-2009 10H06' CST

Aplicación:

Origenes de Datos

Crear

Nombre	Aplicación	Ubicación de JNDI	Pool de Conexiones	Gestionado por OC4J	Probar Conexión	Suprimir
"OracleDS"	default	jdbc:OracleDS	"Example Connection Pool"	✓		

Pools de Conexiones

Crear

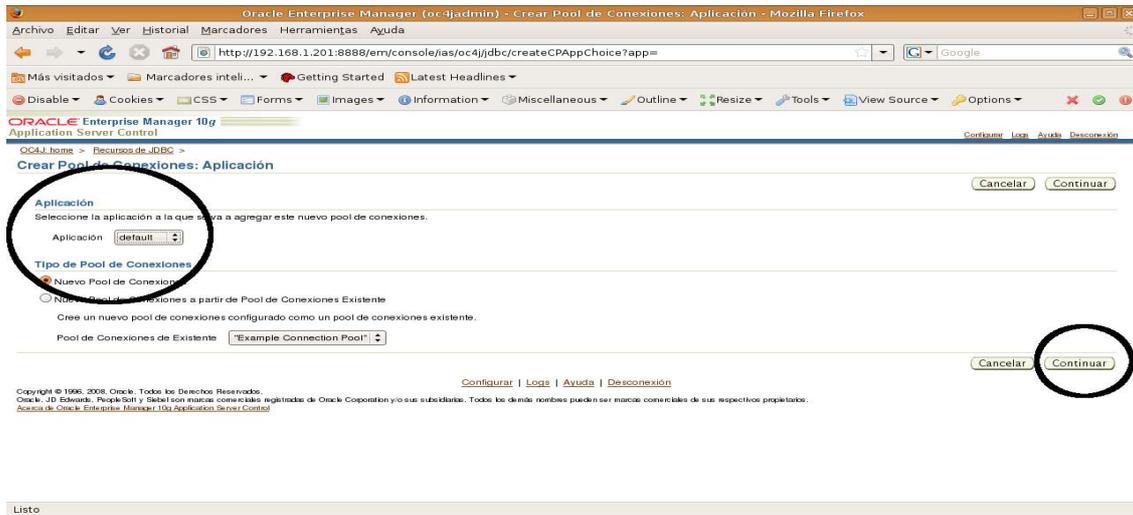
Nombre	Aplicación	Clase de Fábrica de Conexiones	Controlar Rendimiento	Probar Conexión	Refrescar Pool de Conexiones	Suprimir
"Example Connection Pool"	default	oracle.jdbc.pool.OracleDataSource				

Configurar | Logs | Ayuda | Desconexión

Copyright © 1996, 2008, Oracle. Todos los Derechos Reservados.
Oracle, JD Edwards, PeopleSoft y Siebel son marcas comerciales registradas de Oracle Corporation y/o sus subsidiarias. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.
Acronia de Oracle Enterprise Manager 10g Application Server Control

Listo

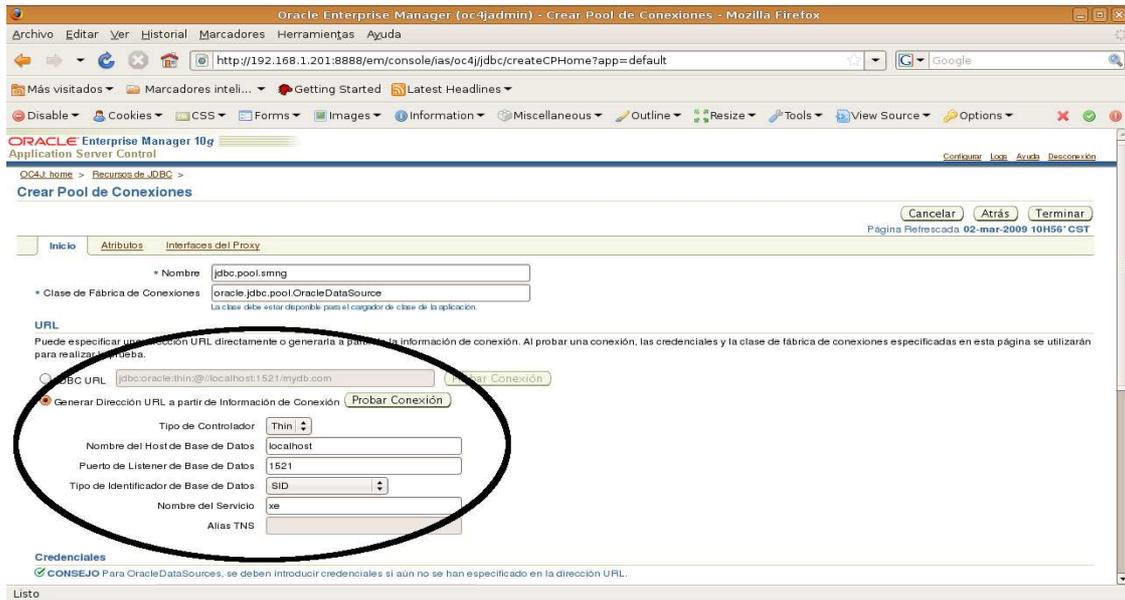
Dentro de la opción de “Crear Pool de Conexiones: Aplicación”, seleccionar en el campo Aplicación la opción por “default”, en el campo de “Tipo de Pool de Conexiones” elegir la opción de “Nuevo Pool de Conexiones”, en el campo de “Pool de conexiones Existentes” dejar la opción por default (Example Connection Pool); después seleccionar “Continuar”.



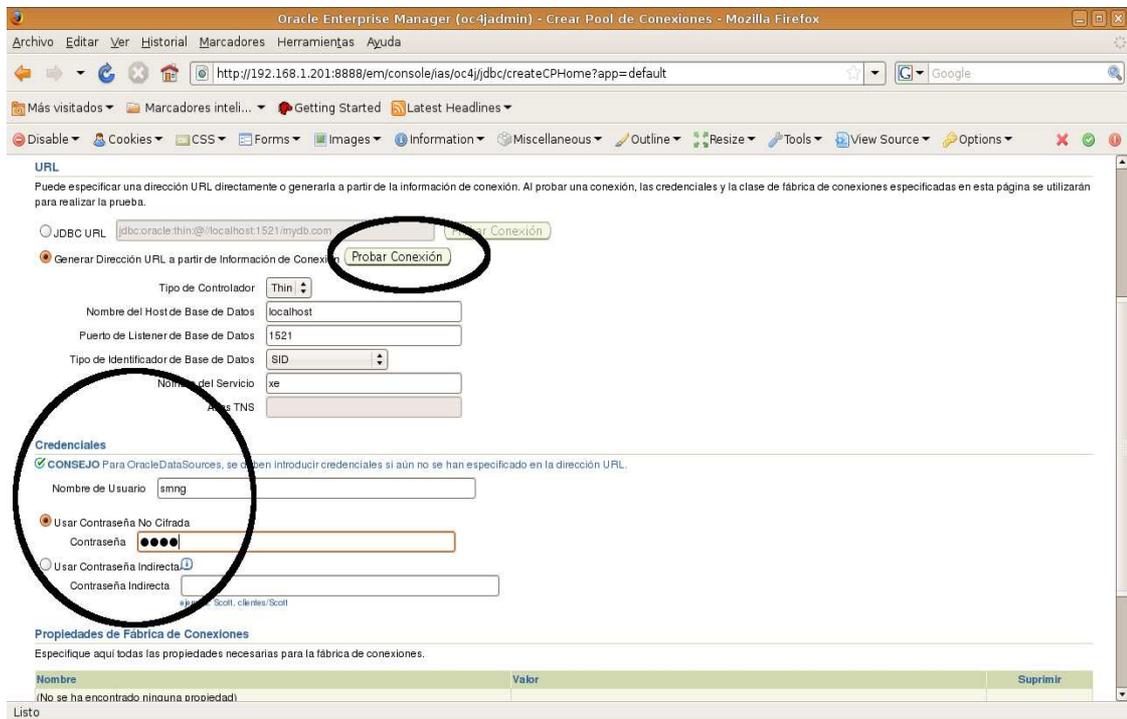
Dentro de la funcionalidad de “Crear Pool de Conexiones”, en el campo de “Nombre” escribir el nombre del pool: **“jdbc.pool.saiec”**, en el campo de “Clase de Fabrica de Conexiones” escribir: **“oracle.jdbc.pool.OracleDataSource”**. Seleccionar la opción de “Generar Dirección URL a partir de información de conexión” y llenar los campos con la siguiente información forma:

- Tipo de controlador: **thin**
- Nombre del Host de Base de Datos: **localhost <Dirección IP o Nombre del Servidor de Base de Datos>**
- Punto de Listener de Base de Datos: **1521**
- Tipo de identificación Base de Datos: **SID**
- Nombre del Servicio: **xe <Nombre de la instancia de la base de datos>**
- Alias TNS: campo vacío

The screenshot shows the Oracle Enterprise Manager 10g console in a Mozilla Firefox browser window. The page title is "Oracle Enterprise Manager (oc4jadmin) - Crear Pool de Conexiones". The browser address bar shows the URL: `http://192.168.1.201:8888/em/console/ias/oc4jjdbc/createCPHome?app=default`. The page content includes a navigation menu with "Inicio", "Atributos", and "Interfaces del Proxy". The "Interfaces del Proxy" tab is active, and the "Nombre" field is circled in black, containing the text "jdbc.pool.smng". Below this, the "Clase de Fábrica de Conexiones" field contains "oracle.jdbc.pool.OracleDataSource". The "URL" section has a radio button selected for "JDBC URL" with the value "jdbc:oracle:thin:@localhost:1521/mydb.com" and a "Probar Conexión" button. Other fields include "Tipo de Controlador" (Thin), "Nombre del Host de Base de Datos", "Puerto de Listener de Base de Datos", "Tipo de Identificador de Base de Datos" (Nombre del Servicio), "Nombre del Servicio", and "Alias TNS". A "Credenciales" section at the bottom has a checked checkbox and a message: "CONSEJO Para OracleDataSources, se deben introducir credenciales si aún no se han especificado en la dirección URL." The status bar at the bottom left says "Listo".



En la sección de "Credenciales", en el campo "Nombre de usuario" escribir el nombre del usuario de la base de datos, por ejemplo "saiec", enseguida seleccionar la opción de "Usar Contraseña No Cifrada", y en el campo de "Contraseña" escribir la contraseña del usuario de la base de datos. Una vez capturada esta información, se puede comprobar la conexión a la base de datos, seleccionando la opción de "Probar Conexión".

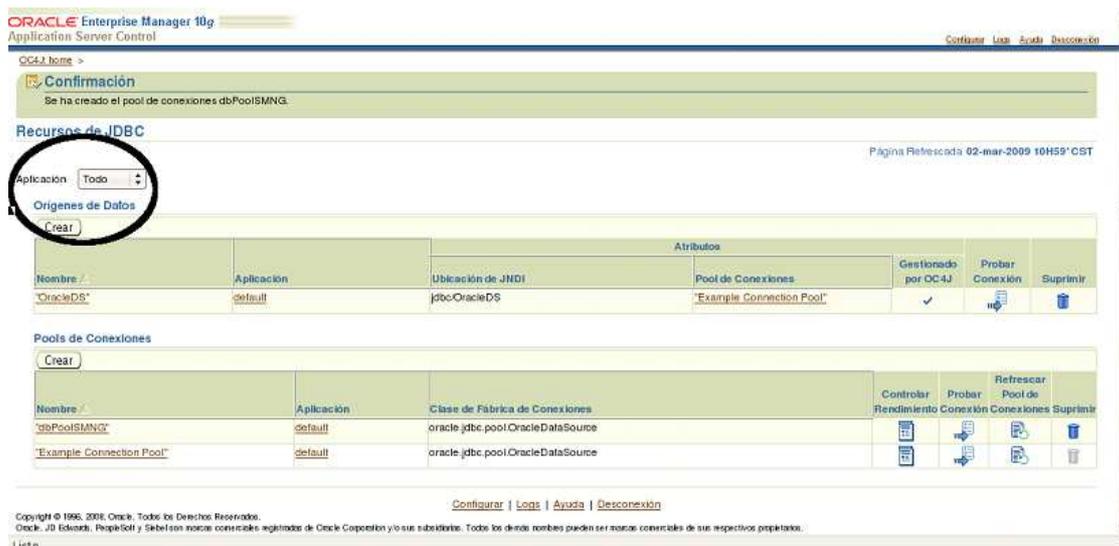


Al ejecutar esta opción deberá aparecer un mensaje del éxito de la operación: "La conexión se ha establecido correctamente", si es así, se debe seleccionar la opción de "Terminar" para concluir esta configuración, de lo contrario se debe verificar la información capturada en los parámetros de conexión para asegurarse de que sea correcta.

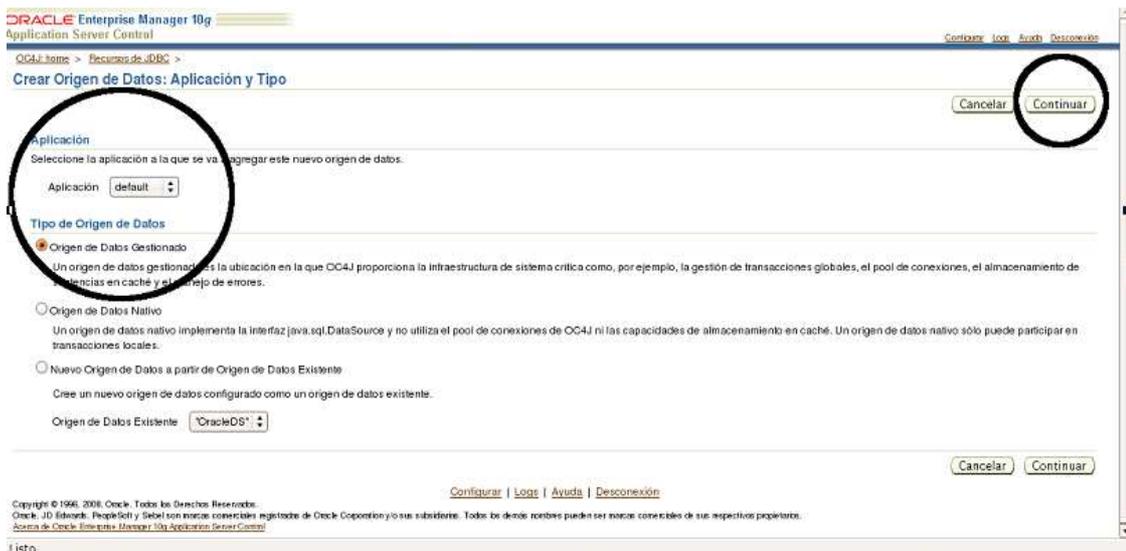


Configuración del Origen de Datos (DataSource)

Para la configuración de un Datasource se deben seguir los siguientes pasos. En la funcionalidad de Recursos de JDBC, en la opción “Orígenes de Datos”, seleccionar la opción de “Crear”.



Una vez dentro de esta opción (Crear Orígenes de Datos), en el campo Aplicación se debe seleccionar la opción “default”, y en el tipo de Origen de Datos se debe seleccionar la opción “Origen de Datos Gestionado”, para después seleccionar “Continuar”.



En la pantalla siguiente de la opción de “Crear Orígenes de Datos: Origen de Datos Gestionado”, llenar los campos solicitados de la siguiente información:

- Nombre: **jdbc.ds.saiec**
- Ubicación de JNDI: **jdbc/DS_SAIEC**
- Nivel de Transacción: seleccionamos “**Transacciones Globales y Locales**”
- Pool de Conexiones: seleccionamos “**jdbc.pool.saiec**”
- Time out de conexión de segundos: 0 (cero)

Terminada esta captura, seleccionar la opción de Terminar, para concluir la configuración del DataSource.

ORACLE Enterprise Manager 10g
Application Server Control

OC4J home > Recursos de JDBC >

Crear Origen de Datos: Origen de Datos Gestionado

Cancelar Atrás Terminar

Aplicación default

* Nombre

* Ubicación de JNDI

Nivel de Transacción: Transacciones Globales y Locales

Pool de Conexiones: Example Connection Pool

* Timeout de Conexión (segundos): 0

► Credenciales

Configurar | Logs | Ayuda | Desconexión

Copyright © 1996, 2008, Oracle. Todos los Derechos Reservados.
Oracle, JD Edwards, PeopleSoft y Siebel son marcas comerciales registradas de Oracle Corporation y/o sus subsidiarias. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.
Acordado Oracle Enterprise Manager 10g Application Server Control

Listo

ORACLE Enterprise Manager 10g
Application Server Control

OC4J home > Recursos de JDBC >

Crear Origen de Datos: Origen de Datos Gestionado

Cancelar Atrás Terminar

Aplicación default

* Nombre: jdbc.ds.smng

* Ubicación de JNDI: jdbc:DS_SMNG

Nivel de Transacción: Transacciones Globales y Locales

Pool de Conexiones: dbPoolSMNG

* Timeout de Conexión (segundos): 0

► Credenciales

Configurar | Logs | Ayuda | Desconexión

Copyright © 1996, 2008, Oracle. Todos los Derechos Reservados.
Oracle, JD Edwards, PeopleSoft y Siebel son marcas comerciales registradas de Oracle Corporation y/o sus subsidiarias. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.
Acordado Oracle Enterprise Manager 10g Application Server Control

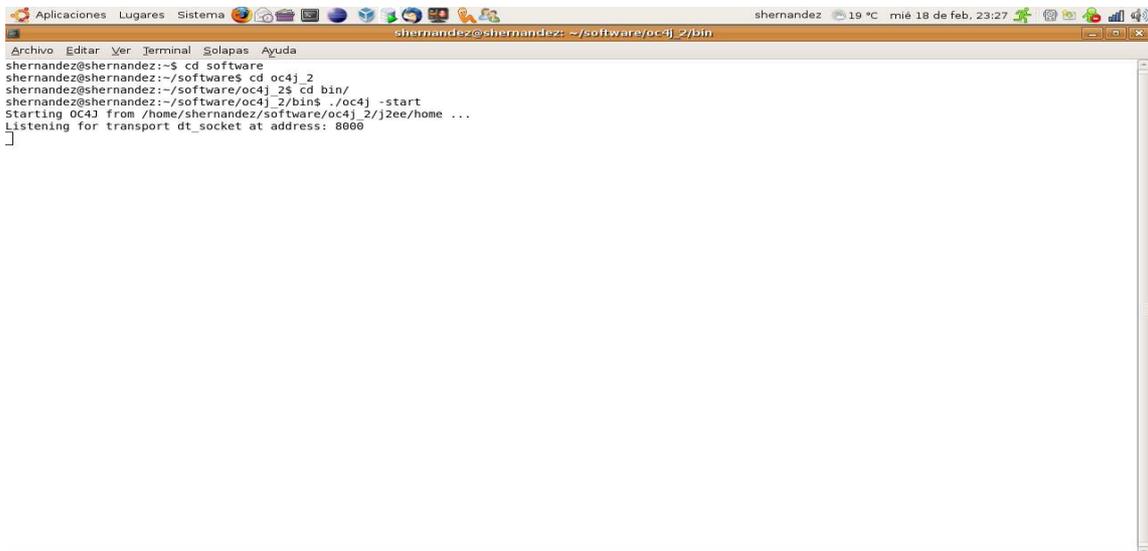
Listo

Instalación de la aplicación (Despliegue)

Levantar el OAS

Para levantar el servidor de aplicaciones se debe ejecutar el siguiente comando, sobre la ruta donde se tiene instalado.

>./oc4j -start



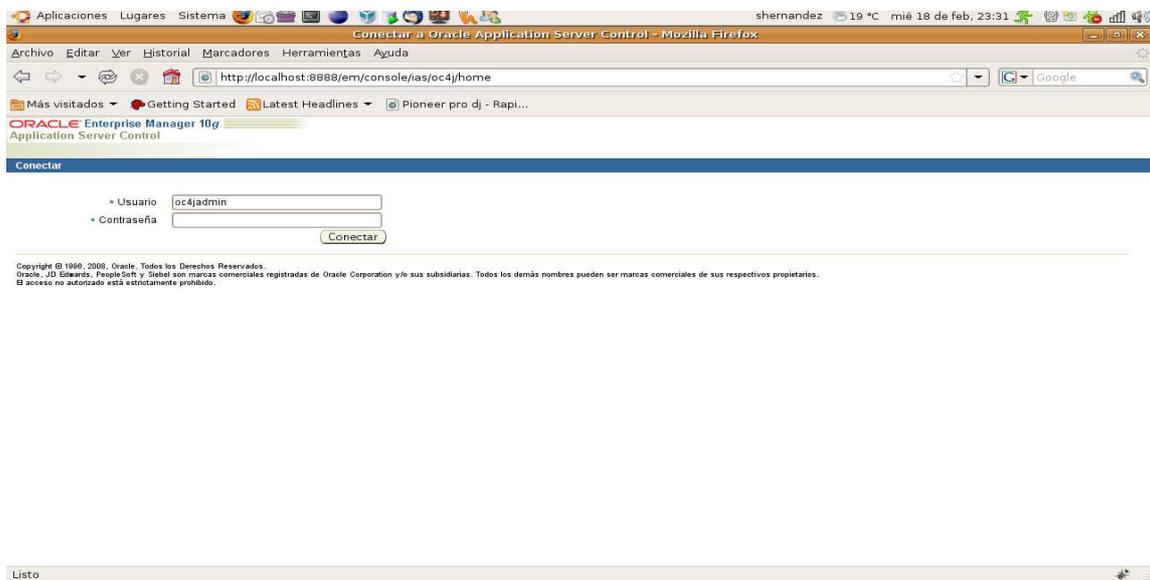
```
shernandez@shernandez: ~/software/oc4j_2/bin
shernandez@shernandez:~$ cd software
shernandez@shernandez:~/software$ cd oc4j_2
shernandez@shernandez:~/software/oc4j_2$ cd bin/
shernandez@shernandez:~/software/oc4j_2/bin$ ./oc4j -start
Starting OC4J from /home/shernandez/software/oc4j_2/j2ee/home ...
Listening for transport dt_socket at address: 8080
]
```



```
shernandez@shernandez: ~/software/oc4j_2/bin
Archivo Editar Ver Terminal Solapas Ayuda
#4 /home/shernandez/software/jdk1.5.0_16/jre/lib/i386/xawt/libmawt.so [0x7cb15d77]
#5 /home/shernandez/software/jdk1.5.0_16/jre/lib/i386/xawt/libmawt.so [0x7cb15ef3]
#6 /home/shernandez/software/jdk1.5.0_16/jre/lib/i386/xawt/libmawt.so(Java_sun_awt_X11GraphicsEnvironment_initDisplay+0x26) [0x7cb16136]
#7 [0xb134e2bb]
#8 [0xb1345b6b]
#9 [0xb1345b6b]
#10 [0xb1343236]
#11 /home/shernandez/software/jdk1.5.0_16/jre/lib/i386/server/libjvm.so [0xb760feec]
#12 /home/shernandez/software/jdk1.5.0_16/jre/lib/i386/server/libjvm.so [0xb77dfae8]
#13 /home/shernandez/software/jdk1.5.0_16/jre/lib/i386/server/libjvm.so [0xb760fd1f]
#14 /home/shernandez/software/jdk1.5.0_16/jre/lib/i386/server/libjvm.so(JVM_DoPrivileged+0x32d) [0xb766d82d]
#15 /home/shernandez/software/jdk1.5.0_16/jre/lib/i386/libjava.so(Java_java_security_AccessController_doPrivileged_Ljava_security_PrivilegedAction_2+0x3d) [0xb72e236d]
#16 [0xb134d7fa]
#17 [0xb1345a94]
#18 [0xb1343236]
#19 /home/shernandez/software/jdk1.5.0_16/jre/lib/i386/server/libjvm.so [0xb760feec]
Locking assertion failure. Backtrace:
#0 /usr/lib/libxcb-xlib.so.0 [0x80c00767]
#1 /usr/lib/libxcb-xlib.so.0(xcb_xlib_lock+0x2e) [0x80c0081e]
#2 /usr/lib/libX11.so.6 [0x7bcd9518]
#3 /usr/lib/libX11.so.6(XGetVisualInfo+0x26) [0x7bcd00a6]
#4 /home/shernandez/software/jdk1.5.0_16/jre/lib/i386/xawt/libmawt.so [0x7cb150b9]
#5 /home/shernandez/software/jdk1.5.0_16/jre/lib/i386/xawt/libmawt.so [0x7cb15303]
#6 /home/shernandez/software/jdk1.5.0_16/jre/lib/i386/xawt/libmawt.so [0x7cb15fa1]
#7 /home/shernandez/software/jdk1.5.0_16/jre/lib/i386/xawt/libmawt.so(Java_sun_awt_X11GraphicsEnvironment_initDisplay+0x26) [0x7cb16136]
#8 [0xb134e2bb]
#9 [0xb1345b6b]
#10 [0xb1345b6b]
#11 [0xb1343236]
#12 /home/shernandez/software/jdk1.5.0_16/jre/lib/i386/server/libjvm.so [0xb760feec]
#13 /home/shernandez/software/jdk1.5.0_16/jre/lib/i386/server/libjvm.so [0xb77dfae8]
#14 /home/shernandez/software/jdk1.5.0_16/jre/lib/i386/server/libjvm.so [0xb760fd1f]
#15 /home/shernandez/software/jdk1.5.0_16/jre/lib/i386/server/libjvm.so(JVM_DoPrivileged+0x32d) [0xb766d82d]
#16 /home/shernandez/software/jdk1.5.0_16/jre/lib/i386/libjava.so(Java_java_security_AccessController_doPrivileged_Ljava_security_PrivilegedAction_2+0x3d) [0xb72e236d]
#17 [0xb134d7fa]
#18 [0xb1345a94]
#19 [0xb1343236]
09/02/18 23:27:42 Oracle Containers for J2EE 10g (10.1.3.4.0) initialized
```

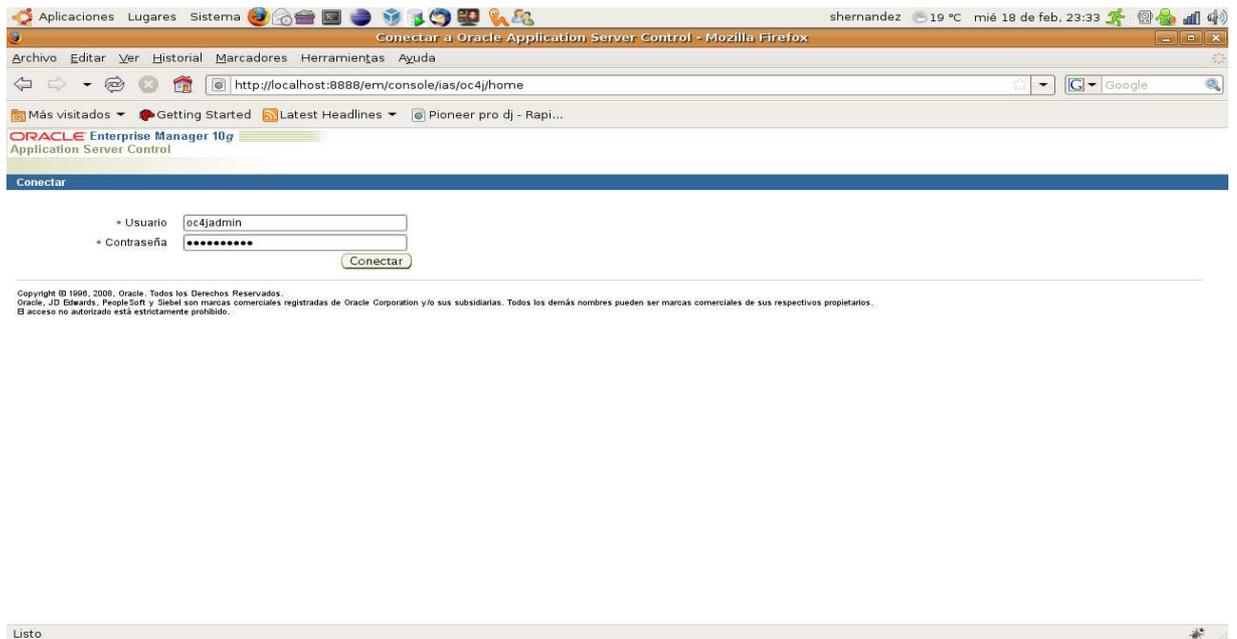
Ingresar a la consola del servidor en la siguiente URL: <http://<IP del servidor OAS>:<Puerto OAS>/em>

Por ejemplo: <http://localhost:8888/em>



Proporcionar el usuario y password de acceso para ingresar. Por ejemplo:

- Usuario= oc4jadmin
- Password= password definido al momento de instalar el OAS



Sistema de Administración de Equipo de Cómputo

Fundación UNAM

The screenshot shows the Oracle Enterprise Manager (OEM) console interface. The browser window title is "Oracle Enterprise Manager (oc4jadmin) - OC4J: home - Mozilla Firefox". The address bar shows "http://localhost:8888/em/console/ias/oc4j/home". The page title is "ORACLE Enterprise Manager 10g Application Server Control". The main content area is titled "OC4J: home" and includes a navigation bar with "Inicio", "Aplicaciones", "Servicios Web", "Rendimiento", and "Administración". The "Aplicaciones" menu item is highlighted. Below the navigation bar, there is a "General" section with a status icon and buttons for "Parar" and "Reiniciar". The status information includes: Estado: Arriba, Hora de Inicio: 18-feb-2009 23H27' CST, Versión: 10.1.3.4.0, Directorio Raíz de Oracle: home/shernandez/software/oc4j_2, Host: localhost, Máquinas Virtuales: 1, and Notificaciones: 0. To the right of the status information is a "Respuesta y Carga" section with a performance graph showing "Tiempo de Procesamiento de Solicitud (segundos)" and "Solicitudes por Segundo" over time. The graph shows a peak in processing time around 21:34:20 on 18-feb-2009. At the bottom of the page, there is a copyright notice: "Copyright © 1996, 2008, Oracle. Todos los Derechos Reservados. Oracle, JD Edwards, PeopleSoft y Siebel son marcas comerciales registradas de Oracle Corporation y/o sus subsidiarias. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios. Fuente de Oracle Enterprise Manager 10g Application Server Control".

En el home del OAS, ingresar a la opción identificada como Aplicaciones.

Sistema de Administración de Equipo de Cómputo

Fundación UNAM

The screenshot shows the Oracle Enterprise Manager 10g Application Server Control interface for the OC4J: home component. The browser address bar shows `http://localhost:8888/em/console/ias/oc4j/home`. The page title is "OC4J: home". The navigation menu includes "Inicio", "Aplicaciones", "Servicios Web", "Rendimiento", and "Administración". The "Aplicaciones" tab is selected and circled in black. The "General" section shows the component status as "Arriba" (Up) with a green checkmark icon. The "Parar" (Stop) and "Reiniciar" (Restart) buttons are visible. The "Respuesta y Carga" (Response and Load) section contains a line graph showing "Tiempo de Procesamiento de Solicitud (segundos)" (Request Processing Time) and "Solicitudes por Segundo" (Requests per Second) over time. The graph shows a flat line at 0.0 on the y-axis (0.0 to 0.3) and a flat line at 0 on the x-axis (21:34:00 to 20:00:20 on 18-feb-2009). The footer includes copyright information for Oracle and the text "Listo" in the bottom left corner.

En la opción de Aplicaciones, seleccionar la operación identificada como “Desplegar”

Sistema de Administración de Equipo de Cómputo

Fundación UNAM

OC4J: home

Esta página muestra las aplicaciones J2EE y los componentes de aplicación (Módulos EJB, Módulos WAR, Módulos de Adaptador de Recursos) desplegados para esta instancia de OC4J.

Ver: **Aplicaciones**

Seleccionar	Nombre	Estado	Hora de Inicio	Solicitudes Activas	Tiempo de Procesamiento de Solicitud (segundos)	Métodos de EJB Activos	MBeans Definidos por la Aplicación
<input type="checkbox"/>	▼ Todas las Aplicaciones						
<input type="checkbox"/>	ascontrol	↑	18-feb-2009 23H27: CST	1	0,12	0	
<input type="checkbox"/>	▼ default	↑	18-feb-2009 23H27: CST	0	0,00	0	
<input type="checkbox"/>	▶ Servicios de Software Intermedio						

CONSEJO Si para una aplicación principal (como la aplicación por defecto), Enterprise Manager parará automáticamente todas las aplicaciones secundarias que dependen de ella. Del mismo modo, si inicia una aplicación secundaria, Enterprise Manager iniciará automáticamente la aplicación principal necesaria.

Copyright © 1996, 2008, Oracle. Todos los Derechos Reservados.
Oracle, JD Edwards, PeopleSoft y Siebel son marcas comerciales registradas de Oracle Corporation y/o sus subsidiarias. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.
Agencia de Oracle Enterprise Manager 10g Application Server Control

A continuación se presentará la pantalla del primer paso para generar el despliegue de la aplicación.

Desplegar: Seleccionar Archivado

Paso 1 de 3

Archivar
Se pueden desplegar los siguientes tipos de archivado: Aplicación J2EE (archivos EAR), Módulos Web (archivos WAR), Módulos EJB (archivos JAR de EJB) y Módulos de Adaptador de Recursos (archivos RAR).

El archivado está en el host local. Cargue el archivado en el servidor en el que se está ejecutando Application Server Control.
Ubicación del Archivado

El archivado ya está en el servidor en el que se está ejecutando Application Server Control.
Ubicación en el Servidor
La ubicación en el servidor debe ser la ruta de acceso absoluta o la ruta de acceso relativa de /j2ee/home

Plan de Despliegue
El plan de despliegue es un archivo XML que contiene los valores de despliegue de una aplicación. Si no tiene un plan de despliegue, se creará uno automáticamente durante el proceso de despliegue. Posteriormente en el plan de despliegue, puede editar y guardar opcionalmente el plan de despliegue para volver a utilizarlo en un próximo despliegue de esta aplicación.

Crear automáticamente un nuevo plan de despliegue.
Los valores del plan de despliegue se basarán en los valores por defecto de OC4J y en la información contenida en el archivado

El plan de despliegue está en el host local. Cargue el plan de despliegue en el servidor en el que se está ejecutando Application Server Control.
Ubicación del Plan

El plan de despliegue ya está en el servidor en el que se está ejecutando Application Server Control.
Ubicación en el Servidor
La ubicación en el servidor debe ser la ruta de acceso absoluta o la ruta de acceso relativa de /j2ee/home

Paso 1 de 3

Copyright © 1996, 2008, Oracle. Todos los Derechos Reservados.
Oracle, JD Edwards, PeopleSoft y Siebel son marcas comerciales registradas de Oracle Corporation y/o sus subsidiarias. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.
Agencia de Oracle Enterprise Manager 10g Application Server Control

Para continuar con el despliegue de la aplicación se debe oprimir la opción de Examinar para elegir el archivo que contiene la aplicación WEB. Se debe buscar el archivo **saiec.war** que fue generado como resultado de la compilación y generación de la aplicación.

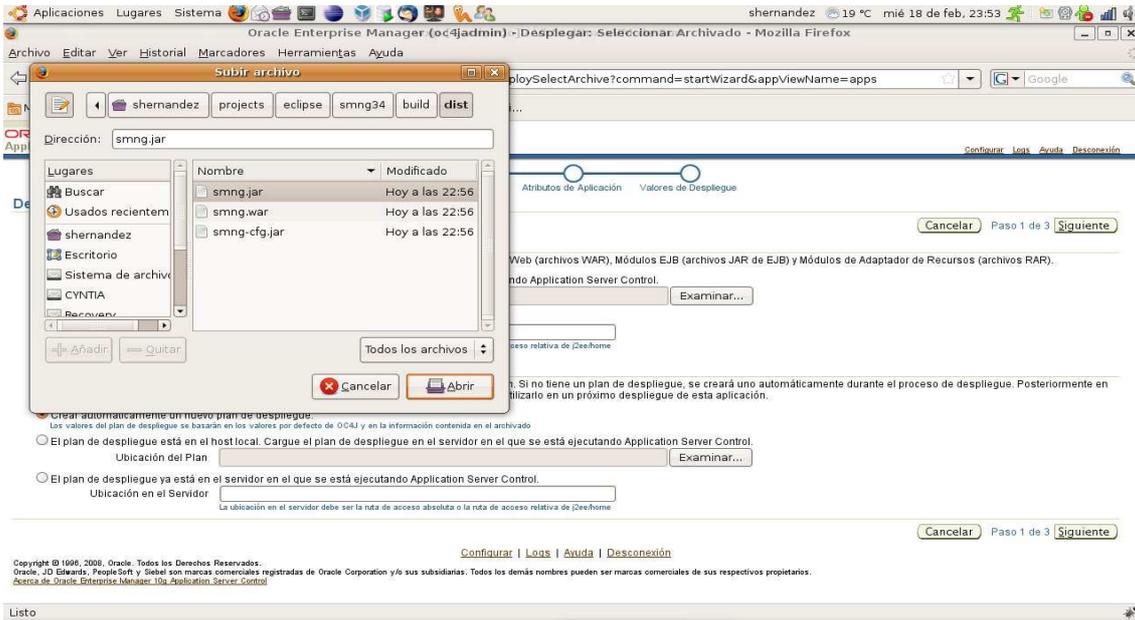
Por ejemplo:

`/home/shernandez/projects/eclipse/smng34/build/dist/saiec.war`

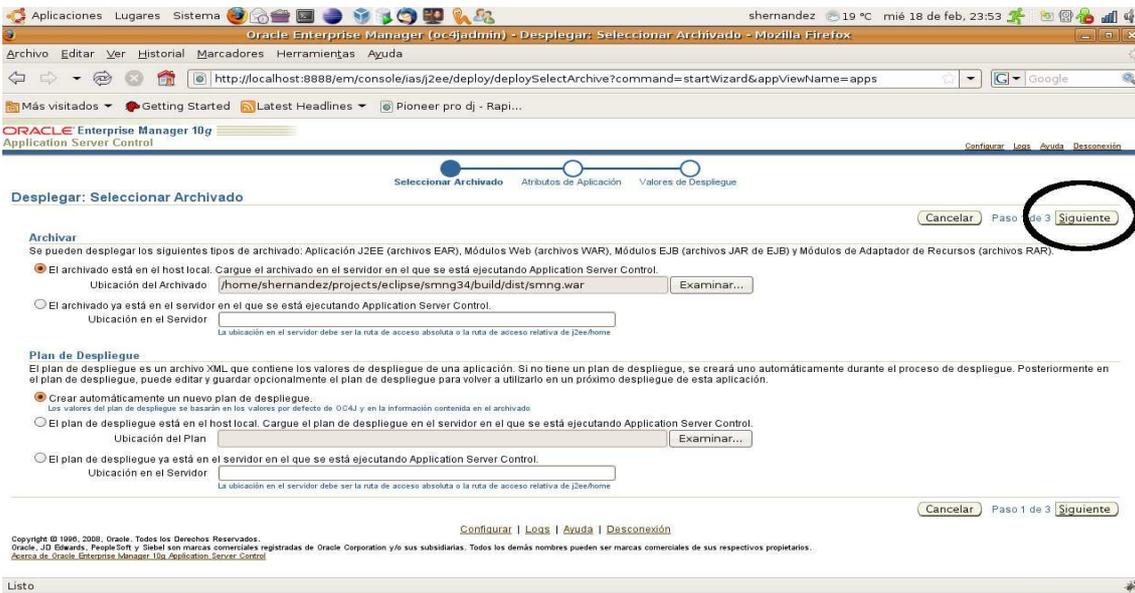
The screenshot shows the Oracle Enterprise Manager 10g Application Server Control interface. The title bar reads "ORACLE Enterprise Manager 10g Application Server Control". The breadcrumb navigation shows "Configurar | Inicio | Ayuda | Desconexión". The main navigation bar has three steps: "Seleccionar Archivado" (active), "Atributos de Aplicación", and "Valores de Despliegue". The current step is "Desplegar: Seleccionar Archivado", with "Cancelar" and "Paso 1 de 3 | Siguiente" buttons. The "Archivar" section has two radio buttons: "El archivado está en el host local. Cargue el archivado en el servidor en el que se está ejecutando Application Server Control." (selected) and "El archivado ya está en el servidor en el que se está ejecutando Application Server Control." Below the first option is a text field for "Ubicación del Archivado" with an "Examinar..." button circled in black. Below the second option is a text field for "Ubicación en el Servidor" with a note: "La ubicación en el servidor debe ser la ruta de acceso absoluta o la ruta de acceso relativa de J2eeHome". The "Plan de Despliegue" section has two radio buttons: "Crear automáticamente un nuevo plan de despliegue." (selected) and "El plan de despliegue está en el host local. Cargue el plan de despliegue en el servidor en el que se está ejecutando Application Server Control." Below the first option is a text field for "Ubicación del Plan" with an "Examinar..." button. Below the second option is a text field for "Ubicación en el Servidor" with a note: "La ubicación en el servidor debe ser la ruta de acceso absoluta o la ruta de acceso relativa de J2eeHome". At the bottom, there are "Configurar | Inicio | Ayuda | Desconexión" links, "Cancelar" and "Paso 1 de 3 | Siguiente" buttons, and a status bar that says "Listo".

Sistema de Administración de Equipo de Cómputo

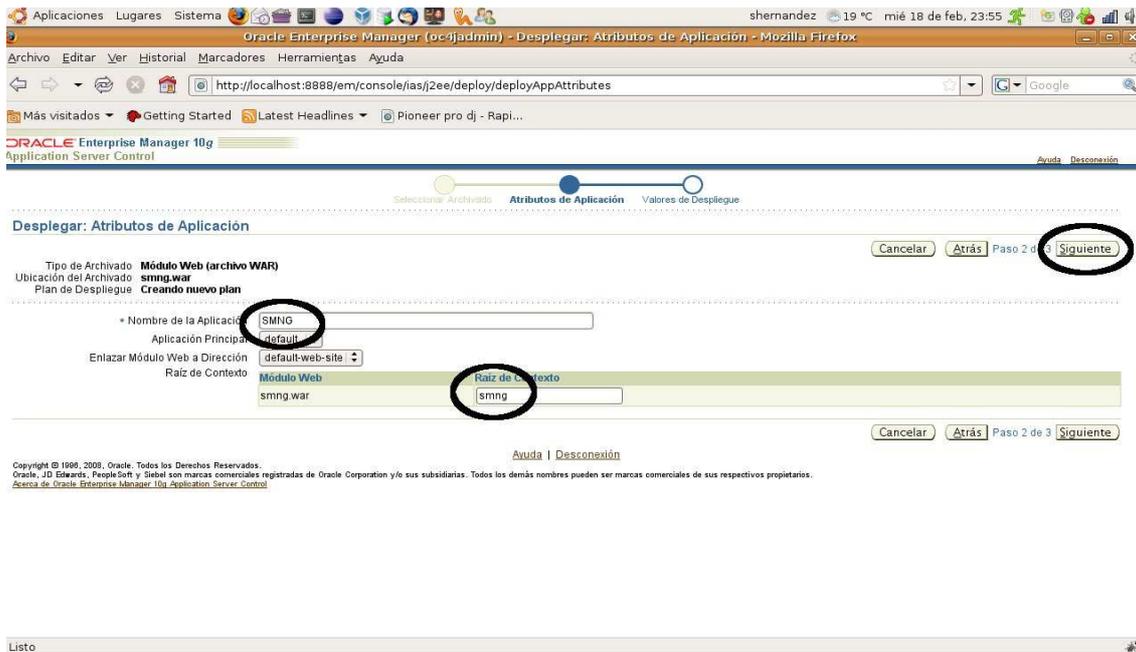
Fundación UNAM



Con el archivo referenciado seleccionar la opción “Siguiete”



A continuación se despliega la pantalla de Atributos de la Aplicación, en donde se solicita la captura del nombre de la aplicación para el nuevo plan a generar. El nombre de la aplicación a capturar debe ser **“saiec”**. Presionar la opción de siguiente.



A continuación se presenta la pantalla de los Valores de Despliegue. Se debe seleccionar la opción de “Configurar Carga de Clase” y desmarcar dentro de esta pantalla el campo “Heredar importaciones de bibliotecas compartidas de la aplicación principal” y seleccione la opción de “Aceptar”.

Sistema de Administración de Equipo de Cómputo

Fundación UNAM

Oracle Enterprise Manager 10g
Application Server Control

Desplegar: Valores de Despliegue

Cancelar Atrás Paso 3 de 3 Desplegar

Tipo de Archivarado: **Módulo Web (archivo WAR)**
 Ubicación del Archivarado: **smng.war**
 Plan de Despliegue: **Creando nuevo plan**

Nombre de la Aplicación: **SMNG**
 Aplicación Principal: **default**
 Enlazar Módulo Web a Dirección: **default-web-site**
 Raíz de Contexto: **smng**

Tareas de Despliegue
 La siguiente tabla proporciona un juego de tareas de despliegue comunes que puede que desee realizar para esta aplicación. Sólo están activadas las tareas que se aplican a la aplicación actual.

Nombre de la Tarea	Ir a Tarea	Descripción
Asignar Referencias de Entorno		Asigne las referencias de entorno de la aplicación (por ejemplo, los orígenes de datos) a entidades físicas presentes actualmente en el entorno operativo.
Seleccionar Proveedor de Seguridad		Un proveedor de seguridad actúa como el origen de los usuarios y los grupos disponibles durante la asignación de roles de seguridad.
Asignar Roles de Seguridad		Asigne los roles de seguridad mostrados por la aplicación a los usuarios y grupos existentes. La lista de usuarios y grupos se obtiene del proveedor de seguridad seleccionado para la aplicación.
Configurar EJB		Configure Enterprise JavaBean en la aplicación.
Configurar Clusters		Configure los clusters de la aplicación.
Configurar Carga de Clase		Manipule la configuración de la aplicación.

Edición de Plan de Despliegue Avanzada
 Haga clic en Editar Plan de Despliegue para definir opciones de despliegue más avanzadas. [Editar Plan de Despliegue](#)

Guardar Plan de Despliegue
 Después de realizar los cambios, puede guardar el plan de despliegue en el disco local. A continuación, puede utilizar el plan de despliegue guardado para volver a desplegar esta aplicación. [Guardar Plan de Despliegue](#)

Listo

Oracle Enterprise Manager 10g
Application Server Control

Valores de Despliegue: Configurar Carga de Clase

Cancelar Aceptar

Tipo de Archivarado: **Módulo Web (archivo WAR)**
 Ubicación del Archivarado: **smng.war**
 Plan de Despliegue: **Creando nuevo plan**

Nombre de la Aplicación: **SMNG**
 Aplicación Principal: **default**
 Enlazar Módulo Web a Dirección: **default-web-site**
 Raíz de Contexto: **smng**

Importar Bibliotecas Compartidas
 La siguiente tabla muestra las bibliotecas compartidas instaladas en esta instancia de OC4J. Seleccione Importar para declarar la dependencia de la aplicación de una biblioteca compartida. Opcionalmente, especifique la versión máxima o mínima para importarla.

Habilitar importaciones de bibliotecas compartidas de la aplicación principal
 CONSEJO Si está activada, los futuros cambios en las importaciones de bibliotecas compartidas de la aplicación principal se aplicarán a esta aplicación.

Biblioteca Compartida	Versiones Disponibles	Versión Mínima a Usar	Versión Máxima a Usar	Importado por Aplicación Principal	Importar
apache.commons.logging	1.0.4	<input type="text"/>	<input type="text"/>		<input type="checkbox"/>
global.libraries	1.0	<input type="text"/>	<input type="text"/>	✓	<input checked="" type="checkbox"/>
global.tag.libraries	1.0	<input type="text"/>	<input type="text"/>	✓	<input checked="" type="checkbox"/>
oracle.cache	10.1.3	<input type="text"/>	<input type="text"/>	✓	<input checked="" type="checkbox"/>
oracle.dms	3.0	<input type="text"/>	<input type="text"/>	✓	<input checked="" type="checkbox"/>
oracle.gdk	10.1.0_2	<input type="text"/>	<input type="text"/>	✓	<input checked="" type="checkbox"/>
oracle.http.client	10.1.3	<input type="text"/>	<input type="text"/>	✓	<input checked="" type="checkbox"/>
oracle.jdbc	10.1.0_2	<input type="text"/>	<input type="text"/>	✓	<input checked="" type="checkbox"/>

Listo

Sistema de Administración de Equipo de Cómputo

Fundación UNAM

Valores de Despliegue: Configurar Carga de Clase

Tipo de Archivarado: **Módulo Web (archivo WAR)**
 Ubicación del Archivarado: **smng.war**
 Plan de Despliegue: **Creando nuevo plan**

Nombre de la Aplicación: **SMNG**
 Aplicación Principal: **default**
 Enlazar Módulo Web a Dirección: **default-web-site**
 Raíz de Contexto: **smng**

Importar Bibliotecas Compartidas
 La siguiente tabla muestra las bibliotecas compartidas instaladas en esta instancia de OC4J. Seleccione Importar para declarar la dependencia de la aplicación de una biblioteca compartida. Opcionalmente, especifique una versión máxima o mínima para importarla.

Heredar importaciones de bibliotecas compartidas de la aplicación principal
 CONSEJO Si está activada, los futuros cambios en las importaciones de bibliotecas compartidas de la aplicación principal se aplicarán a esta aplicación.

Biblioteca Compartida	Versiones Disponibles	Versión Mínima a Usar	Versión Máxima a Usar	Importado por Aplicación Principal	Importar
apache.commons.logging	1.0.4				<input type="checkbox"/>
global.libraries	1.0			✓	<input type="checkbox"/>
global.tag.libraries	1.0			✓	<input type="checkbox"/>
oracle.cache	10.1.3			✓	<input type="checkbox"/>
oracle.dms	3.0			✓	<input type="checkbox"/>
oracle.gdk	10.1.0_2			✓	<input type="checkbox"/>
oracle.http.client	10.1.3			✓	<input type="checkbox"/>
oracle.jdbc	10.1.0_2			✓	<input type="checkbox"/>

Valores de Despliegue: Configurar Carga de Clase

Tipo de Archivarado: **Módulo Web (archivo WAR)**
 Ubicación del Archivarado: **smng.war**
 Plan de Despliegue: **Creando nuevo plan**

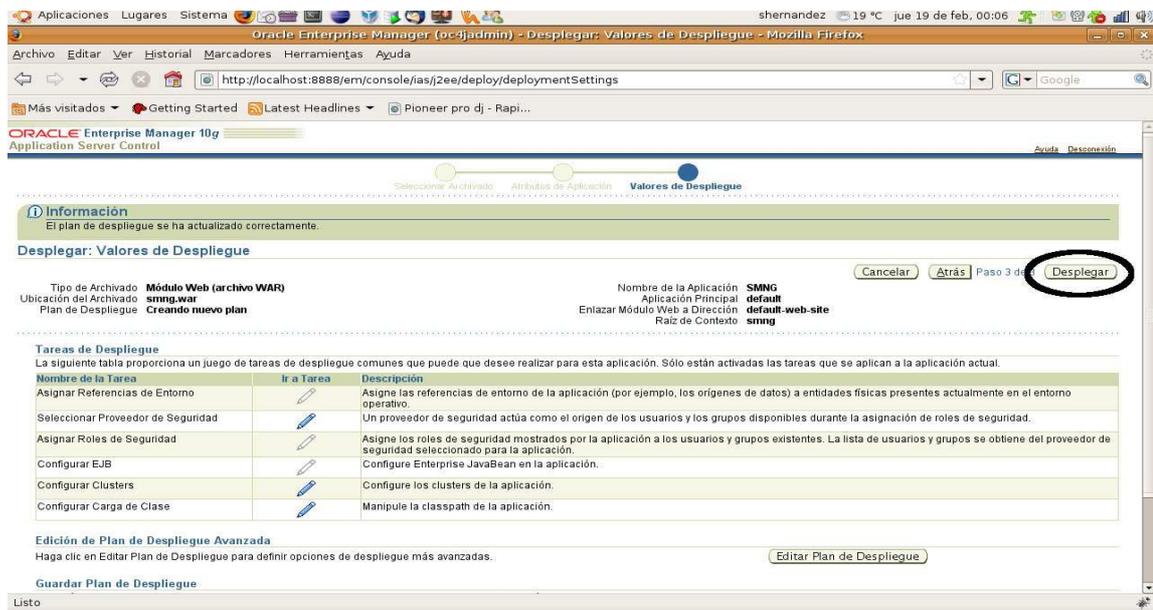
Nombre de la Aplicación: **SMNG**
 Aplicación Principal: **default**
 Enlazar Módulo Web a Dirección: **default-web-site**
 Raíz de Contexto: **smng**

Importar Bibliotecas Compartidas
 La siguiente tabla muestra las bibliotecas compartidas instaladas en esta instancia de OC4J. Seleccione Importar para declarar la dependencia de la aplicación de una biblioteca compartida. Opcionalmente, especifique una versión máxima o mínima para importarla.

Heredar importaciones de bibliotecas compartidas de la aplicación principal
 CONSEJO Si está activada, los futuros cambios en las importaciones de bibliotecas compartidas de la aplicación principal se aplicarán a esta aplicación.

Biblioteca Compartida	Versiones Disponibles	Versión Mínima a Usar	Versión Máxima a Usar	Importado por Aplicación Principal	Importar
apache.commons.logging	1.0.4				<input type="checkbox"/>
global.libraries	1.0			✓	<input type="checkbox"/>
global.tag.libraries	1.0			✓	<input type="checkbox"/>
oracle.cache	10.1.3			✓	<input type="checkbox"/>
oracle.dms	3.0			✓	<input type="checkbox"/>
oracle.gdk	10.1.0_2			✓	<input type="checkbox"/>
oracle.http.client	10.1.3			✓	<input type="checkbox"/>
oracle.jdbc	10.1.0_2			✓	<input type="checkbox"/>

De regreso en la pantalla de Valores de Despliegue, se elige la opción “Desplegar” para concluir con el despliegue de la aplicación.



Por ultimo, se debe presentar la pantalla que indica que la aplicación ha sido desplegada correctamente.

Sistema de Administración de Equipo de Cómputo

Fundación UNAM

Oracle Enterprise Manager 10g
Application Server Control

OC4J: home

Inicio **Aplicaciones** Servicios Web Rendimiento Administración

General

Estado: **Arriba** [Parar] [Reiniciar]

Hora de inicio: 18-feb-2009 23H27' CST

Versión: 10.1.3.4.0

Directorio Raíz de Oracle: /home/shernandez/software/oc4j_2

Host: localhost

Máquinas Virtuales: 1

Notificaciones: 0

Respuesta y Carga

22:13:42:03:23:40:00
18-feb-2009 19

■ Tiempo de Procesamiento de Solicitud (segundos)
■ Solicitudes por Segundo

Configurar | Logs | Ayuda | Desconexión

Copyright © 1996, 2008, Oracle. Todos los Derechos Reservados.
Oracle, JD Edwards, PeopleSoft y Siebel son marcas comerciales registradas de Oracle Corporation y/o sus subsidiarias. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.
Acersa de Oracle Enterprise Manager 10g Application Server Control

Listo

Oracle Enterprise Manager 10g
Application Server Control

OC4J: home

Inicio Aplicaciones **Servicios Web** Rendimiento Administración

Esta página muestra las aplicaciones J2EE y los componentes de aplicación (Módulos EJB, Módulos WAR, Módulos de Adaptador de Recursos) desplegados para esta instancia de OC4J.

Ver: Aplicaciones

[Iniciar] [Parar] [Reiniciar] [Anular Despliegue] [Volver a Desplegar] | [Desplegar]

Seleccionar Todo | No Seleccionar Nada | Ampliar Todo | Reducir Todo

Seleccionar	Nombre	Estado	Hora de Inicio	Solicitudes Activas	Tiempo de Procesamiento de Solicitud (segundos)	Métodos de EJB Activos	MBeans Definidos por la Aplicación
<input type="checkbox"/>	▼ Todas las Aplicaciones						
<input type="checkbox"/>	ascontrol	↑	18-feb-2009 23H27' CST	1	0,00	0	
<input type="checkbox"/>	▼ default						
<input type="checkbox"/>	default	↑	18-feb-2009 23H27' CST	0	0,00	0	
<input type="checkbox"/>	SMNG	↑	18-feb-2009 00H09' CST	0	0,00	0	
<input type="checkbox"/>	▶ Servicios de Software Intermedio						

CONSEJO Si para una aplicación principal (como la aplicación por defecto), Enterprise Manager parará automáticamente todas las aplicaciones secundarias que dependan de ella. Del mismo modo, si inicia una aplicación secundaria, Enterprise Manager iniciará automáticamente la aplicación principal necesaria.

Configurar | Logs | Ayuda | Desconexión

Copyright © 1996, 2008, Oracle. Todos los Derechos Reservados.
Oracle, JD Edwards, PeopleSoft y Siebel son marcas comerciales registradas de Oracle Corporation y/o sus subsidiarias. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.
Acersa de Oracle Enterprise Manager 10g Application Server Control

Listo

De la lista de aplicaciones mostradas, marcar la aplicación **saiec** y seleccionar la opción identificada como “Volver a Desplegar”.

Sistema de Administración de Equipo de Cómputo

Fundación UNAM

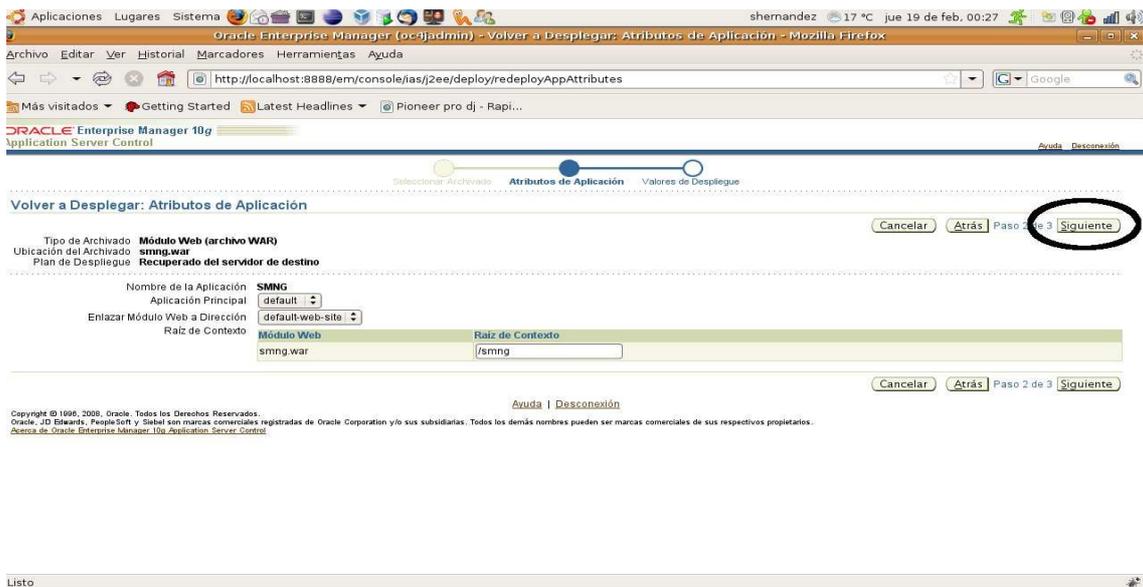
The screenshot shows the Oracle Enterprise Manager (OC4J) console interface. At the top, there are navigation tabs: Inicio, Aplicaciones, Servicios Web, Rendimiento, and Administración. Below the tabs, a text box states: "Esta página muestra las aplicaciones J2EE y los componentes de aplicación (Módulos EJB, Módulos WAR, Módulos de Adaptador de Recursos) desplegados para esta instancia de OC4J." Below this, there are several action buttons: Iniciar, Parar, Reiniciar, Anular Despliegue, Volver a Desplegar, and Desplegar. The "Desplegar" button is circled in red. Below the buttons is a table with columns: Seleccionar, Nombre, Estado, Hora de Inicio, Solicitudes Activas, Tiempo de Procesamiento de Solicitud (segundos), Métodos de EJB Activos, and MBeans Definidos por la Aplicación. The table lists three applications: "ascontrol", "default", and "SIMJG". The "SIMJG" application has a checked checkbox in the "Seleccionar" column, which is also circled in red. Below the table, there is a "CONSEJO" section with a warning icon and text: "Si para una aplicación principal (como la aplicación por defecto), Enterprise Manager parará automáticamente todas las aplicaciones secundarias que dependen de ella. Del mismo modo, si inicia una aplicación secundaria, Enterprise Manager iniciará automáticamente la aplicación principal necesaria." At the bottom of the console, there are links for Configuración, Logs, Ayuda, and Desconexión, and a footer with copyright information.

Seleccionar	Nombre	Estado	Hora de Inicio	Solicitudes Activas	Tiempo de Procesamiento de Solicitud (segundos)	Métodos de EJB Activos	MBeans Definidos por la Aplicación
<input type="checkbox"/>	▼ Todas las Aplicaciones						
<input type="checkbox"/>	ascontrol	↑	18-feb-2009 23H27' CST	1	0,11	0	
<input type="checkbox"/>	▼ default	↑	18-feb-2009 23H27' CST	0	0,00	0	
<input checked="" type="checkbox"/>	SIMJG	↑	18-feb-2009 00H09' CST	0	0,00	0	
<input type="checkbox"/>	► Servicios de Software Intermedio						

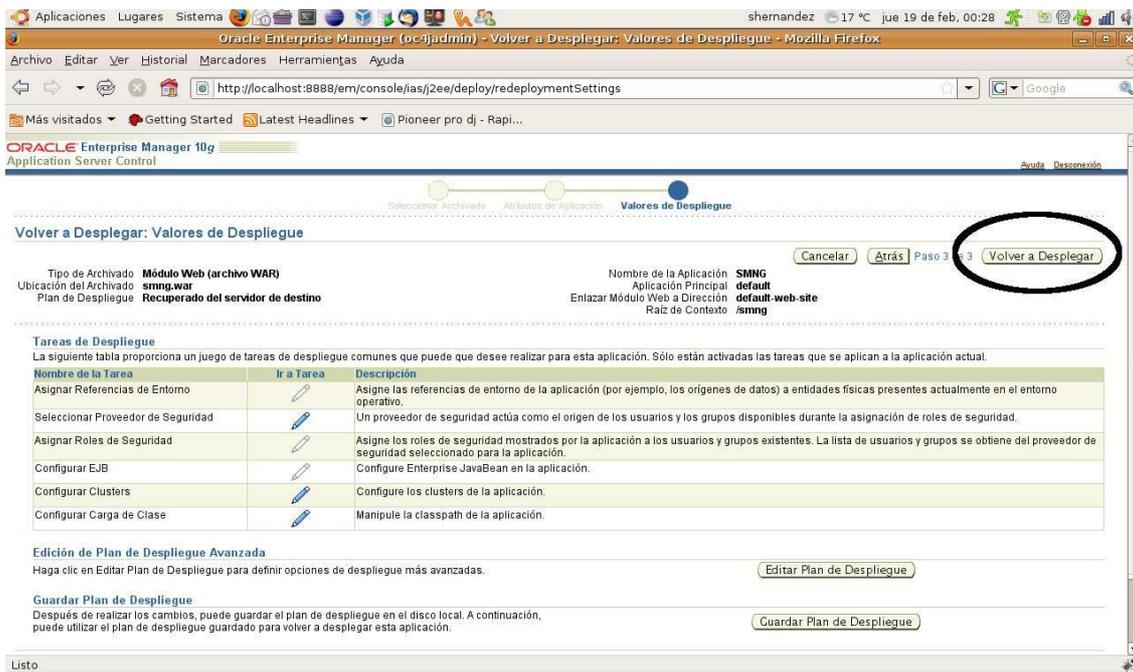
En la pantalla de Despliegue en la opción de Examinar, buscar el archivo de la aplicación WEB (saiec.war), previamente generado, y seleccionar siguiente.



Se presenta a continuación la pantalla Atributos de Aplicación, para el caso de re instalación de la aplicación no se modifica información y se debe seleccionar la opción "Siguiente".



Se presenta la pantalla de Valores de Despliegue, en esta nuevamente no se modifica información y se debe seleccionar la opción “Volver a Desplegar”.



Por ultimo, se presenta la pantalla que indica que la aplicación ha sido desplegada correctamente

