



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

DESARROLLO DEL SISTEMA AUTOMATIZADO DE FORMAS UNICAS PARA LA CONTRATACIÓN DE PERSONAL ACADÉMICO

PROYECTO DE

T E S I S

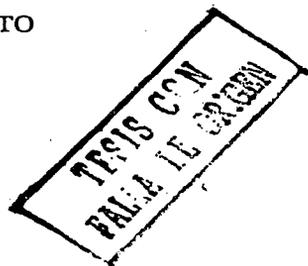
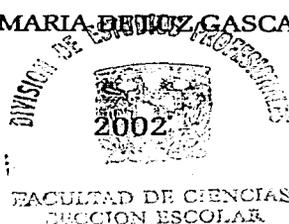
QUE PARA OBTENER EL TÍTULO DE LIC. EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A :

ROBERTO ROSENBERG GONZÁLEZ

DIRECTOR DE TESIS:

M. EN I. MARIA DE LOS RÍOS CASCA SOTO





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.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

M. EN C. ELENA DE OTEYZA DE OTEYZA

Jefa de la División de Estudios Profesionales de la
Facultad de Ciencias

Presente

Comunicamos a usted que hemos revisado el trabajo escrito:

**"Desarrollo del Sistema Automatizado de Formas Unicas para la
Contratación del Personal Académico".**

realizado por Rosenberg González Roberto

con número de cuenta 9561605-0 , quién cubrió los créditos de la carrera de
Ciencias de la Computación.

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis
Propietario

M. en I. María de Luz Gasca Soto

Propietario

M. en C. María Guadalupe Elena Ibarquengoitia González

Propietario

Dr. Luis Fernando Magaña Solís

Suplente

Dra. Amparo López Gaona

Suplente

M. en C. Javier García García

Consejo Departamental de

M. en C. María Guadalupe Elena Ibarquengoitia González

FACULTAD DE CIENCIAS

CONSEJO DEPARTAMENTAL DE

MATEMÁTICAS

**Desarrollo del Sistema Automatizado de
Formas Únicas para la Contratación
de Personal Académico (SAFU)**

SIA 2000

" Sistema Integral de Administración 2000 "

Roberto Rosenberg González

23 enero, 2002

Copyright ©2001 Universidad Nacional Autónoma de México, Fac. de Ciencias;
Roberto Rosenberg González y todos los contribuyentes de SIA2000. Derechos
Reservados.

Agradecimientos

La mayor parte del material usado en este proyecto viene de dos libros de Bases de Datos:

Abraham Silberschatz/Henry F. Korth/S. Sudarshan <no email>
Database System Concepts. Third Edition. McGraw-Hill

C. J. Date <no email>
An Introduction to Database Systems. 7th Edition. Addison-Wesley

Si usted está interesado en los fuentes de este documento, puede encontrar una versión actualizada para L^AT_EX 2_ε por Roberto Rosenberg González en <http://sonim.fciencias.unam.mx/~roberto/SAFU>

Mientras desarrollaba este proyecto, hubo gente que siempre me ayudó. En todo momento tuve un gran apoyo. Las siguientes personas me ayudaron con las correcciones, apoyo moral y sugerencias para mejorar este proyecto. Ellas pusieron un gran esfuerzo para ayudar a que este proyecto estuviera en el estado actual. Quiero agradecer sinceramente a todos ellos. Naturalmente, todos los errores que ustedes puedan encontrar en este proyecto son míos.

Dr. Luis Fernando Magaña Solís, M. en I. María De Luz Gasca Soto, M. en C. María Guadalupe E. Ibarguengoitia González, Dra. Amparo López Gaona, M. en C. Javier García García, M. en C. Virginia Abrín Batule, C.P. Carlos Angeles Medina, José Antonio López Saucedo, Mat. Victor Hugo Dorantes González, Mat. Margarita Elvira Chávez Cano, Mario Arturo Pérez Rangel, Norma Cordero Sánchez, Soledad Bárcenas del Olmo, Act. José Antonio Climent Hernández, Pablo Rosell González, Renato Olivares Becerril, Edith Palomera Mancilla, Sergio Ramos Chávez, Rosa María Flores Rojas, María del Carmen Monroy Nieto, Alicia Hernández Rojas, Nelly Laura Reyes Quiroz y María de los Angeles Sánchez Hernández.

Como una pequeña muestra de mi cariño, agradecimiento y amor infinito, por darme la oportunidad de vivir, por el apoyo brindado, y porque hoy veo llegar a su fin una de las metas de mi vida, le agradezco a mi madre donde quiera que se encuentre la orientación y consejos que siempre me supo dar para salir adelante.

Gracias a mi padre por ayudarme a hacer posible un logro más; el cual no será el último pero quizá el más importante. Gracias por la fe que depositaste en mi y por dármelo todo sin esperar nada a cambio más que el orgullo de hacerme un triunfador. Gracias.

Por último, quiero agradecer a Dios, ya que en silencio me ha acompañado a lo largo de mi vida y sin pedirme nada a cambio hoy me regala la alegría de haber realizado uno más de mis anhelos y la oportunidad de ser alguien importante en la vida. Guarda mi corazón cerca de ti y guíame día con día en el camino que lleva hacia ti.

Roberto Rosenberg González

"Yo soy yo, pero yo sin el otro no soy el que fui, ni seré el que sería de continuar con el otro."

Resumen

El presente documento es el desarrollo de un sistema que le permitirá al Departamento de Personal Académico de la Facultad de Ciencias cumplir con sus funciones y procesos de forma automatizada. Se provee una introducción detallada al campo del desarrollo de software a través del uso de metodologías y herramientas que facilitan la creación de sistemas, así como también la teoría de sistemas de bases de datos necesaria para integrar un sistema con alguna base de datos.

Se comienza el desarrollo considerando de forma muy importante a las metodologías de ingeniería de software, y de la misma forma se aplica la teoría de bases de datos, la de ingeniería de hardware y la ingeniería humana para crear un sistema robusto y de alta calidad.

Finalmente se presenta el sistema terminado con sus posibles problemas y soluciones, además de considerar nuevas mejoras al sistema.

Índice General

Prefacio

xvii

1	Introducción	1
1.1	Definiciones	1
1.1.1	Sistema	1
1.1.2	Software y Hardware	2
1.1.3	Usuarios y Documentación	3
1.1.4	Bases de Datos y Procedimientos	4
1.2	Ingeniería de Software	5
1.3	Ingeniería de Hardware	8
1.4	Ingeniería Humana	8
1.5	Consideraciones	9
2	Ingeniería de Software	13
2.1	Ingeniería de Software	13
2.1.1	Ciclo de Vida	14
2.1.2	Modelos del Proceso de Desarrollo de Software	17
2.1.3	Administración en el Desarrollo de Software	21
2.1.4	Métrica en el Software	23
2.2	Diseño un Sistema	24
2.2.1	UML	25
3	Bases de Datos	33
3.1	Sistemas de Bases de Datos	33
3.2	Base de Datos	37
3.2.1	Ventajas	37
3.2.2	Independencia de Datos	38
3.3	Modelo Relacional y otros Modelos	39
3.4	Lenguajes de Base de Datos	43
3.4.1	Lenguaje de Definición de Datos	43
3.4.2	Lenguaje de Manipulación de Datos	44
3.4.3	SQL	44
3.5	Sistema de Administración de Base de Datos	45

3.6	Estructura de Datos Relacional	47
3.7	Base de Datos Relacionales	49
4	Análisis del Sistema	51
4.1	Origen del Sistema	51
4.2	Introducción	52
4.3	Especificación de Requerimientos	52
4.3.1	Requerimientos de Software	57
4.3.2	Requerimientos de Interfaz	58
4.3.3	Requerimientos de Seguridad	58
4.3.4	Restricciones del Sistema	59
4.3.5	Requerimientos Especiales del Sistema	59
4.3.6	Referencias y Fuentes de Información	59
4.3.7	Diagramas de Casos de Uso	59
4.3.8	Detalle de Casos de Uso	66
4.4	Análisis	94
4.4.1	Diagramas de Paquetes y Clases	94
4.4.2	Modelo de Seguridad de Datos	98
4.4.3	Diagramas de Paquetes y Capas	99
5	Diseño del Sistema	101
5.1	Definición de Políticas y Tácticas Generales	101
5.2	Definición de Nodos del Sistema	102
5.3	Definición de Subsistemas	103
5.4	Definición de Clases	104
5.5	Mecanismos Genéricos de Diseño	104
5.6	Posibilidades de Re-Uso	104
5.7	Diagramas de Diseño	104
5.7.1	Diagrama de Paquetes y Capas	105
5.7.2	Diagramas de Instalación	109
5.7.3	Diagramas de Secuencia	110
5.7.4	Diagramas Entidad Relación	130
6	Prototipo del Sistema	133
6.1	Prototipo	133
6.2	Plan de Pruebas	173
6.3	Problemas y Soluciones	174
	Conclusiones	177
A	Detalle de Casos de Uso	179
B	Diagramas de Secuencia	209
	Bibliografía	233

Índice de Figuras

1.1	Hardware más común de una computadora.	3
2.1	Modelo Cascada.	19
2.2	Modelo de Prototipos.	20
2.3	Modelo de Mejoramiento Iterativo.	21
2.4	Modelo de Espiral.	22
2.5	Definición de Clases.	26
2.6	Símbolo de Herencia.	27
2.7	Símbolo de Agregación/Composición.	28
2.8	Símbolo de Asociación.	28
2.9	Símbolo de Dependencia.	28
2.10	Símbolo de Actor.	29
2.11	Símbolo de Caso de Uso.	29
2.12	Símbolos de Relaciones.	30
2.13	Simbología del Diagrama de Interacción.	30
2.14	Simbología del Diagrama de Paquetes.	31
2.15	Simbología del Diagrama de Instalación.	31
3.1	Modelo Entidad Relación.	41
3.2	Modelo de Red.	42
3.3	Modelo Jerárquico.	43
3.4	Sistema Administrador de Bases de Datos.	47
4.1	Funcionamiento General de SAFU	60
4.2	Diagrama de Casos de SAFU 1er. Nivel	61
4.3	Diagrama de Casos de SAFU 2do. Nivel Opción "F1"	62
4.4	Diagrama de Casos de SAFU 2do. Nivel Opción "F2"	63
4.5	Diagrama de Casos de SAFU 2do. Nivel Opción "F3"	64
4.6	Diagrama de Casos de SAFU 2do. Nivel Opción "F4"	65
4.7	Diagrama de Paquetes de SAFU	94
4.8	Diagrama de Clases de SAFU (Vista General)	95
4.9	Diagrama de Interfaz Humana de SAFU	96
4.10	Diagrama de Dominio del Problema de SAFU	97
4.11	Modelo de Datos de SAFU	98

4.12	Modelo de Seguridad de SAFU	98
4.13	Diagrama de Arquitectura de SAFU	99
5.1	Definición de Arquitectura del SAFU	105
5.2	Diagrama de Paquetes y Capas Detallado del SAFU	106
5.3	Diagrama Detallado de la Capa Presentación	107
5.4	Diagrama Detallado de la Capa Dominio del Problema	108
5.5	Diagrama Detallado de la Capa Manejo de Datos	108
5.6	Diagrama de Instalación del SAFU	109
5.7	Diagrama de Secuencia F1.1 Seguridad Alta	110
5.8	Diagrama de Secuencia F1.2 Seguridad Alta	110
5.9	Diagrama de Secuencia F1.2 Seguridad Final	111
5.10	Diagrama de Secuencia F1.3 Seguridad Alta	111
5.11	Diagrama de Secuencia F1.3 Seguridad Final	112
5.12	Diagrama de Secuencia F1.4 Seguridad Alta	112
5.13	Diagrama de Secuencia F1.5 Seguridad Alta	113
5.14	Diagrama de Secuencia F1.6 Seguridad Alta	113
5.15	Diagrama de Secuencia F2.1 Seguridad Alta	114
5.16	Diagrama de Secuencia F2.2 Seguridad Alta	114
5.17	Diagrama de Secuencia F2.3 Seguridad Alta	115
5.18	Diagrama de Secuencia F2.4 Seguridad Alta	115
5.19	Diagrama de Secuencia F2.4 Seguridad Final	116
5.20	Diagrama de Secuencia F2.5 Seguridad Alta	116
5.21	Diagrama de Secuencia F2.6 Seguridad Alta	117
5.22	Diagrama de Secuencia F2.7 Seguridad Alta	117
5.23	Diagrama de Secuencia F2.8 Seguridad Alta	118
5.24	Diagrama de Secuencia F2.9 Seguridad Alta	118
5.25	Diagrama de Secuencia F3.1 Seguridad Alta	119
5.26	Diagrama de Secuencia F3.2 Seguridad Alta	119
5.27	Diagrama de Secuencia F3.3 Seguridad Alta	120
5.28	Diagrama de Secuencia F3.4 Seguridad Alta	120
5.29	Diagrama de Secuencia F3.5 Seguridad Alta	121
5.30	Diagrama de Secuencia F3.6 Seguridad Alta	121
5.31	Diagrama de Secuencia F3.7 Seguridad Alta	122
5.32	Diagrama de Secuencia F3.8 Seguridad Alta	122
5.33	Diagrama de Secuencia F3.9 Seguridad Alta	123
5.34	Diagrama de Secuencia F3.10 Seguridad Alta	123
5.35	Diagrama de Secuencia F3.11 Seguridad Alta	124
5.36	Diagrama de Secuencia F3.12 Seguridad Alta	124
5.37	Diagrama de Secuencia F3.13 Seguridad Alta	125
5.38	Diagrama de Secuencia F3.14 Seguridad Alta	125
5.39	Diagrama de Secuencia F3.15 Seguridad Alta	126
5.40	Diagrama de Secuencia F4.1 Seguridad Alta	126
5.41	Diagrama de Secuencia F4.1 Seguridad Final	127

5.42	Diagrama de Secuencia F4.2 Seguridad Alta	127
5.43	Diagrama de Secuencia F4.3 Seguridad Alta	128
5.44	Diagrama de Secuencia F4.3 Seguridad Final	128
5.45	Diagrama de Secuencia F4.4 Seguridad Alta	129
5.46	Diagrama de Secuencia F5 Seguridad Alta	129
5.47	Diagrama E-R de la Estructura del SAFU.	130
5.48	Diagrama E-R de la Seguridad del SAFU.	131
6.1	Sistema SIA2000	133
6.2	Sistema SAFU. Login y Password	134
6.3	Sistema SAFU. Validación Login y Password	134
6.4	Sistema SAFU. Menú Principal	135
6.5	Sistema SAFU. Menú Datos Generales	135
6.6	Sistema SAFU. Forma Alta de Datos Generales	136
6.7	Sistema SAFU. Validación Datos Generales 1	137
6.8	Sistema SAFU. Validación Datos Generales 2	138
6.9	Sistema SAFU. Agregando Datos Generales	139
6.10	Sistema SAFU. Forma Modificación Datos Generales	140
6.11	Sistema SAFU. Validación Modificación Datos Generales	140
6.12	Sistema SAFU. Modificación Datos Generales	141
6.13	Sistema SAFU. Modificando Datos Generales	142
6.14	Sistema SAFU. Forma Consulta Datos Generales	143
6.15	Sistema SAFU. Ejecutando Consulta Datos Generales	144
6.16	Sistema SAFU. Consulta Datos Generales	145
6.17	Sistema SAFU. Forma Alta Colonias	145
6.18	Sistema SAFU. Forma Alta Poblaciones	146
6.19	Sistema SAFU. Forma Alta Nacionalidades	146
6.20	Sistema SAFU. Menú Contratos	147
6.21	Sistema SAFU. Alta de Contratos	148
6.22	Sistema SAFU. Alta de Contratos 2	148
6.23	Sistema SAFU. Forma de Alta de Contratos	149
6.24	Sistema SAFU. Validación Alta de Contratos	150
6.25	Sistema SAFU. Agregando Contrato	151
6.26	Sistema SAFU. Generación de Forma Única	152
6.27	Sistema SAFU. Modificación de Contratos	153
6.28	Sistema SAFU. Validación de Modificación de Contratos 1	153
6.29	Sistema SAFU. Validación de Modificación de Contratos 2	154
6.30	Sistema SAFU. Generación de Contrato Modificado	155
6.31	Sistema SAFU. Eliminación de Contratos	156
6.32	Sistema SAFU. Ejecución de Eliminación de Contratos 1	157
6.33	Sistema SAFU. Ejecutando de Eliminación de Contratos 2	158
6.34	Sistema SAFU. Eliminando Contrato	159
6.35	Sistema SAFU. Consulta de Contratos	160
6.36	Sistema SAFU. Ejecutando Consulta de Contratos	161

6.37	Sistema SAFU. Impresión de Contratos	162
6.38	Sistema SAFU. Ejecutando Impresión de Contratos	163
6.39	Sistema SAFU. Alta de Claves y Categorías	164
6.40	Sistema SAFU. Alta de Asignaturas	164
6.41	Sistema SAFU. Alta de Sueldos	165
6.42	Sistema SAFU. Alta de Horarios	165
6.43	Sistema SAFU. Menú Catálogos	166
6.44	Sistema SAFU. Modificación de Colonias	167
6.45	Sistema SAFU. Modificación de Poblaciones	167
6.46	Sistema SAFU. Modificación de Nacionalidades	168
6.47	Sistema SAFU. Modificación de Claves y Categorías	168
6.48	Sistema SAFU. Modificación de Asignaturas	169
6.49	Sistema SAFU. Modificación de Sueldos	169
6.50	Sistema SAFU. Modificación de Horarios	170
6.51	Sistema SAFU. Ver Catálogos	171
6.52	Sistema SAFU. Ejecutando Ver Catálogos	172
6.53	Sistema SAFU. Menú Consultas	173
B.1	Diagrama de Secuencia F1.1 Seguridad Media	211
B.2	Diagrama de Secuencia F1.2 Seguridad Media	211
B.3	Diagrama de Secuencia F1.3 Seguridad Media	212
B.4	Diagrama de Secuencia F1.4 Seguridad Media	212
B.5	Diagrama de Secuencia F1.4 Seguridad Final	213
B.6	Diagrama de Secuencia F1.5 Seguridad Media	213
B.7	Diagrama de Secuencia F1.5 Seguridad Final	214
B.8	Diagrama de Secuencia F1.6 Seguridad Media	214
B.9	Diagrama de Secuencia F1.6 Seguridad Final	215
B.10	Diagrama de Secuencia F2.1 Seguridad Media	215
B.11	Diagrama de Secuencia F2.2 Seguridad Media	216
B.12	Diagrama de Secuencia F2.4 Seguridad Media	216
B.13	Diagrama de Secuencia F2.5 Seguridad Media	217
B.14	Diagrama de Secuencia F2.6 Seguridad Media	217
B.15	Diagrama de Secuencia F2.7 Seguridad Media	218
B.16	Diagrama de Secuencia F2.8 Seguridad Media	218
B.17	Diagrama de Secuencia F2.9 Seguridad Media	219
B.18	Diagrama de Secuencia F3.1 Seguridad Media	219
B.19	Diagrama de Secuencia F3.1 Seguridad Final	220
B.20	Diagrama de Secuencia F3.2 Seguridad Media	220
B.21	Diagrama de Secuencia F3.2 Seguridad Final	221
B.22	Diagrama de Secuencia F3.3 Seguridad Media	221
B.23	Diagrama de Secuencia F3.3 Seguridad Final	222
B.24	Diagrama de Secuencia F3.4 Seguridad Media	222
B.25	Diagrama de Secuencia F3.5 Seguridad Media	223
B.26	Diagrama de Secuencia F3.6 Seguridad Media	223

B.27 Diagrama de Secuencia F3.7 Seguridad Media	224
B.28 Diagrama de Secuencia F3.8 Seguridad Media	224
B.29 Diagrama de Secuencia F3.9 Seguridad Media	225
B.30 Diagrama de Secuencia F3.10 Seguridad Media	225
B.31 Diagrama de Secuencia F3.11 Seguridad Media	226
B.32 Diagrama de Secuencia F3.12 Seguridad Media	226
B.33 Diagrama de Secuencia F3.13 Seguridad Media	227
B.34 Diagrama de Secuencia F3.14 Seguridad Media	227
B.35 Diagrama de Secuencia F3.15 Seguridad Media	228
B.36 Diagrama de Secuencia F4.1 Seguridad Media	228
B.37 Diagrama de Secuencia F4.2 Seguridad Media	229
B.38 Diagrama de Secuencia F4.3 Seguridad Media	229
B.39 Diagrama de Secuencia F4.4 Seguridad Media	230
B.40 Diagrama de Secuencia F5 Seguridad Media	230
B.41 Diagrama de Secuencia F5 Seguridad Final	231

Prefacio

Actualmente vivimos de una forma acelerada y día con día vamos experimentando la automatización de procesos lo que trae consigo el manejo de grandes volúmenes de información y el uso de computadoras que cuentan con sistemas que administran grandes cantidades de información de forma eficiente. Dichos equipos de cómputo cuentan con Sistemas de Bases de Datos que han ido evolucionando. Las aplicaciones que se generan de estos sistemas han ido creciendo: van desde aplicaciones para computadoras personales hasta componentes centrales de modernos ambientes computarizados. Es por esta razón que el conocimiento que debemos tener acerca de los Sistemas de Bases de Datos es parte esencial en muchos ámbitos de la vida diaria. Así, la adquisición, manejo y actualización en el uso de información ha convertido a los Sistemas de Bases de Datos en herramientas de gran trascendencia para las oficinas, escuelas, fábricas, instituciones de servicio y, en general, para la mayoría de las instituciones y personas. Por tanto, el conocimiento, evaluación y selección de alternativas de mayor alcance y utilidad, se convierte en un proceso de vital importancia para el desarrollo tecnológico de cualquier institución, empresa o negocio.

La *Facultad de Ciencias* ha propuesto el desarrollo de un Sistema Integral de Administración que le permitirá cumplir con las necesidades de control de información, seguimientos de procesos y estadísticas relacionadas con la información que maneja y que forma parte de un proyecto de automatización integral y actualización en materia de sistemas y equipos de cómputo, que se ha venido realizando en los últimos años en la UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO (U.N.A.M.).

Este proyecto, representa una propuesta para la automatización de algunas de las actividades y funciones administrativas que se desarrollan para la elaboración de la forma única por parte del Departamento de Personal Académico de la *Facultad de Ciencias* de la U.N.A.M., esta propuesta fue complementada con las observaciones y el análisis que se realizó en conjunto con el personal administrativo de esta área acerca de las necesidades de su departamento; tratando de respetar en lo posible los estándares y los recursos de cómputo, tanto software como hardware, con que cuenta tal departamento.

Esta aplicación, en consecuencia, puede ejecutarse en cualquiera de los

equipos con los que cuenta el Departamento y está diseñada para utilizarse en un ambiente bajo red, multiusuarios y multiplataformas, con niveles de seguridad para control de accesos y funciones de sistema.

Este trabajo le permitirá contar al Departamento de Personal Académico de la *Facultad de Ciencias* de la U.N.A.M. con un sistema controlado a través de la computadora, con el cual podrá manejar eficientemente los datos personales de los maestros, investigadores y ayudantes que soliciten o ya se encuentren contratados por alguno de los departamentos (Biología, Física, Matemáticas) de la Facultad, y los datos propios de cada uno de los contratos que contraigan estas personas con la Institución. Asimismo, le permitirá contar con información estadística al día y en cualquier momento, que le permitirá la elaboración y presentación de informes, así como el análisis y proyecciones de comportamiento de la información.

Por último, el diseño y la programación del sistema fueron enfocados para cubrir en lo posible la mayor cantidad de actividades administrativas del área y para realizar el manejo del sistema de una forma amigable y sencilla, sin necesidad de capacitaciones sofisticadas. Esto se logró gracias a la utilización de herramientas y aplicación de técnicas de la Ingeniería de Software, Análisis de Algoritmos, Programación, Estadística, Diseño de Interfaces y Estructura de Datos. Este proyecto se presenta en 6 capítulos:

- Capítulo 1.** Presenta un breve panorama teórico que introduce de forma general la idea de un sistema de software. Habla también de las ramas de la Ciencias de la Computación que se ocupan en la actualidad para formalizar el desarrollo de un sistema de software.
- Capítulo 2.** Habla de la ingeniería de sistemas por computadora. Explica el ciclo de vida de un sistema de software así como también la mayoría de los factores que influyen a nivel teórico y humano, y que se deben tomar en cuenta para el buen desarrollo de una aplicación.
- Capítulo 3.** Explica con detalle el modelo de bases de datos utilizado para el desarrollo óptimo del sistema, así como también sus diferencias con otros modelos que existen en la actualidad.
- Capítulo 4.** Explica de forma precisa el análisis de requerimientos del sistema, así como las necesidades que tienen los usuarios finales, y que se deben tomar como parte de los requerimientos iniciales del sistema.
- Capítulo 5.** Muestra el diseño del sistema y establece las funciones que debe tener para satisfacer las necesidades de los usuarios finales.
- Capítulo 6.** Muestra un prototipo o implementación inicial del sistema, además de establecer un plan de pruebas para este prototipo. Se incluyen también las soluciones a problemas encontrados en la fase de evaluación del sistema.

Capítulo 1

Introducción

Hoy en día las computadoras están presentes en todos los ámbitos de la vida del ser humano. El pivote elemental de las computadoras es sin duda los sistemas de software; los cuales han ido creciendo en tamaño y complejidad con el fin de resolver más problemas de forma eficiente.

Es por eso que el problema asociado con su desarrollo tiene sus raíces en inadecuados controles de calidad y vicios que se han venido dando a lo largo del tiempo. Por eso, es importante tratar de comprender los errores del pasado para no repetirlos en el futuro. Este sólo será posible teniendo un marco teórico de referencia que nos introduzca de forma general la idea de un sistema de software así como las metodologías básicas para su desarrollo óptimo.

1.1 Definiciones

Esta sección define lo que es un sistema. Informalmente, introduce muchos términos técnicos y pequeñas definiciones que son necesarias para un mejor entendimiento de las ingenierías utilizadas en el proceso de desarrollo de sistemas.

1.1.1 Sistema

Como es costumbre, vienen a la mente varias definiciones de sistema, y probablemente todas son adecuadas. Si nos remitimos a una enciclopedia, podremos encontrar en ella que su definición es “un conjunto de reglas o principios sobre una materia entrelazados entre sí” [1]. Otros libros lo definen como “un conjunto u ordenación de cosas relacionadas de tal manera que forman una unidad o un todo orgánico”, también se puede definir como “un conjunto de hechos, principios, reglas, etcétera, clasificados y ordenados de tal manera que muestran un plan lógico uniendo las diferentes partes”; o incluso podemos decir de una manera simple que es “un método o plan

de clasificación u ordenación" [2]. Utilizaremos la siguiente definición: Un *sistema* es una reunión o conjunto de elementos relacionados [3].

Los elementos de un sistema pueden ser *conceptos*, en cuyo caso hablaríamos de un sistema conceptual. Un lenguaje es un ejemplo de sistema conceptual. Los elementos de un sistema también pueden ser *objetos*, como por ejemplo, una computadora compuesta de varias partes. Finalmente, los elementos de un sistema pueden ser *sujetos*, como los de un equipo de fútbol. Concluyendo, podemos decir entonces que un sistema puede estructurarse por conceptos, objetos y sujetos, como en un sistema hombre-máquina, que comprende las tres clases de elementos. Por tanto, un sistema es un agregado de entidades, viviente o no viviente o ambas. Incluso los sistemas pueden componerse de otros sistemas que llamamos *subsistemas*. En la mayoría de los casos, podemos pensar en sistemas grandes, los cuales comprenden otros sistemas y que llamamos el *sistema total* o el *sistema integral*. Uno de los problemas al tratar de sistemas se deriva de nuestra incapacidad para saber qué tanto descomponer un sistema en módulos (sistemas componentes), o qué tanto componer u organizar un sistema en sistemas más grandes.

1.1.2 Software y Hardware

Software es una colección de programas de computadora, procedimientos, reglas, documentación asociada y datos [4]. Sin embargo, hay una distinción entre un programa y programar un sistema. Un programa generalmente está completado en sí mismo, y es generalmente usado por su autor. Usualmente, tiene muy poca documentación u otro tipo de ayuda para que otra gente use el programa. Las fallas que puede tener no son de mucha importancia para el autor y si este falla por completo, el autor sólo corrige ese error y vuelve a ejecutar el programa nuevamente. Estos no son portables, reutilizables y mucho menos de gran uso para otras personas.

Por otra parte, programar un sistema es muy distinto, estos son usados por una gran cantidad de gente con diferentes intereses y conocimientos, y por lo mismo se provee una interfaz amigable para los usuarios. Tiene suficiente documentación para ayudar a diversos usuarios. Son probados antes de ser lanzados al mercado bajo estrictas pruebas de calidad, y comúnmente son portables, se pueden ejecutar en una gran variedad de hardware y continuamente son actualizados con tecnologías de punta.

Claramente, un programa específico para resolver un problema personal y programar un sistema son dos cosas totalmente diferentes. A pesar de esto, el software es esencial haciendo tareas de rutina como mantener archivos en disco y administrar la pantalla, además de desarrollar tareas tan simples como un procesador de textos, administración de bases de datos, etcétera. Partiendo de estas tareas, el software se pueden dividir en dos categorías:

- **Sistemas Operativos** que controlan los recursos y administran los trabajos que lleva acabo la computadora.

- **Programas de Aplicación** que llevan acabo una multitud de tareas para los usuarios de las computadoras.

El *hardware* a su vez se definirá como los componentes físicos o dispositivos electrónicos que forman un sistema de cómputo y que proporcionan la capacidad de computación [4], incluyendo la Unidad de Procesamiento Central (CPU), los dispositivos de almacenamiento interno (RAM y ROM), los dispositivos de almacenamiento externo (Disco Duro, Cintas Magnéticas, Disco Flexibles, CD-ROM), así como los dispositivos de entrada (Ratón, Teclado, Escaner, Palanca de Juegos) y los dispositivos de salida (Impresora, Pantalla).

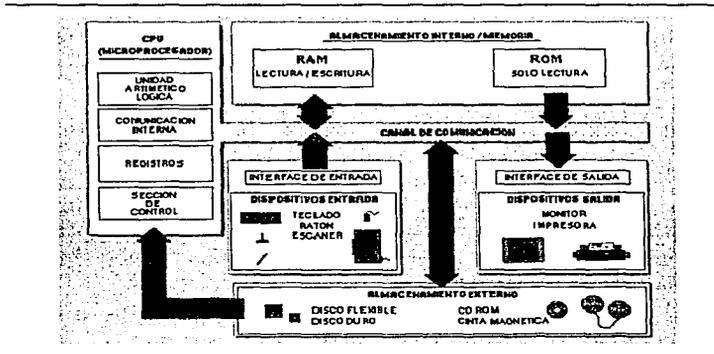


Figura. 1.1: Hardware más común de una computadora.

La Figura 1.1 muestra un esquema que engloba el típico Hardware de una computadora personal.

1.1.3 Usuarios y Documentación

Cuando hablamos del desarrollo de un software, es importante tomar en cuenta muchos aspectos, pero el más importante de estos aspectos es sin duda los *usuarios*, los cuales son los individuos que usan y operan el software y el hardware [2, 6].

En particular, los usuarios que involucran este proyecto se pueden clasificar en:

- **Programadores de Aplicación.**- son profesionales en cómputo que interactúan con el sistema a través de llamadas al sistema por medio de un Lenguaje de Manipulación de Datos, los cuales se encuentran inmersos en una programa escrito en un lenguaje hues-

ped (por ejemplo, Cobol, PL/I, Pascal, C). Estos programas son comúnmente conocidos como programas de aplicación.

- **Usuarios Sofisticados.**- son los que interactúan con el sistema sin escribir programas. En lugar de esto, ellos solicitan sus requerimientos a los programas de aplicación por medio de un Lenguaje de Consulta.
- **Usuarios Especializados.**- que son usuarios sofisticados que escriben aplicaciones especializadas que no entran en un marco de procesamiento de datos tradicional (por ejemplo, datos gráficos y datos de audio).
- **Usuarios Navegadores.**- son usuarios no sofisticados que interactúan con el sistema a través de un programa de navegación en la red.

Para apoyar más a los usuarios y para llevar un mejor control en el desarrollo de un sistema es indispensable contar con la documentación del mismo. La *documentación* consiste en manuales impresos y cualquier otra información descriptiva que explica el uso y la operación del sistema [2]. En la actualidad la mayoría de los sistemas cuentan con manuales escritos en diferentes idiomas, ayuda en línea y, además, algunas compañías proporcionan asistencia telefónica todos los días del año.

1.1.4 Bases de Datos y Procedimientos

Antes de la era de las bases de datos, grandes masas de datos eran almacenadas en sistemas de archivos, estos sistemas contaban con deficiencias, diferencias en estructuras y formatos, así como también inconsistencias y duplicidad en los datos. Con el paso del tiempo y el nacimiento de nuevos sistemas, el usuario se dió cuenta de que los sistemas de archivos con los que contaba no resolvían sus necesidades eficientemente, por lo que decidió crear complejas estructuras de almacenamiento dentro de un único sistema de archivos; de tal forma que fuera más sencillo almacenar y encontrar toda la información que el ser humano guarda para obtener conocimiento del presente, pasado y del posible futuro. Con esto nació lo que hoy en día conocemos como base de datos.

Una *base de datos* es una colección de datos almacenados por medio de un sistema de administración de base de datos [7]. A su vez un *sistema de administración de bases de datos* (DBMS¹) es un programa que almacena y encuentra datos de forma eficiente [7]. Estos sistemas proporcionan servicios que los sistemas de archivos convencionales no ofrecen, como:

¹DBMS -Database-Management System (Sistema de Administración de Bases de Datos)-

- **Consistencia:** el sistema asegura que las actualizaciones a los datos no violen un conjunto de reglas establecidas como propiedades de los datos.
- **Concurrencia:** el sistema permite que varios usuarios puedan leer y actualizar datos simultáneamente.
- **Control de Acceso:** el sistema previene accesos no autorizados con base en los derechos de acceso que tiene cada usuario.
- **Recuperación y Re-uso:** el sistema minimiza la pérdida de datos en caso de que exista un error que detenga de forma anormal el sistema. Inclusive en el caso de daños más extensos, los respaldos y bitácoras recuperan el sistema para prevenir la pérdida de datos.

Es importante reconocer que los DBMS no son la panacea para el problema de administración de datos. Ellos funcionan bien para grandes masas de datos en aplicaciones donde la organización lógica de los datos y el uso concurrente son importantes. Para el problema de administración de datos en menor escala hay otras soluciones que pueden funcionar mejor.

Por último es importante mencionar que para llevar a cabo alguna modificación en la base de datos se necesitan solicitudes explícitas que reciben el nombre de *procedimientos* y se pueden definir como los pasos que definen el uso específico de cada elemento del sistema o el contexto procedimental en que reside el sistema [7].

1.2 Ingeniería de Software

El Software está jugando un rol central en todos los ámbitos de nuestra vida diaria —en el gobierno, bancos y finanzas, educación, transporte, entretenimiento, medicina, agricultura y leyes. El número, tamaño y dominio de los programas que están siendo desarrollados ha crecido drásticamente; como resultado, billones de dólares están siendo gastados en el desarrollo, subsistencia y vida de millones de programas y sistemas computacionales. Desafortunadamente, hay severos problemas en los costos, tiempos y calidad de muchos productos de software; y aún más serio es el efecto que los problemas de calidad pueden tener en los elementos de seguridad crítica del software y que directamente afectan la salud y el bienestar del ser humano.

El principal problema es que el tamaño y la complejidad de los sistemas de software se incrementa, el problema de diseño recae en los algoritmos y en las estructuras de datos: por lo cual, diseñar y especificar la estructura completa del sistema representa un nuevo tipo de problema. Numerosos artículos de investigación incluyen como contenido una organización extensa y una estructura de control global; protocolos de comunicación, sincronización y acceso a datos; asignación de funciones a elementos de diseño;

distribución física; composición en los elementos de diseño; crecimiento y comportamiento; y selección sobre otras alternativas de diseño. Este nivel de diseño ha sido direccionado de muchas maneras incluyendo diagramas informales y términos descriptivos, interconexión de módulos en distintos lenguajes, esqueletos y marcos de referencia para sistemas que pueden ser aplicados a las necesidades de dominios específicos y modelos formales de mecanismos de integración de componentes.

Debido a esta necesidad, la Ingeniería de Software es un pivote de partida en la evolución del software. A pesar de que el software ha llegado a ser un punto crítico en el desarrollo de los más nuevos sistemas creados por el ser humano, los conceptos, los principios, y métodos para la ingeniería aún no están bien definidos de una forma estandar. Es por eso que la falta de consenso para la estandarización de estos aspectos en la práctica de la ingeniería de software y los continuos requisitos establecidos por los nuevos métodos crean confusión y tienen serias consecuencias en la evaluación, adquisición y aplicación del conocimiento de la ingeniería de software.

Para tratar de estandarizar el conocimiento de esta nueva ingeniería en los últimos años se han desarrollado estudios acerca del desarrollo de esta área. A pesar de que ha existido progreso en años recientes, un problema para lograr un avance significativo en el estado de los "componentes de la infraestructura" de la ingeniería de software es la falta de un entendimiento claro y comprensible de la naturaleza y contenido de la misma área.

Es por eso que la aplicación de criterios de evaluación ya mencionados, supone la selección de la configuración de un sistema específico y la especificación del funcionamiento y del rendimiento asignados al hardware, al software, a los usuarios, a las bases de datos, a la documentación y a los procedimientos. Esencialmente, lo que se hace es asignar a cada elemento del sistema un ámbito de funcionamiento y de rendimiento.

La ingeniería del hardware, la ingeniería del software, la ingeniería humana y la ingeniería de bases de datos están para refinar el ámbito y producir un elemento del sistema capaz de funcionar y que pueda ser integrado adecuadamente con otros elementos del sistema.

Durante la ingeniería del sistema se asigna la función y el rendimiento al software. En algunos casos, la función es simplemente la realización de un procedimiento secuencial de manipulación de datos y el rendimiento no queda explícitamente definido, en otros casos, la función es la coordinación y control de otros programas concurrentes y el rendimiento está definido de forma explícita en términos de tiempos de espera y de respuesta.

Si deseamos obtener la función y el rendimiento, se deben construir o adquirir una serie de componentes de software; en donde a diferencia del hardware, los componentes de software están muy poco estandarizados en la mayoría de los casos, por lo que se deben crear componentes a la medida para satisfacer los requisitos asignados al elemento del software que se va a desarrollar.

El elemento de software de un sistema basado en computadora está compuesto por programas, datos y documentación que constituyen el software de la aplicación y el software del sistema. El software de la aplicación aplica los procedimientos requeridos para realizar funciones de procesamiento de la información, mientras que el software del sistema implementa funciones de control que permiten al software de la aplicación comunicarse con otros diversos elementos de software.

Para que un sistema basado en computadoras sea de uso práctico, el software debe proporcionar información o control a otro elemento del sistema o a una fuente externa. Producir información de salida requiere que el software de un formato a los datos que resulten apropiados para el medio de salida y debe saber cómo comunicarse con el dispositivo de salida.

Independientemente del área de aplicación, un programa o sistema basado en computadoras puede ser representado mediante un modelo de entrada-proceso-salida, es decir, el software se usa para adquirir información que puede ser suministrada por alguna fuente externa o por otro elemento del sistema. Cuando un sistema basado en computadoras requiere una interfaz interactiva entre hombre y máquina, el software proporciona la "interacción" de entrada-salida. En el software se realizan los mecanismos de petición y de entrada de datos; con el software se generan las pantallas y los gráficos y, mediante el software, se lleva a cabo la lógica que conduce al usuario a través de la secuencia de pasos interactivos. Cuando se adquieren los datos desde un dispositivo, el software, en forma de controlador, acomoda las características especiales del hardware. Además, el software también se usa para establecer interfaces con las bases de datos, permitiendo a un programa acceder a fuentes de datos preexistentes.

El software aplica los algoritmos de procesamiento requeridos para realizar las funciones del sistema. En general, un algoritmo de procesamiento transforma datos de entrada y produce información o control como salida para otro elemento del sistema o macroelemento. Actualmente, el tipo más común de procesamiento es el procedimiento numérico o no numérico en el que todos los pasos, ciclos y condicionales están predefinidos. Sin embargo, en algunos sistemas basados en computadoras se están introduciendo nuevas categorías de algoritmos de procesamiento, como el software de sistemas expertos, las herramientas tipo "Case", la programación orientada a objetos y las redes neuronales artificiales.

Por último, a diferencia de los algoritmos convencionales, los sistemas expertos utilizan hechos específicos y reglas de inferencia, permitiendo al software mostrar habilidades de diagnóstico parecidas a las humanas en un ámbito de problemas limitado, la programación orientada a objetos asigna propiedades a unidades básicas consideradas por su comportamiento como objetos y, a diferencia de las anteriores, una red neuronal artificial imita las funciones de las neuronas del cerebro humano y ha mostrado buenas expectativas en el reconocimiento de patrones y el aprendizaje automático.

1.3 Ingeniería de Hardware

Una vez que se ha definido la ingeniería del sistema tanto en su análisis como en su función, se asignan funciones al hardware. La primera fase de la ingeniería del hardware comprende la planificación del desarrollo y el análisis de los requisitos del hardware.

La planificación del desarrollo se orienta a establecer el alcance del esfuerzo del hardware, para esto nos hacemos las siguientes preguntas:

- ¿Qué clase de hardware se adapta mejor a las funciones del sistema?
- ¿Qué hardware se puede comprar?
- ¿Cuáles son los proveedores, la disponibilidad y el costo del hardware?
- ¿Qué tipos de interfaces se requieren?
- ¿Qué tenemos que diseñar y construir?
- ¿Cuáles son los problemas potenciales y los recursos requeridos para el sistema?

A continuación debemos establecer una *guía de acciones* para el diseño del hardware y su implementación. El análisis de los requerimientos del hardware se orienta a especificar requisitos funcionales, de rendimiento y de interfaz precisos para todos los componentes del elemento hardware, además, deberán establecerse las restricciones del diseño (por ejemplo, tamaño, entorno) y los criterios de prueba del hardware.

1.4 Ingeniería Humana

Un sistema basado en una computadora casi siempre tiene un elemento humano. Puede ser que una persona interactúe directamente con el hardware y con el software, manteniendo un diálogo que dirija el funcionamiento del sistema; en cualquier caso, el responsable del desarrollo y del mantenimiento del sistema es el usuario.

Antes de que el ingeniero de sistemas pueda asignar una función al elemento humano, se debe especificar la interacción que es necesaria para poder realizar la función. Para hacerlo, se deben entender los "componentes" del elemento humano. Entre los muchos componentes que constituyen el elemento humano se encuentran: la memoria humana y la representación del conocimiento, el pensamiento y el razonamiento, la percepción visual y la construcción del diálogo humano.

La *ingeniería humana* es una actividad multidisciplinaria que aplica un conocimiento derivado de la psicología para especificar y diseñar un producto que sea de alta calidad para los humanos. El proceso de la ingeniería humana comprende los siguientes pasos:

- **Análisis de Actividad.-** Cada actividad asignada a un elemento humano se evalúa en el contexto de la interacción requerida con otros elementos. Cada actividad se subdivide en tarea que son analizadas en etapas posteriores.
- **Análisis y Diseño Sistemático.-** Se define el significado preciso de cada acción requerida por el usuario y de cada acción producida por la máquina. Se establece el diseño de un "diálogo" que comunique adecuadamente la semántica.
- **Diseño Léxico y Sintáctico.-** Se identifica y representa la forma específica de las acciones y órdenes. Después se diseña la implantación en software y en hardware de cada acción u orden.
- **Diseño del Entorno del Usuario.-** El hardware, software y otros elementos del sistema se combinan para formar un entorno de usuario. El entorno puede incluir facilidades físicas, brillo, utilización del espacio, factor de eliminación del stress, etcétera.
- **Creación de Prototipos.-** Es difícil, si no imposible, especificar formalmente una ingeniería humana sin usar un prototipo. La creación de prototipos permite evaluarla desde una perspectiva humana, con una participación activa en lugar de una evaluación pasiva. La creación de prototipos supone una evaluación y una aplicación interactiva de todos los pasos de ingeniería humana anteriores.

1.5 Consideraciones

Para tener un visión más clara del desarrollo de un sistema debemos tomar en cuenta un pequeño conjunto de entornos dentro de los cuales surgen incógnitas que deben ser tomadas en cuenta antes de iniciar el desarrollo del sistema. Tales incógnitas las podemos clasificar en: generales, comerciales, técnicas, humanas, legales, de desarrollo y de interacción con el entorno. Algunos ejemplos de las mismas pueden ser:

Generales

- ¿Puede ser desarrollado el proyecto y su configuración² dentro de los límites preestablecidos de costo y tiempo?
- ¿Cuál es el riesgo asociado a las estimaciones de tiempo y costo?

²Configuración.- Un sistema particular de componentes interrelacionados, tal como como un sistema informático. Configurar un sistema es escoger los componentes de entre una variedad de opciones para crear un sistema particularizado [2].

Comerciales

- ¿Representa la configuración la solución más benéfica?
- ¿Puede ser lanzado con éxito?
- ¿Compensarán los beneficios a los riesgos del desarrollo?

Análisis Técnico

- ¿Existe la tecnología necesaria para desarrollar todos los elementos del sistema?
- ¿Están aseguradas la funcionalidad y el rendimiento?
- ¿Podrá mantenerse correctamente la configuración?
- ¿Existen recursos técnicos?
- ¿Cuál es el riesgo asociado a la tecnología?
- ¿Existe un sistema equivalente?
- ¿Pueden ser adquiridas las parte más importantes del sistema por medio de un tercero?

Evaluación de Desarrollo

- ¿Hay herramientas de programación y equipos de cómputo disponibles?
- ¿Las estimaciones en costo y tiempo dependen del personal que desarrolle el sistema?
- ¿Puede llevarse a cabo adecuadamente una garantía de calidad?

Aspectos Humanos

- ¿Existe personal calificado disponible para el desarrollo del sistema?
- ¿Existen problemas políticos o de cualquier otra índole?
- ¿Comprenden los usuarios lo que va a hacer el sistema?

Interfaces con el Entorno

- ¿Funciona correctamente la interfaz³ con la configuración propuesta y con el entorno externo del sistema?
- ¿Se manejan inteligentemente las comunicaciones máquina-máquina y el hombre-máquina?
- ¿Cumple con las expectativas del usuario?

Legales

- ¿Introduce esta configuración un riesgo de responsabilidad indebido?
- ¿Pueden ser protegidos adecuadamente los aspectos de propiedad?
- ¿La confidencialidad de la información del sistema se puede garantizar?
- ¿Existe una infracción potencial?

³Interfaz.- Una conexión e interacción entre hardware, software y usuario, pueden ser los cables, los conectores, lenguajes, códigos, teclados, ratones y menus empleados para la comunicación entre el usuario y la computadora [2].

Capítulo 2

Ingeniería de Software

La ingeniería de sistemas por computadora representa un amplio campo dentro del cual podemos destacar la Ingeniería de Software, la Ingeniería de Hardware y la Ingeniería Humana, cada una de estas ramas intenta establecer un orden en el desarrollo de sistemas basados en computadoras.

En la actualidad, en los sistemas basados en computadoras el software ha reemplazado al hardware, en el sentido de ser el elemento del sistemas más difícil de planear, con menos posibilidades de éxito que el hardware tanto en tiempo como en dinero y es mucho más peligroso de manejar, además de que se observa que mientras los sistemas basados en computadoras continúan creciendo, tanto en número como en complejidad así como en variedad de aplicaciones, la demanda de software continua sin disminuir.

2.1 Ingeniería de Software

La ingeniería de software es una rama de las ciencias de la computación que ofrece métodos y técnicas para desarrollar y mantener la calidad del software para resolver problemas.

Esta ingeniería trata con áreas de la ciencia de la computación como construcción de compiladores y sistemas operativos; se aplica en ambientes de negocios, investigación científica, recreación, medicina, producción, finanzas, control de tráfico, meteorología, leyes, entre otras. Las soluciones que ofrece un software van desde almacenar registros a sofisticadas descisiones de mercado.

Existen muchas definiciones de la ingeniería de software y cada una enfatiza varios aspectos de ella. Por ejemplo, *von Mayrhauser* [8] nos da las siguientes definiciones:

Definición 1. *Ingeniería de Software* es el estudio de los principios y metodologías para desarrollar y mantener sistemas de software.

Definición 2. *Ingeniería de Software* es la aplicación práctica de conocimiento científico en el diseño y construcción de programas de computadora y la asociación de documentos requeridos para desarrollar, operar y mantenerlos. También se le reconoce como desarrollo de software o producción de software.

Definición 3. *Ingeniería de Software* es el establecimiento de principios de ingeniería y métodos con el fin de obtener económicamente un software reutilizable y que trabaje en máquinas reales.

Estas tres definiciones de ingeniería de software son útiles por varias razones:

- Todas contienen información sobre el significado de la Ingeniería de Software, sus actividades, productos y filosofía.
- Ninguna está equivocada.
- Enfatizan varios aspectos del desarrollo del proceso de software y sus objetivos.
- Como la Ingeniería de Software es una disciplina en evolución. Cuando nuevos términos describen conceptos, principios o metodologías se unen, entonces se da lugar a múltiples definiciones para el mismo término.
- Dan perspectiva histórica.

Tomando como punto de partida estas ideas, podemos decir que para utilizar esta disciplina, se deberán establecer algunos lineamientos basados en algún modelo que ya exista y que al mismo tiempo sea bien conocido por la mayoría de los desarrolladores de software.

2.1.1 Ciclo de Vida

Cuando vamos a desarrollar un programa o construir algo, hay una serie de actividades que debemos realizar explícita o implícitamente. Desde la concepción de la idea de un sistema de software hasta su diseño y liberación para los usuarios, inclusive después de eso, el software para por un desarrollo y evolución gradual. El software tiene un ciclo de vida compuesto de varias fases. Cada una de estas fases da como resultado el desarrollo de cada una de las partes de un sistema asociado con el sistema, como un fragmento de la especificación, un plan de pruebas o el manual de usuario. Las actividades básicas o fases para desarrollar un sistema de software son: *análisis de requerimientos y especificación, diseño de software, implementación, verificación, validación y mantenimiento*. A continuación se describen de forma general estas fases:

Análisis de Requerimientos y Especificación

El propósito de esta fase es indentificar y documentar los requerimientos exactos del sistema. Durante está etapa se desarrolla una descripción bien delimitada del ámbito del esfuerzo del software. De ahí que estos requerimientos sean desarrollados en conjunto con los usuarios y los desarrolladores del software. Para esto se lleva a cabo un análisis de riesgo, se definen los recursos necesarios para desarrollar el software y se establecen las estimaciones de tiempo y costos. El éxito de un sistema se mide en que tan bien se desarrollan estos requerimientos lo más cercano posible a la realidad y que tan bien los usuarios perciben las funciones que ellos realizan con las que les permitirá llevar a cabo el software. Hasta este punto se lleva a cabo la especificación a lo que posteriormente se aplica un análisis de requerimientos que es una parte en el proceso del desarrollo de software donde se define en detalle el elemento del sistema asignado al software. Los requisitos se analizan y se definen uno a uno. Se puede hacer un análisis formal del ámbito de información para establecer modelos de flujo y la estructura de la información. Luego, se amplían esos modelos para convertirlos en una especificación del software. Los requisitos de rendimiento y las limitaciones de recursos se traducen en características para el diseño del software. El análisis global del elemento de software define los criterios de validación que se utilizarán para demostrar que se han podido conseguir los requisitos. El resultado de esta fase es un documento que recibe el nombre de *especificación de requisitos* en donde se establece que es lo que debe hacer el sistema, prototipo de los manuales de usuario, un estudio de estimación de tiempos y costos, requerimientos de comportamiento y vialidad del proyecto [10]. Un punto importante es que este documento no especifica cómo es que el sistema llevará a cabo estos requerimientos.

Diseño de Software

Empezando con la documentación de los requerimientos, los desarrolladores de software diseñan el sistema. El resultado de esta fase es el *documento de especificación de diseño* en donde se indentifican todos los módulos del sistema y sus interfaces [10]. El propósito de esta fase es proveer una descomposición en módulos con base en los requerimientos establecidos. El método de diseño puede tener un gran impacto en la calidad de la aplicación resultante, en particular, en su posible entendimiento y modificación. Inclusive puede también afectar la elección del lenguaje de programación que será usado para la implementación del sistema.

Realización (codificación)

Esta fase comienza con el diseño de especificación con el cual los desarrolladores de software implementan el sistema. La especificación de diseño, en

este caso, establece una estrategia de solución; la meta como fase dentro del desarrollo de software es elegir como, sobre todas las posibles maneras, el sistema debe ser codificado para cumplir con el diseño de especificación. De otra forma, podemos decir que esta fase traduce un conjunto de requisitos estructurados mediante un diseño en el elemento operativo del sistema que llamamos software. Para llevar a cabo esto, los desarrolladores de software generan un programa que use un lenguaje de programación apropiado o una herramienta tipo "Case". La metodología de ingeniería de software contempla la codificación como la consecuencia de un buen diseño. En cuanto al código, se revisa su estilo y claridad, y se comprueba que haya una correspondencia directa con la descripción detallada del diseño. El *listado* en lenguaje fuente de cada componente modular constituye el documento de configuración final de esta etapa de desarrollo.

Verificación y Validación

Esta fase evalúa la calidad de implementación del sistema, que posteriormente es enviado a los clientes. Es importante notar que esta evaluación no se debe concentrar al final de la fase de implementación, más bien debe ocurrir casi al mismo tiempo en cada paso del desarrollo de software para verificar que los pasos intermedios del proceso satisfagan sus objetivos. Es por eso que el ingeniero de software prueba el software para encontrar el mayor número posible de errores antes de que sea puesto en circulación y lo prepara para su lanzamiento. Esto se lleva a cabo por medio de pruebas de unidad con la finalidad de intentar verificar el rendimiento funcional de cada componente modular individual del software, también se realizan pruebas de integración las cuales constituyen un medio de construcción de la arquitectura del software, estas son las pruebas de su funcionamiento, de sus interfaces y la prueba de validación. Para verificar todo esto periódicamente debemos contestar las siguientes dos preguntas:

- ¿Estamos construyendo el producto correctamente?
- ¿Estamos construyendo el producto apropiado?

Tras cada uno de estos pasos de prueba y habiendo respondido las dos preguntas, puede realizarse una depuración-diagnóstico-corrección de defectos. Para los pasos de prueba, se puede desarrollar un plan y un procedimiento de prueba. Una vez terminada la prueba del software, éste está casi preparado para ser entregado a los usuarios finales. Sin embargo, antes de la entrega se llevan a cabo una serie de actividades de garantía de calidad para asegurar que se han generado y catalogado los registros y los documentos internos adecuados, que se haya desarrollado una documentación de alta calidad para el usuario y que hayan establecido mecanismos apropiados de control de configuraciones. Entonces, el software ya puede ser distribuido a

los usuarios finales. Tan pronto como se entregue el software a los usuarios finales, el trabajo del ingeniero de software cambia. En ese momento, el enfoque cambia de la construcción al mantenimiento-corrección de errores, adaptación al entorno y mejoramiento de la funcionalidad del sistema.

Mantenimiento

Habiendo entregado el sistema, algunos cambios en el sistema pueden llegar a ser necesarios debido a la detección de fallas, un deseo de agregar una nueva capacidad o de mejorar alguna de las que ya existen, la necesidad de corregir cambios en el ambiente operacional, reordenar o regenerar índices, o respaldar la información del sistema. Estos cambios necesariamente tienen que llevarse a cabo por medio de mantenimiento al sistema. Lo importante de esta fase puede ser el hecho que los costos de mantenimiento son típicamente tan costos como todas las demas fases combinadas.

2.1.2 Modelos del Proceso de Desarrollo de Software

La meta de cualquier desarrollo es producir algún producto. Un *proceso de desarrollo* es un conjunto de actividades unidas con una relación de orden entre ellas [10], que se lleva a cabo de alguna manera para producir el producto deseado. Un *modelo* es una representación abstracta de un proceso de desarrollo [10].

En el desarrollo de un software la meta es producir un software de alta calidad. Existen diferentes modelos que han sido propuestos para el desarrollo de software. Algunos de ellos son: *el modelo cascada, el modelo de prototipos, el modelo de mejoramiento iterativo y el modelo espiral*. A continuación se describen detalladamente estos modelos.

Modelo Cascada

Es el proceso modelo más simple. Establece que las fases están organizadas con un orden lineal. La secuencia de actividades desarrolladas en el desarrollo de un software usando este modelo son: análisis de requerimientos, planeación de proyecto, diseño de sistema, diseño detallado, codificación y pruebas preliminares, integración del sistema y pruebas finales.

El orden lineal de actividades tiene algunas consecuencias importantes. Primero, identificar claramente el final de una fase y el inicio de otra; aquí se emplean ciertos mecanismos de certificación para el final de cada fase. Usualmente estos mecanismos son la verificación y validación. Lo único que hacen es asegurar que el resultado final de una fase sea consistente con los requerimientos iniciales de la fase siguiente. Estos resultados intermedios entre dos fases se conocen como *productos intermedios* y usualmente son documentos. Para la fase de codificación el resultado es código. Desde este

punto de vista, el resultado de un proyecto de software no sólo es el programa final, este viene acompañado de manuales de usuario, documentos de análisis, documentos de diseño, implementación, plan de pruebas y resultados finales. Segundo, los resultados intermedios del modelo cascada son:

- Documentación de Requerimientos
- Plan del Proyecto
- Documento de Diseño del Sistema
- Documento de Diseño Detallado
- Plan de Pruebas y Reporte de Pruebas
- Código Final
- Manual del Usuario
- Reporte de Repaso

Tercero, las limitaciones del modelo cascada son:

1. Asume que los requerimientos de un sistema pueden seguir siendo los mismo antes de que el diseño comience.
2. Si los requerimientos no sufren modificaciones usualmente requiere elegir el hardware específico para los requerimientos de ese momento.
3. Estipula que los requerimientos deben ser completamente especificados antes de que el desarrollo pueda iniciar.

La Figura 2.1 muestra todas las fases de desarrollo y resultados intermedios del modelo cascada.

Modelo de Prototipos

La meta de un desarrollo basado en prototipos es eliminar las primeras dos limitaciones del modelo cascada. Es decir, en lugar de que los requerimientos sigan siendo iguales antes de cualquier diseño o implementación, se construye un *prototipo* que ayuda a entender los requerimientos. Este prototipo se construye basándose en los requerimientos conocidos. Esta construcción requiere de diseño, implementación y pruebas, pero cada fase no es llevada a cabo formalmente. El prototipo se prueba con el cliente y está interacción le permite al cliente entender mejor los requerimientos del sistema que desea.

Este modelo es atractivo para sistemas grandes y complejos donde no hay un manual de procesos o un sistema ya existente que determine parte de los requerimientos.

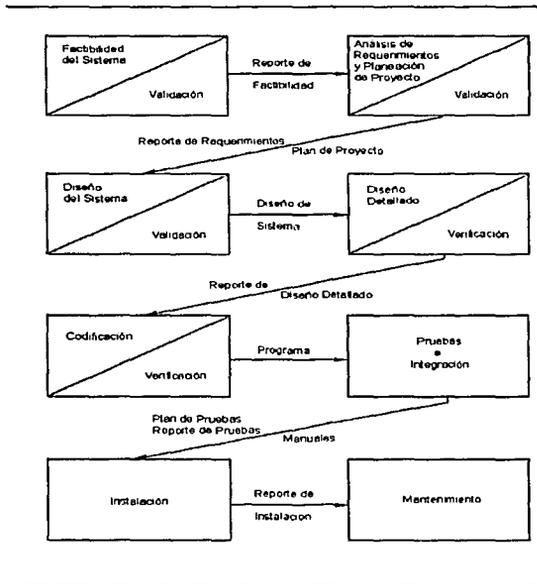


Figura. 2.1: Modelo Cascada.

Otra razón para utilizar el modelo de prototipos es el costo. Esto se debe a que los prototipos son generalmente sistemas no completados y muchos de los detalles no se contruyen en ellos.

La principal meta de un prototipo es proveer un sistema funcional. Además los costos de pruebas y documentos detallados se reducen. Esto ayuda a optimizar los costos del desarrollo del prototipo. Y por otro lado los desarrolladores ganan experiencia cuando desarrollan prototipos, que es necesaria para el desarrollo final.

El resultado final después de haber aplicado este modelo, será un sistema más reutilizable y mejor diseñado ya que se adapta mejor a las necesidades del cliente, además por medio de los prototipos intermedios será más sencillo encontrar y corregir errores y malfuncionamientos en el programa final.

La Figura 2.2 muestra las fases de desarrollo que se deben llevar a cabo en un desarrollo por prototipos.

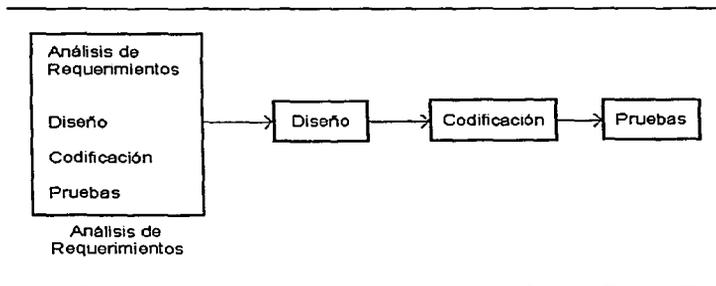


Figura 2.2: Modelo de Prototipos.

Modelo de Mejoramiento Iterativo

Este modelo elimina la tercera limitante del modelo cascada y combina los beneficios del modelo de prototipos y del modelo cascada. La idea básica es que un software puede desarrollarse de forma incremental, donde cada incremento agrega alguna capacidad funcional nueva al sistema hasta que se complete totalmente.

En cada paso, extensiones y modificaciones de diseño pueden aflorar. Esto puede resultar en mejores pruebas, ya que estas son incrementales. Además como en los prototipos, el incremento permite al usuario darse cuenta cuales podrían ser los requerimientos finales del sistema.

El primer paso de este modelo, es una implementación inicial de un subconjunto de todo el problema. Este subconjunto es aquel que contiene los principales aspectos claves para entender e implementar el problema. De ahí se crea una *lista de control de proyecto*, con el fin de definir el orden en que las mejoras deben llevarse a cabo y al mismo tiempo da una idea de que tan lejos está el desarrollador de concluir el sistema.

Cada paso consiste en realizar una mejora y removerla de la lista, esto se da diseñando, implementando y probando la mejora; y desarrollando el análisis parcial del sistema obtenido hasta este paso. Esto es lo que conocemos como fase de diseño, fase de implementación y fase de análisis. Este proceso se repite hasta que la lista de control esté vacía, con lo que obtenemos como resultado la implementación final.

Se puede dar que una tarea de la lista sea rediseñar componentes defectuosos encontrados en el análisis. Algo que es importante recalcar es que cada entrada de la lista se desarrolla en un sólo paso interactivo y debe ser lo suficientemente simple para ser entendido completamente. Esto minimiza los errores y reduce el trabajo de rediseño.

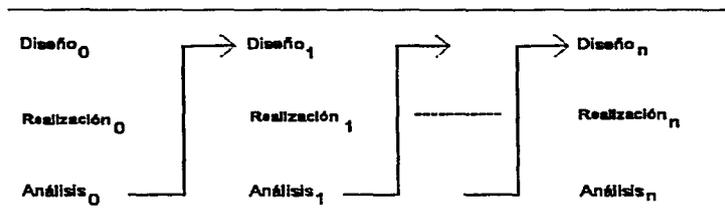


Figura. 2.3: Modelo de Mejoramiento Iterativo.

La Figura 2.3 muestra un esquema general de como se lleva a cabo el desarrollo de un software por medio del modelo de mejoramiento iterativo.

Modelo Espiral

En este modelo como su nombre lo sugiere, las actividades puede ser organizadas como una *espiral*. La espiral tiene varios ciclos. Su dimensión radial representa el costo de cada paso y su dimensión angular el progreso hecho para completar cada ciclo de la espiral.

Cada ciclo en la espiral comienza con la identificación de objetivos para ese ciclo y las diferentes alternativas para tomar esos objetivos son tomadas como reglas de control. Después se evalúan estas alternativas basadas en los objetivos. Esto también involucra incógnitas y riesgos. El siguiente paso es desarrollar estrategias que resuelvan las incógnitas y minimicen los riesgos.

El riesgo natural de este modelo permite acomodar y mezclar especificaciones, prototipos, simulaciones o aproximaciones que resuelvan el ciclo que se está desarrollando. Algo importante de este modelo es que cada ciclo de la espiral se completa por medio de una revisión que cubre todos los productos desarrollados durante el ciclo, incluyendo planes para el siguiente ciclo. Este modelo funciona bien tanto para desarrollo como para mejora de proyectos.

Como es un modelo relativamente nuevo, puede compaginar con otros modelos o estrategias, además de desarrollar actividades incorpora administración y planeación de actividades. Este modelo se recomienda para proyectos del alto riesgo.

La Figura 2.4 muestra el desarrollo de software usando el modelo espiral.

2.1.3 Administración en el Desarrollo de Software

Una administración integral es parte del desarrollo de software. El desarrollo de un proyecto grande involucra muchas personas que van a trabajar

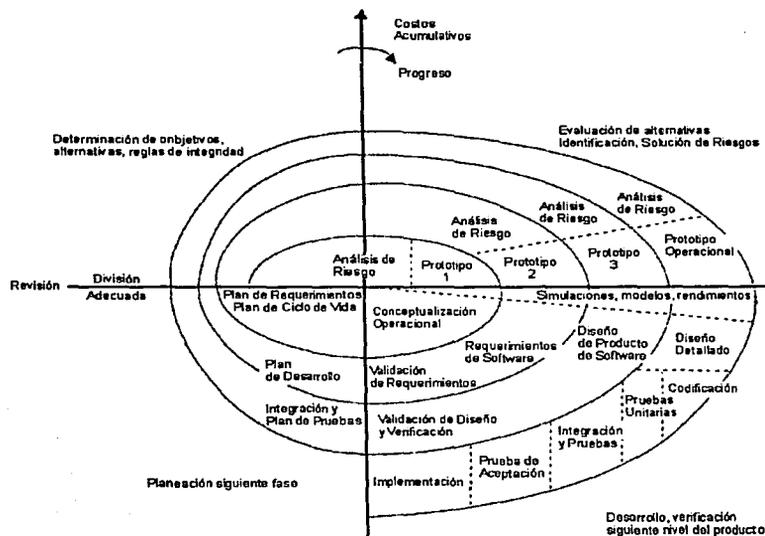


Figura. 2.4: Modelo de Espiral.

por un período largo de tiempo. Para que el trabajo de todas estas personas sea eficiente, se tienen que organizar apropiadamente y el trabajo debe ser dividido en diferentes partes, donde cada persona puede trabajar independientemente. Todo este trabajo debe ser monitoreado, y en caso de existir algún problema, se deben tomar las acciones correctivas necesarias para asegurar el progreso del proyecto.

Un hecho importante es que los procesos modelos no especifican cómo dividir los recursos de un proyecto para sus diferentes actividades. Tampoco especifican un esquema de trabajo para las actividades; es entonces cuando el administrador del proyecto debe tomar estas decisiones. Además de esto, las principales directrices del administrador del proyecto son:

- **Administrar los procesos.** El administrador debe asegurar que el proceso de desarrollo siga los lineamientos establecidos. Para esto el administrador debe preguntarse ¿cómo se que el proceso modelo especificado se esta siguiendo correctamente?

- **Colocar los recursos.** El administrador debe colocar los recursos óptimamente mientras se minimizan los requerimientos generales del proyecto.
- **Establecer un esquema de desarrollo.** El administrador determina un esquema de desarrollo realista y luego asegura que éste sea respetado.

Ahora bien, las actividades de la administración pueden dividirse en dos categorías: planeación del proyecto y monitoreo y control del mismo. La *planeación* junta actividades de la administración que tuvieron lugar antes que la fase de desarrollo comenzara. El *monitoreo y control* toma lugar cuando comienzan las actividades de desarrollo del proyecto.

La administración del proyecto comienza con la planeación, que es quizás la responsabilidad más grande de toda la administración. La meta de la planeación es desarrollar un plan de desarrollo para el software que involucra aspectos como portabilidad, re-uso, eficiencia y costos. Una planeación apropiada es un ingrediente crítico para el éxito de un proyecto. Esta planeación usualmente se produce antes que las actividades de desarrollo comienzan y se actualiza conforme avanza el desarrollo del proyecto.

Las actividades más fuertes de la planeación son los costos, la estimación de recursos, el desarrollo de un esquema de trabajo, la determinación del avance del trabajo basándose en cada fase del modelo empleado, la agrupación del personal involucrado en el desarrollo del proyecto, los controles de calidad, el plan de control y el monitoreo.

2.1.4 Métrica en el Software

La *métrica* del software es una medida de cuantificación que se usa para medir diferentes características del software o del proceso de desarrollo [4].

Todas las disciplinas en ingeniería tienen métricas para cuantificar características de sus productos y la ingeniería de software no podía ser la excepción. Se puede decir que la métrica del software es un área que comienza a surgir. Y como el software no tiene atributos físicos, las métricas comunes no ayudan mucho a cuantificar las diferentes características del software. Se han propuesto un número de métricas tales como el tamaño, la complejidad y el re-uso. Un ejemplo de estas métricas puede ser:

Tamaño:	Pequeño	Medio	Grande	Muy Grande
(LC)	Menor de 2K	2K-8K	8K-32K	Mayor de 32K
Líneas de Código				

Para el desarrollo de software se emplean dos tipos de métricas: *métrica del producto* y *métrica del proceso*. La primera se usa para cuantificar características del producto que está en desarrollo. La segunda se utiliza para

cuantificar características del ambiente del proceso que se esta empleando para desarrollar el software. También esta última mide consideraciones de productividad, costos, requerimientos de recursos y el efecto de desarrollar técnicas y herramientas durante el proceso.

Si una métrica no se mide directamente, llamamos a la métrica *indirecta*. Esto se ocupa para algunas cualidades de software, algunos ejemplos son: cuando se desea predecir la calidad del software tanto en las primeras etapas de desarrollo como después del diseño, en la predicción de modelos, etcétera.

Este tipo de métricas son importantes para la administración del proyecto, en especial para el monitoreo y control. Algo que es importante reconocer es que las métricas sólo proveen una escala de cuantificación de cualidades.

2.2 Diseño un Sistema

Una organización de software exitosa es aquella que consistentemente deposita la calidad del software en las necesidades de los usuarios. Para esto es necesario modelar una sólida arquitectura que nos permita llevar a cabo el desarrollo de un software con rapidez, eficiencia, efectividad y un mínimo de errores y fallas.

El modelado es una parte central de todas las actividades que dirigen el buen desarrollo de un software. El hombre construye modelos para comunicar las estructuras que desea, para expresar el comportamiento de sus sistemas, para visualizar y controlar la arquitectura de un software, así como también para poder entender mejor como desarrollar las ideas que esta construyendo y al mismo tiempo administrar los riesgos.

Con todo esto podemos decir que un *modelo*, se define como la simplificación de la realidad [12]. Un "buen" modelo provee los planos de construcción de un sistema, el cual, debe incluir aquellos elementos que tienen grandes efectos sobre el sistema y omitir aquellos que son irrelevantes. Un modelo puede ser *estructural*, en donde se enfatiza la organización del sistema, o bien puede mostrar *el comportamiento* enfatizando la dinámica del sistema.

Para poder construir un modelo existen cuatro principios básicos que hay que tener presentes:

1. La elección de los modelos tiene una fuerte influencia en como se ataca al problema y en como se le da forma a las soluciones.
2. Cada modelo puede ser expresado en diferentes niveles de abstracción.
3. Los mejores modelos son aquellos que se conectan con la realidad.
4. Un sólo modelo no es suficiente. Cada sistema se desarrolla de una mejor forma a través de un conjunto de modelos independientes.

2.2.1 UML

Teniendo como antecedente los problemas ya mencionados en la ingeniería de software y la inquietud de usar los nuevos lenguajes de programación para construir sistemas, a partir de 1998, *Grady Booch* (Rational Software Corporation), *Ivar Jacobson* (Objectory) y *James Rumbaugh* (General Electric) [12] se unieron para crear un lenguaje que permitiera:

- Modelar sistemas, desde los conceptos hasta la ejecución de un artefacto, usando técnicas orientadas a objetos.
- Administrar y dirigir las escalas de complejidad en la herencia y sistemas de misión crítica.
- Crear un lenguaje que fuera útil para humanos y máquinas.

El lenguaje que nació a partir de estas ideas se conoce con el nombre de UML¹. Es un lenguaje gráfico creado especialmente para visualizar, construir, especificar y documentar cada una de las partes que comprende el desarrollo de software.

UML es un lenguaje porque provee un vocabulario y las reglas para combinar las palabras de ese vocabulario con el propósito de comunicar. Se usa para visualizar ya que es un lenguaje gráfico a partir del cual se generan los planos de un sistema, aunque es importante mencionar que este lenguaje es más que un conjunto de símbolos gráficos. Se usa para especificar ya que se construyen con el modelos precisos, completos y sin ambigüedades. Se usa para construir ya que permite la generación de código fuente orientado a objetos. Se usa para documentar ya que permite generar la documentación de la arquitectura de un sistema y todos sus detalles, así mismo expresa los requerimientos, desarrollo y pruebas de un sistema dentro de un proyecto o plan de desarrollo.

UML entrega una forma de modelar las cosas conceptuales como lo son procesos de negocios y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de bases de datos y componentes de software para re-uso. Para poder entender este lenguaje es necesario conocer y entender la notación que ocupa en los diagramas. Cada uno de los diagramas tiene un objetivo específico dentro del desarrollo de software. Los diagramas más importantes dentro de UML son:

- Diagramas de Clases
- Diagramas de Casos de Uso
- Diagramas de Secuencia o Interacción

¹UML -Unified Modeling Language (Lenguaje de Modelamiento Unificado)-

- Diagramas de Paquetes
- Diagramas de Instalación

Diagramas de Clases

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contención. Un *diagrama de clases* está compuesto por los siguientes elementos:

- **Clase:** es la unidad básica que encapsula toda la información de un Objeto (un *objeto* es un ejemplar de una clase [12]). A través de ella podemos modelar el entorno en estudio. Esta compuesta por: atributos, métodos y visibilidad. En UML, una clase es representada por un rectángulo que posee tres divisiones: *superior, intermedio e inferior*.

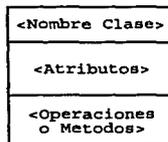


Figura. 2.5: Definición de Clases.

La figura anterior ilustra gráficamente una Clase, en donde:

- **Superior:** contiene el nombre de la Clase
- **Intermedio:** contiene los atributos (o variables del ejemplar) que caracterizan a la Clase, pueden ser:
 - * **Públicos (+):** indican que el atributo será visible tanto dentro como fuera de la clase, es decir es accesible desde todos lados.
 - * **Privados (-):** indican que el atributo sólo será accesible desde dentro de la clase, sólo sus métodos lo pueden acceder.
 - * **Protegidos (#):** indican que el atributo no será accesible desde fuera de la clase, pero si podrá ser accedido por métodos de la clase además de las subclases que se deriven.
- **Inferior:** contiene los métodos u operaciones, los cuales son la forma como interactúa el objeto con su entorno dependiendo de la visibilidad que puede ser:
 - * **Pública (+):** indica que el método será visible tanto dentro como fuera de la clase, es decir, será accesible desde todos lados.

- * **Privada (-):** indica que el método será sólo accesible desde dentro de la clase, sólo otros métodos de la clase lo pueden acceder.
- * **Protegida (#):** indica que el método no será accesible desde fuera de la clase, pero sí podrá ser accesado por métodos de las subclases que se deriven.
- **Relaciones:** son las interrelaciones que pueden existir entre dos o más clases (cada una con características y objetivos diferentes). Cada relación tiene una cardinalidad que indica el grado y nivel de dependencia, esta se anota en los extremos de la relación y pueden ser:
 - **uno a muchos:** 1..*, (1..n)
 - **0 a muchos:** 0..*, (0..n)
 - **número fijo:** m (m denota el número)

Los tipos de relación que pueden existir entre las clases son:

- **Herencia (Especialización/Generalización):** indica que una subclase hereda los métodos y atributos especificados por una Super Clase, por ende la Subclase además de poseer sus propios métodos y atributos, poseerá las características y atributos visibles de la Super Clase (pública, protegida). Este tipo de relación se indica por medio de la siguiente simbología:



Figura. 2.6: Símbolo de Herencia.

- **Agregación:** para modelar objetos complejos, no bastan los tipos de datos básicos que proveen los lenguajes: enteros, reales y secuencias de caracteres. Cuando se requiere componer objetos que son ejemplares de clases definidas por el desarrollador del sistema, tenemos dos posibilidades:
 - * **Por Valor.** Este es un tipo de relación estática, en donde el tiempo de vida del objeto incluido está condicionado por el tiempo de vida que lo incluye. Este tipo de relación es comúnmente llamada **Composición**. El Objeto base se construye a partir del objeto incluido, es decir, es “parte/todo”.
 - * **Por Referencia.** Este es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye. Este tipo de relación es comúnmente llamada **Agregación**. El Objeto base utiliza al incluido para su funcionamiento.



Figura. 2.7: Símbolo de Agregación/Composición.

La simbología de este tipo de relaciones es la siguiente:

- **Asociación:** permite asociar objetos que colaboran entre si. Cabe destacar que no es una relación fuerte, es decir, el tiempo de vida de un objeto no depende del otro. La simbología que se utiliza para esta relación es la siguiente:



Figura. 2.8: Símbolo de Asociación.

- **Dependencia o Instanciación (uso):** representa un tipo de relación muy particular, en la que una clase genera ejemplares, los cuales dependen de otros objetos o clases. La simbología que se utiliza para esta relación es la siguiente:



Figura. 2.9: Símbolo de Dependencia.

Casos Particulares:

- **Clase Abstracta:** indica que la clase no puede generar ejemplares pues posee métodos abstractos (aún no definidos, es decir, sin implementación). La única forma de utilizarla es definiendo subclases, que implementan los métodos abstractos definidos. Se denota con el nombre de la clase y de los métodos con letra "itálica".
- **Clase Parametrizada:** indica que se necesitan pasar parámetros a la clase para poder ser utilizada. Se denota con un subcuadro en el extremo superior de la clase, donde se especifican los parámetros que deben ser pasados a la clase para que esta pueda crear ejemplares.

Diagramas de Casos de Uso

El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan. Un diagrama de casos de uso consta de los siguientes elementos:

- **Actor:** es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra *rol*, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema. La simbología que se usa para representar un actor es la siguiente:



Figura. 2.10: Símbolo de Actor.

- **Caso de Uso:** es una operación o tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso. La simbología que se usa para representar un caso de uso es la siguiente:



Figura. 2.11: Símbolo de Caso de Uso.

- **Relaciones:** es el tipo de interacción que puede existir entre los mismos casos de uso y actores. Los tipos de relaciones que pueden existir son:
 - **Asociación:** es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación o caso de uso.
 - **Dependencia o Creación de Ejemplares:** es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se crea.
 - **Generalización:** es la relación más utilizada, cumple con una doble función dependiendo de su estereotipo, que puede ser de **Uso** o de **Herencia**. Es de **Uso** cuando se tiene un conjunto de características que son similares en más de un caso de uso y no se desea mantener copiada la descripción de la característica. Es de **Herencia** cuando un caso de uso es similar a otro. Solo está orientada para casos de uso.

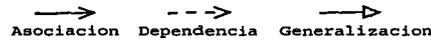


Figura. 2.12: Símbolos de Relaciones.

Diagramas de Interacción o Secuencia

El diagrama de interacción o secuencia, representa la forma en como un Cliente (Actor) u Objetos (Clases) se comunican entre sí en petición a un evento. Esto implica recorrer toda la secuencia de llamadas, de donde se obtienen las reponsabilidades claramente.

Dicho diagrama puede ser obtenido de dos partes, desde el Diagrama Estático de Clases o desde el Diagrama de Casos de Uso. Los componentes de un diagrama de interacción son:

- Un Objeto o Actor
- Mensaje de un objeto a otro objeto
- Mensaje de un objeto a sí mismo.

Los símbolos que se ocupan para representar cada uno de estos elementos son:



Figura. 2.13: Simbología del Diagrama de Interacción.

Las líneas punteadas representan las llamadas a métodos del objeto.

Diagramas de Paquetes

El diagrama de paquetes es una agrupación de elementos de modelado. Los paquetes por sí sólo pueden estar anidados con otros paquetes. Un paquete puede contener paquetes subordinados así como otros diferentes tipos de elementos de modelado, esto puede suceder ya que todos los elementos de modelado de UML pueden ser organizados en paquetes.

Hay que notar que los paquetes poseen elementos de modelado y fragmentos de modelado, estos son las bases para el control de la configuración,

almacenamiento, y control de acceso. Cada elemento puede ser directamente poseído por un sólo paquete, por lo que la jerarquía de un paquete es estrictamente un árbol. De cualquier forma, los paquetes también pueden ser referenciados por otros paquetes, los cuales pueden ser modelados usando algún tipo de estereotipo. Otra clase de dependencias entre paquetes usualmente implica que una o más dependencias existen entre los elementos.

La simbología que usan estos diagramas es la siguiente:

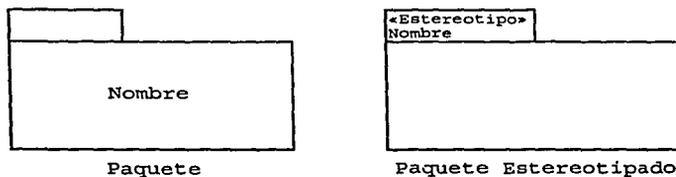


Figura. 2.14: Simbología del Diagrama de Paquetes.

Diagramas de Instalación

Los diagramas de instalación muestran la configuración de los elementos de procesamiento en tiempo de ejecución, componentes de software, procesos y objetos que viven en estos. Los componentes ejemplares de software representan el comportamiento en tiempo de ejecución de las unidades de código. Los componentes se conectan a otros componentes a través de la simbología de dependencia ya vista con anterioridad.

Estos diagramas utilizan la siguiente simbología de representación:

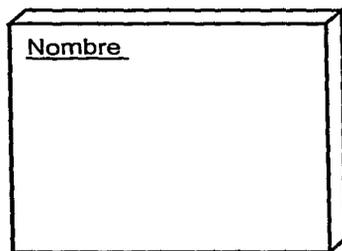


Figura. 2.15: Simbología del Diagrama de Instalación.

Capítulo 3

Bases de Datos

Los sistemas de bases de datos están diseñados para administrar grandes volúmenes de información, ese almacén de información es a menudo crucial para el funcionamiento general de una aplicación de cómputo mayor. La administración de datos envuelve tanto la definición de las estructuras de almacenamiento de la información como la provisión de mecanismos para la manipulación de los datos. Además, los sistemas de bases de datos deben proveer seguridad para la información almacenada. La ingeniería de bases de datos es una disciplina técnica que se aplica una vez que se ha definido el ámbito y flujo de la información. Por ello, es de gran importancia en el desarrollo de un software el definir la información que va a contener la base de datos, los tipos de peticiones que se podrían procesar, la manera en que se accederá a los datos y la capacidad de la base de datos.

3.1 Sistemas de Bases de Datos

Un *sistema de base de datos* es básicamente una colección de datos interrelacionados y un conjunto de programas para guardar y acceder datos [6]. La colección de datos, usualmente es conocida como la *base de datos*, la cual contiene información acerca de una empresa en particular. La finalidad general del sistema de base de datos es almacenar información y permitir a los usuarios recuperar, guardar y actualizar datos con base en peticiones solicitadas y hacer que la información esté disponible cuando se le solicite. La información en cuestión puede ser cualquier cosa que resulte de importancia para un usuario u organización; en otras palabras, todo lo que sea necesario para auxiliarse en el proceso general de su administración. Por otro lado, la base de datos es simplemente una especie de archivero electrónico; dicho de otra manera es un lugar donde se almacena un conjunto de archivos de datos grabados en la computadora. Los usuarios del sistema pueden realizar una variedad de operaciones sobre dichos archivos, por ejemplo:

- Agregar nuevos archivos vacíos a la base de datos

- Insertar datos dentro de los archivos existentes
- Recuperar datos de los archivos existentes
- Modificar datos en archivos existentes
- Eliminar datos de los archivos existentes
- Eliminar archivos existentes de la base de datos

Un sistema de base de datos comprende cuatro componentes principales: **datos, hardware, software y usuarios.**

Datos

En general, los *datos* de una base de datos –por lo menos de un sistema grande– serán tanto integrados como compartidos. Al ser integrados, queremos decir que podemos imaginar a la base de datos como una unificación de varios archivos que de otro modo serían distintos, con una redundancia entre ellos eliminada al menos parcialmente. Cuando decimos compartidos, hablamos de que las piezas individuales de la base de datos pueden ser compartidas por diferentes usuarios y que cada uno de ellos puede tener acceso a la misma pieza de datos, probablemente con fines diferentes.

Hardware

Los componentes de *hardware* del sistema constan de:

- Los procesadores de hardware y la memoria principal asociada a los usuarios para apoyar la ejecución del software del sistema de base de datos.
- Los volúmenes de almacenamiento secundario –principalmente discos magnéticos– que se emplean para contener los datos almacenados, junto con los dispositivos asociados de Entrada y Salida, E/S, controladores de dispositivos y canales de E/S entre otros.

Software

Entre la base de datos física – es decir, los datos como están almacenados físicamente– y los usuarios del sistema, hay una capa de *software* conocida de manera distinta como el *administrador de base de datos*; o más comúnmente como el *sistema de administración de base de datos* (DBMS¹) el cual es un programa que almacena y accesa datos de forma eficiente [7]. El DBMS maneja todas las solicitudes de acceso a la base de datos, por lo tanto su función principal consiste en ocultar a los usuarios de la base de datos los

¹DBMS –DataBase Management System (Sistema Administrador de Base de Datos)–

detalles al nivel de hardware. En otras palabras, el DBMS ofrece a los usuarios una percepción de la base de datos que está, en cierto modo, por encima del nivel del hardware y que maneja las operaciones del usuario expresadas en términos de ese nivel más alto de percepción.

Usuarios

Dentro de un sistema de administración de bases de datos consideramos tres clases grandes de usuarios, los cuales interactúan:

- **Programadores de aplicaciones.**- son responsables de escribir los programas de aplicación de base de datos en algún lenguaje de programación. Tales programas pueden ser aplicaciones convencionales por lotes o pueden ser aplicaciones en línea, cuyo propósito es permitir al usuario final el acceso a la base de datos desde una estación de trabajo o terminal en línea.
- **Usuarios finales.**- son quienes interactúan con el sistema desde estaciones de trabajo o terminales en línea. Un usuario final puede tener acceso a la base de datos a través de las aplicaciones en línea, o bien pueden usar una interfaz proporcionada como parte integral del software del sistema de base de datos. Estas interfaces controladas por menús o por formularios tienden a facilitar el uso a personas que no cuentan con una capacitación formal. En contraste, las interfaces controlados por comandos tienden a requerir cierta experiencia profesional.
- **Administrador de Datos (DA²).**- es la persona que toma las decisiones de estrategia y política con respecto a los datos de la empresa. **Administrador de la Base de Datos (DBA³).**- es el responsable del control general del sistema a nivel técnico.

El DBA está encargado de crear la base de datos real e implementar los controles técnicos necesarios para hacer cumplir las diversas decisiones de políticas hechas por el administrados de datos. Es responsable también de asegurar que el sistema opere con el rendimiento adecuado y de proporcionar una variedad de otros servicios técnicos. Por lo regular tiene a su disposición un equipo de programadores de sistemas y otros asistentes técnicos; sin embargo para fines de simplicidad, es conveniente suponer que el DBA es de hecho un solo individuo. En general, las tareas que desarrolla el DBA comprenden al menos las siguientes:

- **Definir el esquema conceptual.**

²DA -Data Administrator (Administrador de Datos)-

³DBA - Database Administrator (Administrador de Bases de Datos)-

Es trabajo del administrador de datos decidir exactamente el contenido de la base de datos. En otras palabras, identificar las entidades de interés para la empresa e identificar la información que hay que registrar de dichas entidades. Por lo general, a este proceso se le conoce con el nombre de *diseño lógico* —en ocasiones conceptual— de la base de datos. Una vez hecho esto, el DBA creará el esquema conceptual correspondiente.

- **Definir el esquema interno.**

El DBA también decidirá la forma en que va a ser representada la información en la base de datos. A este proceso se le conoce como *diseño físico* de la base de datos. Una vez realizado el diseño físico, el DBA deberá crear la definición de la estructura de almacenamiento correspondiente, es decir, el esquema interno.

- **Establecer un enlace con los usuarios.**

Es asunto del DBA enlazarse con los usuarios para asegurar que los datos necesarios estén disponibles, así como también escribir los esquemas externos necesarios. Otros aspectos de la función de enlace con los usuarios incluyen la asesoría sobre el diseño de aplicaciones; una capacitación técnica; ayuda en la determinación y resolución de problemas; así como otros servicios profesionales similares.

- **Definir las restricciones de seguridad y de integridad.**

El DBA debe establecer los niveles de acceso así como cada una de las posibles funciones que tendrán los niveles de acceso. De igual forma, se debe asegurar de llevar un estricto control de los cambios en la información a través de monitoreo que permita conocer cada una de las acciones llevadas a cabo por cada uno de los usuarios en el sistema. Esto se hace con el fin de reconocer los posibles autores de cambios en la información dentro del sistema. Definir este tipo de restricciones puede ser visto como parte del esquema conceptual.

- **Definir las políticas de carga y descarga de datos.**

Una vez que la empresa se compromete con un sistema de base de datos, se vuelve drásticamente dependiente del funcionamiento exitoso de dicho sistema. En el caso de que se produzca un daño en cualquier parte de la base de datos —ocasionado, por ejemplo, por un error humano o por una falla en el hardware o en el sistema operativo— resulta esencial poder reparar los datos afectados con el mínimo de demora y con tan poco efecto como sea posible sobre el resto del sistema. El DBA debe definir e implementar un esquema apropiado de control de daños que comprenda la descarga periódica

de la base de datos en un dispositivo de almacenamiento de respaldo y la carga de la base de datos cuando sea necesario, a partir del respaldo más reciente.

- **Supervisar el rendimiento y responder a los cambios.**

El DBA es responsable de organizar el sistema de tal manera que se obtenga el rendimiento "ideal para la empresa" y de hacer los ajustes que sean apropiados conforme las necesidades cambien. Todo cambio a nivel interno de almacenamiento físico debe estar acompañado por el cambio correspondiente en la definición de la transformación conceptual, de manera que el esquema conceptual se mantenga constante.

3.2 Base de Datos

Resulta normal referirse a los datos de la base de datos como "persistentes". Por este motivo podemos definir una *base de datos* como un conjunto de datos persistentes que es utilizado por los sistemas de aplicación de alguna empresa⁴ [11]. Por persistentes debemos entender de una forma intuitiva, que los datos de la base de datos difieren de otros más efímeros. Siendo más precisos decimos que los datos de la base de datos "persisten" debido a que una vez que son aceptados por el DBMS para entrar en la base de datos, en lo sucesivo sólo pueden ser removidos de la base de datos por alguna solicitud explícita (procedimiento) al DBMS.

3.2.1 Ventajas

En general, las propiedades de una base de datos pueden ser tan simples o tan complejas como queramos. Pero si hablamos de sus ventajas tal vez sea más fácil apreciarlas comparando un sistema de bases de datos contra los métodos tradicionales basados en papel, para llevar un registro. He aquí algunas:

- **Compactación:** No hay necesidad de archivos en papel voluminosos.
- **Velocidad:** La máquina puede recuperar y actualizar datos más rápidamente que un humano.
- **Menos trabajo laborioso:** Se puede eliminar gran parte del trabajo de llevar los archivos a mano. Las tareas mecánicas siempre las realizan mejor las máquinas.
- **Disponibilidad y Veracidad:** En el momento que la necesitemos, tendremos a nuestra disposición información precisa y actualizada.

⁴Empresa.- es simplemente un término genérico conveniente para identificar cualquier organización independiente de cualquier tipo.

- **Datos compartidos:** Varios usuarios pueden consultar, agregar y modificar datos al mismo tiempo.
- **Minimización de redundancia:** Tratar que se repitan lo menos posible información en la base de datos.
- **Evitar inconsistencias:** Asegurar, hasta cierto grado, que los datos que estén en la base de datos estén relacionados correctamente.
- **Manejo de transacciones⁵ más eficiente:** Asegurar que las operaciones que realizan los usuarios en la base de datos se concluyan siempre, o en caso de existir un error, que estas no se lleven a cabo.
- **Integridad:** Tratar que los datos que estén dentro del sistema sean congruentes con las restricciones de tipo de dato, tamaño, etcétera.
- **Control Seguridad:** Asegurar que los usuarios que usan el sistema tengan acceso a las funciones que les correspondan dependiendo de su nivel de acceso.

3.2.2 Independencia de Datos

Las aplicaciones actuales (al menos implantadas en sistemas menos modernos, por oposición a los sistemas de administración de base de datos más recientes) con frecuencia dependen de los datos. Dicho de otra forma, los requerimientos de la aplicación en cuestión determinan la forma de organizar los datos en almacenamiento secundario y la técnica para acceder a ellos. El conocimiento de esa información de los datos y esa técnica de acceso forman parte integral de la lógica y el código de esa aplicación.

La habilidad para modificar un esquema de definición, en un nivel de diseño, sin afectar a otros esquemas de definición recibe el nombre de *independencia de datos*.

Existen dos niveles de independencia de datos:

- **Independencia de Datos Física.-** es la habilidad para modificar el esquema físico sin que los programas de aplicación tengan que ser reescritos. Modificaciones en el diseño físico son ocasionalmente necesarios para mejorar el desempeño.
- **Independencia de Datos Lógica.-** es la habilidad para modificar el esquema lógico sin que los programa de aplicación tengan que ser reescritos. Modificaciones en el diseño lógico son necesarias cuando la estructura lógica de la base de datos es alterada.

Para lograr la independencia de los datos tenemos que considerar que:

⁵Transacción.- es una colección de operaciones que se llevan a cabo con un orden lógico para llevar a cabo una función de la base de datos.

1. Cada aplicación requiere una vista diferente de los mismos datos.
2. El DBA debe tener la libertad para modificar la estructura de almacenamiento o la técnica de acceso (o las dos cosas) para adaptarlas a cambios en los requerimientos, sin tener que modificar las aplicaciones ya existentes.

De lo anterior se desprende que la independencia de los datos es un objeto primordial de los sistemas de base de datos. Esta *independencia* puede definirse como la inmunidad de las aplicaciones ante los cambios en la estructura de almacenamiento y la técnica de acceso [6], lo cual implica, por supuesto, que las aplicaciones en cuestión no dependan de una estructura de almacenamiento o una técnica de acceso específica.

3.3 Modelo Relacional y otros Modelos

En una base de datos lo más importante son los datos. La palabra *datos* se deriva del vocablo latín para “dar”; por lo tanto, los datos son en realidad *hechos dados*, a partir de los cuales los usuarios pueden inferir hechos adicionales. Siendo tan importantes, resulta primordial elegir el modelo de datos más adecuado para poder organizarlos dentro de la base de datos.

Un *modelo de datos* es una definición lógica, independiente y abstracta de los objetos, operadores y demás que en conjunto constituyen la máquina abstracta con la que interactúan los usuarios [11]. Los objetos de un modelo de datos nos permiten modelar la estructura de los datos, mientras que los operadores nos permitirán modelar su comportamiento.

Una razón por la cual los sistemas de bases de datos se han vuelto tan dominantes en todo el mundo, es que manejan en forma muy directa la interpretación precedente de los datos. En especial los productos DBMS que se basan en el *modelo de datos relacional* (los “sistemas relacionales”) han venido a dominar el mercado de las bases de datos. Lo que es más, la mayor parte de la investigación sobre bases de datos en los últimos 30 años, se ha basado en este modelo. De hecho, la presentación del modelo relacional en 1969-70 fue de manera innegable el evento más importante en toda la historia de las bases de datos. Por otro lado, el modelo relacional está sólidamente fundamentado en la lógica y en las matemáticas y por lo tanto ofrece un vehículo ideal para la enseñanza de los principios de las bases de datos. Por estas razones, usaremos este modelo como pilar para el desarrollo del proyecto que nos ocupa en el resto del trabajo.

Estos modelos usan una colección de tablas para representar datos y relaciones junto con los datos. Cada tabla tiene múltiples columnas, y cada columna representa en general un atributo, el cual tiene un nombre único. Los operadores de alcance del usuario generan tablas a partir de las existentes. Estos sistemas reciben el apelativo de “relaciones” porque la palabra

“relación” no es más que un término matemático para referirse a una tabla. De hecho, en casi todas las situaciones prácticas los términos “relación” y “tabla” pueden tomarse como sinónimos.

Para distinguir los sistemas relacionales de los no relacionales, debe hacerse la siguiente consideración. Como ya se dijo, el usuario de un sistema relacional percibe los datos como tablas, únicamente. En cambio, el usuario de un sistema no relacional percibe otras estructuras de datos como pueden ser árboles, gráficas, listas ligadas, etcétera.

Los sistemas de bases de datos pueden clasificarse en forma conveniente de acuerdo con la descripción de datos, relaciones de los datos, semántica de los datos y consistencia. Las clasificaciones que se proponen caen en tres grupos:

1. **Modelo Lógico Orientado a Objetos.** Se usan para describir datos en los niveles lógico y de vistas. Se caracterizan por el hecho de proveer capacidades de estructuración algo flexibles y permiten que la validación de datos se defina explícitamente. Algunos de los modelos que recaen en esta clasificación son:
 - Modelo Entidad-Relación
 - Modelo Orientado a Objetos
 - Modelo Semántico de Datos
 - Modelo Funcional de Datos

2. **Modelo Lógico Basado en Registros.** Se usan para describir datos en los niveles lógico y de vistas. Se caracterizan por el hecho de especificar toda la estructura lógica de la base de datos y proveer una descripción de alto nivel para la implementación. Algunos de los modelos de registros más aceptados son:
 - Modelo Relacional
 - Modelo de Red
 - Modelo Jerárquico

3. **Modelo de Datos Físico.** Se usan para describir datos en los niveles más bajos. Se caracterizan por capturar aspectos de la implementación de la base de datos. Hay muy pocos modelos físicos en uso, dos de los más usados son:
 - Modelo Unificado
 - Modelo de Memoria

Modelo Entidad Relación

Este modelo está basado en la percepción del mundo real que consiste en una colección de objetos básicos, llamados *entidades*, y *relaciones* entre estos objetos. Una entidad es una "cosa" u "objeto" real o abstracto que se distingue de otros objetos. Cada entidad se describe por medio de un conjunto de *atributos*; por ejemplo: la entidad *persona* tiene los atributos *nombre-persona*, *rfc*, *domicilio*, *ciudad*. Una relación es una asociación entre varias entidades; ejemplo: la entidad *persona* y la entidad *cuenta* tienen la relación *depósitos*.

Además de la entidades y relaciones, el modelo Entidad-Relación representa la forma y las reglas de validación de como se debe estructurar el contenido de la base de datos. Para esto se ocupan los diagramas Entidad-Relación por medio los cuales se expresa gráficamente la estructura lógica de la base de datos. En este diagrama cada componente se etiqueta de forma única.

La Figura 3.1 muestra el diagrama entidad relación del ejemplo mencionado anteriormente.

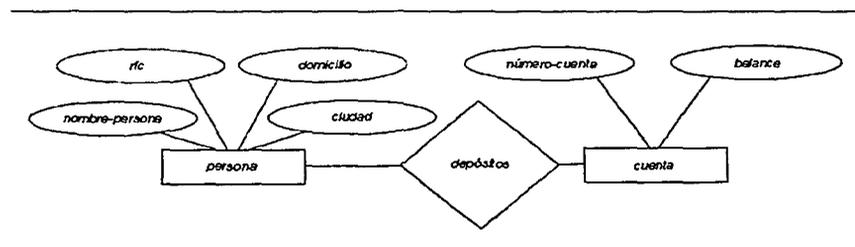


Figura. 3.1: Modelo Entidad Relación.

Modelo Orientado a Objetos

Este modelo está basado en objetos. Cada objeto contiene valores almacenados en *ejemplos de variables*. También cada objeto contiene cuerpo que está formado por código y que es el encargado de operar con el objeto. A estos cuerpos formados por código les llamamos *métodos*.

Los objetos que contienen los mismos tipos de valores y los mismos métodos son agrupados en *clases*. Una clase puede ser vista como una definición de tipos para los objetos. La única forma en que un objeto pueda acceder datos de otro objeto es por medio de la invocación de un método de ese otro objeto. Esto recibe el nombre de *envío de mensajes* al objeto.

Para ilustrar este modelo podemos considerar un objeto que represente una *cuenta de banco*. Este objeto contiene variables generadas a partir de un *número de cuenta y balance*. Y también contiene un método de *pago-de-intereses*.

A diferencia del modelo entidad relación, cada objeto tiene su propio identificador único, independiente de los valores que contenga. Esto quiere decir que a pesar que dos objetos contengan el mismo valor, siempre son distintos. Esta distinción en los objetos se mantiene a nivel físico a través de la asignación de identificadores físicos para cada objeto.

Modelo de Red

En este modelo los datos son representados por medio de registros, las relaciones entre los datos se representan por medio de *ligas*, y que pueden ser vistas también como apuntadores. Los registros son almacenados como *colecciones de gráficas* arbitrariamente. Un ejemplo de este modelo puede ser:

La Figura 3.2 muestra un ejemplo de una base de datos basada en el modelo de red.

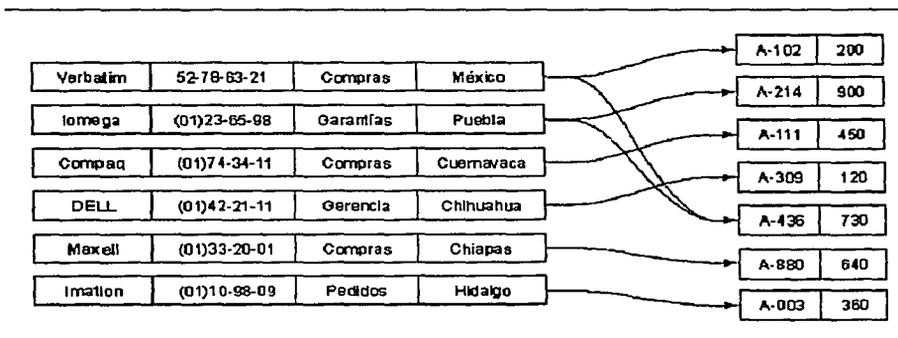


Figura. 3.2: Modelo de Red.

Modelo Jerárquico

Este modelo es similar al modelo de red en el sentido en que los datos y las relaciones entre los datos son representados por registros y ligas, respectivamente. Sin embargo, difiere del modelo anterior en que los registros son organizados como una colección de árboles en lugar de gráficas.

La Figura 3.3 muestra un ejemplo de una base de datos contruida con el modelo jerárquico.

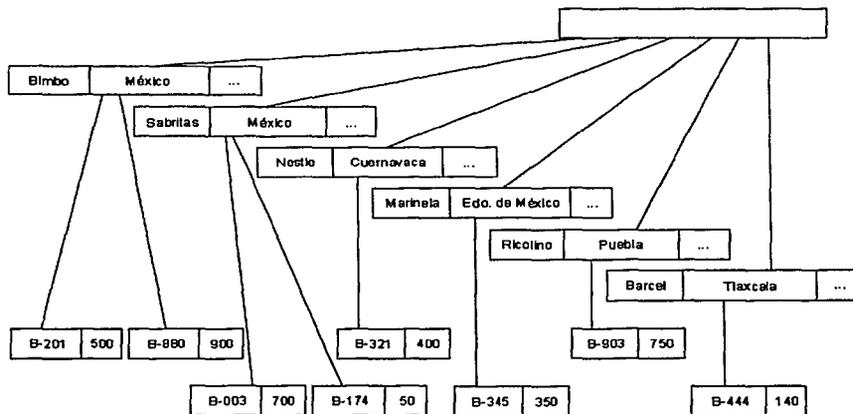


Figura. 3.3: Modelo Jerárquico.

3.4 Lenguajes de Base de Datos

Los sistemas de base de datos proveen dos diferentes tipos de lenguajes: el primero para especificar el esquema de la base de datos, el segundo para expresar las consultas y actualizaciones hechas a la base de datos.

3.4.1 Lenguaje de Definición de Datos

Un *lenguaje de definición de datos* (DDL⁶) es un lenguaje por medio del cual se especifica el esquema de una base de datos por medio de un conjunto de expresiones. El resultante de compilar el conjunto de expresiones de este lenguaje es un conjunto de tablas en archivo especial que recibe el nombre de *diccionario de datos* o *directorio de datos*.

El diccionario de datos es un archivo que contiene *metadatos*, es decir, datos acerca de los datos. Este archivo se lee antes que cualquier dato de la base de datos sea leído o modificado en el sistema de base de datos.

⁶DDL -Data Definition Language (Lenguaje de Definición de Datos)-

Las estructuras de almacenamiento y los métodos de acceso usados por el sistema de base de datos son especificados por un conjunto de definiciones en un tipo especial de DDL llamado *lenguaje de almacenamiento y definición de datos*. Como resultado de compilar estas definiciones se obtiene un conjunto de instrucciones para especificar la implementación detallada de los esquemas de la base de datos.

3.4.2 Lenguaje de Manipulación de Datos

Cuando hablamos de manipulación de datos, entendemos:

- Accesar datos almacenados en una base de datos
- Insertar nuevos datos en una base de datos
- Borrar datos de una base de datos
- Modificar datos de una base de datos

Por lo mismo, un *lenguaje de manipulación de datos* es un lenguaje que permite a los usuarios acceder o manipular datos organizados por un modelo de datos apropiado. Básicamente hay dos tipos:

- **Procedurales** que requieren de un usuario para especificar qué datos se necesitan y cómo conseguirlos.
- **No Procedurales** los cuales requieren de un usuario para especificar qué datos se necesitan sin especificar cómo conseguirlos.

3.4.3 SQL

SQL es el lenguaje estándar para trabajar con bases de datos relaciones y es soportado prácticamente por todos los productos del mercado. Originalmente fue desarrollado en ©*IBM Research* a principios de los años setenta; su nombre oficial es **Estándar Internacional del Lenguaje de Base de Datos SQL**. Aunque en la actualidad existen numerosas versiones de SQL, todos estos están constituidos principalmente por:

- **Lenguaje de Definición de Datos (DDL)**.- provee comandos para definir esquemas de relación, eliminación de relaciones, creación de índices y modificación de esquemas de relación.
- **Lenguaje de Manipulación de Datos Interactivo (DML)**.- incluye un lenguaje de consultas basado en el álgebra relacional y el cálculo relacional. Incluye comandos para insertar, borrar y modificar datos en la base de datos.
- **Agregado DML**.- está diseñado para uso general con lenguajes de programación, como PL/I, Cobol, Pascal, Fortran y C.

- **Definición de Vistas.**- una *vista* es un conjunto de comandos de SQL con un orden específico que se usa para facilitar el uso de cualquier serie de comandos que se repitan constamente por los usuarios.
- **Seguridad.**- incluye comandos para especificar accesos a relaciones y vistas.
- **Integridad.**- incluye comandos para especificar principios de integridad, es decir, que los datos que van a ser almacenados en la base de datos cumplan con ciertas reglas establecidas por el DBA.
- **Control de Transacciones.**- incluye comandos para especificar el principio y el fin de las transacciones.

Por último, debemos reconocer que SQL está muy lejos de ser el lenguaje relacional perfecto; padece de faltas tanto de omisión como de comisión. Esto se debe a que a veces es necesario adaptar la teoría en que se basa a la programación.

3.5 Sistema de Administración de Base de Datos

Un *sistema de administración* de base de datos (DBMS) consiste de una colección de datos interrelacionados y un conjunto de programas para acceder esos datos [6]. La principal meta de un DBMS es la de proveer un ambiente que al mismo tiempo sea conveniente y eficiente para acceder y almacenar la información en la base de datos. De manera conceptual, lo que sucede es lo siguiente:

1. Un usuario emite un petición de acceso, utilizando algún sublenguaje de datos específico, por lo regular SQL.
2. El DBMS intercepta esta petición y la analiza.
3. El DBMS inspecciona, en su momento el esquema externo para ese usuario, la transformación externa-conceptual correspondiente, el esquema conceptual, la transformación conceptual-interna y la definición de la estructura de almacenamiento.
4. El DBMS ejecuta las operaciones necesarias sobre las bases de datos almacenadas

Profundizando con más detalle en las funciones del DBMS tenemos las siguientes:

Definición de Datos

El DBMS debe ser capaz de aceptar definiciones de datos (esquemas externos, el esquema conceptual, el esquema interno y todas las transformaciones respectivas) en la forma fuente y convertirlas a la forma objeto correspondiente.

Manipulación de Datos

El DBMS debe ser capaz de manejar peticiones para recuperar, actualizar o eliminar datos existentes en la base de datos o agregar nuevos datos.

Optimización y Ejecución

Las peticiones resultantes de la manipulación de datos, planeadas o no, deben ser procesadas por el componente *optimizador*, cuya finalidad es determinar una forma eficiente de implementar la petición. Las peticiones optimizadas se ejecutan entonces bajo el control del *administrador en tiempo de ejecución*.

Seguridad e Integridad de los Datos

El DBMS debe vigilar las peticiones del usuario y rechazar todo intento de violar las restricciones de seguridad y de integridad definidas por el DBA. Estas tareas pueden realizarse durante el tiempo de compilación, de ejecución o entre ambos.

Recuperación de Datos y Concurrencia

El DBMS debe imponer ciertos controles de recuperación y concurrencia; o más probablemente, algún otro componente de software relacionado, denominado comúnmente *administrador de transacciones* o *monitor de procesamiento de transacciones*.

Diccionario de Datos

El DBMS debe proporcionar una función de *diccionario de datos*. Este diccionario puede ser visto como una base de datos por derecho propio (una base de datos del sistema más que una del usuario). El diccionario contiene "información acerca de los datos" (*metadatos* o *descriptores*); es decir, *definiciones* de otros objetos del sistema, en lugar de simples "datos en bruto". En particular, todos los diversos esquemas y transformaciones (externos, conceptuales) y todas las diversas restricciones de seguridad y de integridad, serán almacenadas en el diccionario, tanto en forma fuente como objeto.

Rendimiento

El DBMS debe realizar todas las tareas antes identificadas de la manera más eficientemente posible. Esto involucra el aprovechamiento del espacio de la mejor manera posible, la ejecución de transacciones en el menor tiempo posible, minimizar el desperdicio de recursos al máximo, etcétera.

La Figura 3.4 muestra la interacción del DBMS con las aplicaciones, los usuario finales y la base de datos.

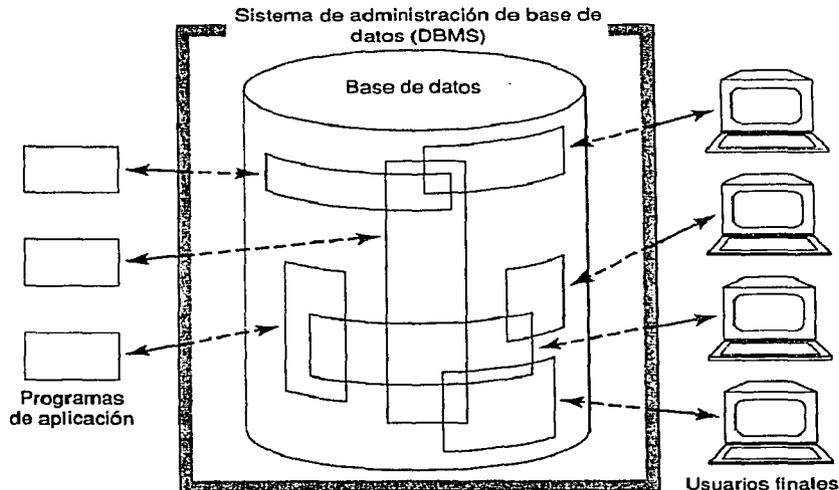


Figura. 3.4: Sistema Administrador de Bases de Datos.

3.6 Estructura de Datos Relacional

El modelo relacional se ha establecido por si mismo como el modelo de datos primario para aplicaciones de procesamiento de datos comerciales. Una teoría substancial existe en las bases de datos relacionales. Está teoría asiste en el diseño de las base de datos relacionales y en el progreso eficiente de las peticiones de los usuarios por información de la base de datos, de ahí que en la actualidad el modelo relacional sea usado en numerosas aplicaciones fuera del dominio tradicional del procesamiento de datos.

Los sistemas relacionales se basan en un fundamento formal, o teórico, denominado *modelo relacional de datos*. De manera intuitiva, lo que esta afirmación significa es que en dichos sistemas existe:

1. **Aspecto estructural:** El usuario percibe la información de la base de datos como tablas únicamente.
2. **Aspecto de integridad:** Estas tablas satisfacen ciertas restricciones de integridad, algunos ejemplos pueden ser: cardinalidad, pertenencia, relación numérica, etcétera.
3. **Aspecto de manipulación:** Los operadores disponibles para que el usuario manipule estas tablas. En particular, tres operadores son importantes:
 - La operación *restringir* o seleccionar
 - La operación *proyectar* o extraer
 - La operación *juntar* o reunir

Para dar una definición del modelo relacional para fines de futuras referencias (no obstante que la denificación es bastante abstracta), podemos decir que el **modelo relacional** consta de cinco componentes:

1. Un conjunto abierto de **tipos escalares**, incluyendo en particular el tipo *lógico* o *valor verdadero*.
2. Un **generador de tipos de relación** y una interpretación propuesta de dichos tipos.
3. Herramientas para definir **variables de relación** de dichos tipos de relación generados.
4. Una operación **asignación relacional** para asignar valores de relación a las variables de relación mencionadas.
5. Un conjunto abierto de **operadores relacionales** genéricos para derivar valores de relación de otros valores de relación.

Como podemos ver, el modelo relacional es mucho más que sólo "tablas con restringir, proyectar y juntar", aunque a menudo se le caracteriza informalmente de esa manera.

Una vez conocida su definición podemos mencionar de la misma forma sus componentes principales:

- Una *relación* corresponde a lo que hasta ahora hemos llamado en general una tabla.

- Una *tupla* corresponde a una fila de esa tabla y un *atributo* a una columna. El número de tuplas se denomina *cardinalidad* y el número de atributos se llama *grado*.
- La *llave primaria* 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.
- Un *dominio* es una colección de valores, de los cuales uno o más atributos (columnas) obtienen sus valores reales.

Tipos de Relaciones

Para concluir esta sección identificaremos algunos de los diferentes tipos de relaciones que pueden existir en un sistema relacional.

1. **Relaciones Base** (llamadas también relaciones reales): Una relación base corresponde al concepto de tabla en SQL; es decir, es una relación autónoma, con nombre. En otras palabras, las relaciones base son aquellas cuya importancia, para la aplicación en cuestión, es tal que el diseñador de la base de datos ha decidido darles un nombre y hacerlas parte directa de la base de datos en sí, a diferencia de otras relaciones cuya naturaleza es más efímera; por ejemplo, el resultado de una consulta.
2. **Vistas** (llamadas también relaciones virtuales): Una vista es una relación derivada, con un nombre, representada dentro del sistema exclusivamente mediante su definición en términos de otras relaciones con nombre; no poseen datos almacenados propios, separados y distinguibles, a diferencia de las relaciones base.
3. **Instantáneas**: Una instantánea (*snapshot*) es también una relación derivada, con nombre, como una vista. Pero a diferencia de las vistas, las instantáneas son reales, no virtuales; es decir, están representadas no sólo por su definición en términos de otras relaciones con nombre, sino también por sus propios datos almacenados.

3.7 Base de Datos Relacionales

Una *base de datos relacional* es una base de datos percibida por el usuario como una colección de relaciones normalizadas de diversos grados que varía con el tiempo. Esta consiste de una colección de *tablas*, donde a cada uno se le asigna un único nombre.

Resumiendo, podemos decir que en términos tradicionales, una *relación* se asemeja a un archivo, una *tupla* a un registro (ocurrencia de no tipo) y un

atributo a un campo (tipo, no ocurrencia). Pero estas correspondencias son aproximadas, en el mejor de los casos. Una relación no debe considerarse como sólo un archivo, sino más bien como un archivo disciplinado, siendo el resultado de esa disciplina una simplificación considerable de las estructuras de datos con las cuales debe interactuar el usuario, lo cual a su vez simplifica los operadores requeridos para manejar estructuras.

Para concluir este capítulo, y nuestro análisis de la estructura de datos relacional, haremos un resumen de las características principales de los archivos relacionales, los cuales se distinguen de los archivos tradicionales (no disciplinados). Estas características no son las únicas pero sí las más importantes que podemos resaltar en estos archivos.

1. Cada "archivo" contiene sólo un tipo de registros.
2. Los campos no tienen un orden específico, de izquierda a derecha.
3. Los registros no tienen un orden específico, de arriba hacia abajo.
4. Cada campo tiene un solo valor.
5. Los registros poseen un campo identificador único (o combinación de campos) llamado clave primaria.

Así, todos los datos de una base de datos relacional se representan de una y sólo una manera, a saber, por su valor explícito, esta característica se denomina en ocasiones "principio básico del modelo relacional". En particular, las conexiones lógicas dentro de una relación y entre las relaciones se representan mediante esos valores; no existen "ligas" o apuntadores visibles para el usuario, ni grupos repetitivos visibles para el usuario.

Capítulo 4

Análisis del Sistema

El término "análisis en ingeniería de software" ha sido definido como una variedad de procedimientos y técnicas que tratan de proveer la mayor cantidad posible de información acerca de un problema, la cual es fundamental para el desarrollo de un software. Actualmente todos los métodos usados para el desarrollo de un software cuentan con una etapa de análisis que se aproxima mucho a lo que la ingeniería de software propone de forma ideal de muchas maneras. El decidir que método, técnica o modelo usar para desarrollar un software mejora la calidad del producto, pero muchas veces esto no es una tarea fácil. Al final, el método elegido deberá reducir los costos de desarrollo, optimizar los tiempos de trabajo, mejorar la eficiencia de trabajo, etcétera. En especial el análisis es el pilar de mayor importancia en el desarrollo de un sistema, ya que a partir de los resultados de esta etapa, se basan y se llevan a cabo todas las demás.

4.1 Origen del Sistema

El Departamento de Personal Académico (DPA) de la Facultad de Ciencias, se vió en la necesidad de contar con un sistema de información que le permitiera realizar el registro, la consulta y manipulación de datos y formas únicas con las que trabaja, aprovechando el uso de los recursos de cómputo con los que cuenta actualmente, de manera que le permita realizar todas sus funciones de forma eficiente. Esta necesidad surge con la finalidad de sustituir el software con el que cuenta actualmente este departamento, el cual, presenta diversas limitaciones que van: desde un manejo inadecuado de la información hasta la falta de funciones básicas que son necesarias para que el personal de este departamento lleve a cabo todas sus funciones de una forma rápida y eficiente.

Ante esta situación, la Coordinación de Servicios de Cómputo (CSC) de esta facultad, se dió a la tarea de desarrollar un sistema de información basado en computadora que le permita al DPA la captura de la información

una sola vez, el manejo de la información rápida y eficientemente, la eliminación de errores inherentes al manejo de expedientes y grandes volúmenes de datos, la reducción de respuesta en la elaboración de contratos y entrega de informes y estadísticas de la información que maneja, así como también la consolidación y centralización de la información de forma segura a través de niveles de acceso a funciones del sistema y métodos de respaldo en tiempos establecidos.

4.2 Introducción

La etapa de análisis tiene como objetivo primordial el presentar de una forma clara y precisa todos los requerimientos identificados para el desarrollo del sistema de información requerido, con el fin de que se realice una revisión formal de estos, antes de continuar con las siguientes fases.

Los requerimientos identificados deberán satisfacer los aspectos fundamentales de almacenamiento de información, proceso de datos, mecanismos internos de validación, así como la producción de los datos de salida necesarios para el óptimo aprovechamiento del sistema.

Será necesario proporcionar todos los elementos identificados para llevar a cabo una completa revisión del sistema, antes de iniciar la programación del mismo. De esta forma, se garantiza que lo que el sistema va a realizar es lo que realmente necesitan los usuarios.

Cabe mencionar que toda la información que aquí se presenta, es el resultado del análisis detallado, el cual se llevo a cabo con todas las personas que intervienen de manera directa o indirecta en el desarrollo de este proyecto.

4.3 Especificación de Requerimientos

La especificación de requerimientos del sistema comenzó con las entrevistas llevadas a cabo con el personal del Departamento de Personal Académico, encargado del manejo de la información base y de controlar los trámites y procesos posteriores que siguen las formas únicas de contrato de cada una de las personas académicas que trabaja para la Facultad de Ciencias. Se recabaron formas únicas de contrato para ser analizadas y poder obtener la información necesaria para dar de alta un contrato, intentando que esta información se asemejará lo más posible a la forma única.

Con base en estas entrevistas y a la documentación obtenida, el sistema a realizar tiene como objetivo principal: *la creación de las formas únicas de contrato por medio de la computadora*. Para esto es importante tomar en cuenta los siguientes puntos:

1. Identificar las necesidades de los usuarios de este departamento.
2. Evaluar la viabilidad del sistema.

3. Realizar un análisis técnico y económico.
4. Asignar funciones al software, al hardware, a los usuarios, a la base de datos y a otros elementos del sistema.
5. Establecer restricciones de costo y tiempo.
6. Crear una definición del sistema que sea la base para todo el trabajo posterior de desarrollo.

Estos puntos son importantes, ya que por medio de ellos se podrá alcanzar con éxito el objetivo del sistema. Sin embargo, no es lo único que hay que tomar en cuenta. Se requiere experiencia, tanto en software como en hardware, así como en ingeniería humana y en bases de datos. Tomando en cuenta esto, podemos establecer una serie de lineamientos que nos permitan conocer como debemos diseñar y codificar este sistema de información:

- El sistema deberá ejecutarse de manera óptima en los equipos con los que cuenta actualmente el Departamento de Personal Académico, sin necesidad de adquirir elementos de Software o Hardware adicionales a los actuales.
- El sistema deberá ejecutarse en versión multiusuarios bajo una red de área amplia o local y bajo el hardware de red con el que cuenta actualmente este departamemto.
- El sistema deberá ser de acceso sencillo y con una interfase de usuario amigable. Esta interfaz debe ser amigable para todos los niveles de usuarios.
- La capacitación para el manejo del sistema deberá ser fácil y rápida, sin necesidad de que el personal operativo del mismo cuente con conocimientos técnicos o especiales para su manejo.
- El sistema no deberá de representar ninguna inversión de tipo económica adicional a la realizada a la fecha para el manejo de la información.
- El sistema deberá garantizar su actualización y crecimiento futuro, de acuerdo al desarrollo de las necesidades del área.
- El sistema deberá ser desarrollado a la medida de las necesidades del área a automatizar.

Lista de Características

De acuerdo a las entrevistas realizadas con el personal del Departamento de Personal Académico se pudo establecer lo siguiente:

1. El sistema permitirá almacenar la información de cada trabajador académico de forma estructurada y reciente.

La información (**Datos Personales**) que se almacenará en el sistema de cada trabajador es la siguiente:

- *Expediente Laboral*: es un número asignado por la Dirección General de Personal Académico (DGPA) de la UNAM, el cual, identifica de manera única a cada uno de los empleados.
- *Apellido Paterno*: es el nombre de familia por parte del padre del empleado.
- *Apellido Materno*: es el nombre de familia por parte de la madre del empleado.
- *Nombre(s)*: es el patronímico o patronímicos del empleado.
- *Domicilio*: es el lugar donde reside el empleado.
- *Colonia*: es parte del domicilio del empleado e identifica de forma más general al domicilio.
- *Población*: es parte del domicilio del empleado e identifica de forma más general a la colonia, también es conocido como delegación o municipio respectivamente.
- *Código Postal*: es la clave numérica que identifica el destino de la correspondencia y envíos del empleado.
- *Nacionalidad*: es el país de nacimiento natural o adoptado del empleado.
- *Sexo*: es la condición orgánica del empleado.
- *Estado Civil*: es la condición marital del empleado.
- *Teléfono*: es un número donde se puede localizar al empleado en caso de alguna necesidad.
- *Correo Electrónico*: es una dirección electrónica a donde se le puede enviar información al empleado en caso de ser necesario. Este atributo no es parte de la forma única como tal, pero se anexo a los datos personales para poder mantener informado al empleado de todas aquellas cuestiones laborales que le aquejan.

2. El sistema permitirá la creación y almacenamiento de formas únicas de contrato de forma estructurada y eficiente.

La información (**Datos Contractuales**) que se utilizarán para la creación y almacenamiento de las formas únicas es la siguiente:

- *Personal*: es el tipo de trabajo que lleva a cabo el empleado.

- *Número de Oficio*: es un número asignado por la Dirección General de Personal Académico de la UNAM, el cual, identifica de forma única a cada uno de los contratos laborales de los empleados.
- *Lugar*: es el lugar, facultad y división de estudios donde se expide el contrato del empleado.
- *Solicitud*: es el tipo de contrato del empleado.
- *Causa*: es la explicación detallada del tipo de contrato del empleado.
- *Clave y Categoría*: es la clave asociada al nombramiento con el que cuenta el empleado.
- *Asignatura*: es el nombre de la materia que imparte el empleado.
- *Sueldo o Remuneración*: es el salario que recibe el empleado mensualmente por los servicios que presta.
- *Afecta Nómina*: es el atributo que informa al Departamento de Contabilidad y Presupuesto y la Dirección General de Personal Académico si el contrato afecta la nómina del lugar donde labora el empleado.
- *Horas Laboradas*: es el número de horas teóricas y prácticas que imparte el empleado.
- *Fecha del Movimiento*: es la fecha de inicio y fecha de fin del contrato del empleado.
- *Código Programático*: es la clave numérica de la partida económica asignada por el Departamento de Contabilidad y Presupuesto al contrato del empleado.
- *Número de Registro de Plaza*: es un número que describe de forma más general el trabajo que realiza el empleado.
- *Horario*: es el horario semanal detallado en el cual el empleado otorga sus servicios a la facultad.
- *Observaciones*: son comentarios que se pueden agregar para detallar más algunos aspectos relevantes del contrato del empleado.
- *Fecha de Alta de Contrato*: es el día en el cual se elabora el contrato del empleado.

Además de esta información la forma única cuenta con otra información que es más constante, pero que del mismo modo es indispensable para dar de alta un contrato. Esta información es la siguiente:

- *Nombre del Director*: es el nombre completo y firma de conformidad del contrato del director de la facultad, escuela, centro de trabajo o instituto donde labora el empleado.

- *Director General de Personal*: es el nombre completo y firma de conformidad del contrato de la persona que ocupa ese puesto actualmente en la universidad.
 - *Secretario Administrativo de la UNAM*: es el nombre completo y firma de conformidad del contrato de la persona que ocupa ese puesto actualmente en la universidad.
 - *Interesado*: es el nombre completo y firma de conformidad del empleado.
3. El sistema permitirá la modificación de la información almacenada en el sistema.
- Para realizar modificaciones a la información almacenada en el sistema de un empleado, se requiere que:
- (a) los datos personales del empleado estén dados de alta.
 - (b) el empleado cuente al menos con una forma única de contrato dentro del sistema.
4. El sistema permitirá realizar consultas a la información almacenada en el sistema.

Las consultas se dividen en dos tipos: *sencillas* y *avanzadas*.

Estas se podrán realizar tanto en los datos personales, datos contractuales o ambos.

A partir de esta información es posible estructurar un esquema general por medio del cual se construya un diseño estructural de base de datos que permita organizar esta información de forma eficiente. Anexo a este análisis llevado a cabo en cada una de las formas únicas, el personal del Departamento de Personal Académico mencionó algunas inquietudes en cuanto a las funciones del sistema. A continuación se enlistan cada una de ellas:

- * La posibilidad de dar de alta hasta dos materias por contrato con sus respectivas observaciones.
- * La posibilidad en sus casos respectivos de realizar altas y bajas de contrato en la misma forma única.
- * La posibilidad de modificar y eliminar información dentro del sistema.
- * La posibilidad de generar historiales académicos laborales de cada uno de los empleados, en caso de ser solicitados.
- * La posibilidad de generar consultas simples y complejas para poder generar reportes o estadísticas en caso de ser solicitados.

- * La posibilidad de imprimir la forma única en pantalla y papel con sus respectivas copias necesarias.

Cada una de estas inquietudes se convierten en requerimientos iniciales del sistema, en donde cada requerimiento representa nuevos atributos que serán necesarios anexar al diseño estructural de la base de datos antes de iniciar la programación del sistema.

4.3.1 Requerimientos de Software

En base a las entrevistas, análisis de documentos e inquietudes mencionadas por el personal del DPA de la Facultad de Ciencias, podemos establecer los lineamientos sobre los cuales vamos basar el desarrollo de este sistema de información.

1. El sistema será desarrollo en lenguaje HTML¹ y PHP². Algunos de las razones para usar estos lenguajes son las siguientes:
 - La integración de PHP con páginas HTML es muy fácil y directa. Esto significa que hay una clara separación de código y datos. Además PHP es un lenguaje de *scripting* diseñado especialmente para internet.
 - La facilidad de aprendizaje o legibilidad de código en PHP y HTML es rápida y sencilla.
 - Los módulos de acceso a base de datos de PHP funcionan muy bien y en general no tienen problemas con las interfaces a bases de datos.
2. Para el manejo de bases de datos se utilizará PostgreSQL, el cual es un "sofisticado" DMBS³ Relacional que soporta todas las construcciones de SQL, incluyendo subselecciones, transacciones y definición de tipos y funciones. Además su distribución libre permite minimizar los costos del sistema de información.
3. El sistema será desarrollado bajo plataforma ©RedHat Linux como servidor, mientras que los usuarios del sistema trabajarán bajo ambiente ©Windows como clientes.

¹HTML -HyperText Markup Language (Lenguaje de Formato de Documentos para Hipertexto)-

²PHP -Professional Hypertext Preprocessor (Pre-procesador de Hipertexto Profesional)-

³DBMS -DataBase Management System (Sistema Administrador de Bases de Datos)-

4.3.2 Requerimientos de Interfaz

La interfaz de usuario estará conformada por un navegador de internet, dentro del cual se ejecuta el sistema. Este sistema de información está basado en páginas HTML semidinámicas, ya que estas cambian dependiendo del tipo de modificación o consulta que le solicite el usuario al sistema. La interfaz de usuario deberá contar con las siguientes características:

1. El sistema deberá ser lo más sencillo y amigable posible, lo que le permita al usuario aprender el uso del sistema sin necesidad de capacitaciones sofisticadas.
2. El sistema deberá estar organizado de tal manera que sean minimizados los errores del usuario.
3. Los menús que presente el sistema deberán estar organizados de la mejor manera posible, lo que le debe permitir al usuario llevar a cabo todas sus tareas de forma rápida y sencilla.

Para la construcción de las páginas HTML del sistema se usarán aplicaciones tales como The GIMP⁴ y EMACS⁵.

4.3.3 Requerimientos de Seguridad

La seguridad del sistema estará basada en grupos o niveles de acceso, donde cada grupo o nivel de acceso tendrá asignado un conjunto determinado de funciones. Además de esto el sistema debe tener las siguientes características de seguridad:

1. El sistema deberá contar con mecanismos estándar de *login* y *password* por usuario, el cual determinará el nivel de acceso al sistema.
2. El sistema deberá llevar un registro detallado de cada uno de los movimientos hechos por los usuarios del sistema.
3. Toda transacción dentro del sistema deberá ser segura, para esto será necesario hacer uso de metodologías de encriptación que permitan enviar y recibir información entre el cliente y el servidor sin problemas de seguridad.
4. Las bases de datos utilizadas por el sistema deberán ser protegidas por mecanismos estándares de *login* y *password* para mantenerlas protegidas de usuarios ajenos al sistema.

⁴GIMP -The GNU Image Manipulation Program (Programa Manipulador de Imágenes de GNU)-

⁵EMACS.- Editor de textos en tiempo real y más de GNU

4.3.4 Restricciones del Sistema

A pesar de que el sistema cuenta con niveles de seguridad de acceso, bitácoras de registro y métodos de encriptación, este cuenta con las siguientes restricciones:

1. El sistema no permitirá la eliminación de datos personales del sistema.
2. El sistema no generará ni imprimirá las credenciales académicas para empleados.
3. El sistema no generará las fotos de los empleados requeridas como parte visual de los datos personales.
4. El sistema no controlará ni calculará la nómina de los empleados académicos.
5. El sistema será realizado para usuarios de habla hispana.

4.3.5 Requerimientos Especiales del Sistema

Estos requerimientos son parte del desarrollo del sistema, pero se les considera especiales ya que por medio de estos, los usuarios tendrán a la mano la información necesaria para resolver dudas sobre el manejo del sistema. A continuación se enlistan estos requerimientos:

1. Manual del Usuario Impreso.
2. Módulo de Ayuda dentro del sistema.

4.3.6 Referencias y Fuentes de Información

La información expuesta, las formas únicas obtenidas y las entrevistas realizadas con los usuarios del Departamento de Personal Académico son las fuentes de información y referencia en que están basados estos requerimientos. Esta información pretende cumplir inicialmente con los requerimientos necesarios para este departamento pueda alcanzar a la brevedad posible y con sus recursos, el proceso de automatización de sus funciones bajo ambiente de cómputo.

Posteriormente se realizará conjuntamente con los usuarios del sistema el análisis de las funciones que ellos desean incorporar al sistema.

4.3.7 Diagramas de Casos de Uso

A continuación se presentan cada uno de los caso de uso del sistema SAFU. Estos diagramas nos ayudarán a entender de forma más clara las funciones del sistema.

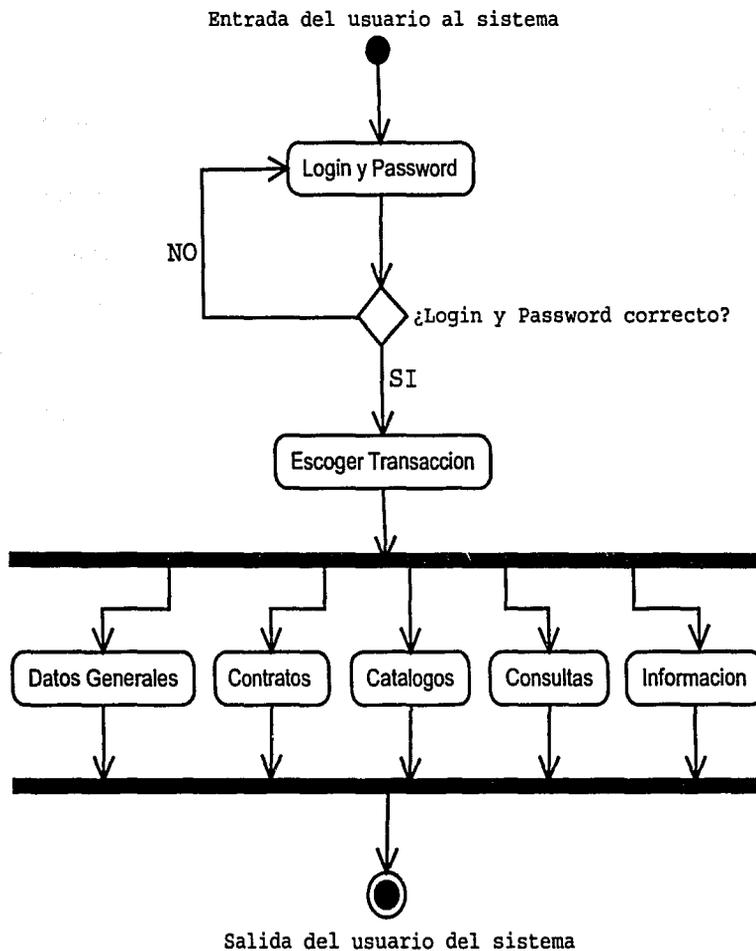
PROCESO DE USO DE SAFU

Figura. 4.1: Funcionamiento General de SAFU

DIAGRAMA DE CASOS DE USO 1er. NIVEL

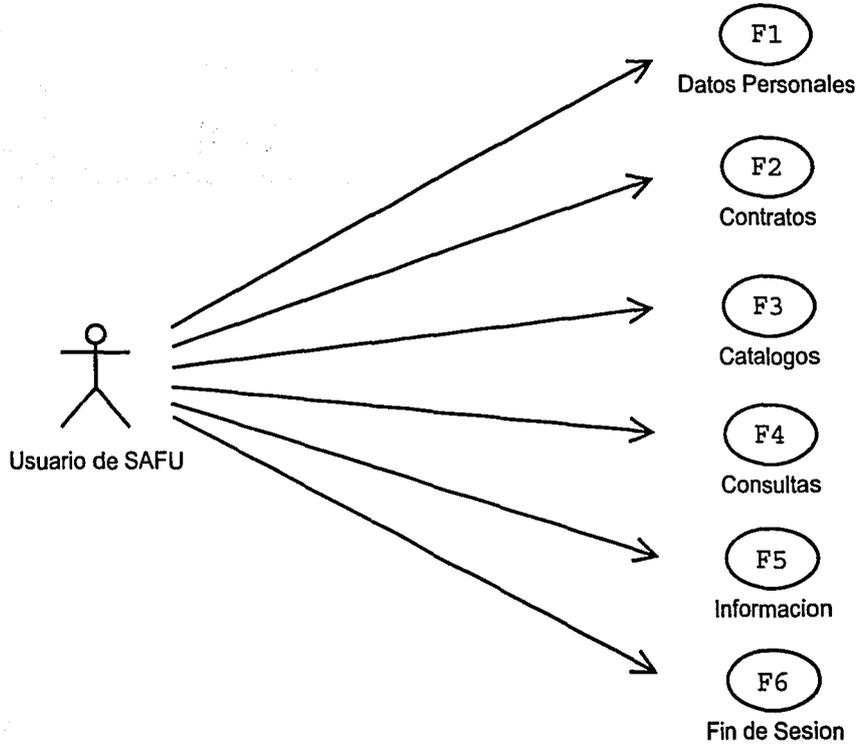


Figura. 4.2: Diagrama de Casos de SAFU 1er. Nivel

DIAGRAMA DE CASOS DE USO
2do. NIVEL Opción "F1"

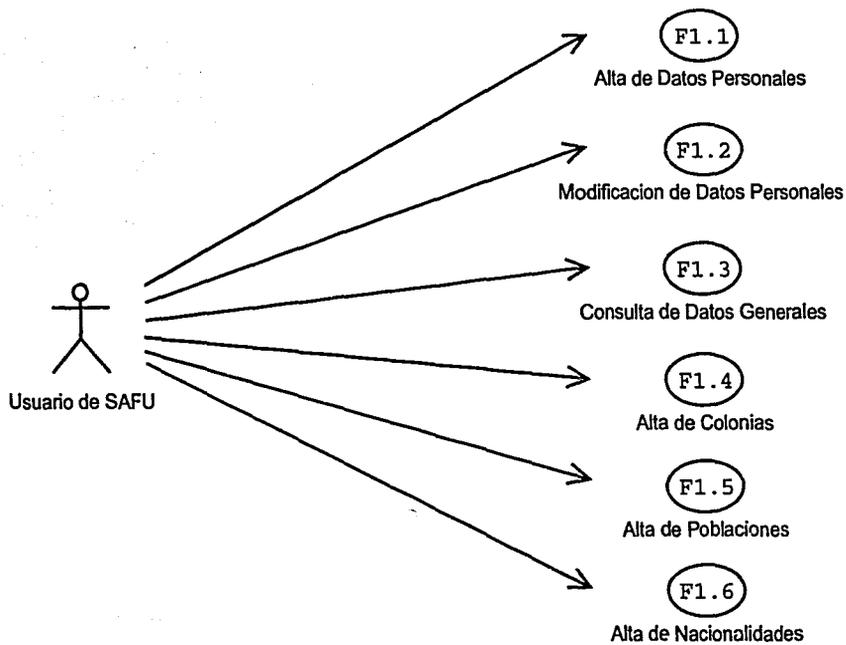


Figura. 4.3: Diagrama de Casos de SAFU 2do. Nivel Opción "F1"

DIAGRAMA DE CASOS DE USO

2do. NIVEL Opción "F2"

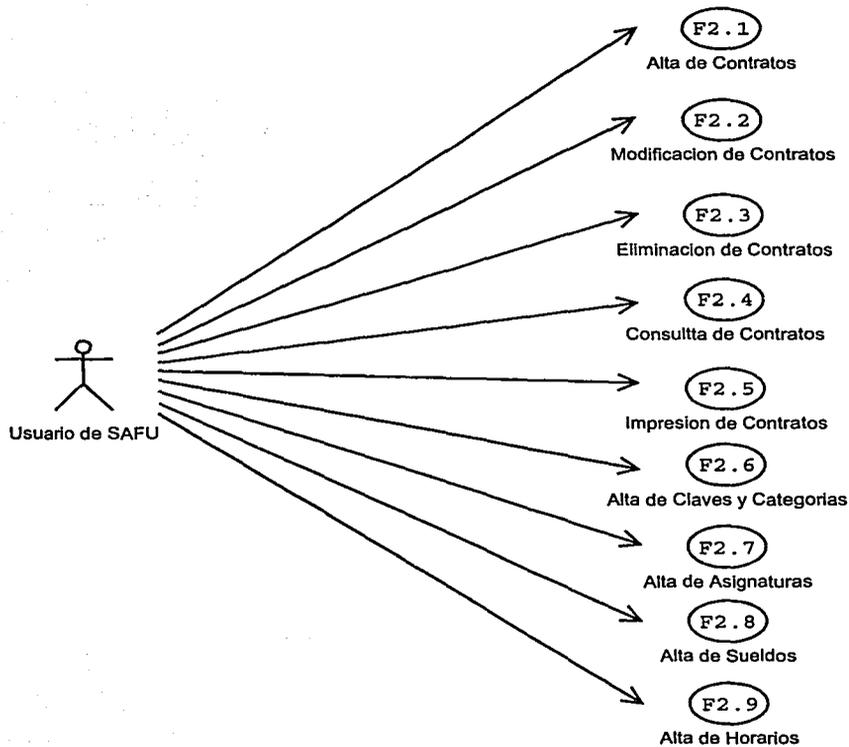


Figura. 4.4: Diagrama de Casos de SAFU 2do. Nivel Opción "F2"

DIAGRAMA DE CASOS DE USO

2do. NIVEL Opción "F3"

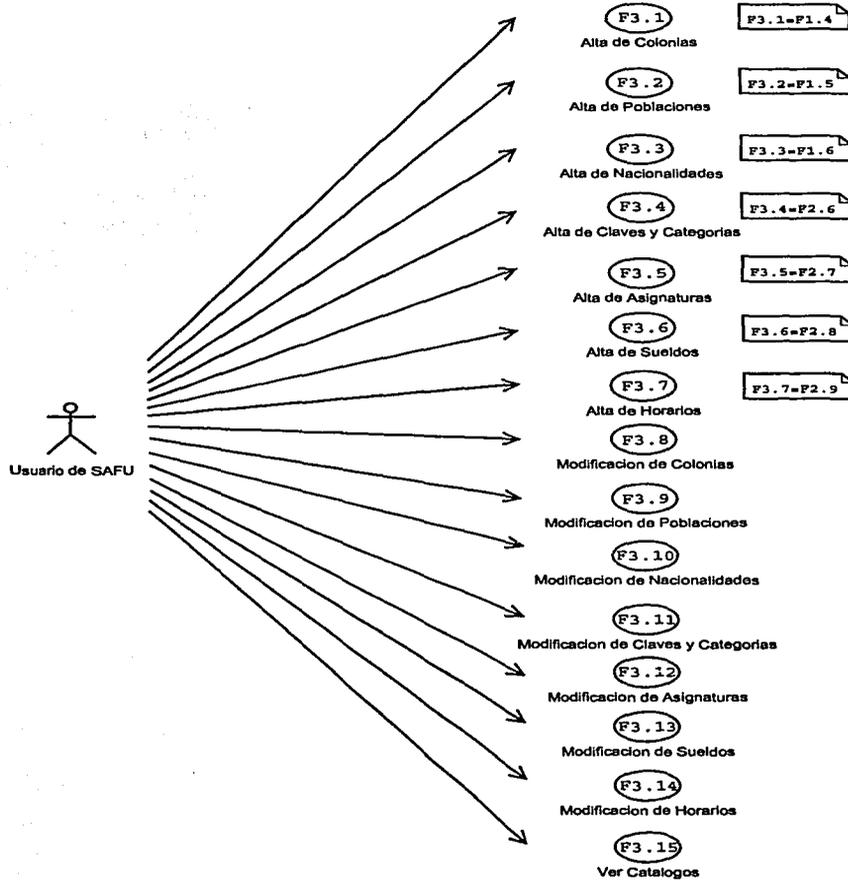


Figura. 4.5: Diagrama de Casos de SAFU 2do. Nivel Opción "F3"

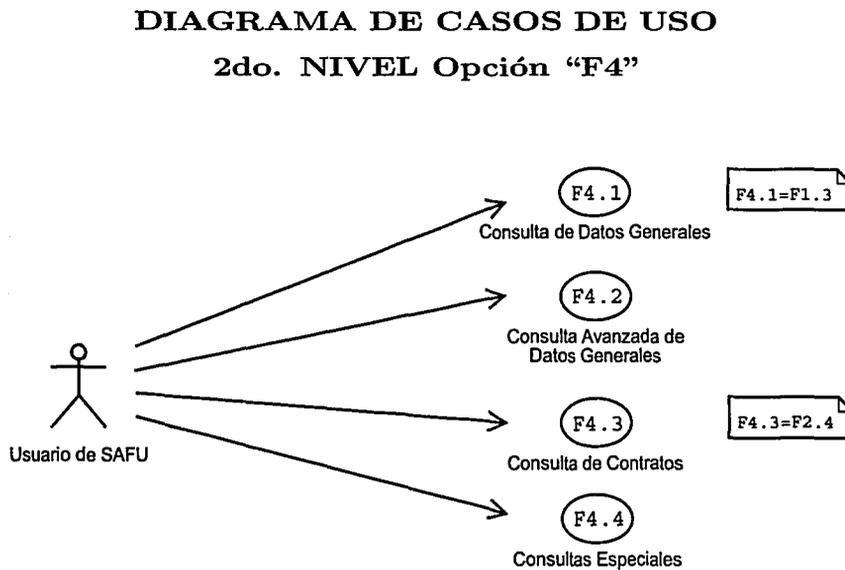


Figura. 4.6: Diagrama de Casos de SAFU 2do. Nivel Opción "F4"

4.3.8 Detalle de Casos de Uso

A continuación se presenta una descripción de los casos de uso más relevantes y significativos del sistema, los demás se encuentran en el Apéndice A.

CASO DE USO: Alta de Datos Generales

IDENTIFICADOR: F1.1

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso de uso se agregan los datos generales de un nuevo empleado al sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Ingresar datos del nuevo empleado	2	Procesa datos del nuevo empleado
		3	Se agrega a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	Paso
E1	Expediente existente	Mensaje de Expediente incorrecto	2
E2	Expediente inválido	Mensaje de Expediente incorrecto	2
E3	Ap.Paterno inválido	Mensaje de Ap.Paterno incorrecto	2
E4	Ap.Materno inválido	Mensaje de Ap.Materno incorrecto	2
E5	Nombre1 inválido	Mensaje de Nombre1 incorrecto	2
E6	Nombre2 inválido	Mensaje de Nombre2 incorrecto	2
E7	Calle y Número inválido	Mensaje de Dirección incorrecta	2
E8	Código Postal inválido	Mensaje de Código Postal incorrecto	2
E9	RFC inválido	Mensaje de RFC incorrecto	2
E10	Teléfono inválido	Mensaje de Teléfono incorrecto	2
E11	Correo Electrónico inválido	Mensaje de Correo Electrónico incorrecto	2

CASO DE USO: Modificación de Datos Generales

IDENTIFICADOR: F1.2

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso de uso se modifican los datos generales de un empleado en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige los datos generales a modificar y empleado	2	Construye forma de datos generales a modificar
3	Ingresa los nuevos datos generales	4	Procesa los datos generales modificados
		5	Modifica los datos generales en la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E1	Expediente existente	Mensaje de Expediente incorrecto	4
E2	Expediente inválido	Mensaje de Expediente incorrecto	4
E3	Ap.Paterno inválido	Mensaje de Ap.Paterno incorrecto	4
E4	Ap.Materno inválido	Mensaje de Ap.Materno incorrecto	4
E5	Nombre1 inválido	Mensaje de Nombre1 incorrecto	4
E6	Nombre2 inválido	Mensaje de Nombre2 incorrecto	4
E7	Calle y Número inválido	Mensaje de Dirección incorrecta	4
E8	Código Postal inválido	Mensaje de Código Postal incorrecto	4
E9	RFC inválido	Mensaje de RFC incorrecto	4
E10	Teléfono inválido	Mensaje de Teléfono incorrecto	4
E11	Correo Electrónico inválido	Mensaje de Correo Electrónico incorrecto	4
E12	Expediente inexistente	Mensaje de Expediente incorrecto	1
E13	RFC inexistente	Mensaje de RFC incorrecto	1

CASO DE USO: Modificación de Datos Generales

IDENTIFICADOR: F1.2

ACTORES: Profesor
Ayudante

DESCRIPCIÓN: En este caso de uso se modifican los datos generales de un empleado en el sistema.

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige los datos generales a modificar	2	Construye forma de datos generales a modificar
3	Ingresa los nuevos datos generales	4	Procesa los datos generales modificados
		5	Modifica los datos generales en la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E1	Expediente existente	Mensaje de Expediente incorrecto	4
E2	Expediente inválido	Mensaje de Expediente incorrecto	4
E3	Ap.Paterno inválido	Mensaje de Ap.Paterno incorrecto	4
E4	Ap.Materno inválido	Mensaje de Ap.Materno incorrecto	4
E5	Nombre1 inválido	Mensaje de Nombre1 incorrecto	4
E6	Nombre2 inválido	Mensaje de Nombre2 incorrecto	4
E7	Calle y Número inválido	Mensaje de Dirección incorrecta	4
E8	Código Postal inválido	Mensaje de Código Postal incorrecto	4
E9	RFC inválido	Mensaje de RFC incorrecto	4
E10	Teléfono inválido	Mensaje de Teléfono incorrecto	4
E11	Correo Electrónico inválido	Mensaje de Correo Electrónico incorrecto	4
E12	Expediente inexistente	Mensaje de Expediente incorrecto	1
E13	RFC inexistente	Mensaje de RFC incorrecto	1

CASO DE USO: Consulta de Datos Generales

IDENTIFICADOR: F1.3 = F4.1

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso de uso se consultan los datos generales de un empleado en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe el dato y elige el patrón a buscar	2	Busca el dato de acuerdo al patrón
		3	Presenta resultados de búsqueda en pantalla

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E14	Datos no encontrados	Mensaje de búsqueda de datos no encontrados	2

CASO DE USO: Consulta de Datos Generales

IDENTIFICADOR: F1.3 = F4.1

ACTORES: Profesor
Ayudante

DESCRIPCIÓN: En este caso de uso se consultan los datos generales de un empleado en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
		1	Busca datos del Usuario
		2	Presenta resultados de búsqueda en pantalla

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E14	Datos no encontrados	Mensaje de búsqueda de datos no encontrados	1

CASO DE USO: Alta de Colonias

IDENTIFICADOR: F1.4 = F3.1

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso se agrega una colonia nueva en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe la colonia nueva	2	Procesa la colonia nueva
		3	Se agrega la colonia nueva a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E15	Colonia existente	Mensaje de Colonia incorrecta	2
E16	Colonia inválida	Mensaje de Colonia incorrecta	2

CASO DE USO: Alta de Poblaciones

IDENTIFICADOR: F1.5 = F3.2

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso se agrega una población nueva en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe la población nueva	2	Procesa la población nueva
		3	Se agrega la población nueva a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E17	Población existente	Mensaje de Población incorrecta	2
E18	Población inválida	Mensaje de Población incorrecta	2

CASO DE USO: Alta de Nacionalidades

IDENTIFICADOR: F1.6 = F3.3

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso se agrega una nacionalidad nueva en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe la nacionalidad nueva	2	Procesa la nacionalidad nueva
		3	Se agrega la nacionalidad nueva a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E19	Abreviatura existente	Mensaje de Abreviatura incorrecta	2
E20	Abreviatura inválida	Mensaje de Abreviatura incorrecta	2
E21	País existente	Mensaje de País incorrecto	2
E22	País inválido	Mensaje de País incorrecto	2
E23	Nacionalidad existente	Mensaje de Nacionalidad incorrecta	2
E24	Nacionalidad inválida	Mensaje de Nacionalidad incorrecta	2

CASO DE USO: Alta de Contratos

IDENTIFICADOR: F2.1

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso de uso se agregan los datos contractuales de un empleado al sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Ingresar datos del nuevo contrato	2	Procesa datos del nuevo contrato
		3	Se agrega a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E25	Oficio existente	Mensaje de Oficio incorrecto	2
E26	Oficio inválido	Mensaje de Oficio incorrecto	2
E27	Solicitud inválida	Mensaje de Solicitud incorrecta	2
E28	Causa inválida	Mensaje de Causa incorrecta	2
E29	Num.Plaza inválida	Mensaje de Num.Plaza incorrecta	2
E30	Hor.Teóricas inválidas	Mensaje de Hor.Teóricas incorrectas	2
E31	Hor.Prácticas inválidas	Mensaje de Hor.Prácticas incorrectas	2
E32	Fecha Inicio de Contrato inválida	Mensaje de Fecha Inicio de Contrato incorrecta	2
E33	Fecha Fin de Contrato inválida	Mensaje de Fecha Fin de Contrato incorrecta	2
E34	Código Programático inválido	Mensaje de Código Programático incorrecto	2
E35	Horario inválido	Mensaje de Horario inválido	2

CASO DE USO: Modificación de Contratos

IDENTIFICADOR: F2.2

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso de uso se modifican los datos contractuales de un empleado en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige los datos contractuales a modificar	2	Construye forma de datos contractuales a modificar
3	Ingresa los nuevos datos contractuales	4	Procesa los datos contractuales modificados
		5	Modifica los datos contractuales en la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E25	Oficio existente	Mensaje de Oficio incorrecto	4
E26	Oficio inválido	Mensaje de Oficio incorrecto	4
E27	Solicitud inválida	Mensaje de Solicitud incorrecta	4
E28	Causa inválida	Mensaje de Causa incorrecta	4
E29	Num.Plaza inválida	Mensaje de Num.Plaza incorrecta	4
E30	Hor.Teóricas inválidas	Mensaje de Hor.Teóricas incorrectas	4
E31	Hor.Prácticas inválidas	Mensaje de Hor.Prácticas incorrectas	4
E32	Fecha Inicio de Contrato inválida	Mensaje de Fecha Inicio de Contrato incorrecta	4
E33	Fecha Fin de Contrato inválida	Mensaje de Fecha Fin de Contrato incorrecta	4
E34	Código Programático inválido	Mensaje de Código Programático incorrecto	4
E35	Horario inválido	Mensaje de Horario inválido	4
E12	Expediente inexistente	Mensaje de Expediente incorrecto	1
E13	RFC inexistente	Mensaje de RFC incorrecto	1

CASO DE USO: Eliminación de Contratos

IDENTIFICADOR: F2.3

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso se elimina un contrato del sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe el dato y elige el patrón a buscar	2	Busca contratos de acuerdo al dato y al patrón
		3	Presenta resultados de búsqueda en pantalla
4	Escoge contrato a eliminar	5	Elimina el contrato de la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E36	Contratos no encontrados	Mensaje de búsqueda de contratos no encontrados	2

CASO DE USO: Consulta de Contratos

IDENTIFICADOR: F2.4 = F4.3

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso de uso se consultan los contratos de un empleado del sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe el dato y elige el patrón a buscar	2	Busca contratos de acuerdo al dato y al patrón
		3	Presenta resultados de búsqueda en pantalla
4	Escoge contrato a consultar	5	Presenta contrato en pantalla

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E36	Contratos no encontrados	Mensaje de búsqueda de contratos no encontrados	2

CASO DE USO: Consulta de Contratos

IDENTIFICADOR: F2.4 = F4.3

ACTORES: Profesor
Ayudante

DESCRIPCIÓN: En este caso de uso se consultan los contratos de un empleado del sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
		1	Busca contratos del empleado
		2	Presenta resultados de búsqueda en pantalla

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E36	Contratos no encontrados	Mensaje de búsqueda de contratos no encontrados	1

CASO DE USO: Impresión de Contratos

IDENTIFICADOR: F2.5

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso de uso se imprimen los contratos de un empleado del sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe el dato y elige el patrón a buscar	2	Busca contratos de acuerdo al dato y al patrón
		3	Presenta resultados de búsqueda en pantalla
4	Escoge contrato a consultar	5	Presenta contrato en pantalla
6	Imprime contrato		

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E36	Contratos no encontrados	Mensaje de búsqueda de contratos no encontrados	2

**ESTA TESIS NO SALE
DE LA BIBLIOTECA**

CASO DE USO: Alta de Claves y Categorías

IDENTIFICADOR: F2.6 = F3.4

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso se agrega una clave y categoría nueva en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe los datos de la clave y categoría nueva	2	Procesa los datos de la clave y categoría nueva
		3	Se agrega la clave y categoría nueva a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E37	Clave y Categoría existente	Mensaje de Clave y Categoría incorrecta	2

CASO DE USO: Alta de Asignaturas

IDENTIFICADOR: F2.7 = F3.5

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso se agrega una asignatura nueva en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe los datos de la nueva	2	Procesa los datos de la asignatura nueva
		3	Se agrega la asignatura nueva a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E38	Clave existente	Mensaje de Clave incorrecta	2
E39	Clave incorrecta	Mensaje de Clave incorrecta	2
E40	Nombre incorrecto	Mensaje de Nombre incorrecto	2
E41	Créditos incorrectos	Mensaje de Créditos incorrectos	2
E30	Hor. Teóricas inválidas	Mensaje de Hor. Teóricas incorrectas	2
E31	Hor. Prácticas inválidas	Mensaje de Hor. Prácticas incorrectas	2

CASO DE USO: Alta de Sueldos

IDENTIFICADOR: F2.8 = F3.6

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso se agrega un sueldo nuevo en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe los datos del sueldo nuevo	2	Procesa los datos del sueldo nuevo
		3	Se agrega el sueldo nuevo a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E42	Sueldo existente	Mensaje de Sueldo incorrecto	2
E43	Sueldo incorrecto	Mensaje de Sueldo incorrecto	2
E44	Vigencia incorrecta	Mensaje de Vigencia incorrecta	2

CASO DE USO: Alta de Horarios

IDENTIFICADOR: F2.9 = F3.7

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso se agrega un horario nuevo en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe el horario nuevo	2	Procesa el horario nuevo
		3	Se agrega el horario nuevo a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E45	Horario existente	Mensaje de Horario incorrecto	2
E46	Horario incorrecto	Mensaje de Horario incorrecto	2

CASO DE USO: Modificación de Colonias

IDENTIFICADOR: F3.8

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso se modifica una colonia en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige la colonia a modificar	2	Construye la forma de la colonia a modificar
3	Modifica la colonia	4	Procesa la colonia modificada
		5	Modifica la colonia en la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E15	Colonia existente	Mensaje de Colonia incorrecta	2
E16	Colonia incorrecta	Mensaje de Colonia incorrecta	2

CASO DE USO: Modificación de Poblaciones

IDENTIFICADOR: F3.9

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso se modifica una población en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige la población a modificar	2	Construye la forma de la población a modificar
3	Modifica la población	4	Procesa la población modificada
		5	Modifica la población en la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E17	Población existente	Mensaje de Población incorrecta	2
E18	Población incorrecta	Mensaje de Población incorrecta	2

CASO DE USO: Modificación de Nacionalidades

IDENTIFICADOR: F3.10

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso se modifica una nacionalidad en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige la nacionalidad y los datos a modificar	2	Construye la forma de la nacionalidad a modificar
3	Modifica los datos de la nacionalidad	4	Procesa los datos de la nacionalidad modificada
		5	Modifica los datos de la nacionalidad en la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E19	Abreviatura existente	Mensaje de Abreviatura incorrecta	4
E20	Abreviatura inválida	Mensaje de Abreviatura incorrecta	4
E21	País existente	Mensaje de País incorrecto	4
E22	País inválido	Mensaje de País incorrecto	4
E23	Nacionalidad existente	Mensaje de Nacionalidad incorrecta	4
E24	Nacionalidad inválida	Mensaje de Nacionalidad incorrecta	4

CASO DE USO: Modificación de Claves y Categorías

IDENTIFICADOR: F3.11

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso se modifica una clave y categoría en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige la clave y categoría a modificar	2	Construye la forma de la clave y categoría a modificar
3	Modifica la clave y categoría	4	Procesa la clave y categoría modificada
		5	Modifica la clave y categoría en la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E37	Clave y Categoría existente	Mensaje de Clave y Categoría incorrecta	4

CASO DE USO: Modificación de Asignaturas

IDENTIFICADOR: F3.12

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso se modifica una asignatura en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige los datos de la asignatura a modificar	2	Construye la forma de la asignatura a modificar
3	Modifica los datos de la asignatura	4	Procesa los datos de la asignatura modificada
		5	Modifica la asignatura en la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E38	Clave existente	Mensaje de Clave incorrecta	4
E39	Clave incorrecta	Mensaje de Clave incorrecta	4
E40	Nombre incorrecto	Mensaje de Nombre incorrecto	4
E41	Créditos incorrectos	Mensaje de Créditos incorrectos	4
E30	Hor.Teóricas inválidas	Mensaje de Hor.Teóricas incorrectas	4
E31	Hor.Prácticas inválidas	Mensaje de Hor.Prácticas incorrectas	4

CASO DE USO: Modificación de Sueldos

IDENTIFICADOR: F3.13

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso se modifica un sueldo en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige los datos del sueldo a modificar	2	Construye la forma del sueldo a modificar
3	Modifica los datos del sueldo	4	Procesa los datos del sueldo modificado
		5	Modifica el sueldo en la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E42	Sueldo existente	Mensaje de Sueldo incorrecto	4
E43	Sueldo incorrecto	Mensaje de Sueldo incorrecto	4
E44	Vigencia incorrecta	Mensaje de Vigencia incorrecta	4

CASO DE USO: Modificación de Horarios

IDENTIFICADOR: F3.14

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso se modifica un horario en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige el horario a modificar	2	Construye la forma del horario a modificar
3	Modifica el horario	4	Procesa el horario modificado
		5	Modifica el horario en la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E45	Horario existente	Mensaje de Horario incorrecto	4
E46	Horario incorrecto	Mensaje de Horario incorrecto	4

CASO DE USO: Ver Catálogos

IDENTIFICADOR: F3.15

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso se consultan con detalle
cada catálogo en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige el catálogo	2	Muestra el catálogo elegido

EXCEPCIONES: Ninguna

CASO DE USO: Consulta Avanzada de Datos Generales

IDENTIFICADOR: F4.2

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso de uso se consultan de forma más detallada los datos generales de un empleado en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe el dato y elige los patrones a buscar	2	Busca los posibles datos que se aproximen de acuerdo a los patrones
		3	Presenta resultados de búsqueda en pantalla

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E14	Datos no encontrados	Mensaje de búsqueda de datos no encontrados	2

CASO DE USO: Consultas Especiales

IDENTIFICADOR: F4.4

ACTORES: Administrador del SAFU
Jefe del DPA

DESCRIPCIÓN: En este caso de uso se consultan los datos generales y los contratos de cada empleado de forma específica en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe los dato y elige los patrones a buscar	2	Busca los posibles datos y contratos que se aproximen de acuerdo a los patrones
		3	Presenta resultados de búsqueda en pantalla

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E14	Datos no encontrados	Mensaje de búsqueda de datos no encontrados	2
E36	Contratos no encontrados	Mensaje de búsqueda de contratos no encontrados	2
E47	Datos y Contratos no encontrados	Mensaje de búsqueda de datos y contratos no encontrados	2

4.4 Análisis

Una vez completada la especificación de requerimientos, estableceremos de forma más clara los paquetes, clases y modelo de seguridad del sistema.

4.4.1 Diagramas de Paquetes y Clases

DIAGRAMA DE PAQUETES

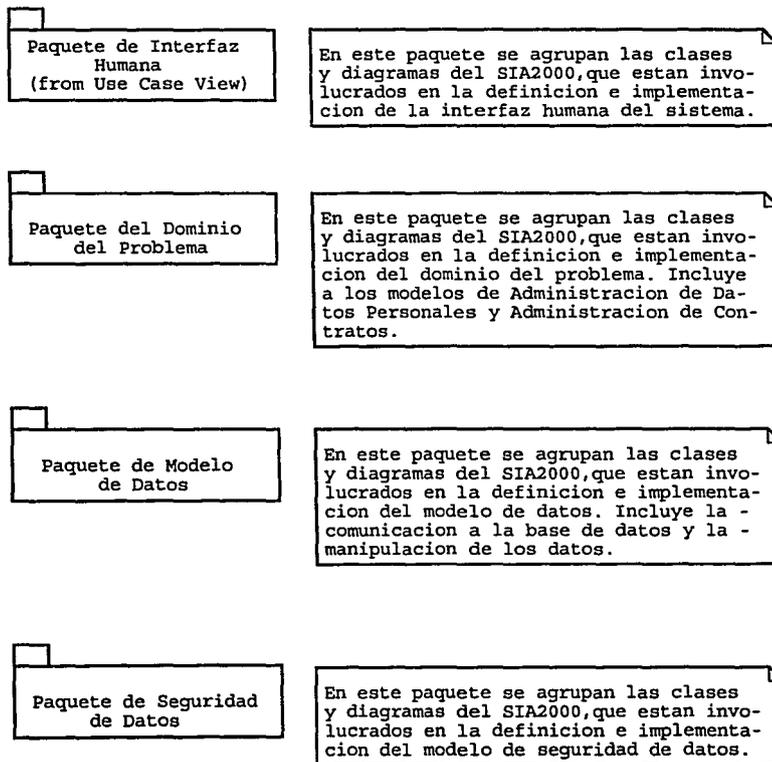


Figura. 4.7: Diagrama de Paquetes de SAFU

DIAGRAMA DE CLASES VISTA GENERAL DEL SISTEMA

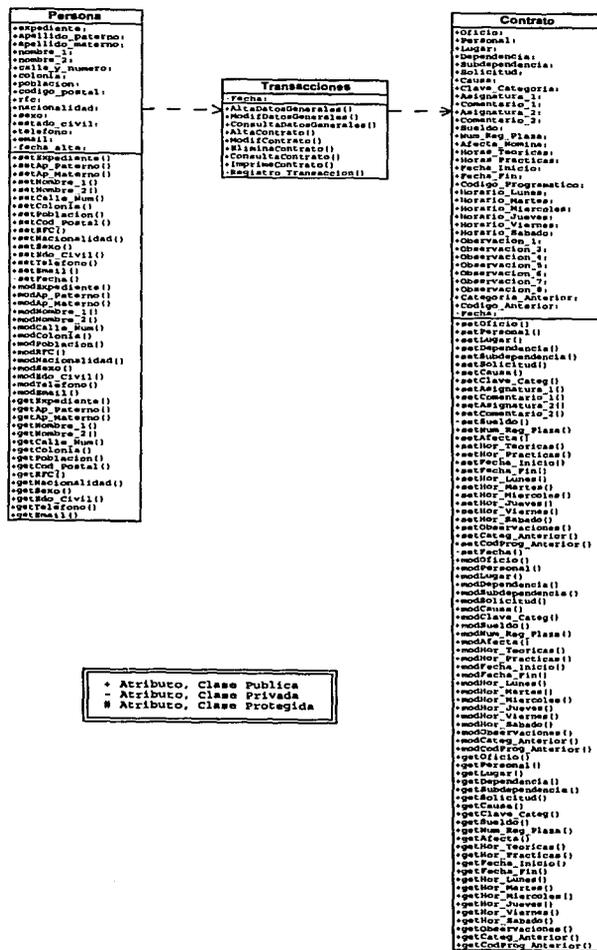


Figura. 4.8: Diagrama de Clases de SAFU (Vista General)

DIAGRAMA DE CLASES DOMINIO DEL PROBLEMA

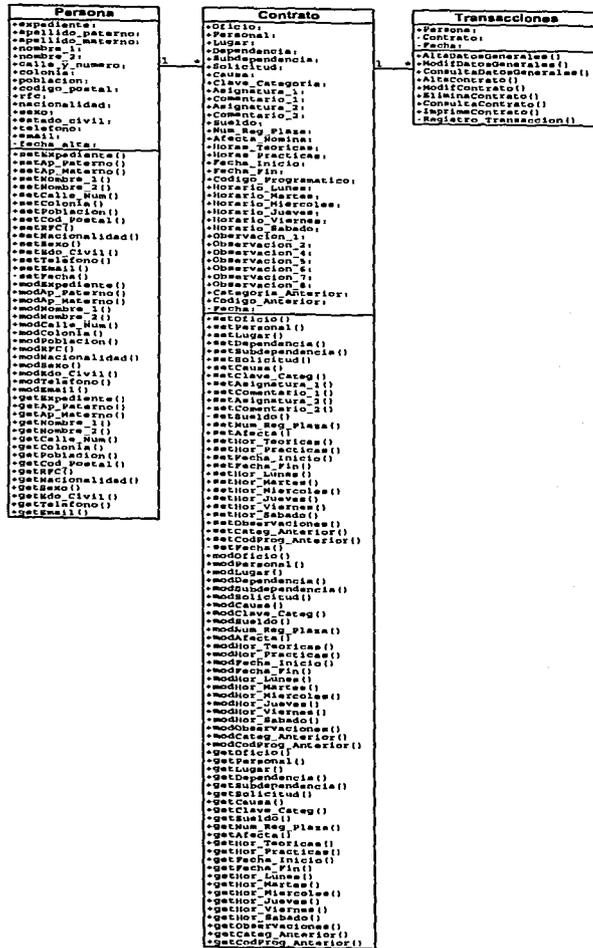


Figura. 4.10: Diagrama de Dominio del Problema de SAFU

DIAGRAMA DE CLASES MODELO DE DATOS

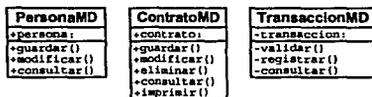


Figura. 4.11: Modelo de Datos de SAFU

4.4.2 Modelo de Seguridad de Datos

MODELO DE SEGURIDAD

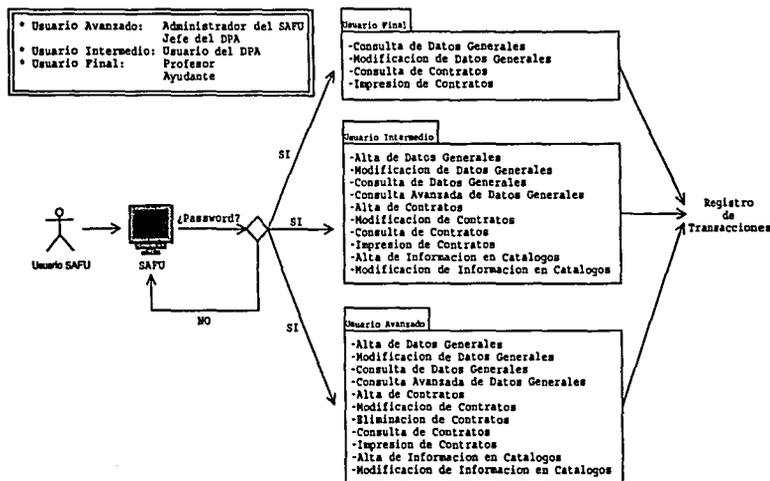


Figura. 4.12: Modelo de Seguridad de SAFU

4.4.3 Diagramas de Paquetes y Capas

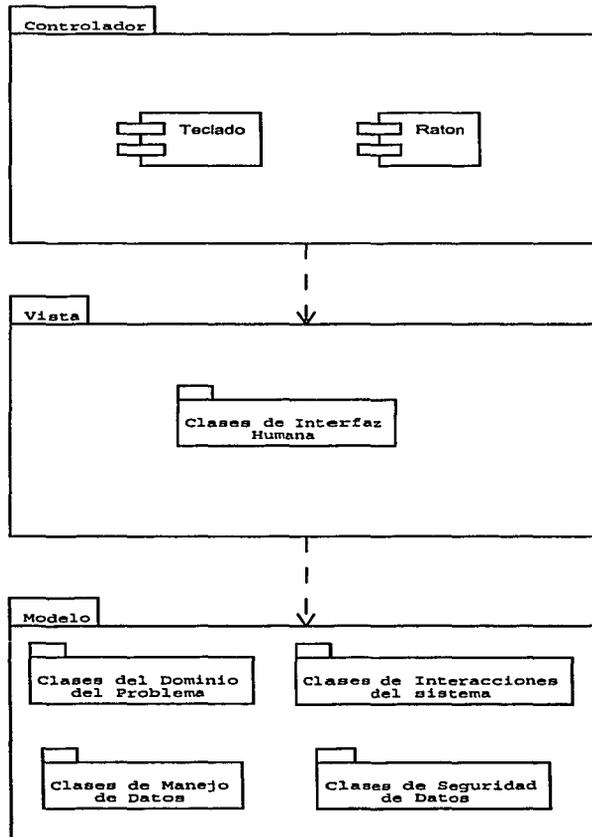
**DIAGRAMA DE PAQUETES Y CAPAS (RELAJADO)
DEFINICIÓN DE ARQUITECTURA: MODELO VISTA
CONTROLADOR Y CAPAS**

Figura. 4.13: Diagrama de Arquitectura de SAFU

Capítulo 5

Diseño del Sistema

El objetivo de esta etapa es producir un "buen" diseño como sea posible. Es por eso que en esta etapa se examinan las conclusiones que han sido generadas a partir de la observación, especificación y análisis del problema, y se usan para identificar formas de soporte que son requeridas por la metodología de desarrollo que se esta usando. Existen algunos otros factores cuya influencia se ve reflejada en el diseño de software pero que no modifica de forma importante el análisis realizado con anterioridad.

5.1 Definición de Políticas y Tácticas Generales

Estas políticas tiene como objetivo primordial establecer los lineamientos base para el desarrollo del SAFU. A continuación se enlistan cada una de ellas:

1. **Definición del Ambiente de Implementación:** El sistema será desarrollado en una plataforma UNIX, utilizando el sistema operativo Linux ©RedHat 7.1 y con los lenguajes de programación PHP¹ V3.0.18, HTML² V4.01, y el sistema manejador de base de datos PostgreSQL V.7.0.3-8 ó posterior. Dado las características de PHP y HTML, el sistema podrá utilizarse en cualquier sistema operativo que cuente con un navegador de internet (©Microsoft Internet Explorer 4.5 o posterior, ©Netscape Communicator 4.76 o posterior).
2. **Manejo de Datos:** Se creará una base de datos utilizando el Modelo Relacional de Datos. Para el manejo de datos, se utilizará el sistema manejador de bases de datos PostgreSQL bajo la plataforma de desarrollo eligada.

¹PHP -Professional Hypertext Preprocessor (Pre-procesador de Hipertexto Profesional)-

²HTML -HyperText Markup Language (Language de Formato de Documentos para Hipertexto)-

3. **Manejo de Memoria:** El manejo de memoria estará controlado por el sistema operativo UNIX (Linux ©RedHat 7.1) junto con las rutinas de control de memoria propias de PostgreSQL, PHP y HTML. Estas rutinas de control de memoria estarán administradas por el servidor de páginas web ©APACHE.
4. **Manejo y Detección de Errores:** Se usarán rutinas implementadas en PHP para el manejo de errores, así mismo se utilizará el manejo de excepciones de PostgreSQL y se crearán las funciones que sean necesarias para que el sistema sea robusto.
5. **Manejo del Control:** El control se llevará a cabo utilizando menus de opciones, así como también niveles de acceso y registro de transacciones en todo momento.
6. **Formato para la Documentación:** Se utilizarán las convenciones de codificación que proponen los desarrolladores de PHP y HTML.
7. **Formato para la Interfaz Humano-Computadora:** Se realizará un sistema que funciona bajo el ambiente de Internet, con vistas o formas estructuradas, manejo de eventos propios de una aplicación de internet, así como pantallas sencillas y de fácil entendimiento. Estas formas o vistas serán programadas por medio de HTML y se harán dinámicas mezclando código de PHP dentro de ellas.

5.2 Definición de Nodos del Sistema

Definiremos conceptualmente la arquitectura física del SAFU por medio de nodos. Un *nodo* es un elemento de hardware sobre el cual puede ejecutarse un elemento de software. Este sistema contará con un solo nodo de las siguientes características:

- **Capacidad:** La capacidad mínima de hardware requerida para instalar y correr el sistema SAFU es: Computadora ©Intel Pentium III o equivalente con reloj interno de 800MHz, 128MB de Memoria RAM, 10GB de disco duro, disco floppy de 3.5, monitor Super VGA, CDROM 52x, tarjeta de video 8MB de memoria, tarjeta de red de 10-100MBs, teclado español de 101 teclas y ratón de tres o dos botones.
- **Tipo de Conexión:** Se requiere de una conexión de red de por lo menos 10MBs, ya que el sistema correrá bajo el modo "Cliente-Servidor". Sin embargo, si el sistema no contará con conexiones de red, este seguiría funcionando en modo "Stand-alone" sin afectar el funcionamiento del sistema.

- **Protocolos:** El sistema requiere del protocolo TCP/IP³ debido a que requiere conexión de red. Del mismo modo se utilizará el protocolo de comunicación del manejador de la base de datos PostgreSQL, SSL⁴ para el envío y recepción de información de forma segura y HTTP⁵ para el funcionamiento propio del sistema.
- **Respaldo de Datos:** Se harán respaldos periódicos de los datos del sistema contenidos en la base de datos. La frecuencia de los respaldos la definirá el DBA del sistema, sin embargo, se recomienda hacer al menos un respaldo diario. También se harán respaldos del sistema completo (incluyendo base datos, archivos ejecutables o interpretados e imágenes). El DBA definirá la frecuencia de estos respaldos pero se recomienda hacerlo por lo menos una o dos veces al mes. Los respaldos se harán en cualquier medio de almacenamiento no volátil. Se aconseja hacerlo en el disco duro, cintas magnéticas o discos compactos grabables, ya que son confiables y de bajo costo.

5.3 Definición de Subsistemas

Para el desarrollo del SAFU se utilizará el patrón de arquitectura del *Modelo-Vista-Controlador* así como el patrón de *Capas Relajado*. Esto se debe a que el sistema es totalmente interactivo y no existirán restricciones entre las capas del sistema.

- **Modelo-Vista-Controlador.** Se utilizará este patrón arquitectónico ya que el sistema es interactivo, lo que hace ideal el uso de este modelo. Bajo estas circunstancias, el sistema tendrá cuatro componentes (capas), cada uno representado en el diagrama de Paquetes y Capas (ver diagrama de Paquetes y Capas anexo). Cada paquete o capa está compuesto a su vez de uno o varios componentes. En particular, el paquete que representa al Modelo, esta formado por tres grandes componentes o capas que son: el modelo persona, el modelo contrato y el modelo del manejo de datos (transacciones).
- **Capas (Relajado).** Se ocupará también este patrón ya que el SAFU se puede descomponer en subareas que tienen un nivel de abstracción particular. Especialmente la capa de Modelo, la cual a su vez se subdivide en la capa de datos generales, la capa de datos contractuales y la capa del dominio del problema.

³TCP/IP -Transfer Control Protocol/Internet Protocol (Protocolo de Control de Transferencias/Protocolo de Internet)-

⁴SSL -Secure Socket Layer (Capa de Conexión Segura)-

⁵HTTP -HyperText Transfer Protocol (Protocolo de Transferencia de HiperTexto)-

5.4 Definición de Clases

De acuerdo con los patrones de arquitectura que se tomarán de guía para el diseño y desarrollo del sistema, se pueden identificar cuatro grandes clases:

- Clase del Dominio del Problema
- Clase de Interacciones del Sistema
- Clase del Manejo de Datos
- Clase de Seguridad de Datos

Estos cuatro tipo de clases presentadas, se identifican y se agrupan en paquetes en los que, cada una de las clases de su tipo, son clases candidatas.

5.5 Mecanismos Genéricos de Diseño

Para el manejo de persistencia en la base de datos, se usarán los mecanismos convencionales implementados dentro del sistema manejador de bases de datos PostgreSQL. Para el manejo y recuperación de errores, se usarán las rutinas de manejo de errores de PostgreSQL, así como funciones implementadas en PHP, lo que mejora la robustez del sistema. En lo referente a la administración de transacciones, se utilizará el sistema manejador de bases de datos ya mencionado anteriormente. Y por último en cuanto a los mecanismos de seguridad, se aplicarán los sistemas de respaldos descritos con anterioridad, además de mecanismos de encriptación en los envíos, recepción y almacenamiento de datos, así como niveles de seguridad y registro de toda transacción realizada por cada uno de los usuarios del sistema.

5.6 Posibilidades de Re-Uso

Dado las características del sistema y los componentes de hardware y software, el sistema será totalmente portable y re-usable. En la medida de lo posible se generarán bibliotecas de funciones generales, que permitan el desarrollo de otros módulos anexos a este sistema.

5.7 Diagramas de Diseño

A continuación se presentan los diagramas de paquetes y capas con más detalle, así como los diagramas de secuencia y diagramas entidad-relación de las bases de datos usadas por el sistema.

5.7.1 Diagrama de Paquetes y Capas

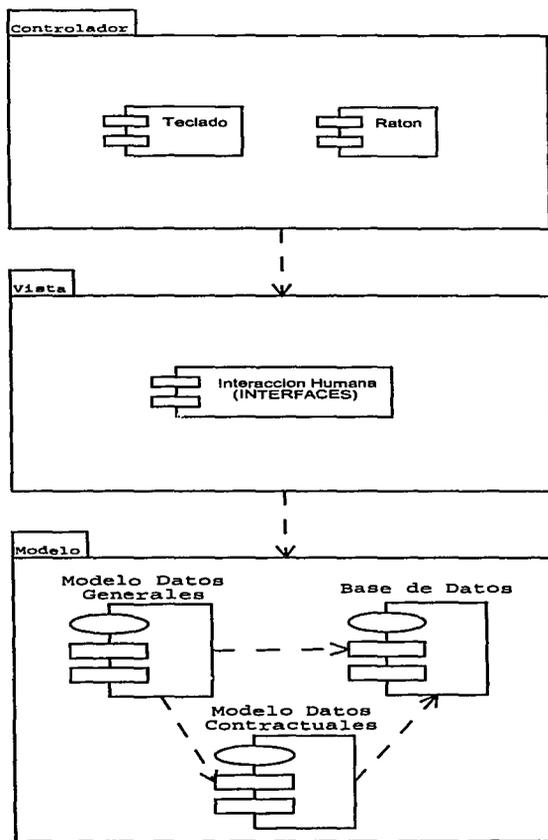
**DIAGRAMA DE PAQUETES Y CAPAS
DEFINICIÓN DE ARQUITECTURA: MODELO VISTA
CONTROLADOR Y CAPAS RELAJADO**

Figura. 5.1: Definición de Arquitectura del SAFU

**DIAGRAMA DE PAQUETES Y CAPAS DETALLADO
ARQUITECTURA: MODELO VISTA-CONTROLADOR Y
CAPAS RELAJADO**

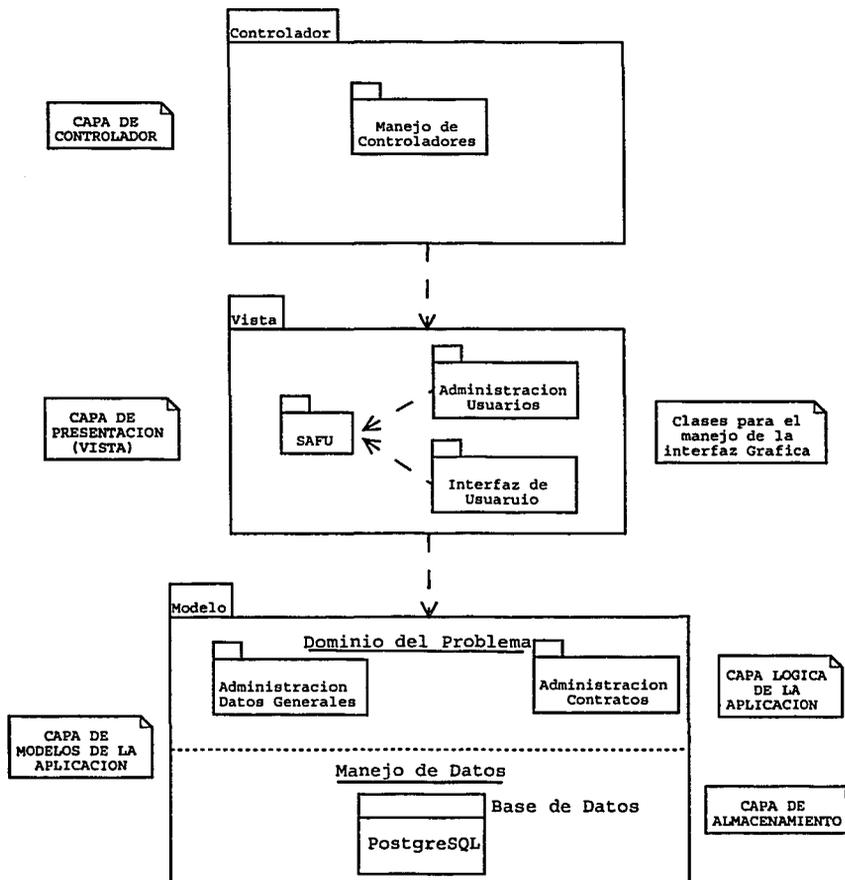


Figura. 5.2: Diagrama de Paquetes y Capas Detallado del SAFU

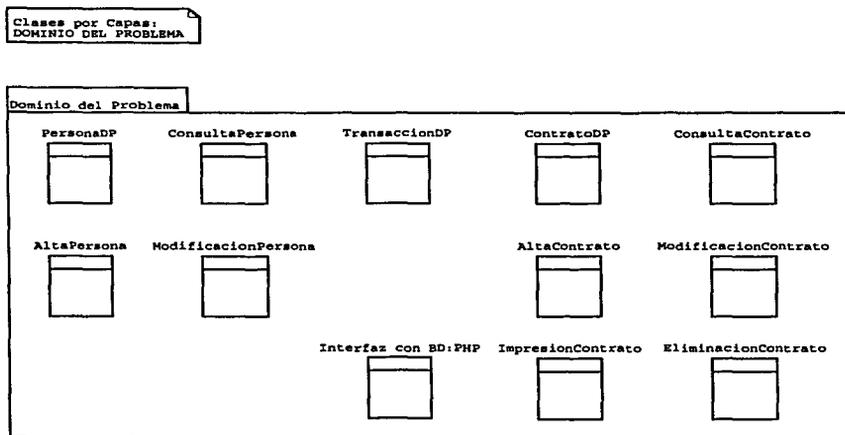


Figura. 5.4: Diagrama Detallado de la Capa Dominio del Problema

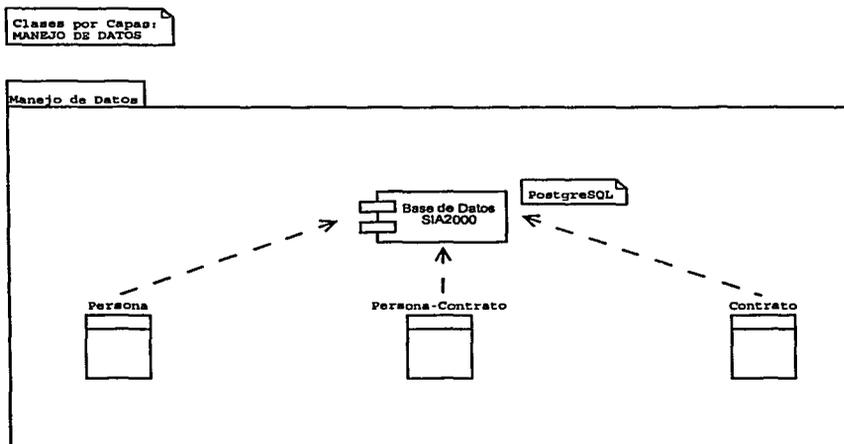


Figura. 5.5: Diagrama Detallado de la Capa Manejo de Datos

5.7.2 Diagramas de Instalación

DIAGRAMA DE INSTALACIÓN

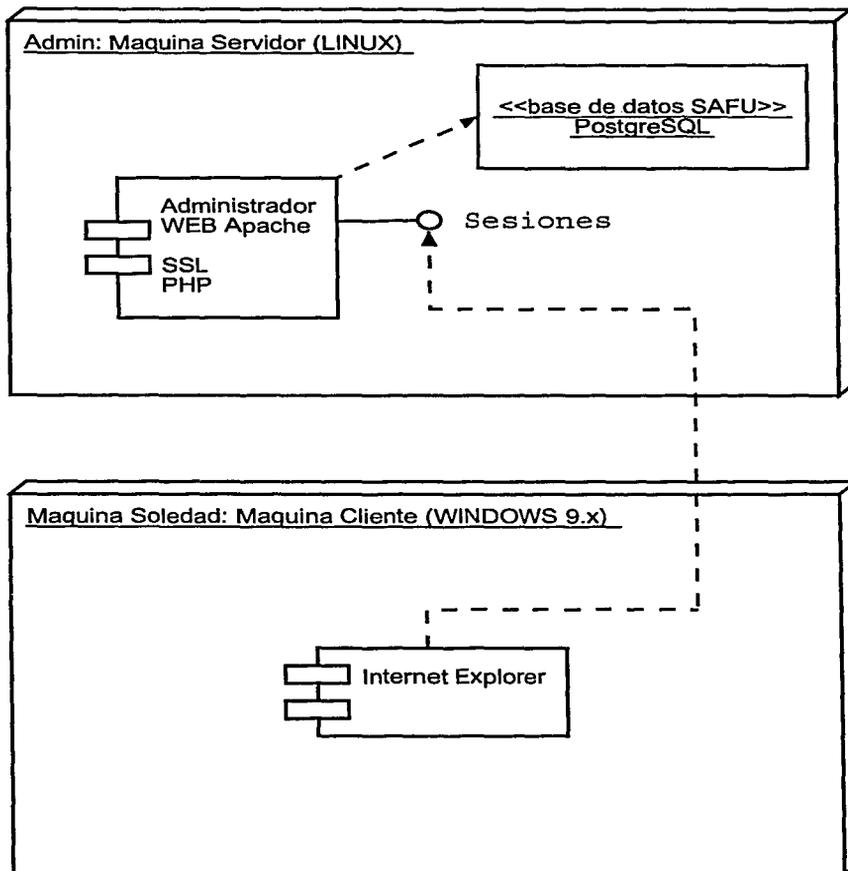


Figura. 5.6: Diagrama de Instalación del SAFU

5.7.3 Diagramas de Secuencia

A continuación se presentan los diagramas de secuencias de los casos de uso más relevantes y significativos del sistema, los demás se encuentran en el Apéndice B.

DIAGRAMAS DE SECUENCIA ALTA DE DATOS GENERALES

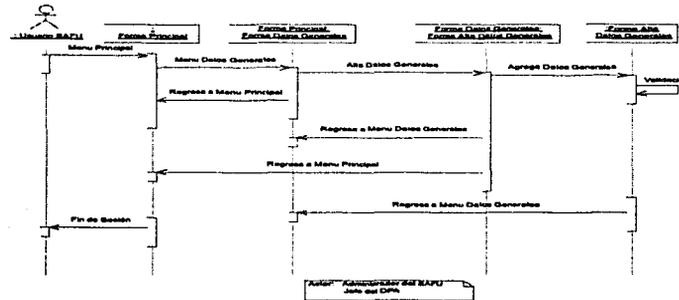


Figura. 5.7: Diagrama de Secuencia F1.1 Seguridad Alta

DIAGRAMAS DE SECUENCIA MODIFICACIÓN DE DATOS GENERALES

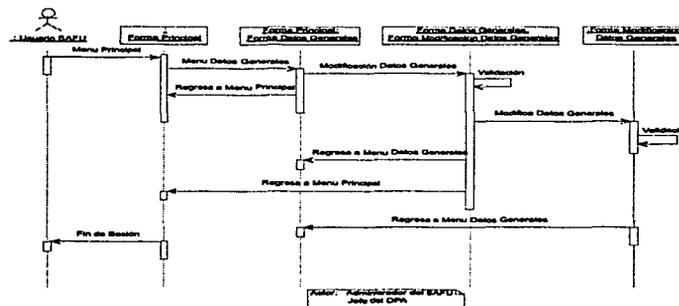


Figura. 5.8: Diagrama de Secuencia F1.2 Seguridad Alta

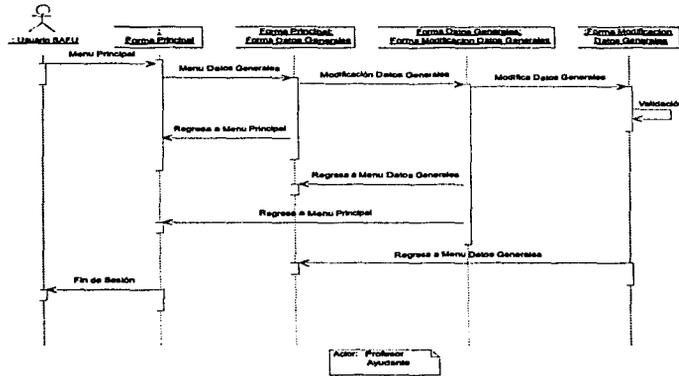


Figura. 5.9: Diagrama de Secuencia F1.2 Seguridad Final

DIAGRAMAS DE SECUENCIA CONSULTA DE DATOS GENERALES

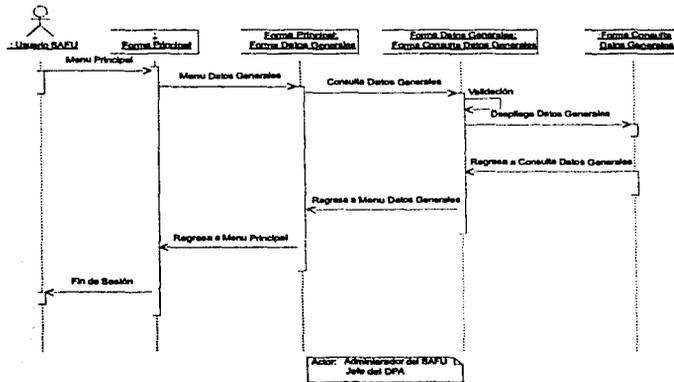


Figura. 5.10: Diagrama de Secuencia F1.3 Seguridad Alta

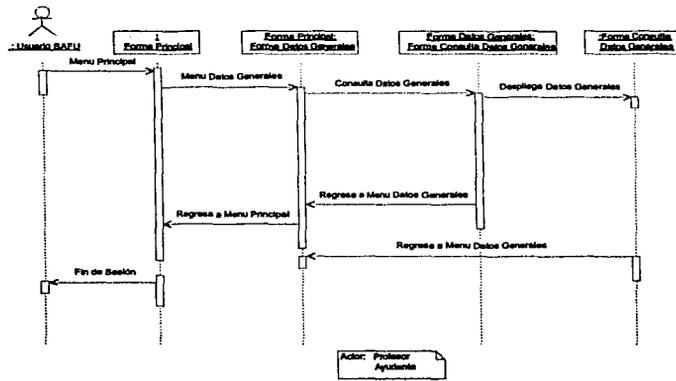


Figura. 5.11: Diagrama de Secuencia F1.3 Seguridad Final

DIAGRAMAS DE SECUENCIA ALTA DE COLONIAS

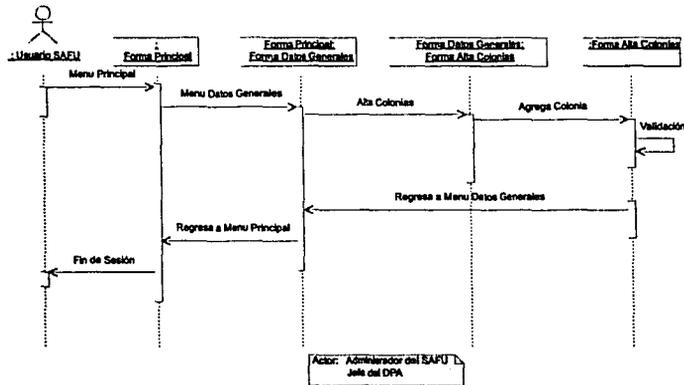


Figura. 5.12: Diagrama de Secuencia F1.4 Seguridad Alta

DIAGRAMAS DE SECUENCIA ALTA DE POBLACIONES

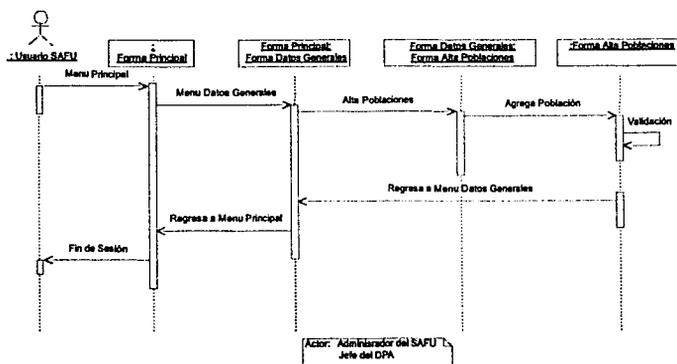


Figura. 5.13: Diagrama de Secuencia F1.5 Seguridad Alta

DIAGRAMAS DE SECUENCIA ALTA DE NACIONALIDADES

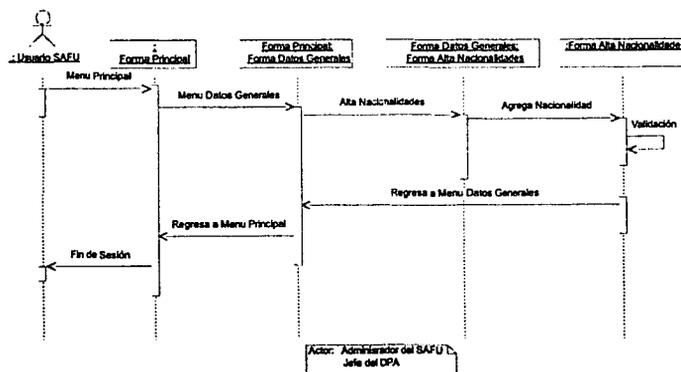


Figura. 5.14: Diagrama de Secuencia F1.6 Seguridad Alta

DIAGRAMAS DE SECUENCIA ALTA DE CONTRATOS

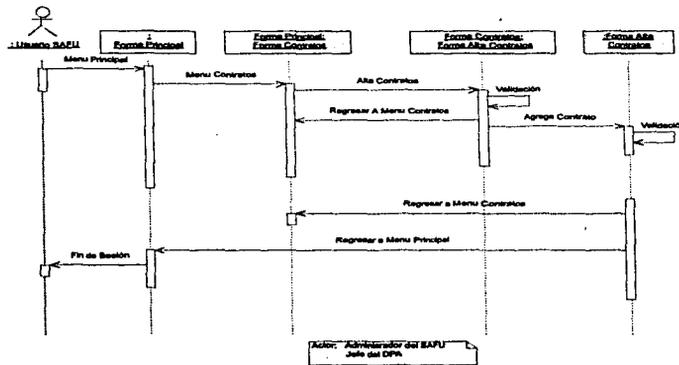


Figura. 5.15: Diagrama de Secuencia F2.1 Seguridad Alta

DIAGRAMAS DE SECUENCIA MODIFICACIÓN DE CONTRATOS

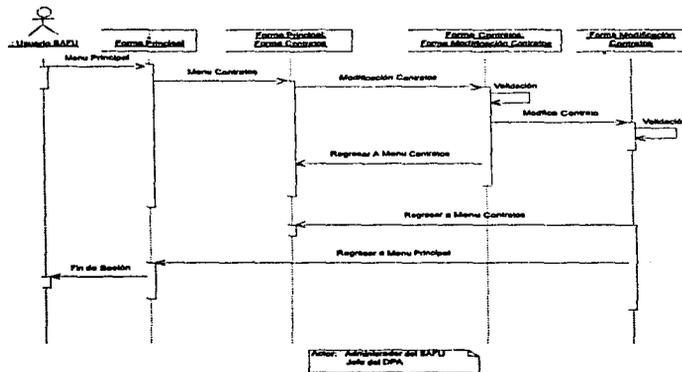


Figura. 5.16: Diagrama de Secuencia F2.2 Seguridad Alta

DIAGRAMAS DE SECUENCIA ELIMINACIÓN DE CONTRATOS

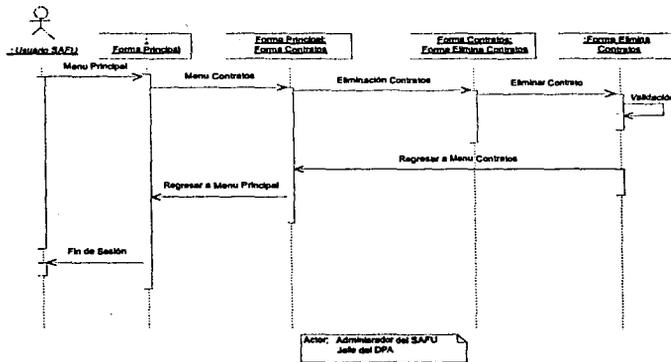


Figura. 5.17: Diagrama de Secuencia F2.3 Seguridad Alta

DIAGRAMAS DE SECUENCIA CONSULTA DE CONTRATOS

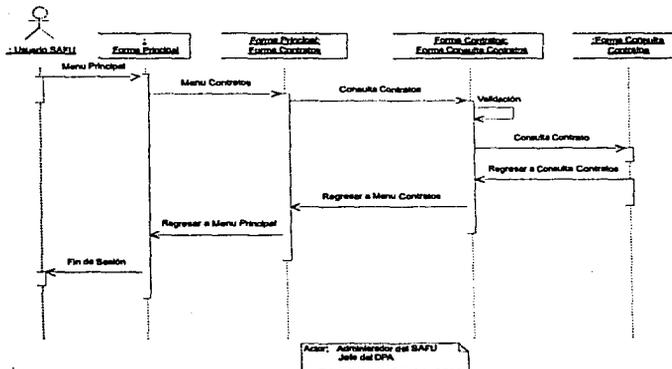


Figura. 5.18: Diagrama de Secuencia F2.4 Seguridad Alta

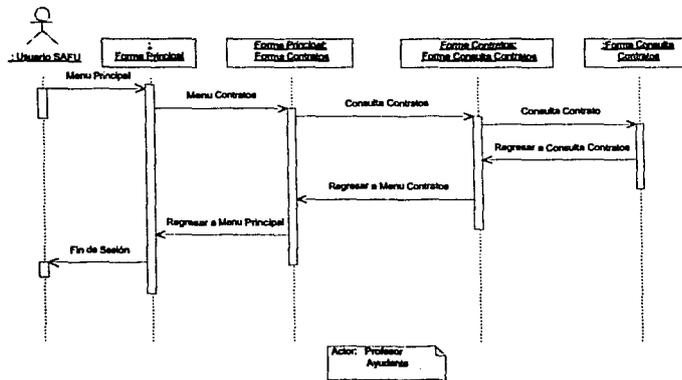


Figura. 5.19: Diagrama de Secuencia F2.4 Seguridad Final

DIAGRAMAS DE SECUENCIA IMPRESIÓN DE CONTRATOS

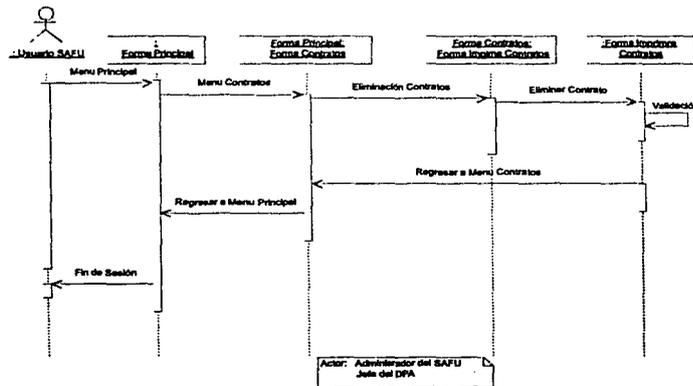


Figura. 5.20: Diagrama de Secuencia F2.5 Seguridad Alta

DIAGRAMAS DE SECUENCIA ALTA DE CLAVES Y CATEGORÍAS

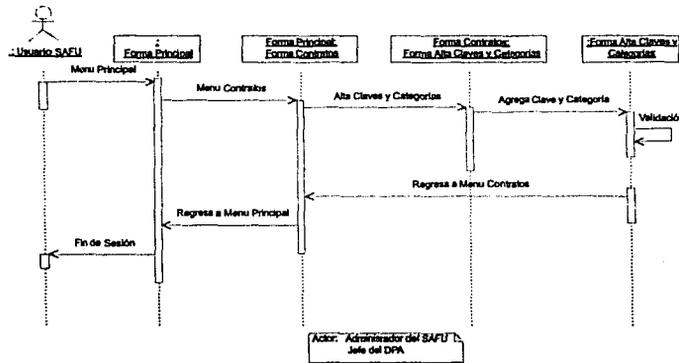


Figura. 5.21: Diagrama de Secuencia F2.6 Seguridad Alta

DIAGRAMAS DE SECUENCIA ALTA DE ASIGNATURAS

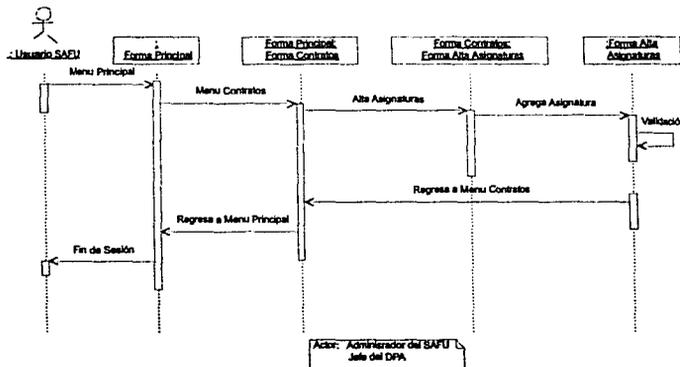


Figura. 5.22: Diagrama de Secuencia F2.7 Seguridad Alta

DIAGRAMAS DE SECUENCIA ALTA DE SUELDOS

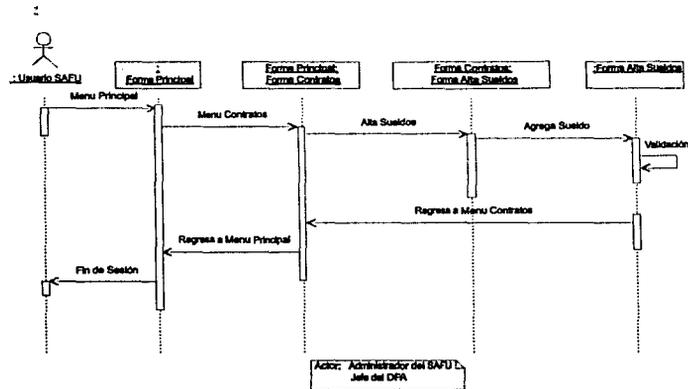


Figura. 5.23: Diagrama de Secuencia F2.8 Seguridad Alta

DIAGRAMAS DE SECUENCIA ALTA DE HORARIOS

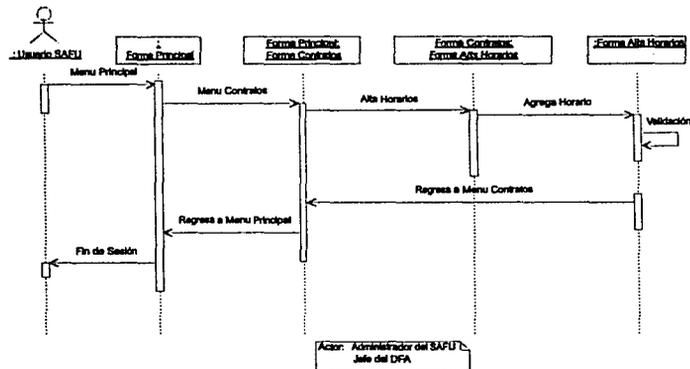


Figura. 5.24: Diagrama de Secuencia F2.9 Seguridad Alta

DIAGRAMAS DE SECUENCIA ALTA DE COLONIAS

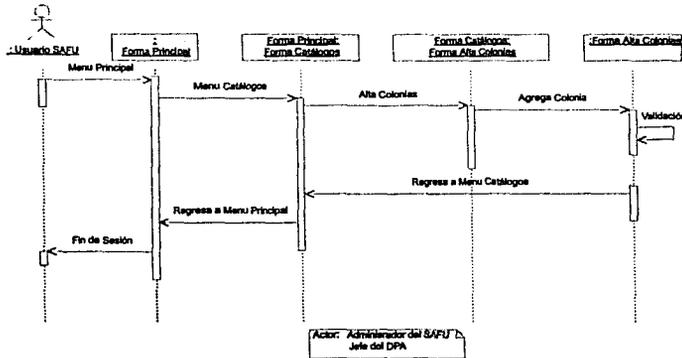


Figura. 5.25: Diagrama de Secuencia F3.1 Seguridad Alta

DIAGRAMAS DE SECUENCIA ALTA DE POBLACIONES

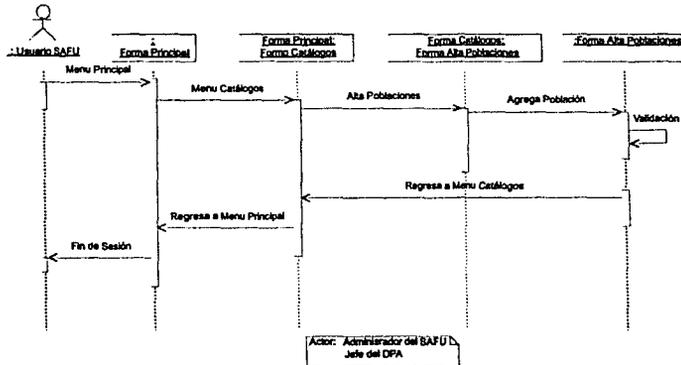


Figura. 5.26: Diagrama de Secuencia F3.2 Seguridad Alta

DIAGRAMAS DE SECUENCIA ALTA DE NACIONALIDADES

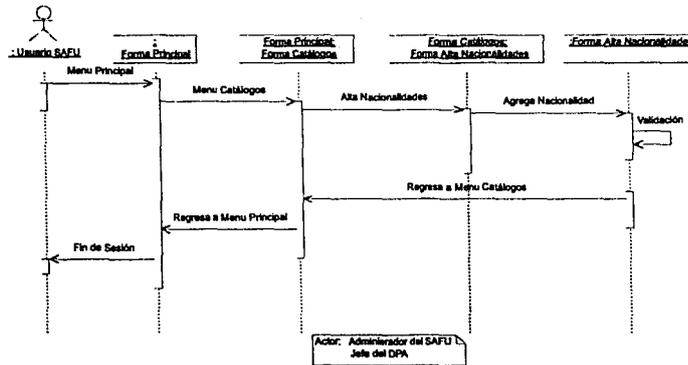


Figura. 5.27: Diagrama de Secuencia F3.3 Seguridad Alta

DIAGRAMAS DE SECUENCIA ALTA DE CLAVES Y CATEGORÍAS

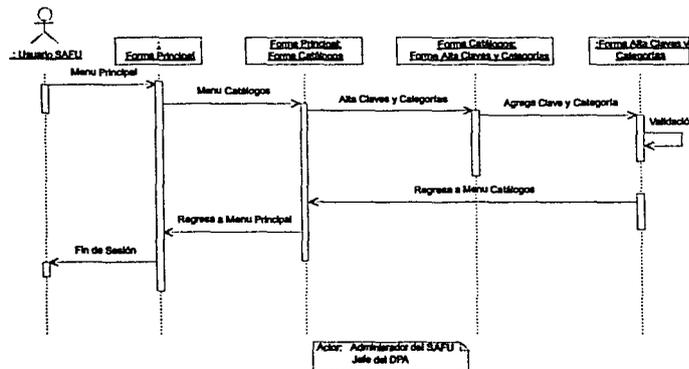


Figura. 5.28: Diagrama de Secuencia F3.4 Seguridad Alta

DIAGRAMAS DE SECUENCIA ALTA DE ASIGNATURAS

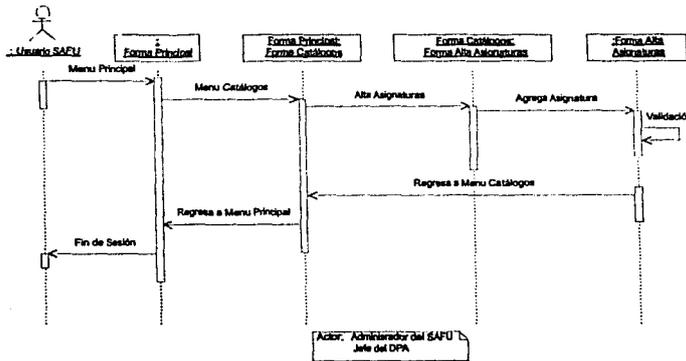


Figura. 5.29: Diagrama de Secuencia F3.5 Seguridad Alta

DIAGRAMAS DE SECUENCIA ALTA DE SUELDOS

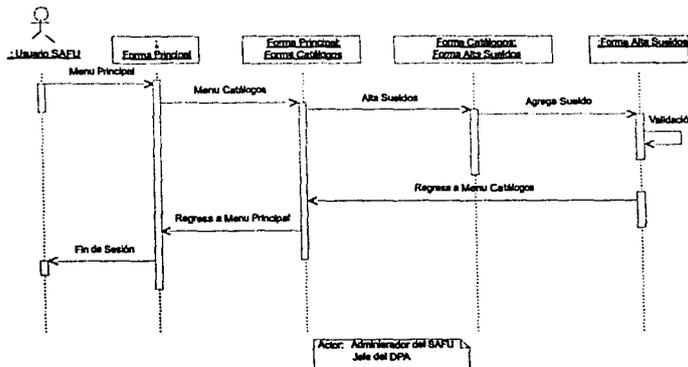


Figura. 5.30: Diagrama de Secuencia F3.6 Seguridad Alta

DIAGRAMAS DE SECUENCIA ALTA DE HORARIOS

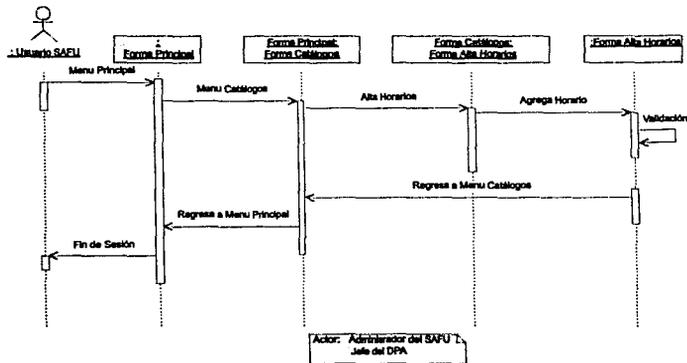


Figura. 5.31: Diagrama de Secuencia F3.7 Seguridad Alta

DIAGRAMAS DE SECUENCIA MODIFICACIÓN DE COLONIAS

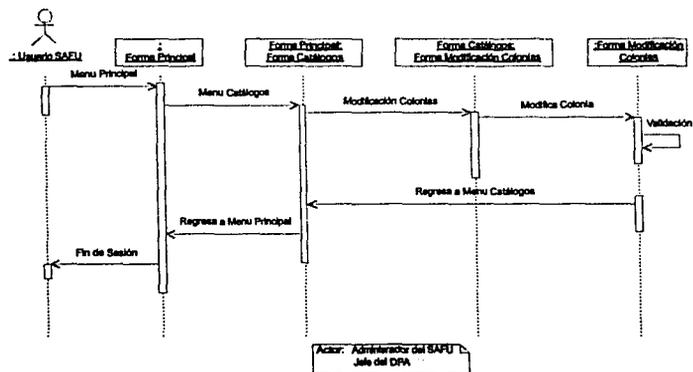


Figura. 5.32: Diagrama de Secuencia F3.8 Seguridad Alta

DIAGRAMAS DE SECUENCIA MODIFICACIÓN DE POBLACIONES

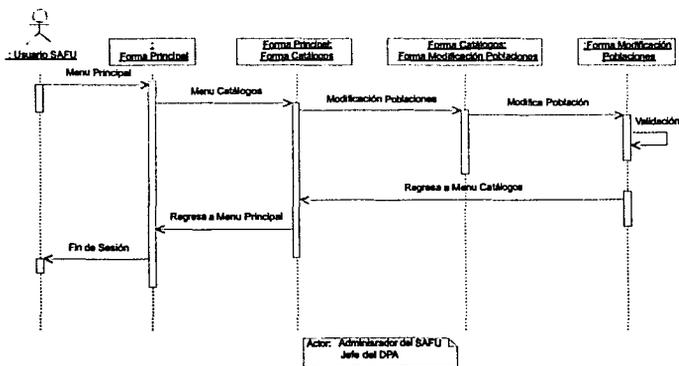


Figura. 5.33: Diagrama de Secuencia F3.9 Seguridad Alta

DIAGRAMAS DE SECUENCIA MODIFICACIÓN DE NACIONALIDADES

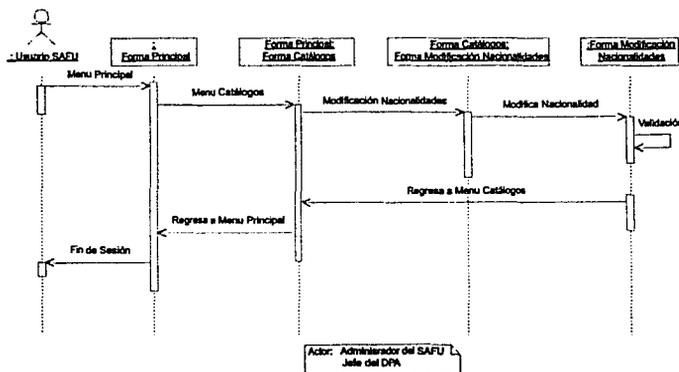


Figura. 5.34: Diagrama de Secuencia F3.10 Seguridad Alta

DIAGRAMAS DE SECUENCIA MODIFICACIÓN DE CLAVES Y CATEGORÍAS

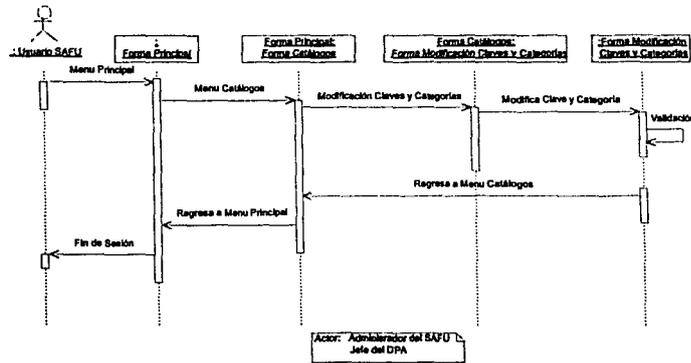


Figura. 5.35: Diagrama de Secuencia F3.11 Seguridad Alta

DIAGRAMAS DE SECUENCIA MODIFICACIÓN DE ASIGNATURAS

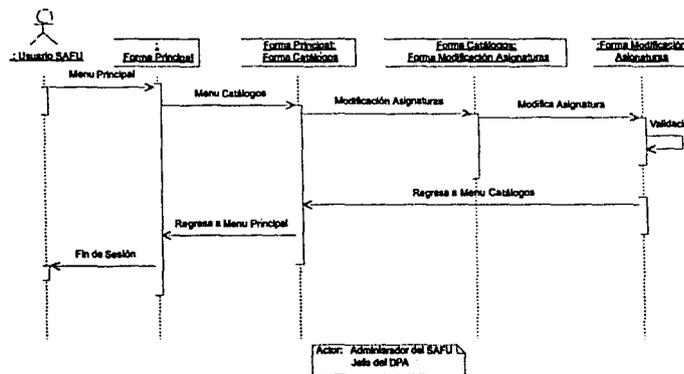


Figura. 5.36: Diagrama de Secuencia F3.12 Seguridad Alta

DIAGRAMAS DE SECUENCIA MODIFICACIÓN DE SUELDOS

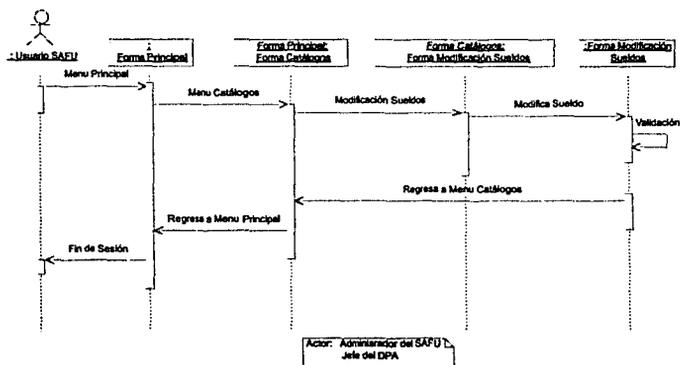


Figura. 5.37: Diagrama de Secuencia F3.13 Seguridad Alta

DIAGRAMAS DE SECUENCIA MODIFICACIÓN DE HORARIOS

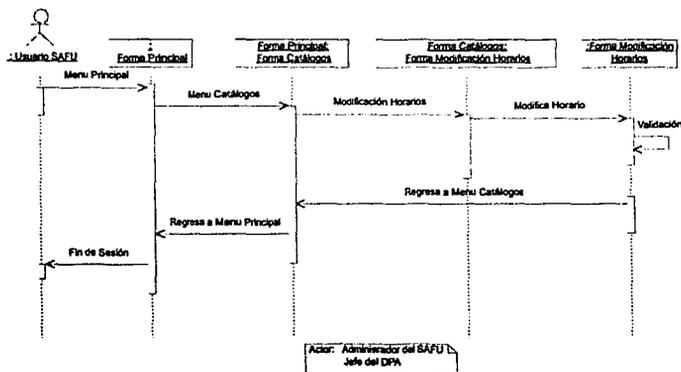


Figura. 5.38: Diagrama de Secuencia F3.14 Seguridad Alta

DIAGRAMAS DE SECUENCIA VER CATÁLOGOS

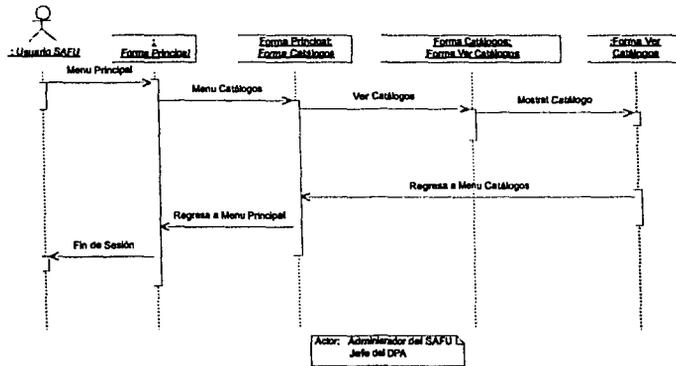


Figura. 5.39: Diagrama de Secuencia F3.15 Seguridad Alta

DIAGRAMAS DE SECUENCIA CONSULTA DATOS GENERALES

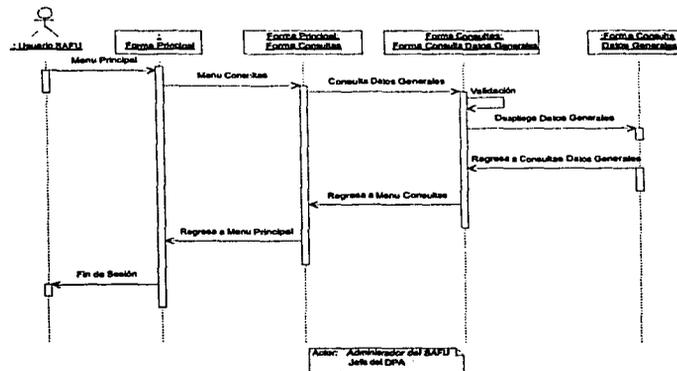


Figura. 5.40: Diagrama de Secuencia F4.1 Seguridad Alta

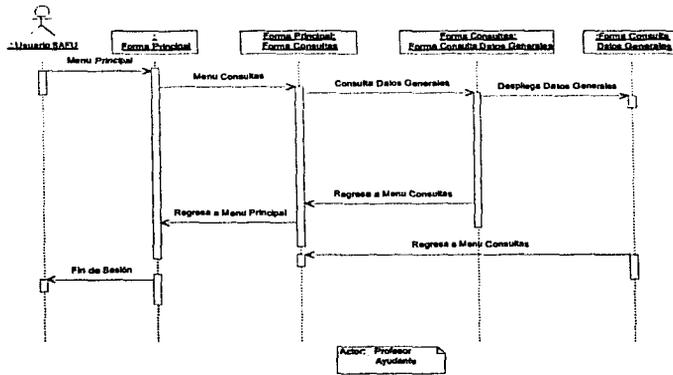


Figura. 5.41: Diagrama de Secuencia F4.1 Seguridad Final

DIAGRAMAS DE SECUENCIA CONSULTA AVANZADA DE DATOS GENERALES

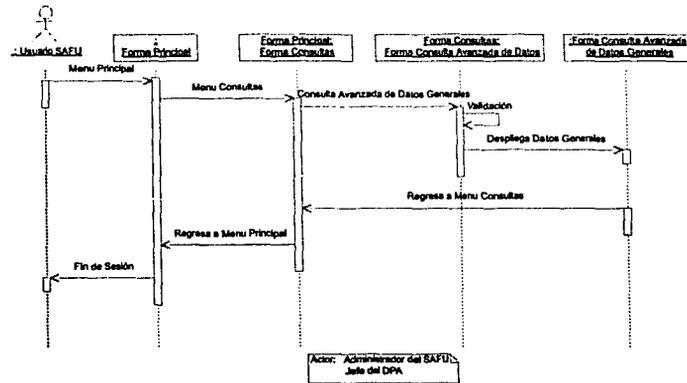


Figura. 5.42: Diagrama de Secuencia F4.2 Seguridad Alta

DIAGRAMAS DE SECUENCIA CONSULTA DE CONTRATOS

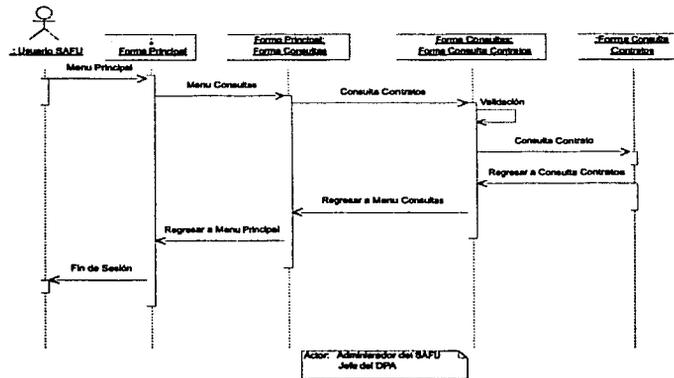


Figura. 5.43: Diagrama de Secuencia F4.3 Seguridad Alta

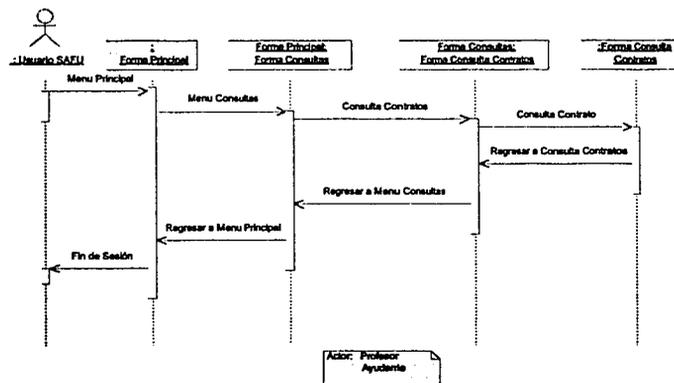


Figura. 5.44: Diagrama de Secuencia F4.3 Seguridad Final

DIAGRAMAS DE SECUENCIA CONSULTAS ESPECIALES

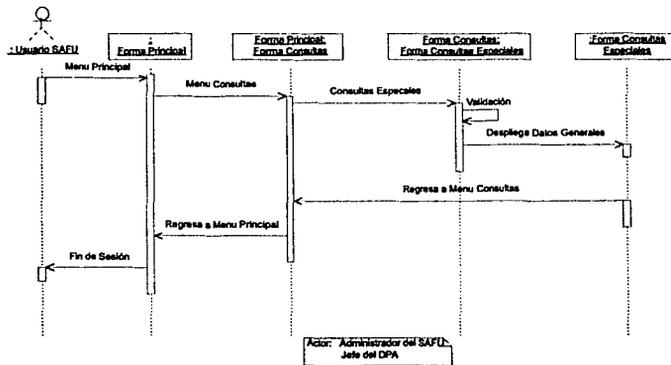


Figura. 5.45: Diagrama de Secuencia F4.4 Seguridad Alta

DIAGRAMAS DE SECUENCIA INFORMACIÓN

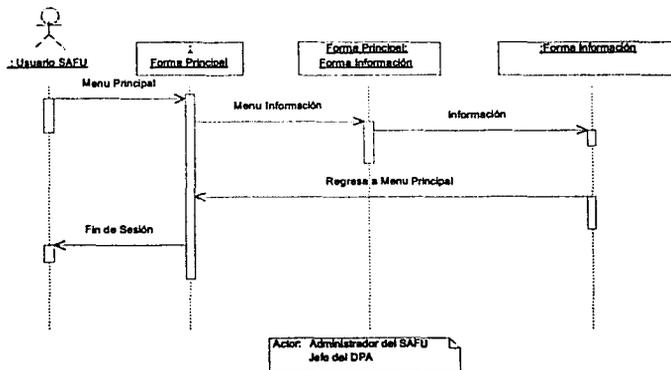


Figura. 5.46: Diagrama de Secuencia F5 Seguridad Alta

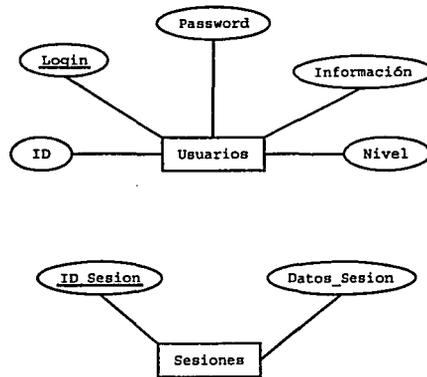
**DIAGRAMA ENTIDAD-RELACIÓN
SEGURIDAD DE LA BASE DE DATOS**

Figura. 5.48: Diagrama E-R de la Seguridad del SAFU.

Capítulo 6

Prototipo del Sistema

El objetivo de esta etapa es mostrar el resultado de la codificación del sistema. El desarrollo de un prototipo es una idea atractiva para sistemas grandes y complejos. Se requiere del análisis, diseño y codificación para poder crearlo, y por medio de él, los desarrolladores se dan cuenta si han alcanzado los requerimientos y especificaciones de los usuarios, así como el cliente se da una mejor idea del sistema que desea. Aunque un prototipo no es el resultado final de un sistema, es importante reconocer que este se aproxima mucho al resultado final, pero se requieren de varios de ellos para poder entregar un sistema casi perfecto y de alta calidad.

6.1 Prototipo

A continuación se presentan algunas pantallas del prototipo preliminar del sistema SAFU.

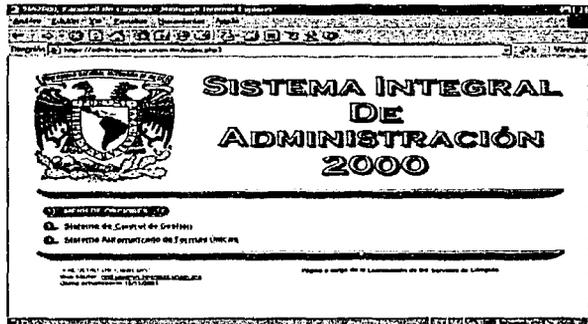


Figura. 6.1: Sistema SIA2000

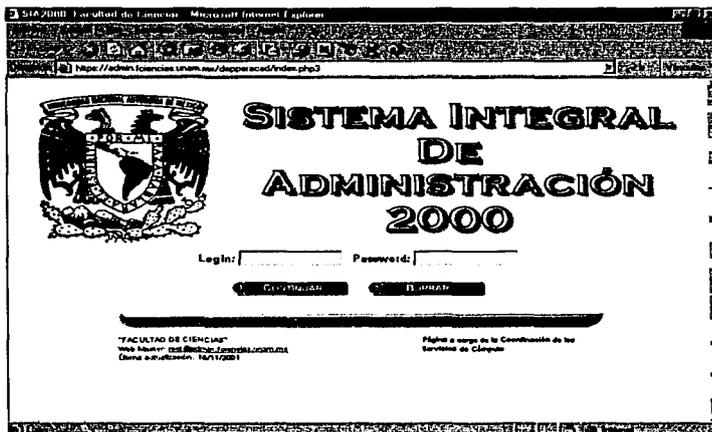


Figura. 6.2: Sistema SAFU. Login y Password

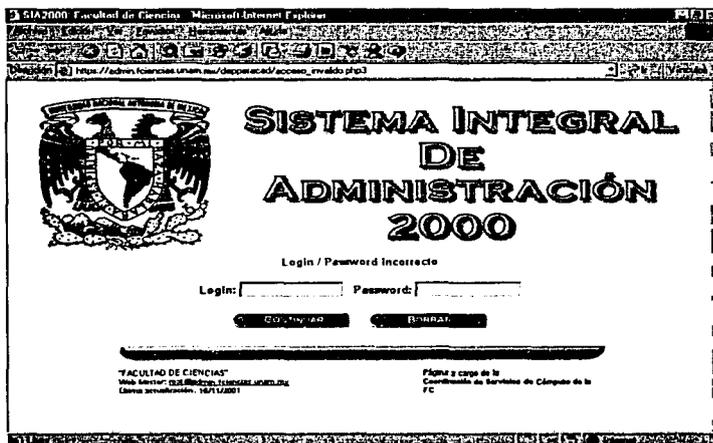


Figura. 6.3: Sistema SAFU. Validación Login y Password

SIAMARU, Validación de Datos Generales: Protocolo Internet Explorer

https://cedrar.licenciacion.unam.mx/dep/validacion/validacion.cgi.php?2

Alta de Datos Generales

NOTA: El campo Expediente es o se llena cuando se cuenta con el número de ingreso otorgado por la Dirección General de Personal (DGP), en caso contrario este campo vacío.

Expediente NO válido
 Expediente NO válido
 Expediente NO válido

Apellido Paterno NO válido
 Apellido Materno NO válido
 Primer Nombre NO válido

Segundo Nombre NO válido

NOTA: Los campos de Apellido Paterno, Apellido Materno, Primer Nombre y Segundo Nombre se deben escribir comenzando con la primera letra mayúscula, en caso de ser un apellido o nombre compuesto cada parte del apellido o nombre(s) se escribirá empezando con mayúscula. Ejemplo: (De La Grca)

Calle y Número NO válido

Colonia
 Población
 Código Postal NO válido

Nacionalidad
 RFC NO válido
 Teléfono NO válido

Sexo
 Estado Civil

Correo Electrónico
 Correo Electrónico NO válido

Religión
 Religión

COORDINACIÓN DE SERVICIOS DE COMPUTO FACULTAD DE CIENCIAS
Carrera de Ingeniería en Informática

000077

Figura. 6.7: Sistema SAFU. Validación Datos Generales 1

3 GIA2000 Validation de Datos Generales Microsoft Internet Explorer

Alta de Datos Generales

USUARIO NUEVO AGREGADO

Los siguientes Datos Generales fueron agregados a la base de datos. El nuevo usuario debe pasar a la Coordinación de Servicios de Computo para que le sea asignado un "login" y "password" con el cual podrá acceder al sistema así como también se le tendrá una foto de identificación personal que firmará parte de su expediente.

1) Número de Expediente	309706
2) Apellido Paterno	Rodriguez
3) Apellido Materno	Orcivaldez
4) Primer Nombre	Rodolfo
5) Segundo Nombre	Ninguno
6) Calle y Número	Condellar a #140 Depto. I
7) Colonia	Miraflores
8) Municipio	Ocoyucán
9) Código Postal	24370
10) Nacionalidad	Mexicana
11) RFC	ROOR760507
12) Teléfono	56823356
13) Sexo	Variado
14) Estado Civil	Soltero
15) Correo Electrónico	rodg@p.fernandezuramex

COORDINACIÓN DE SERVICIOS DE COMPUTO, FACULTAD DE DERECHO
Calle de los Reyes 141172X1

000079

Figura. 6.9: Sistema SAFU. Agregando Datos Generales

Modificación de Datos Generales

Escoge los Datos Generales que se desean modificar y proporciona el número de expediente ó RFC de la persona a la cual se le van a modificar los datos.

RFC Expediente

Apellido Paterno

Apellido Materno

Primer Nombre

Segundo Nombre

Calle y Número

Colonia

Población

Teléfono

Nacionalidad

RFC

Estado Civil

Sexo

Selecciona Todos Limpia Todo

Cancelar Aceptar Validar Datos

COORDINACIÓN DE SERVICIOS DE CÓMPUTO, FACULTAD DE CIENCIAS
Calle Arredondo, 1851/2001

00083

Figura. 6.10: Sistema SAFU. Forma Modificación Datos Generales

Modificación de Datos Generales

Escoge los Datos Generales que se desean modificar y proporciona el número de expediente ó RFC de la persona a la cual se le van a modificar los datos.

RFC Expediente

Apellido Paterno

Apellido Materno

Primer Nombre

Segundo Nombre

Calle y Número

Colonia

Población

Teléfono

Nacionalidad

RFC

Estado Civil

Sexo

Selecciona Todos Limpia Todo

Cancelar Aceptar Validar Datos

COORDINACIÓN DE SERVICIOS DE CÓMPUTO, FACULTAD DE CIENCIAS
Calle Arredondo, 1851/2001

00083

RFC NO válido

Figura. 6.11: Sistema SAFU. Validación Modificación Datos Generales

SINCD31: Modificación de Datos Generales de Datos Generales - Modificación de Datos

http://www.sincd31.com.ar/usuarios/usuarios/modificacion_datos_generales.php

Modificación de Datos Generales

DATOS GENERALES MODIFICADOS !!!

Los siguientes Datos Generales de expediente 009706 fueron modificados en la base de datos.

<input type="checkbox"/> Número Expediente	009706
<input type="checkbox"/> Apellido Paterno	Rosenberg
<input type="checkbox"/> Apellido Materno	Gonzalez
<input type="checkbox"/> Primer Nombre	Roberto
<input type="checkbox"/> Segundo Nombre	Ninguno
<input type="checkbox"/> Calle	Candelaria # 43 Dept. 1
<input type="checkbox"/> Ciudad	Montevideo
<input type="checkbox"/> Fecha Nac.	05/03/73
<input type="checkbox"/> Estado Postal	91000
<input type="checkbox"/> Nacionalidad	Venezolana
<input type="checkbox"/> P.D.	RODRIGUEZ
<input type="checkbox"/> Teléfono	56883306
<input type="checkbox"/> Sexo	Varonil
<input type="checkbox"/> Estado Civil	Soltero
<input type="checkbox"/> E-mail	rob@t.Usuarios.LaBata

CONFIDENCIAL !!

000007 000008

COMUNIDAD DE SERVICIOS DE COMPUTO FACULTAD DE CIENCIAS
Calle Azules 11 - 111000

000007

Figura. 6.13: Sistema SAFU. Modificando Datos Generales

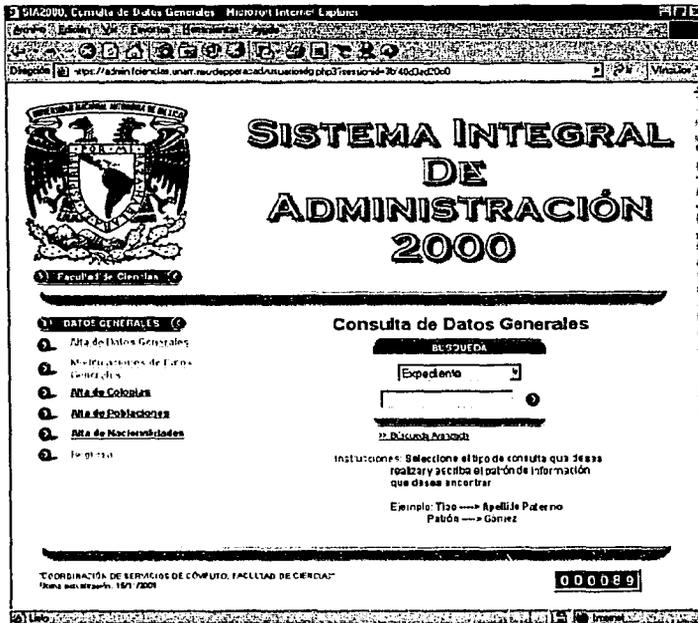


Figura. 6.14: Sistema SAFU. Forma Consulta Datos Generales

Consulta de Datos Generales

Buscador: Ex: Expediente

Mostrar Resultados

Expediente	Apellido Paterno	Apellido Materno	Primer Nombre	Segundo Nombre
107	Hernández	Reyes	Rodrigo	
299	Martínez	Lira	Rodrigo	
*31	Martínez		Rodrigo	
*60	Ortega	Martínez	Rodrigo	
260	Ponce	Juárez	Rodrigo	
273	Soto	Orozco	Rodrigo	
*24078	Cabrera	Vargas	Rodrigo	
094747	Hernández	José	Rodrigo	Tito
803389	Martínez	Lira	Rodrigo	
801718	Pomero	Ramírez	Rodrigo	
809706	Rosenberg	Orozco	Rodrigo	
890333	Díaz	Balón	Rodrigo	
730989	Bayona	Bola	Rodrigo	
*00830	Berra	Herrera	Rodrigo	

[Inicio](#)
[Nuestra Información](#)
[Menu Consultas](#)

INSTITUCIÓN DESENVOLUPAMIENTO DE CAPITAL HUMANO, FACULTAD DE D.D. CIVIL
 Lima, Perú - 15/11/2011

000001

Figura. 6.15: Sistema SAFU. Ejecutando Consulta Datos Generales

CONSULTA DE DATOS GENERALES

BUSQUEDA

Buscar: Experiencia

Nombre Completo: Rosenberg Ouzalix Roberto RFC: RO09760507

Domicilio: Candelaria #140 Depto. 1
Colonia Abasco
Delegación (Declarada) Cuernavaca
C.P. 62470

Nacionalidad: Mexicana Teléfono: 50893330 Sexo: Masculino Estado Civil: Soltero

Correo Electrónico: ros@ch.fciencias.unam.mx

Continuar

COORDINACIÓN DE SERVICIOS DE CÓMPUTO, FACULTAD DE CIENCIAS
Última actualización: 16/1/2001

Figura. 6.16: Sismeta SAFU. Consulta Datos Generales

Alta de Colonia

Identificador: 7884

Colonia: Cobahla

Menu Catalogos Continuar Pasado de 700

COORDINACIÓN DE SERVICIOS DE CÓMPUTO, FACULTAD DE CIENCIAS
Última actualización: 16/1/2001

Figura. 6.17: Sistema SAFU. Forma Alta Colonias

The screenshot shows a web browser window titled "Alta de Poblaciones" with a Microsoft Internet Explorer address bar. The page content includes a title "Alta de Poblaciones", a "Identificador" field with the value "2287", and a "Fecha de nacimiento" field. Below these are navigation links: "Inicio", "Manual de Usuario", "Consultar", "Registrar", and "Cancelar". At the bottom, there is a footer with the text "COORDINACIÓN DE SERVICIOS DE CÓMPUTO, FACULTAD DE CIENCIAS" and "Última actualización: 16/11/2001", along with a small box containing the number "000091".

Figura. 6.18: Sistema SAFU. Forma Alta Poblaciones

The screenshot shows a web browser window titled "Alta de Nacionalidades" with a Microsoft Internet Explorer address bar. The page content includes a title "Alta de Nacionalidades", an "Identificador" field with the value "180", and fields for "Apellido", "Nombre", and "Nacionalidad". Below these are navigation links: "Inicio", "Manual de Usuario", "Consultar", "Registrar", and "Cancelar". At the bottom, there is a footer with the text "COORDINACIÓN DE SERVICIOS DE CÓMPUTO, FACULTAD DE CIENCIAS" and "Última actualización: 16/11/2001", along with a small box containing the number "000091".

Figura. 6.19: Sistema SAFU. Forma Alta Nacionalidades

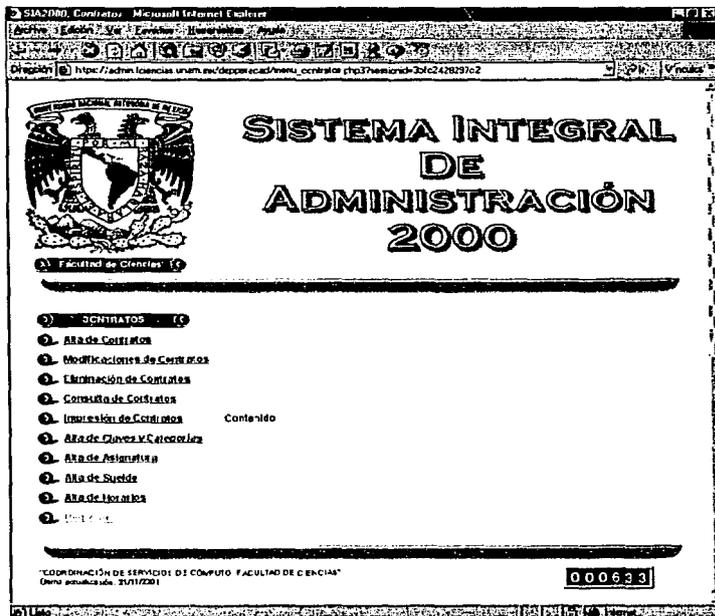


Figura. 6.20: Sistema SAFU. Menú Contratos

SIA2000 Alta de Contratos - Microsoft Internet Explorer

http://admin.fciencias.unam.mx/deppes/acad/contratos/pealtacontratos.php?2

SISTEMA INTEGRAL DE ADMINISTRACIÓN 2000

Facultad de Ciencias

CONTRATOS

- 1 Modificaciones a Contratos
- 2 Eliminación de Contratos
- 3 Consulta de Contratos
- 4 Información de Contratos
- 5 Alta de Claves y Categorías
- 6 Alta de Asignatura
- 7 Alta de Solicitud
- 8 Alta de Usuario
- 9 HELP/PM

Alta de Contratos

Inicio

RFC: []

[Continuar]

Instrucciones: Seleccione al usuario al que se le dará de alta un contrato.

Ejemplo: Tipo → TEC Usuario → ABCDE10203

COORDINACIÓN DE SERVICIOS DE COMPUTO, FACULTAD DE CIENCIAS
Código actualización: 31/1/2001

000633

Figura. 6.21: Sistema SAFU. Alta de Contratos

SIA2000 Alta de Contratos - Microsoft Internet Explorer

http://admin.fciencias.unam.mx/deppes/acad/contratos/pealtacontratos.php?3

Alta de Contratos

USUARIO

Usuario: [] Pw: [RFC]

Apellido Paterno: Rosenberg Apellido Materno: González Primer Nombre: Roberto Segundo Nombre: []

[Continuar]

COORDINACIÓN DE SERVICIOS DE COMPUTO, FACULTAD DE CIENCIAS
Código actualización: 31/1/2001

000633

Figura. 6.22: Sistema SAFU. Alta de Contratos 2

ALTA DE CONTRATOS



U.N.A.M.

Número de Contrato: []

Número de Expediente: []

ME PERMITE PRESENTAR A UD. EL SIGUIENTE MOVIMIENTO:

1) Lugar de Tránsito: [Universidad Nacional Autónoma De México]

2) Dependencia: [Facultad De Ciencias]

3) Subdependencia: [División De Estudios Profesionales]

4) Tipo de Contratación:

<input type="radio"/> Alta	<input type="radio"/> Baja	<input type="radio"/> Licencia Con Buitas	<input type="radio"/> Licencia Sin Buitas
<input type="radio"/> Nueva	<input type="radio"/> Remuneración Adicional	<input type="radio"/> Reclasificación	<input type="radio"/> Año Sabático
<input type="radio"/> Nuevo Ingreso	<input type="radio"/> Retorno	<input type="radio"/> Aumento De Horas	<input type="radio"/> Transferencia
<input type="radio"/> Defunción	<input type="radio"/> Estudios En El País	<input type="radio"/> Promoción	<input type="radio"/> Promoción
<input type="radio"/> Comisión Oficial	<input type="radio"/> Retiro	<input type="radio"/> Separación	<input type="radio"/> Ausencia
<input type="radio"/> Regularización	<input type="radio"/> Alivio Contemplativo	<input type="radio"/> Propensión	<input type="radio"/> Cambio De Los Hijos
<input type="radio"/> Complementaria	<input type="radio"/> Cargo Público	<input type="radio"/> Prorroga Licencia	<input type="radio"/> Elaboración de Teses
<input type="radio"/> Nombramiento De Rector	<input type="radio"/> Examen Profesional	<input type="radio"/> Servicio Social	<input type="radio"/> Cambio Cod. Prog.
<input type="radio"/> Licencia Sindical	<input type="radio"/> Función Acad. Admin.		
<input type="radio"/> Complement Académica			

5) Clave y Categoría: [Elige Clave y Categoría]

6) Asignatura 1: [Asignatura]

7) Asignatura 2: [Ninguna]

8) Comentario 1: []

9) Comentario 2: []

10) Num. Reg. Alza: [00000 - 00]

11) Afecta Nómina: SI No

12) Horas Teóricas: [0]

13) Horas Prácticas: [0]

14) Fecha de Inicio de Contrato: [DIA / MES / AÑO]

15) Fecha de Fin de Contrato: [DIA / MES / AÑO]

16) Costo de Proceso Inicial: []

17) Horario del Lunes: [00 : 00 - 00 : 00]

18) Horario del Martes: [00 : 00 - 00 : 00]

19) Horario del Miércoles: [00 : 00 - 00 : 00]

20) Horario del Jueves: [00 : 00 - 00 : 00]

21) Horario del Viernes: [00 : 00 - 00 : 00]

22) Horario del Sábado: [00 : 00 - 00 : 00]

23) Observaciones:

[Continuar] [Finalizar]

CODOR 30

COMISIÓN DE SERVICIOS DE COMPUTO, FACULTAD DE CIENCIAS
Dirección: Av. 20 de Noviembre 2171/2001

Figura. 6.23: Sistema SAFU. Forma de Alta de Contratos

Validación de Alta de Contratos - Microsoft (Internal Explorer)

Alta de Contratos

Personal Académico

Numero de Oficio: 100100

Numero de Expediente: 009708

ME PERMITO PRESENTAR A UD. EL SIGUIENTE MOVIMIENTO:

Lugar de Tránsito: UNAM. Dependencia: Facultad De Ciencias. Situación: División De Estudios Profesionales

Tipo de Contrato: Alta. Clave: Prorroga

Clave y Categoría: D-1300 - Ayudante De Profesor De Asignatura B Intérno

Asignatura: Inteligencia Artificial

Comentario 1: Ninguna

Asignatura 2: Ninguna

Comentario 2: Ninguna

Num. Req. Para: Ninguno

Salario o Remuneración: \$2860 9C Dos Mil Novecientos Sesenta pesos 00/100 M.N.

Alcance Nomina: B. Horas Técnicas: 10.0. Horas Piblicas: 10.0

Fecha de Inicio de Contrato: 07/01/2002. Fecha de Fin del Contrato: 07/08/2002. Código Proposicional: 1004411.01.142.01

Horario del Lunes: 00:00 - 00:00. Horario del Martes: 12:00 - 13:00. Horario del Miércoles: 00:00 - 00:00

Horario del Jueves: 12:00 - 13:00. Horario del Viernes: 00:00 - 00:00. Horario del Sábado: 00:00 - 00:00

Observaciones:

Revise los datos nuevamente y si son correctos oprima "Almacenar Contrato" Si hay algún error en ellos oprima "Regresar"

Almacenar Contrato

COORDINACIÓN DE SERVICIOS DE COMPUTO, FACULTAD DE CIENCIAS. Última actualización: 31/11/2001. 000633

Figura. 6.25: Sistema SAFU. Agregando Contrato

SAF2000: Modificación de Contratos - Microsoft Internet Explorer

Inicio | Edición | Ver | Formato | Herramientas | Ayuda

Dirección: https://admin.ciencias.unam.mx/depoeac/mod_contratos.php?version=3c0c78d34c5f

Modificación de Contratos

Escoga los Datos del contrato que se desean modificar y proporcione el número de expediente ó RFC de la persona a la cual se le va a modificar el contrato.

RFC Expediente: [RFC:]

<input type="checkbox"/> 1) Número de Oficio	<input type="checkbox"/> 4) Puesto	<input type="checkbox"/> 7) Subdependencia
<input type="checkbox"/> 2) Tipo de Solución	<input type="checkbox"/> 5) Causa	<input type="checkbox"/> 8) Clase y Categoría
<input type="checkbox"/> 3) Asignatura 1	<input type="checkbox"/> 6) Comentario	<input type="checkbox"/> 9) Asignatura 2
<input type="checkbox"/> 4) Comentario 2	<input type="checkbox"/> 10) Num. Reg. Plaza	<input type="checkbox"/> 10) Alerta Nomina
<input type="checkbox"/> 5) Horas Teóricas	<input type="checkbox"/> 11) Horas Prácticas	<input type="checkbox"/> 11) Sujeto o Remuneración
<input type="checkbox"/> 6) Fecha de Inicio de Contrato	<input type="checkbox"/> 12) Fecha de Fin de Contrato	<input type="checkbox"/> 12) Cálculo Programado
<input type="checkbox"/> 7) Horario del Lunes	<input type="checkbox"/> 13) Horario del Martes	<input type="checkbox"/> 13) Horario del Miércoles
<input type="checkbox"/> 8) Horario del Jueves	<input type="checkbox"/> 14) Horario del Viernes	<input type="checkbox"/> 14) Horario del Sábado
<input type="checkbox"/> 9) Observaciones		

Continuar | Restablecer

COORDINACIÓN DE SERVICIOS DE COMPUTO, FACULTAD DE CIENCIAS
 Última actualización: 03/12/2001

001513

Figura. 6.27: Sistema SAFU. Modificación de Contratos

SAF2000: Validación de Modificación De Contratos - Microsoft Internet Explorer

Inicio | Edición | Ver | Formato | Herramientas | Ayuda

Dirección: https://admin.ciencias.unam.mx/depoeac/mod_contratos.php?version=3c0c78d34c5f&action=1

Modificación de Contratos

Escoga los Datos del contrato que se desean modificar y proporcione el número de expediente ó RFC de la persona a la cual se le va a modificar el contrato.

RFC Expediente: [RFC:]

RFC NO válido

<input type="checkbox"/> 1) Número de Oficio	<input type="checkbox"/> 4) Puesto	<input type="checkbox"/> 7) Subdependencia
<input type="checkbox"/> 2) Tipo de Solución	<input type="checkbox"/> 5) Causa	<input type="checkbox"/> 8) Clase y Categoría
<input type="checkbox"/> 3) Asignatura 1	<input type="checkbox"/> 6) Comentario	<input type="checkbox"/> 9) Asignatura 2
<input type="checkbox"/> 4) Comentario 2	<input type="checkbox"/> 10) Num. Reg. Plaza	<input type="checkbox"/> 10) Sujeto o Remuneración
<input type="checkbox"/> 5) Alerta Nomina	<input type="checkbox"/> 11) Horas Teóricas	<input type="checkbox"/> 11) Horas Prácticas
<input type="checkbox"/> 6) Fecha de Inicio de Contrato	<input type="checkbox"/> 12) Fecha de Fin de Contrato	<input type="checkbox"/> 12) Cálculo Programado
<input type="checkbox"/> 7) Horario del Lunes	<input type="checkbox"/> 13) Horario del Martes	<input type="checkbox"/> 13) Horario del Miércoles
<input type="checkbox"/> 8) Horario del Jueves	<input type="checkbox"/> 14) Horario del Viernes	<input type="checkbox"/> 14) Horario del Sábado
<input type="checkbox"/> 9) Observaciones		

Continuar | Restablecer

COORDINACIÓN DE SERVICIOS DE COMPUTO, FACULTAD DE CIENCIAS
 Última actualización: 03/12/2001

001513

Figura. 6.28: Sistema SAFU. Validación de Modificación de Contratos 1

Validación de Modificación de Contratos - Microsoft Internet Explorer

SE. RECTOR DE LA U.N.A.M. PRESENTE.

Modificación de Contratos

Académico 100101

609708

ME PORMENTO PRESENTAR A UO EL SOURCE MOVIMIENTO:

Universidad Nacional Autónoma De México Facultad De Ciencias División De Estudios Profesionales

Alta

Nuevo Ingreso

D-1300 - Asistente De Profes De Asignatura B Interno

Acuicultura

Ninguno

Ninguno

Ninguno

Ninguno

0000-00

\$2878.30 - Dos Mil Setecientos Ochenta y Nueve pesos 20100 M.N.

Atenas Nomina	Horas Teóricas	Horas Prácticas
SI	10	10
Tecnico Inicio Contrato	Tecnico Fin de Contrato	Código Proposición
30/01/2001	25/12/2001	10 02 411.01.142.05
Fecha Inicio Plazas	Fecha Fin Plazas	
00-00-00-00	00-00-00-00	00-00-00-00
Horario de Inicio	Horario del Fin	Horario de Salida
00-00-00-00	00-00-00-00	00-00-00-00

Revisar datos

Revisar los datos nuevamente y si son correctos apriete "Modificar Contrato"
Si hay al glos error en datos apriete "Regresar"

MODIFICAR CONTRATO

REGRESAR

COORDINACIÓN DE EDUCACIÓN DE CÓMPUTO, FACULTAD DE CIENCIAS
(Calle Acatlán, 9317 000)

001534

Figura. 6.29: Sistema SAFU. Validación de Modificación de Contratos 2

SIA 2000 Consulta de Contratos. No se está Internet Explorer

http://www1.cinvestav.unam.mx/departamentos/informacion_contratos.php?sessionid=300007009425



SISTEMA INTEGRAL DE ADMINISTRACIÓN 2000

Facultad de Ciencias

MENU DE OPCIONES

- 1. Alta de Contratos
- 2. Modificación de Contratos
- 3. Consulta de Contratos
- 4. Imprimir de Contratos
- 5. Alta de Clave y Categoría
- 6. Alta de Asesoría
- 7. Alta de Servicio
- 8. Alta de Horario
- 9. Reportes

Eliminación de Contratos

BUQUEDA

Expediente

809706

Instrucciones: Seleccione el tipo de consulta que desea realizar y escriba el patrón de información que desea encontrar.

Ejemplo Tipo → Apellido Paterno
Patrón → Gómez

COORDINACIÓN DE SERVICIOS DE COMPUTO, FACULTAD DE CIENCIAS
Calle de Arroyo 647 0451

001547

Figura. 6.31: Sistema SAFU. Eliminación de Contratos

SIAZ000: Eliminación de Contratos Microsoft Internet Explorer

dominioservicios.unam.mx/Departamento/contabilidad/eliminacioncontrato.php?res=contr-36/Doc-70/54-51/Depo-1/Contas-009700

Eliminación de Contratos

BUSQUEDA

Buscar: Ex: Expediente

Expediente	RFC	Oficio	Fecha	Movimiento
<input type="checkbox"/> 809700	ROOR780507	43	Wed Mar 07	D-1300
<input type="checkbox"/> 809700	ROOR780507	44	Wed Mar 07	D-1300
<input type="checkbox"/> 809700	ROOR780507	45	Wed Mar 07	D-1200
<input type="checkbox"/> 809700	ROOR780507	46	Wed Mar 07	D-1300
<input type="checkbox"/> 809700	ROOR780507	47	Wed Mar 07	D-1200
<input type="checkbox"/> 809700	ROOR780507	48	Wed Mar 07	D-1300
<input type="checkbox"/> 809700	ROOR780507	100100	Mon Mar 12	D-1300
<input type="checkbox"/> 809700	ROOR780507	509	Mon Nov 26	D-1200
<input type="checkbox"/> 809700	ROOR780507	510	Mon Nov 26	D-1200
<input type="checkbox"/> 809700	ROOR780507	518	Tue Nov 27	D-1300
<input type="checkbox"/> 809700	ROOR780507	511	Tue Nov 27	D-1300
<input type="checkbox"/> 809700	ROOR780507	852	Fri Nov 30	D-1300

Eliminar Control

COORDINACIÓN DE SERVICIOS DE COMPUTO, FACULTAD DE CIENCIAS
Última actualización: 03/12/2001

001547

Figura. 6.32: Sistema SAFU. Ejecución de Eliminación de Contratos 1

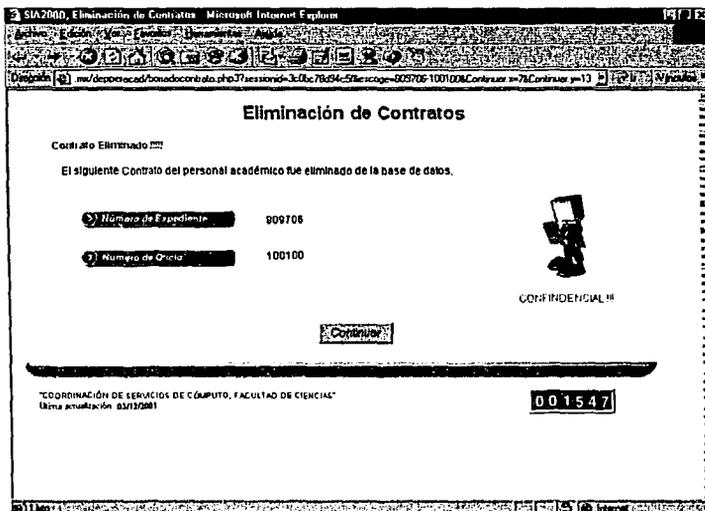


Figura. 6.34: Sistema SAFU. Eliminando Contrato

SIADHU - Consulta de Contratos - Menú de Opciones - Login

https://administracion.unam.mx/dep/aca/c/consulta_contratoscent.php?mes=oct-06&hor=7:05:45



SISTEMA INTEGRAL DE ADMINISTRACIÓN 2000

Facultad de Ciencias

MENÚ DE OPCIONES

- 1. Alta de Contratos
- 2. Modificación de Contratos
- 3. Eliminación de Contratos
- 4. Imposición de Contratos
- 5. Alta de Clases y Categorías
- 6. Alta de Asignaturas
- 7. Alta de Semestre
- 8. Alta de Horario
- 9. Reportes

Consulta de Contratos

BUSQUEDA

Expediente:

Instrucciones: Seleccione el tipo de consulta que desea realizar y escriba el parámetro de información que desea encontrar.

Ejemplo: Tipo → Apellido Paterno
Patrón → Gómez

COORDINACIÓN DE SERVICIOS DE CÓMPUTO, FACULTAD DE CIENCIAS
Cuenta de usuario: 03/13/2001

007547

Figura. 6.35: Sistema SAFU. Consulta de Contratos

SIA2000, Consulta de Contratos - Microsoft Internet Explorer

http://admin.fceencias.unam.mx/depepecad/consultacontratos/contratos.php?res=sond-cc0bc7b94c5f

Consulta de Contratos

BUROFUDA

Busca: Etc. Expediente

SEGUIR PAGANDO				
Expediente	RFC	Oficio	Fecha	Nombre mesero
809706	ROOR760507	43	Wed Mar 07	D-1300
809706	ROOR760507	44	Wed Mar 07	D-1300
809706	ROOR760507	45	Wed Mar 07	D-1200
809706	ROOR760507	46	Wed Mar 07	D-1300
809706	ROOR760507	47	Wed Mar 07	D-1200
809706	ROOR760507	48	Wed Mar 07	D-1300
809706	ROOR760507	539	Mon Nov 26	D-1200
809706	ROOR760507	510	Mon Nov 26	D-1200
809706	ROOR760507	518	Tue Nov 27	D-1300
809706	ROOR760507	511	Tue Nov 27	D-1300
809706	ROOR760507	652	Fri Nov 30	D-1300

[Inicio](#)
[Mostrar Información](#)
[Menu Consultas](#)

COORDINACIÓN DE SERVICIOS DE COMEDIO, FACULTAD DE CIENCIAS
 (Código de autorización: 001547)

Figura. 6.36: Sistema SAFU. Ejecutando Consulta de Contratos

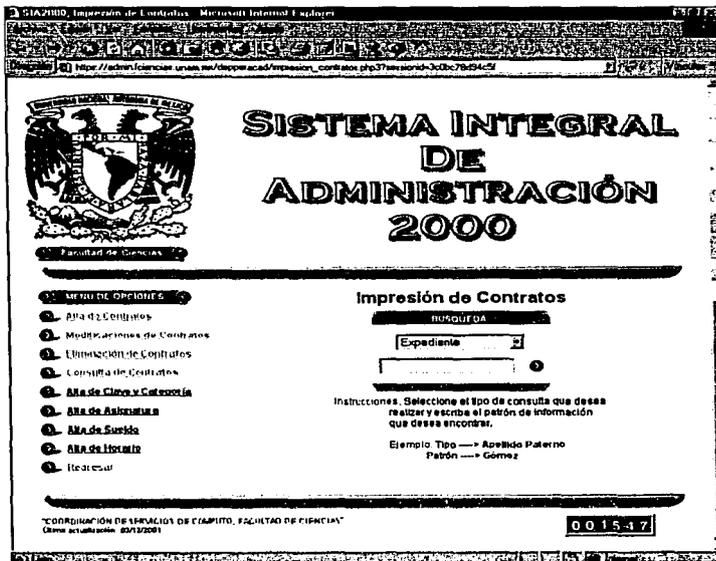


Figura. 6.37: Sistema SAFU. Impresión de Contratos

SIA2000 Impresión de Contratos - Microsoft Internet Explorer

http://admin.licencias.unam.mx/departoac/consultayimpresioncontrato.php3?version=30bc78b94c5f

Impresión de Contratos

DESCUCCION

Buscar: Ent: Expediente

Expediente	RFC	Oficio	Fecha	Nombramiento
<input type="checkbox"/> 809706	ROOR7605C7	43	Wed Mar 37	D-1300
<input type="checkbox"/> 809706	ROOR7605C7	44	Wed Mar 37	D-1300
<input type="checkbox"/> 809706	ROOR7605C7	45	Wed Mar 37	D-1200
<input type="checkbox"/> 809706	ROOR7605C7	43	Wed Mar 37	D-1300
<input type="checkbox"/> 809706	ROOR7605C7	47	Wed Mar 37	D-1200
<input type="checkbox"/> 809706	ROOR7605C7	43	Wed Mar 37	D-1300
<input type="checkbox"/> 809706	ROOR7605C7	508	Mon Nov 26	D-1200
<input type="checkbox"/> 809706	ROOR7605C7	510	Mon Nov 26	D-1200
<input type="checkbox"/> 809706	ROOR7605C7	518	Tue Nov 27	D-1200
<input type="checkbox"/> 809706	ROOR7605C7	511	Tue Nov 27	D-1300
<input type="checkbox"/> 809706	ROOR7605C7	652	Fri Nov 30	D-1300

Mostrar Contratos | Menú Consulta

COORDINACIÓN DE SERVICIOS DE COMPUTO, FACULTAD DE CIENCIAS
 Última actualización: 03/12/2001

001517

Figura. 6.38: Sistema SAFU. Ejecutando Impresión de Contratos

Alta de Claves y Categoría

Nombre:
 Apellido:
 Categoría:
 Nivel:
 Carrera:
 Grado:

Menú Catálogos | Menú Cursos | |

COORDINACIÓN DE SERVICIOS DE CÓMPUTO, FACULTAD DE CIENCIAS
 Última actualización: 23/12/2001 001547

Figura. 6.39: Sistema SAFU. Alta de Claves y Categorías

Alta de Asignatura

Clave:
 Número:
 Nombre:
 Créditos:
 Horas Teóricas:
 Horas Prácticas:
 Programa:
 Carrera:
 Semestre:

Menú Catálogos | Menú Cursos | |

COORDINACIÓN DE SERVICIOS DE CÓMPUTO, FACULTAD DE CIENCIAS
 Última actualización: 23/12/2001 001547

Figura. 6.40: Sistema SAFU. Alta de Asignaturas

The screenshot shows a web browser window titled "Alta de Sueldo" with the URL "http://admfin.ciencias.unam.mx/dgpp/acad/usuarios.php?Tresorero=36Dhc78d94c5". The form contains the following fields:

- Identificador: 1042
- Nombre (Numerado): [Empty]
- Cantidad (Mensual): [Empty]
- Clave (Categoría): D-1200
- Moras: 0
- Fecha de Vigencia: [Empty]

Navigation links include "Inicio", "Inicio de Sesión", "Menu", "Categorias", "Continuar", and "Restablecer". The footer text reads "COORDINACIÓN DE SERVICIOS DE CÓMPUTO, FACULTAD DE CIENCIAS" and "Servicio brindado en 03/12/2001". A session ID "001547" is displayed in a box.

Figura. 6.41: Sistema SAFU. Alta de Sueldos

The screenshot shows a web browser window titled "Alta de Horario" with the same URL as Figure 6.41. The form contains the following fields:

- Identificador: 43
- Horario Inicio: 01:00
- Horario Fin: 01:00

Navigation links include "Inicio", "Inicio de Sesión", "Menu", "Categorias", "Continuar", and "Restablecer". The footer text is identical to Figure 6.41, including "COORDINACIÓN DE SERVICIOS DE CÓMPUTO, FACULTAD DE CIENCIAS" and "Servicio brindado en 03/12/2001". A session ID "001547" is displayed in a box.

Figura. 6.42: Sistema SAFU. Alta de Horarios

SIA 2000 Catalogue - Microsoft Internet Explorer

http://administracion.unari.mg/dgpmacac/menu_catalogos.php?seccion=320b27e654c9

SISTEMA INTEGRAL DE ADMINISTRACIÓN 2000

Facultad de Ciencias

MENU DE OPCIONES

1. Alta de Colonias
2. Alta de Poblaciones
3. Alta de Nacionalidades
4. Alta de Clases y Categorías
5. Alta de Asignaturas
6. Alta de Siestas
7. Alta de Hieratos
8. Modificación de Colonias
9. Modificación de Poblaciones
10. Modificación de Nacionalidades
11. Modificación de Clases y Categorías
12. Modificación de Asignaturas
13. Modificación de Siestas
14. Modificación de Hieratos
15. Ver Catálogos
16. Regresar

Contenido

© 2000 MAC. ÓN DE SERVICIOS DE COMPUTO, FACULTAD DE CIENCIAS
DISEÑO Y DESARROLLO: EPT/2000

001547

Figura. 6.43: Sistema SAFU. Menú Catálogos

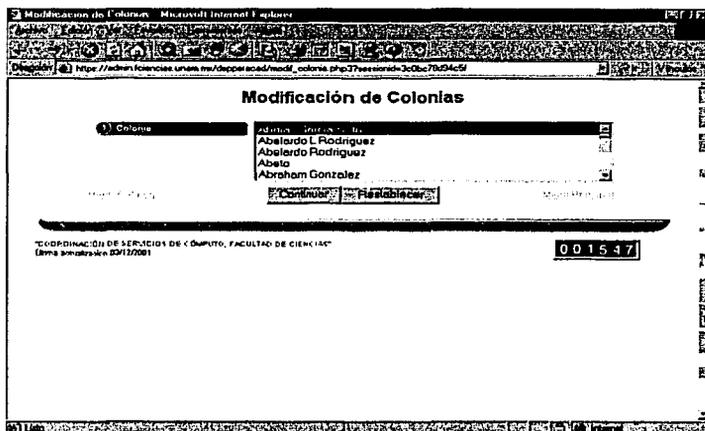


Figura. 6.44: Sistema SAFU. Modificación de Colonias

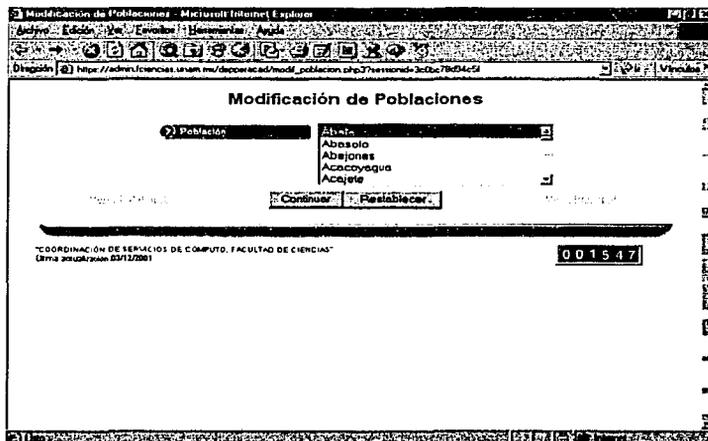


Figura. 6.45: Sistema SAFU. Modificación de Poblaciones

The screenshot shows a web browser window with the title "SAFU Modificación de Nacionalidades". The address bar contains the URL: https://admis.ciencias.unam.mx/despejacad/modif_nacionalidad.php?Theccion=00100789429. The main heading is "Modificación de Nacionalidades". Below the heading, there is a form with several fields and checkboxes. A dropdown menu is open, showing the following options: "Albanesa", "Alemana", "Alto Voltaica", and "Andorrea". To the right of the dropdown, there are checkboxes for "Agrupación", "Pais", and "Nacionalidad". Below the form, there are buttons for "Continuar" and "Cancelar". At the bottom of the page, there is a footer that reads: "COORDINACIÓN DE SERVICIOS DE CÓMPUTO, FACULTAD DE CIENCIAS" and "Última actualización: 02/12/2001". A small box in the bottom right corner contains the number "001417".

Figura. 6.46: Sistema SAFU. Modificación de Nacionalidades

The screenshot shows a web browser window with the title "SAFU Modificación de Claves y Categorías". The address bar contains the URL: https://admis.ciencias.unam.mx/despejacad/modif_clavescategorias.php?Theccion=00100789429. The main heading is "Modificación de Claves y Categorías". Below the heading, there is a form with several fields and checkboxes. A dropdown menu is open, showing the following options: "Ayudante De Profesor De Asignatura A Interino", "Personal Jubilado A", "Personal Jubilado B", and "Profesor Ordinario De Asignatura A Destituido". To the right of the dropdown, there are checkboxes for "Clase", "Nacionalidad", "Categoría", "Agrupación", "Pais", and "Categoría". Below the form, there are buttons for "Continuar" and "Cancelar". At the bottom of the page, there is a footer that reads: "COORDINACIÓN DE SERVICIOS DE CÓMPUTO, FACULTAD DE CIENCIAS" and "Última actualización: 02/12/2001". A small box in the bottom right corner contains the number "001417".

Figura. 6.47: Sistema SAFU. Modificación de Claves y Categorías

The screenshot shows a web browser window with the title 'SAFU2000 Modificación de Asignatura - Microsoft Internet Explorer'. The address bar shows the URL: 'http://admweb/ciencias.unam.mx/depce/acad/modif_asignatura.php?7sessesion=3c10c78d94c5f'. The main content area is titled 'Modificación de Asignatura' and contains the following form elements:

- A text input field for 'Nombre' with the value 'Acuicultura' and a dropdown arrow.
- A grid of checkboxes for selection:
 - Nombre
 - Clase
 - Nivel
 - Créditos
 - Horas Teóricas
 - Horas Prácticas
 - Plan de Estudios
 - Carrera
 - Semestre
- Buttons for 'Continuar' and 'Restablecer'.
- Footer text: 'COORDINACIÓN DE SERVICIOS DE COMPUTO, FACULTAD DE CIENCIAS' and 'Última actualización: 03/11/2001'.

Figura. 6.48: Sistema SAFU. Modificación de Asignaturas

The screenshot shows a web browser window with the title 'SAFU2000 Modificación de Sueldo - Microsoft Internet Explorer'. The address bar shows the URL: 'http://admweb/ciencias.unam.mx/depce/acad/modif_sueldos.php?7sessesion=3c10c78d94c5f'. The main content area is titled 'Modificación de Sueldo' and contains the following form elements:

- A dropdown menu for 'Sueldo (Número)' with the value '15348'.
- A grid of checkboxes for selection:
 - Sueldo (Número)
 - Sueldo (Letras)
 - Clase y Categoría
 - Horas
 - Fecha de Vigencia
- Buttons for 'Continuar' and 'Restablecer'.
- Footer text: 'COORDINACIÓN DE SERVICIOS DE COMPUTO, FACULTAD DE CIENCIAS' and 'Última actualización: 03/11/2001'.
- A small box containing the number '001547'.

Figura. 6.49: Sistema SAFU. Modificación de Sueldos

5/16/2000, Modificación de Horario - Microsoft Internet Explorer

http://sistemas.fciencias.unam.mx/compser/acad/proc01_horario.php?sesion=3c30c79d34c58

Modificación de Horario

Inicio:

Hora de inicio:

Hora de fin:

Selección de día de la semana:

COORDINACIÓN DE SERVICIOS DE CÓMPUTO, FACULTAD DE CIENCIAS

001537

Figura. 6.50: Sistema SAFU. Modificación de Horarios

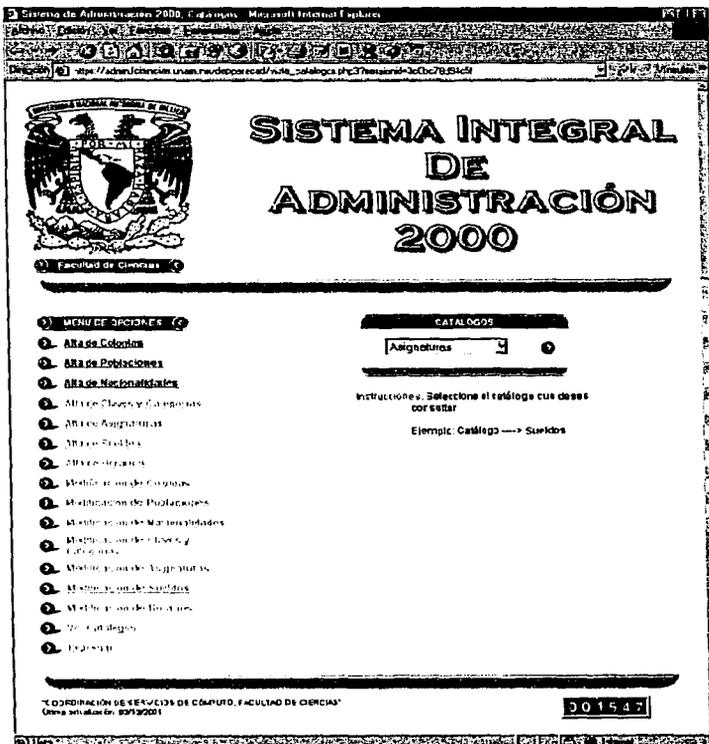


Figura. 6.51: Sistema SAFU. Ver Catálogos

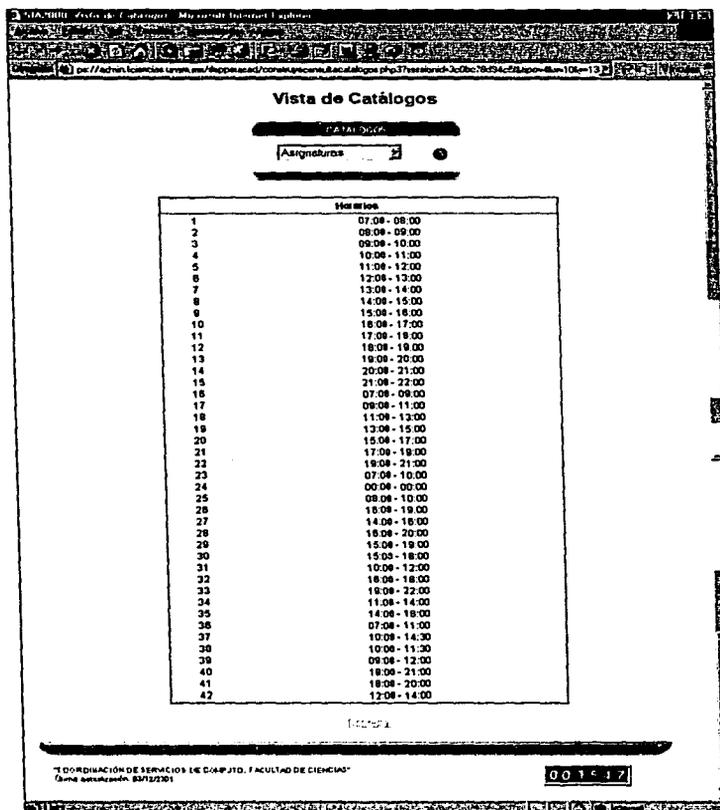


Figura. 6.52: Sistema SAFU. Ejecutando Ver Catálogos

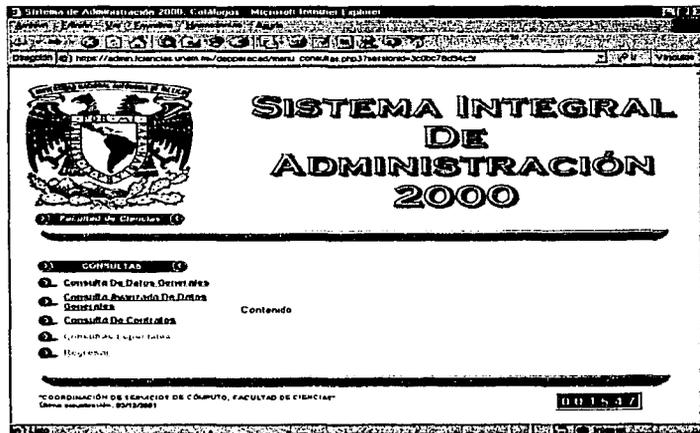


Figura. 6.53: Sistema SAFU. Menú Consultas

6.2 Plan de Pruebas

El plan de pruebas para el SAFU consiste en el levantamiento de las nóminas completas de los departamentos de Física, Matemáticas y Biología de la Facultad de Ciencias para el semestre 2002-I. Dentro de este planteamiento se involucran los siguientes puntos.

1. Alta de Datos Generales del nuevo personal académico.
2. Modificación de Datos Generales del personal académico ya existente.
3. Alta de Contratos del nuevo personal académico.
4. Alta de Contratos del personal académico existente.
5. Modificación, Eliminación e Impresión de los contratos académicos generados según convenga.
6. Alta de datos en los catálogos del sistema.
7. Modificación de datos en los catálogos del sistema.

Este plan de pruebas nos permitirá conocer el comportamiento del sistema bajo situaciones reales, así como también nos podremos dar cuenta de los posibles problemas en su funcionamiento y desempeño. Con él se evaluarán los siguientes aspectos:

- Validación de información.
- Comportamiento en transacciones (Altas, Modificaciones, Bajas).
- Calidad y tiempo de impresión de contratos.
- Funcionamiento general del sistema.
- Seguridad del sistema.

6.3 Problemas y Soluciones

Con base en los puntos de evaluación dentro del SAFU, se encontraron los siguientes problemas:

Problema. Tiempo de impresión de contratos.

Solución. Se aumentó la memoria física de la impresora con la que cuenta el personal del Departamento de Personal Académico, así como también se modificó la configuración de impresión en cada una de las máquinas de ese departamento. De igual forma se modificó la imagen base usada en el sistema para generar los contratos de forma visual.

Problema. Errores de mapeo de tipos de letras en la impresión de contratos.

Solución. Se modificó el tipo de letra "Helvetica" usado en la impresión de contratos a tipo de letra "Estandar".

Problema. Al realizar dos movimientos por contrato no se pueden especificar la horas teóricas y prácticas del segundo movimiento.

Solución. Se anexaron los campos necesarios en la forma de captura de los datos contractuales y en estructura de la base de datos del sistema para capturar las horas teóricas y prácticas del segundo movimiento en un contrato.

Problema. Al realizar alguna modificación en un contrato, se modifica en la imagen del contrato el tipo de solicitud del mismo.

Solución. Se corrigió la forma de modificación de contratos para evitar el cambio de tipo solicitud de contrato en la imagen generada.

Problema. Al modificar el expediente temporal del un empleado al expediente definitivo se pierde el vínculo de los contratos que se le generaron al empleado con el expediente temporal.

Solución. Se corrigió la forma de modificación de datos generales para evitar perder el vínculo entre el expediente temporal de un empleado y sus contratos con el nuevo expediente definitivo.

Conclusiones

Durante el desarrollo de este proyecto de tesis pude observar la importancia de aplicar una metodología en el ciclo de vida de un sistema. Esto me permitió trabajar con una mejor planeación y seguridad; minimizando riesgos y evitando demoras en las etapas de desarrollo del software. El sistema obtenido es un trabajo de calidad que cumple con los requerimientos establecidos para el desarrollo del proyecto.

A pesar de que las nuevas metodologías de ingeniería software van mejorando cada vez más y facilitan la creación de sistemas, estas son cada vez más complejas y se necesita de una amplia experiencia y preparación para poder utilizarlas adecuadamente. A pesar de esto, herramientas como UML ayudan de forma importante al desarrollo de los sistemas de una forma sencilla y organizada. Con esto nos podemos dar cuenta que sin importar el modelo utilizado para desarrollar un sistema, si este se aplica adecuadamente, se obtendrán sistemas de alta calidad, eficientes y que cumplan con los requerimientos y necesidades de los usuarios. Por otro lado las bases teóricas y el modelo de base de datos usado para implantar este sistema favoreció el desarrollo de un sistema robusto y seguro, lo que hace que el sistema construido sea vanguardista en cuanto a tecnología de información.

Finalmente puedo decir que el sistema desarrollado en este proyecto cubre las necesidades de los usuarios de forma clara y precisa. Sin embargo el estado actual del sistema muestra que todavía puede ser mejorado en muchos aspectos. Dentro de los aspectos que se pueden mejorar están el tiempo de respuesta y flujo de datos en las formas interactivas del sistema, las cuales a pesar de ser funcionales y correctas, necesitan ser más dinámicas y versátiles; esto se puede lograr utilizando nuevas tecnologías y algoritmos de programación que las hagan más sencillas e interactivas. También se puede reducir el uso del espacio físico en el servidor para el almacenamiento de contratos digitales, hacer más robusta la atomicidad en cada una de las transacciones que se lleven a cabo y tal vez el aspecto más importante a mejorar sería la conectividad con otros sistemas de la facultad, lo que permitiría realizar una mayor cantidad de trámites interdepartamentales en menos tiempo. Todos estos aspectos podría ser mejorados ya que día con día la tecnología, los nuevos lenguajes de programación y los modelos y técnicas para desarrollar sistemas de software se mejoran y actualizan con el paso del tiempo.

Apéndice A

Detalle de Casos de Uso

A continuación se presenta la parte complementaria del detalle de los casos de uso de la sección 4.3.8. Los casos de uso complementarios son los siguientes:

- Alta de Datos Generales (Seguridad Media)
- Modificación de Datos Generales (Seguridad Media)
- Consulta de Datos Generales (Seguridad Media)
- Alta de Colonias (Seguridad Media)
- Alta de Colonias (Seguridad Final)
- Alta de Poblaciones (Seguridad Media)
- Alta de Poblaciones (Seguridad Final)
- Alta de Nacionalidades (Seguridad Media)
- Alta de Nacionalidades (Seguridad Final)
- Alta de Contratos (Seguridad Media)
- Modificación de Contratos (Seguridad Media)
- Consulta de Contratos (Seguridad Media)
- Impresión de Contratos (Seguridad Media)
- Alta de Claves y Categorías (Seguridad Media)
- Alta de Asignaturas (Seguridad Media)
- Alta de Sueldos (Seguridad Media)
- Alta de Asignaturas (Seguridad Media)

- **Modificación de Colonias (Seguridad Media)**
- **Modificación de Poblaciones (Seguridad Media)**
- **Modificación de Nacionalidades (Seguridad Media)**
- **Modificación de Claves y Categorías (Seguridad Media)**
- **Modificación de Asignaturas (Seguridad Media)**
- **Modificación de Sueldos (Seguridad Media)**
- **Modificación de Horarios (Seguridad Media)**
- **Ver Catálogos (Seguridad Media)**
- **Consulta Avanzada de Datos Generales (Seguridad Media)**
- **Consultas Especiales (Seguridad Media)**

CASO DE USO: Alta de Datos Generales

IDENTIFICADOR: F1.1

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso de uso se agregan los datos generales de un nuevo empleado al sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Ingresar datos del nuevo empleado	2	Procesa datos del nuevo empleado
		3	Se agrega a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	Paso
E1	Expediente existente	Mensaje de Expediente incorrecto	2
E2	Expediente inválido	Mensaje de Expediente incorrecto	2
E3	Ap.Paterno inválido	Mensaje de Ap.Paterno incorrecto	2
E4	Ap.Materno inválido	Mensaje de Ap.Materno incorrecto	2
E5	Nombre1 inválido	Mensaje de Nombre1 incorrecto	2
E6	Nombre2 inválido	Mensaje de Nombre2 incorrecto	2
E7	Calle y Número inválido	Mensaje de Dirección incorrecta	2
E8	Código Postal inválido	Mensaje de Código Postal incorrecto	2
E9	RFC inválido	Mensaje de RFC incorrecto	2
E10	Teléfono inválido	Mensaje de Teléfono incorrecto	2
E11	Correo Electrónico inválido	Mensaje de Correo Electrónico incorrecto	2

CASO DE USO: Modificación de Datos Generales

IDENTIFICADOR: F1.2

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso de uso se modifican los datos generales de un empleado en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige los datos generales a modificar y empleado	2	Construye forma de datos generales a modificar
3	Ingresa los nuevos datos generales	4	Procesa los datos generales modificados
		5	Modifica los datos generales en la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E1	Expediente existente	Mensaje de Expediente incorrecto	4
E2	Expediente inválido	Mensaje de Expediente incorrecto	4
E3	Ap.Paterno inválido	Mensaje de Ap.Paterno incorrecto	4
E4	Ap.Materno inválido	Mensaje de Ap.Materno incorrecto	4
E5	Nombre1 inválido	Mensaje de Nombre1 incorrecto	4
E6	Nombre2 inválido	Mensaje de Nombre2 incorrecto	4
E7	Calle y Número inválido	Mensaje de Dirección incorrecta	4
E8	Código Postal inválido	Mensaje de Código Postal incorrecto	4
E9	RFC inválido	Mensaje de RFC incorrecto	4
E10	Teléfono inválido	Mensaje de Teléfono incorrecto	4
E11	Correo Electrónico inválido	Mensaje de Correo Electrónico incorrecto	4
E12	Expediente inexistente	Mensaje de Expediente incorrecto	1
E13	RFC inexistente	Mensaje de RFC incorrecto	1

CASO DE USO: Consulta de Datos Generales

IDENTIFICADOR: F1.3 = F4.1

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso de uso se consultan los datos generales de un empleado en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe el dato y elige el patrón a buscar	2	Busca el dato de acuerdo al patrón
		3	Presenta resultados de búsqueda en pantalla

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E14	Datos no encontrados	Mensaje de búsqueda de datos no encontrados	2

CASO DE USO: Alta de Colonias

IDENTIFICADOR: F1.4 = F3.1

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso se agrega una colonia nueva en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe la colonia nueva	2	Procesa la colonia nueva
		3	Se agrega la colonia nueva a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E15	Colonia existente	Mensaje de Colonia incorrecta	2
E16	Colonia inválida	Mensaje de Colonia incorrecta	2

CASO DE USO: Alta de Colonias

IDENTIFICADOR: F1.4 = F3.1

ACTORES: Profesor
Ayudante

DESCRIPCIÓN: En este caso se agrega una colonia nueva en el sistema.

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe la colonia nueva	2	Procesa la colonia nueva
		3	Se agrega la colonia nueva a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E15	Colonia existente	Mensaje de Colonia incorrecta	2
E16	Colonia inválida	Mensaje de Colonia incorrecta	2

CASO DE USO: Alta de Poblaciones

IDENTIFICADOR: F1.5 = F3.2

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso se agrega una población nueva en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe la población nueva	2	Procesa la población nueva
		3	Se agrega la población nueva a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E17	Población existente	Mensaje de Población incorrecta	2
E18	Población inválida	Mensaje de Población incorrecta	2

CASO DE USO: Alta de Poblaciones

IDENTIFICADOR: F1.5 = F3.2

ACTORES: Profesor
Ayudante

DESCRIPCIÓN: En este caso se agrega una población nueva en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe la población nueva	2	Procesa la población nueva
		3	Se agrega la población nueva a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E17	Población existente	Mensaje de Población incorrecta	2
E18	Población inválida	Mensaje de Población incorrecta	2

CASO DE USO: Alta de Nacionalidades

IDENTIFICADOR: F1.6 = F3.3

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso se agrega una nacionalidad nueva en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe la nacionalidad nueva	2	Procesa la nacionalidad nueva
		3	Se agrega la nacionalidad nueva a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E19	Abreviatura existente	Mensaje de Abreviatura incorrecta	2
E20	Abreviatura inválida	Mensaje de Abreviatura incorrecta	2
E21	País existente	Mensaje de País incorrecto	2
E22	País inválido	Mensaje de País incorrecto	2
E23	Nacionalidad existente	Mensaje de Nacionalidad incorrecta	2
E24	Nacionalidad inválida	Mensaje de Nacionalidad incorrecta	2

CASO DE USO: Alta de Nacionalidades

IDENTIFICADOR: F1.6 = F3.3

ACTORES: Profesor
Ayudante

DESCRIPCIÓN: En este caso se agrega una nacionalidad nueva en el sistema.

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe la nacionalidad nueva	2	Procesa la nacionalidad nueva
		3	Se agrega la nacionalidad nueva a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E19	Abreviatura existente	Mensaje de Abreviatura incorrecta	2
E20	Abreviatura inválida	Mensaje de Abreviatura incorrecta	2
E21	País existente	Mensaje de País incorrecto	2
E22	País inválido	Mensaje de País incorrecto	2
E23	Nacionalidad existente	Mensaje de Nacionalidad incorrecta	2
E24	Nacionalidad inválida	Mensaje de Nacionalidad incorrecta	2

CASO DE USO: Alta de Contratos

IDENTIFICADOR: F2.1

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso de uso se agregan los datos contractuales de un empleado al sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Ingresar datos del nuevo contrato	2	Procesa datos del nuevo contrato
		3	Se agrega a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E25	Oficio existente	Mensaje de Oficio incorrecto	2
E26	Oficio inválido	Mensaje de Oficio incorrecto	2
E27	Solicitud inválida	Mensaje de Solicitud incorrecta	2
E28	Causa inválida	Mensaje de Causa incorrecta	2
E29	Num.Plaza inválida	Mensaje de Num.Plaza incorrecta	2
E30	Hor.Teóricas inválidas	Mensaje de Hor.Teóricas incorrectas	2
E31	Hor.Prácticas inválidas	Mensaje de Hor.Prácticas incorrectas	2
E32	Fecha Inicio de Contrato inválida	Mensaje de Fecha Inicio de Contrato incorrecta	2
E33	Fecha Fin de Contrato inválida	Mensaje de Fecha Fin de Contrato incorrecta	2
E34	Código Programático inválido	Mensaje de Código Programático incorrecto	2
E35	Horario inválido	Mensaje de Horario inválido	2

CASO DE USO: Modificación de Contratos

IDENTIFICADOR: F2.2

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso de uso se modifican los datos contractuales de un empleado en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige los datos contractuales a modificar	2	Construye forma de datos contractuales a modificar
3	Ingresa los nuevos datos contractuales	4	Procesa los datos contractuales modificados
		5	Modifica los datos contractuales en la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E25	Oficio existente	Mensaje de Oficio incorrecto	4
E26	Oficio inválido	Mensaje de Oficio incorrecto	4
E27	Solicitud inválida	Mensaje de Solicitud incorrecta	4
E28	Causa inválida	Mensaje de Causa incorrecta	4
E29	Num.Plaza inválida	Mensaje de Num.Plaza incorrecta	4
E30	Hor.Teóricas inválidas	Mensaje de Hor.Teóricas incorrectas	4
E31	Hor.Prácticas inválidas	Mensaje de Hor.Prácticas incorrectas	4
E32	Fecha Inicio de Contrato inválida	Mensaje de Fecha Inicio de Contrato incorrecta	4
E33	Fecha Fin de Contrato inválida	Mensaje de Fecha Fin de Contrato incorrecta	4
E34	Código Programático inválido	Mensaje de Código Programático incorrecto	4
E35	Horario inválido	Mensaje de Horario inválido	4
E12	Expediente inexistente	Mensaje de Expediente incorrecto	1
E13	RFC inexistente	Mensaje de RFC incorrecto	1

CASO DE USO: Consulta de Contratos

IDENTIFICADOR: F2.4 = F4.3

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso de uso se consultan los contratos de un empleado del sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe el dato y elige el patrón a buscar	2	Busca contratos de acuerdo al dato y al patrón
		3	Presenta resultados de búsqueda en pantalla
4	Escoge contrato a consultar	5	Presenta contrato en pantalla

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E36	Contratos no encontrados	Mensaje de búsqueda de contratos no encontrados	2

CASO DE USO: Impresión de Contratos

IDENTIFICADOR: F2.5

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso de uso se imprimen los contratos de un empleado del sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe el dato y elige el patrón a buscar	2	Busca contratos de acuerdo al dato y al patrón
		3	Presenta resultados de búsqueda en pantalla
4	Escoge contrato a consultar	5	Presenta contrato en pantalla
6	Imprime contrato		

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E36	Contratos no encontrados	Mensaje de búsqueda de contratos no encontrados	2

CASO DE USO: Alta de Claves y Categorías

IDENTIFICADOR: F2.6 = F3.4

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso se agrega una clave y categoría nueva en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe los datos de la clave y categoría nueva	2	Procesa los datos de la clave y categoría nueva
		3	Se agrega la clave y categoría nueva a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E37	Clave y Categoría existente	Mensaje de Clave y Categoría incorrecta	2

CASO DE USO: Alta de Asignaturas

IDENTIFICADOR: F2.7 = F3.5

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso se agrega una asignatura nueva en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe los datos de la nueva	2	Procesa los datos de la asignatura nueva
		3	Se agrega la asignatura nueva a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E38	Clave existente	Mensaje de Clave incorrecta	2
E39	Clave incorrecta	Mensaje de Clave incorrecta	2
E40	Nombre incorrecto	Mensaje de Nombre incorrecto	2
E41	Créditos incorrectos	Mensaje de Créditos incorrectos	2
E30	Hor.Teóricas inválidas	Mensaje de Hor.Teóricas incorrectas	2
E31	Hor.Prácticas inválidas	Mensaje de Hor.Prácticas incorrectas	2

CASO DE USO: Alta de Sueldos

IDENTIFICADOR: F2.8 = F3.6

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso se agrega un sueldo nuevo en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe los datos del sueldo nuevo	2	Procesa los datos del sueldo nuevo
		3	Se agrega el sueldo nuevo a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E42	Sueldo existente	Mensaje de Sueldo incorrecto	2
E43	Sueldo incorrecto	Mensaje de Sueldo incorrecto	2
E44	Vigencia incorrecta	Mensaje de Vigencia incorrecta	2

CASO DE USO: Alta de Horarios

IDENTIFICADOR: F2.9 = F3.7

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso se agrega un horario nuevo en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe el horario nuevo	2	Procesa el horario nuevo
		3	Se agrega el horario nuevo a la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E45	Horario existente	Mensaje de Horario incorrecto	2
E46	Horario incorrecto	Mensaje de Horario incorrecto	2

CASO DE USO: Modificación de Colonias

IDENTIFICADOR: F3.8

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso se modifica una colonia en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige la colonia a modificar	2	Construye la forma de la colonia a modificar
3	Modifica la colonia	4	Procesa la colonia modificada
		5	Modifica la colonia en la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E15	Colonia existente	Mensaje de Colonia incorrecta	2
E16	Colonia incorrecta	Mensaje de Colonia incorrecta	2

CASO DE USO: Modificación de Poblaciones

IDENTIFICADOR: F3.9

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso se modifica una población en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige la población a modificar	2	Construye la forma de la población a modificar
3	Modifica la población	4	Procesa la población modificada
		5	Modifica la población en la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E17	Población existente	Mensaje de Población incorrecta	2
E18	Población incorrecta	Mensaje de Población incorrecta	2

CASO DE USO: Modificación de Nacionalidades

IDENTIFICADOR: F3.10

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso se modifica una nacionalidad en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige la nacionalidad y los datos a modificar	2	Construye la forma de la nacionalidad a modificar
3	Modifica los datos de la nacionalidad	4	Procesa los datos de la nacionalidad modificada
		5	Modifica los datos de la nacionalidad en la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E19	Abreviatura existente	Mensaje de Abreviatura incorrecta	4
E20	Abreviatura inválida	Mensaje de Abreviatura incorrecta	4
E21	País existente	Mensaje de País incorrecto	4
E22	País inválido	Mensaje de País incorrecto	4
E23	Nacionalidad existente	Mensaje de Nacionalidad incorrecta	4
E24	Nacionalidad inválida	Mensaje de Nacionalidad incorrecta	4

CASO DE USO: Modificación de Claves y Categorías

IDENTIFICADOR: F3.11

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso se modifica una clave y categoría en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige la clave y categoría a modificar	2	Construye la forma de la clave y categoría a modificar
3	Modifica la clave y categoría	4	Procesa la clave y categoría modificada
		5	Modifica la clave y categoría en la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E37	Clave y Categoría existente	Mensaje de Clave y Categoría incorrecta	4

CASO DE USO: Modificación de Asignaturas

IDENTIFICADOR: F3.12

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso se modifica una asignatura en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige los datos de la asignatura a modificar	2	Construye la forma de la asignatura a modificar
3	Modifica los datos de la asignatura	4	Procesa los datos de la asignatura modificada
		5	Modifica la asignatura en la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E38	Clave existente	Mensaje de Clave incorrecta	4
E39	Clave incorrecta	Mensaje de Clave incorrecta	4
E40	Nombre incorrecto	Mensaje de Nombre incorrecto	4
E41	Créditos incorrectos	Mensaje de Créditos incorrectos	4
E30	Hor. Teóricas inválidas	Mensaje de Hor. Teóricas incorrectas	4
E31	Hor. Prácticas inválidas	Mensaje de Hor. Prácticas incorrectas	4

CASO DE USO: Modificación de Sueldos

IDENTIFICADOR: F3.13

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso se modifica un sueldo en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige los datos del sueldo a modificar	2	Construye la forma del sueldo a modificar
3	Modifica los datos del sueldo	4	Procesa los datos del sueldo modificado
		5	Modifica el sueldo en la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E42	Sueldo existente	Mensaje de Sueldo incorrecto	4
E43	Sueldo incorrecto	Mensaje de Sueldo incorrecto	4
E44	Vigencia incorrecta	Mensaje de Vigencia incorrecta	4

CASO DE USO: Modificación de Horarios

IDENTIFICADOR: F3.14

ACTORES: Usuario del SAFU

DESCRIPCIÓN: En este caso se modifica un horario en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige el horario a modificar	2	Construye la forma del horario a modificar
3	Modifica el horario	4	Procesa el horario modificado
		5	Modifica el horario en la base de datos

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E45	Horario existente	Mensaje de Horario incorrecto	4
E46	Horario incorrecto	Mensaje de Horario incorrecto	4

CASO DE USO: Ver Catálogos

IDENTIFICADOR: F3.15

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso se consultan con detalle cada catálogo en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Elige el catálogo	2	Muestra el catálogo elegido

EXCEPCIONES: Ninguna

CASO DE USO: Consulta Avanzada de Datos Generales

IDENTIFICADOR: F4.2

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso de uso se consultan de forma más detallada los datos generales de un empleado en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe el dato y elige los patrones a buscar	2	Busca los posibles datos que se aproximen de acuerdo a los patrones
		3	Presenta resultados de búsqueda en pantalla

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E14	Datos no encontrados	Mensaje de búsqueda de datos no encontrados	2

CASO DE USO: Consultas Especiales

IDENTIFICADOR: F4.4

ACTORES: Usuario del DPA

DESCRIPCIÓN: En este caso de uso se consultan los datos generales y los contratos de cada empleado de forma específica en el sistema

FLUJO:

ACTOR		SISTEMA	
PASO	ACCIÓN	PASO	ACCIÓN
1	Escribe los dato y elige los patrones a buscar	2	Busca los posibles datos y contratos que se aproximen de acuerdo a los patrones
		3	Presenta resultados de búsqueda en pantalla

EXCEPCIONES:

ID	NOMBRE	ACCIÓN	PASO
E14	Datos no encontrados	Mensaje de búsqueda de datos no encontrados	2
E36	Contratos no encontrados	Mensaje de búsqueda de contratos no encontrados	2
E47	Datos y Contratos no encontrados	Mensaje de búsqueda de datos y contratos no encontrados	2

Apéndice B

Diagramas de Secuencia

A continuación se presenta la parte complementaria de los diagramas de secuencia de la sección 5.7.2. Los diagramas de secuencia complementarios son los siguientes:

- Diagrama de Secuencia F1.1 (Seguridad Media)
- Diagrama de Secuencia F1.2 (Seguridad Media)
- Diagrama de Secuencia F1.3 (Seguridad Media)
- Diagrama de Secuencia F1.4 (Seguridad Media)
- Diagrama de Secuencia F1.4 (Seguridad Final)
- Diagrama de Secuencia F1.5 (Seguridad Media)
- Diagrama de Secuencia F1.5 (Seguridad Final)
- Diagrama de Secuencia F1.6 (Seguridad Media)
- Diagrama de Secuencia F1.6 (Seguridad Final)
- Diagrama de Secuencia F2.1 (Seguridad Media)
- Diagrama de Secuencia F2.2 (Seguridad Media)
- Diagrama de Secuencia F2.4 (Seguridad Media)
- Diagrama de Secuencia F2.5 (Seguridad Media)
- Diagrama de Secuencia F2.6 (Seguridad Media)
- Diagrama de Secuencia F2.7 (Seguridad Media)
- Diagrama de Secuencia F2.8 (Seguridad Media)
- Diagrama de Secuencia F2.9 (Seguridad Media)

- Diagrama de Secuencia F3.1 (Seguridad Media)
- Diagrama de Secuencia F3.1 (Seguridad Final)
- Diagrama de Secuencia F3.2 (Seguridad Media)
- Diagrama de Secuencia F3.2 (Seguridad Final)
- Diagrama de Secuencia F3.3 (Seguridad Media)
- Diagrama de Secuencia F3.3 (Seguridad Final)
- Diagrama de Secuencia F3.4 (Seguridad Media)
- Diagrama de Secuencia F3.5 (Seguridad Media)
- Diagrama de Secuencia F3.6 (Seguridad Media)
- Diagrama de Secuencia F3.7 (Seguridad Media)
- Diagrama de Secuencia F3.8 (Seguridad Media)
- Diagrama de Secuencia F3.9 (Seguridad Media)
- Diagrama de Secuencia F3.10 (Seguridad Media)
- Diagrama de Secuencia F3.11 (Seguridad Media)
- Diagrama de Secuencia F3.12 (Seguridad Media)
- Diagrama de Secuencia F3.13 (Seguridad Media)
- Diagrama de Secuencia F3.14 (Seguridad Media)
- Diagrama de Secuencia F3.15 (Seguridad Media)
- Diagrama de Secuencia F4.1 (Seguridad Media)
- Diagrama de Secuencia F4.2 (Seguridad Media)
- Diagrama de Secuencia F4.3 (Seguridad Media)
- Diagrama de Secuencia F4.4 (Seguridad Media)
- Diagrama de Secuencia F5 (Seguridad Media)
- Diagrama de Secuencia F5 (Seguridad Final)

DIAGRAMAS DE SECUENCIA ALTA DE DATOS GENERALES

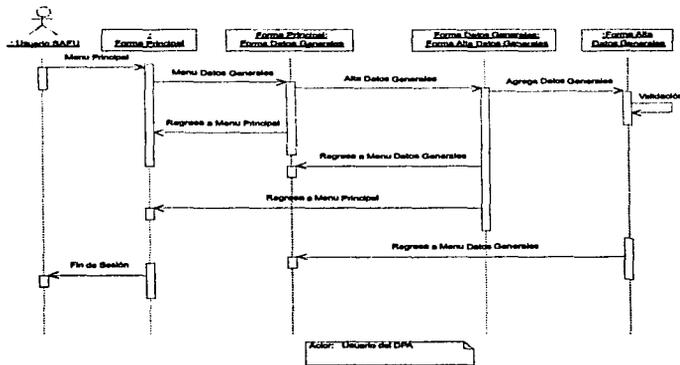


Figura. B.1: Diagrama de Secuencia F1.1 Seguridad Media

DIAGRAMAS DE SECUENCIA MODIFICACIÓN DE DATOS GENERALES

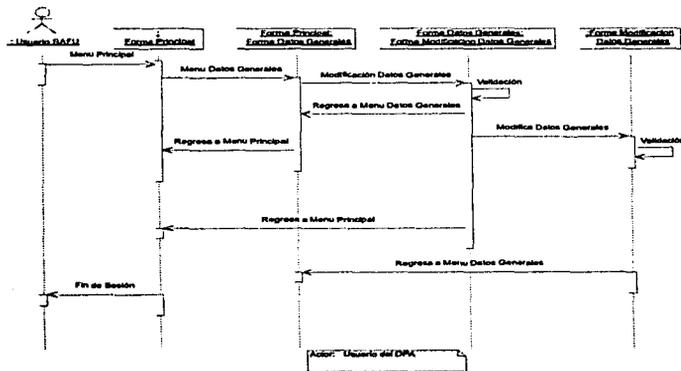


Figura. B.2: Diagrama de Secuencia F1.2 Seguridad Media

DIAGRAMAS DE SECUENCIA CONSULTA DE DATOS GENERALES

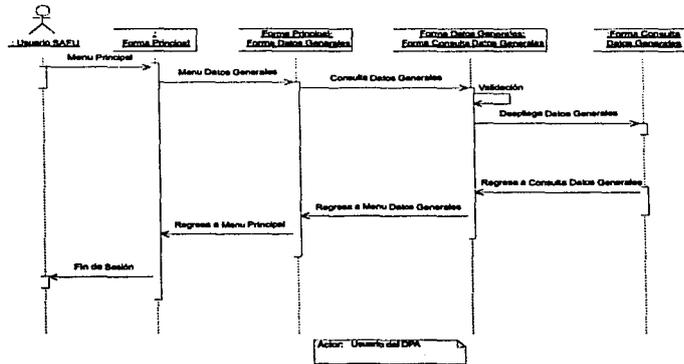


Figura. B.3: Diagrama de Secuencia F1.3 Seguridad Media

DIAGRAMAS DE SECUENCIA ALTA DE COLONIAS

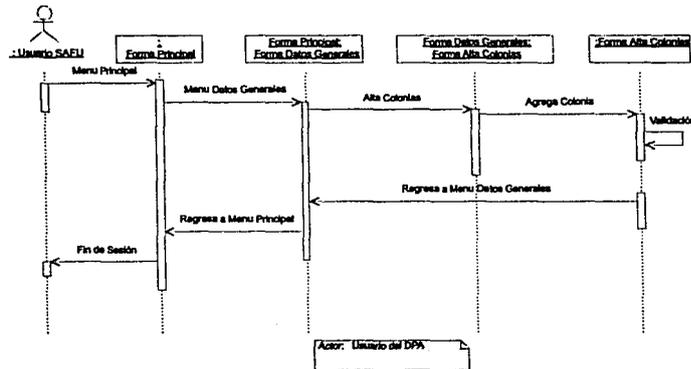


Figura. B.4: Diagrama de Secuencia F1.4 Seguridad Media

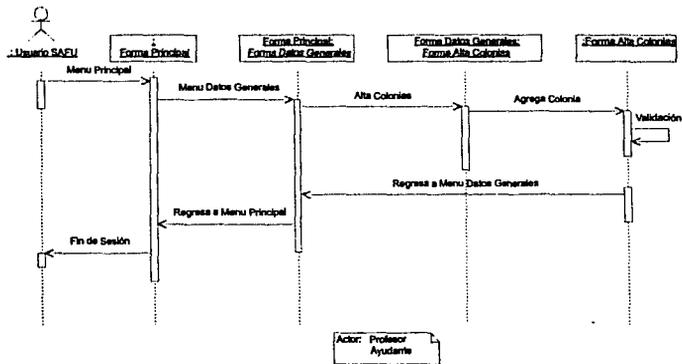


Figura. B.5: Diagrama de Secuencia F1.4 Seguridad Final

DIAGRAMAS DE SECUENCIA ALTA DE POBLACIONES

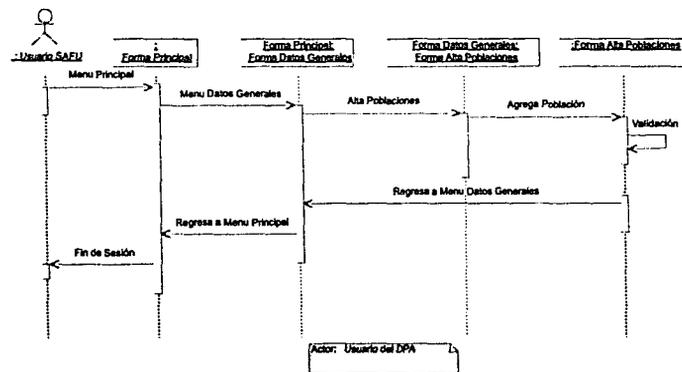


Figura. B.6: Diagrama de Secuencia F1.5 Seguridad Media

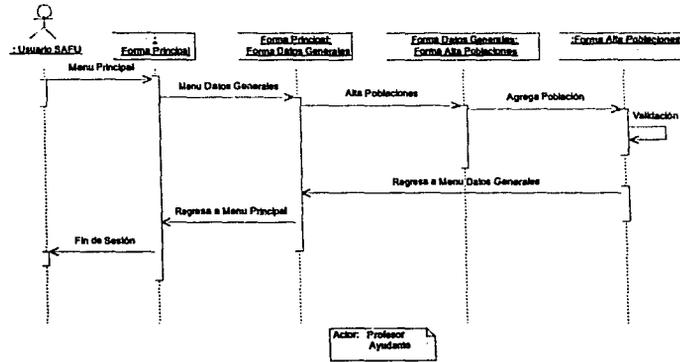


Figura. B.7: Diagrama de Secuencia F1.5 Seguridad Final

DIAGRAMAS DE SECUENCIA ALTA DE NACIONALIDADES

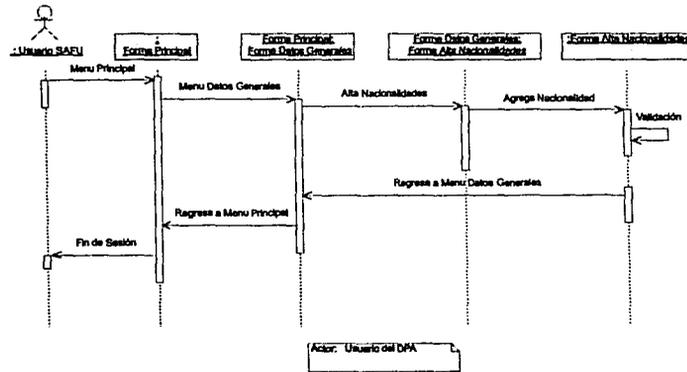


Figura. B.8: Diagrama de Secuencia F1.6 Seguridad Media

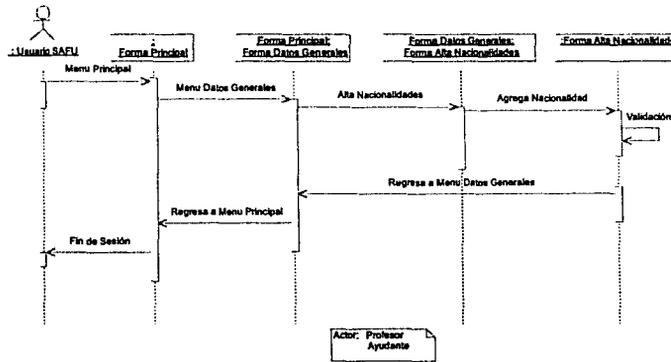


Figura. B.9: Diagrama de Secuencia F1.6 Seguridad Final

DIAGRAMAS DE SECUENCIA ALTA DE CONTRATOS

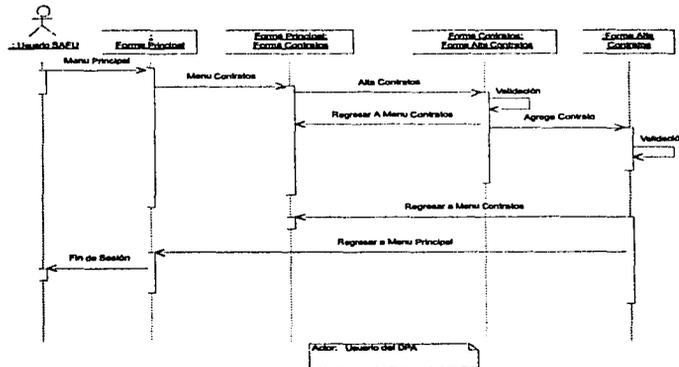


Figura. B.10: Diagrama de Secuencia F2.1 Seguridad Media

DIAGRAMAS DE SECUENCIA MODIFICACIÓN DE CONTRATOS

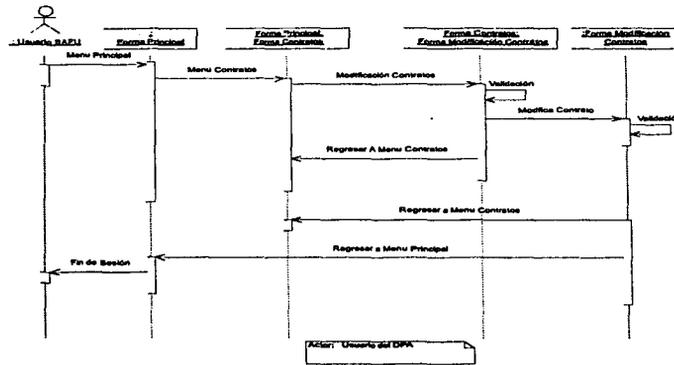


Figura. B.11: Diagrama de Secuencia F2.2 Seguridad Media

DIAGRAMAS DE SECUENCIA CONSULTA DE CONTRATOS

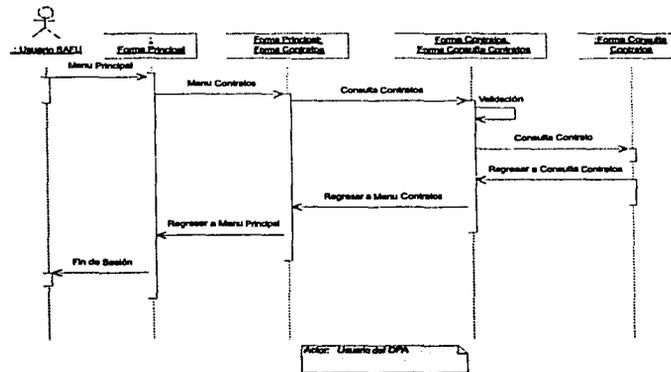


Figura. B.12: Diagrama de Secuencia F2.4 Seguridad Media

DIAGRAMAS DE SECUENCIA IMPRESIÓN DE CONTRATOS

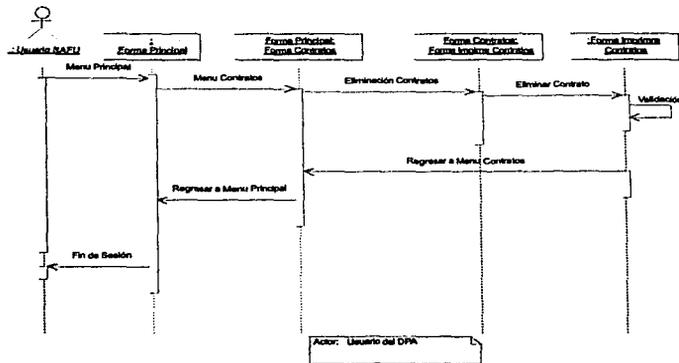


Figura. B.13: Diagrama de Secuencia F2.5 Seguridad Media

DIAGRAMAS DE SECUENCIA ALTA DE CLAVES Y CATEGORÍAS

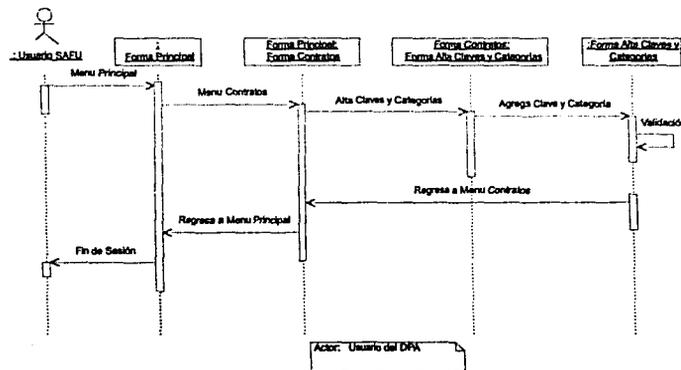


Figura. B.14: Diagrama de Secuencia F2.6 Seguridad Media

DIAGRAMAS DE SECUENCIA ALTA DE ASIGNATURAS

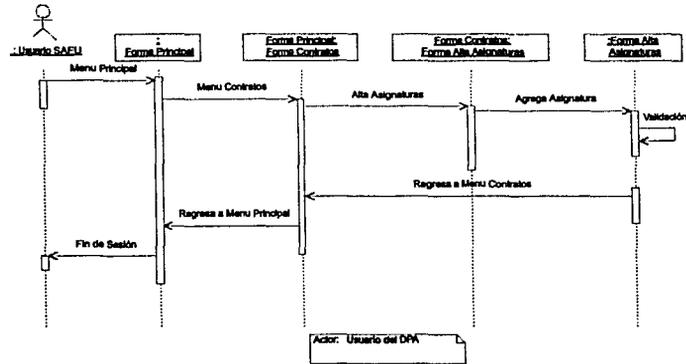


Figura. B.15: Diagrama de Secuencia F2.7 Seguridad Media

DIAGRAMAS DE SECUENCIA ALTA DE SUELDOS

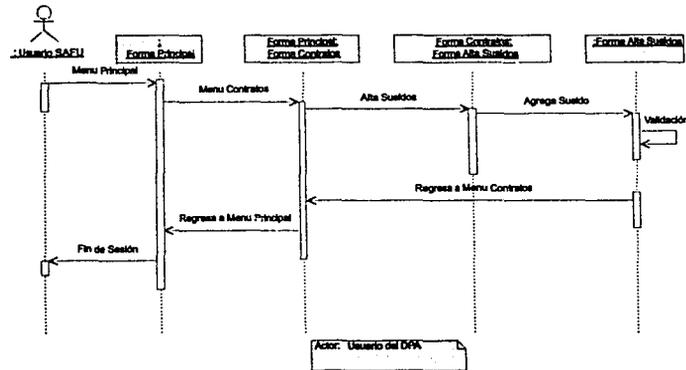


Figura. B.16: Diagrama de Secuencia F2.8 Seguridad Media

DIAGRAMAS DE SECUENCIA ALTA DE HORARIOS

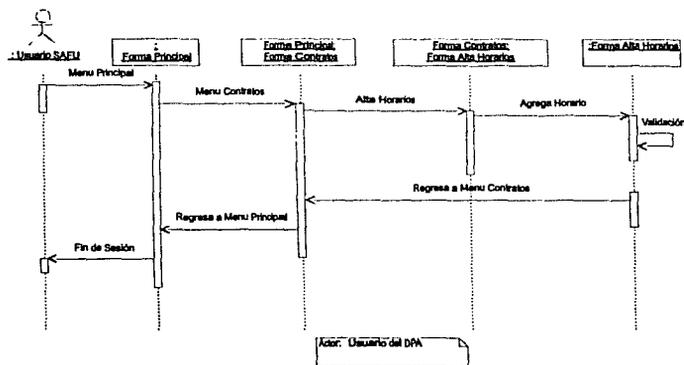


Figura. B.17: Diagrama de Secuencia F2.9 Seguridad Media

DIAGRAMAS DE SECUENCIA ALTA DE COLONIA

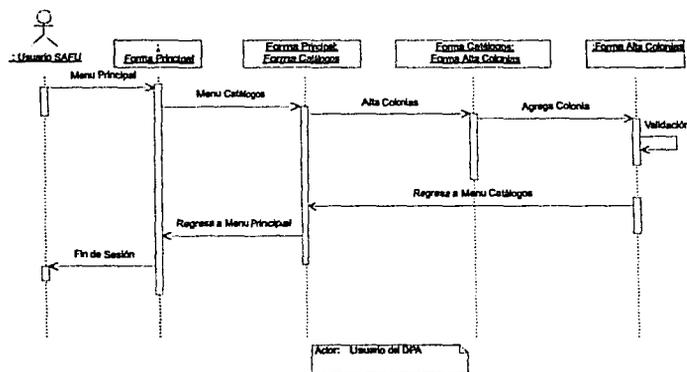


Figura. B.18: Diagrama de Secuencia F3.1 Seguridad Media

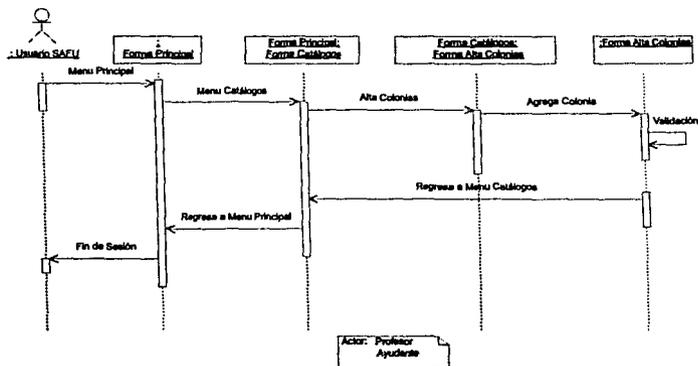


Figura. B.19: Diagrama de Secuencia F3.1 Seguridad Final

DIAGRAMAS DE SECUENCIA ALTA DE POBLACIONES

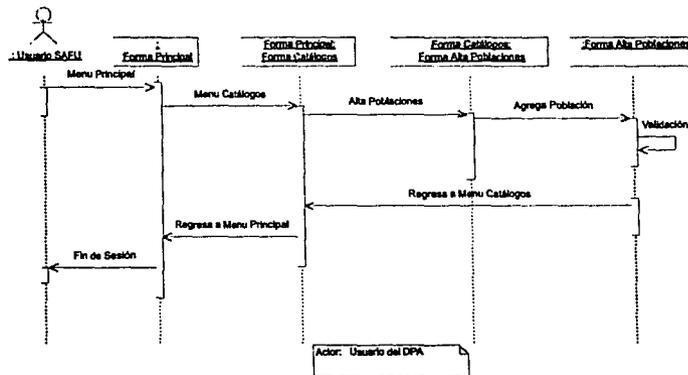


Figura. B.20: Diagrama de Secuencia F3.2 Seguridad Media

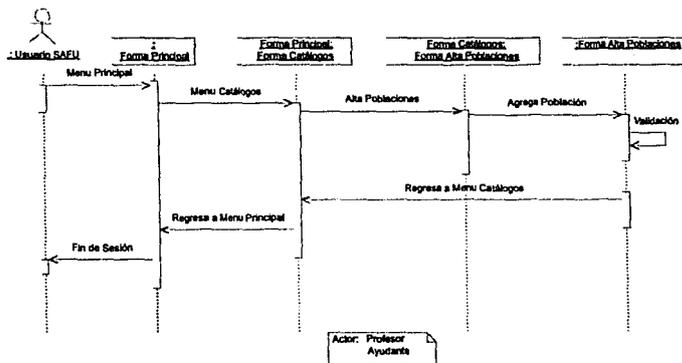


Figura. B.21: Diagrama de Secuencia F3.2 Seguridad Final

DIAGRAMAS DE SECUENCIA ALTA DE NACIONALIDADES

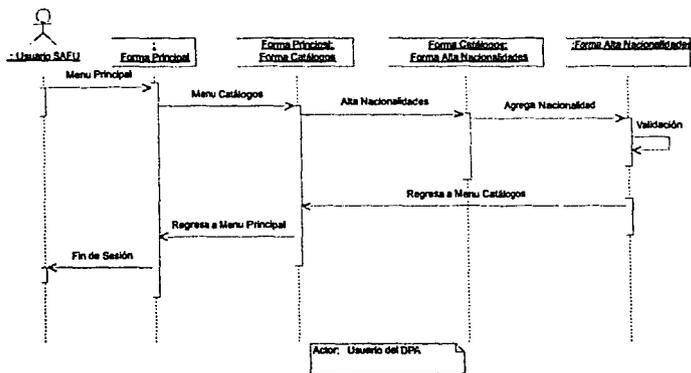


Figura. B.22: Diagrama de Secuencia F3.3 Seguridad Media

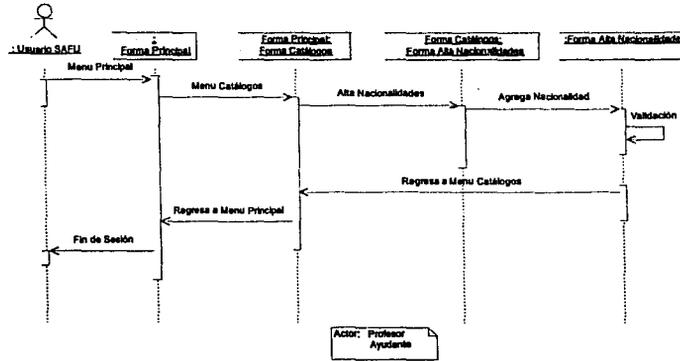


Figura. B.23: Diagrama de Secuencia F3.3 Seguridad Final

DIAGRAMAS DE SECUENCIA ALTA DE CLAVES Y CATEGORÍAS

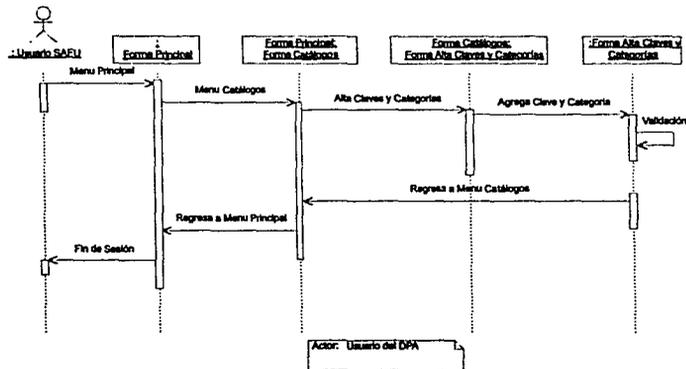


Figura. B.24: Diagrama de Secuencia F3.4 Seguridad Media

DIAGRAMAS DE SECUENCIA ALTA DE ASIGNATURAS

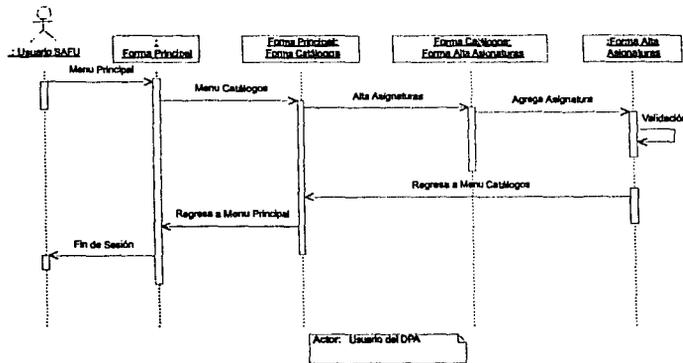


Figura. B.25: Diagrama de Secuencia F3.5 Seguridad Media

DIAGRAMAS DE SECUENCIA ALTA DE SUELDOS

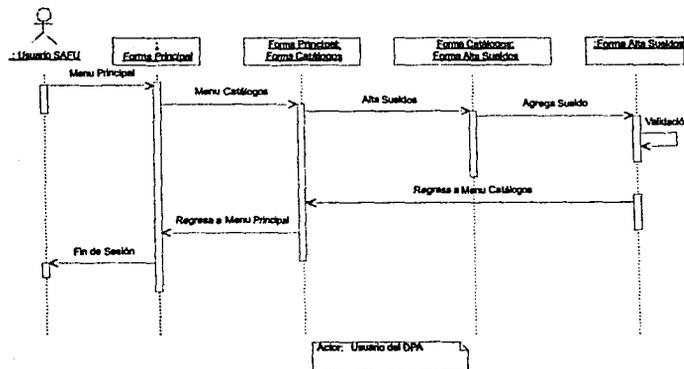


Figura. B.26: Diagrama de Secuencia F3.6 Seguridad Media

DIAGRAMAS DE SECUENCIA ALTA DE HORARIOS

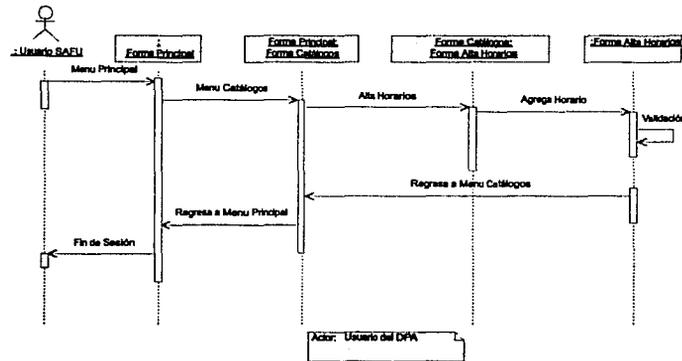


Figura. B.27: Diagrama de Secuencia F3.7 Seguridad Media

DIAGRAMAS DE SECUENCIA MODIFICACIÓN DE COLONIAS

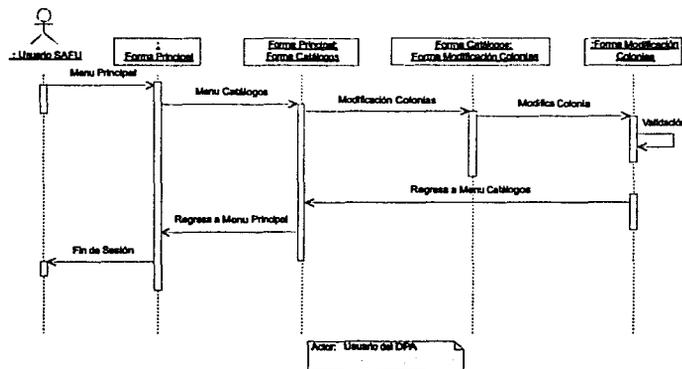


Figura. B.28: Diagrama de Secuencia F3.8 Seguridad Media

DIAGRAMAS DE SECUENCIA MODIFICACIÓN DE POBLACIONES

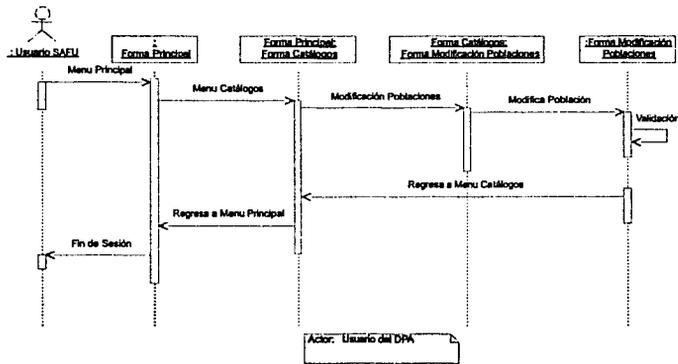


Figura. B.29: Diagrama de Secuencia F3.9 Seguridad Media

DIAGRAMAS DE SECUENCIA MODIFICACIÓN DE NACIONALIDADES

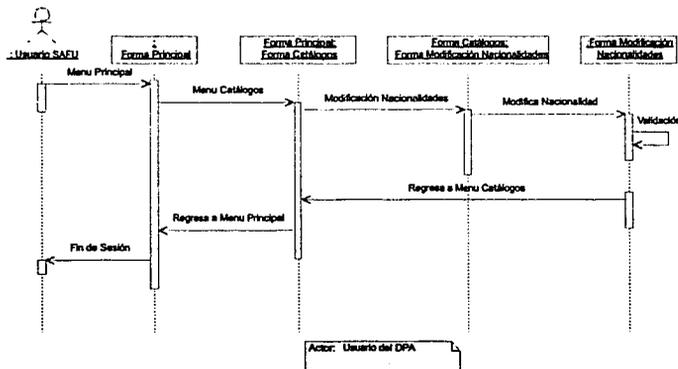


Figura. B.30: Diagrama de Secuencia F3.10 Seguridad Media

DIAGRAMAS DE SECUENCIA MODIFICACIÓN DE CLAVES Y CATEGORÍAS

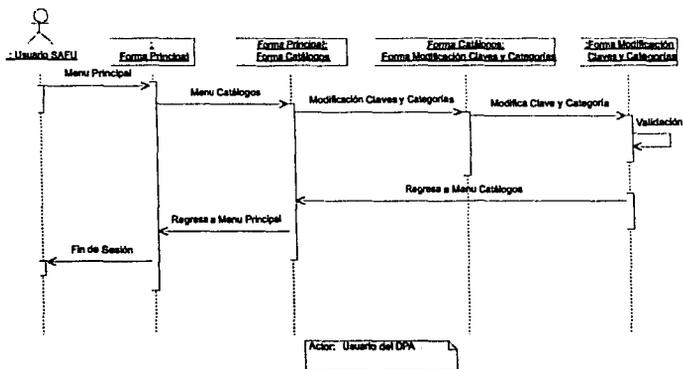


Figura. B.31: Diagrama de Secuencia F3.11 Seguridad Media

DIAGRAMAS DE SECUENCIA MODIFICACIÓN DE ASIGNATURAS

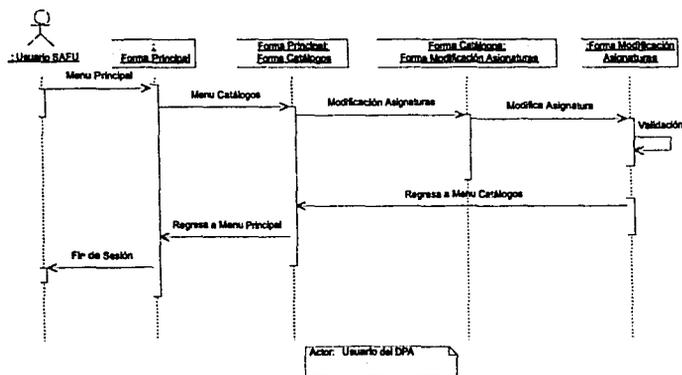


Figura. B.32: Diagrama de Secuencia F3.12 Seguridad Media

DIAGRAMAS DE SECUENCIA MODIFICACIÓN DE SUELDOS

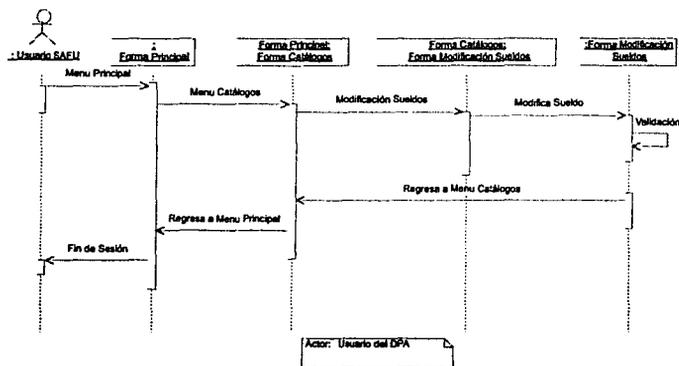


Figura. B.33: Diagrama de Secuencia F3.13 Seguridad Media

DIAGRAMAS DE SECUENCIA MODIFICACIÓN DE HORARIOS

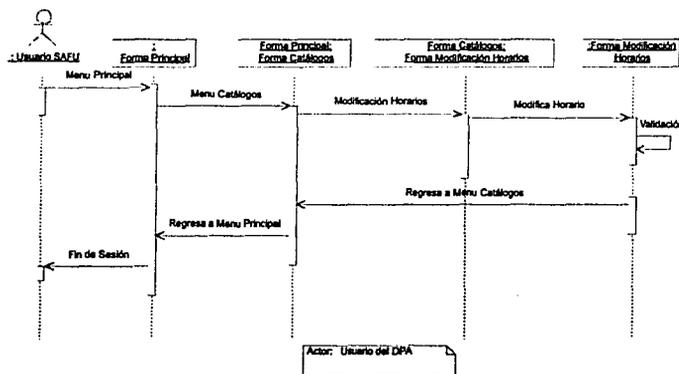


Figura. B.34: Diagrama de Secuencia F3.14 Seguridad Media

DIAGRAMAS DE SECUENCIA VER CATÁLOGOS

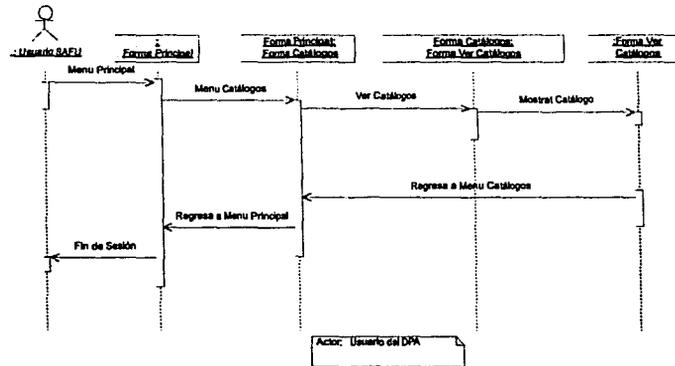


Figura. B.35: Diagrama de Secuencia F3.15 Seguridad Media

DIAGRAMAS DE SECUENCIA CONSULTA DE DATOS GENERALES

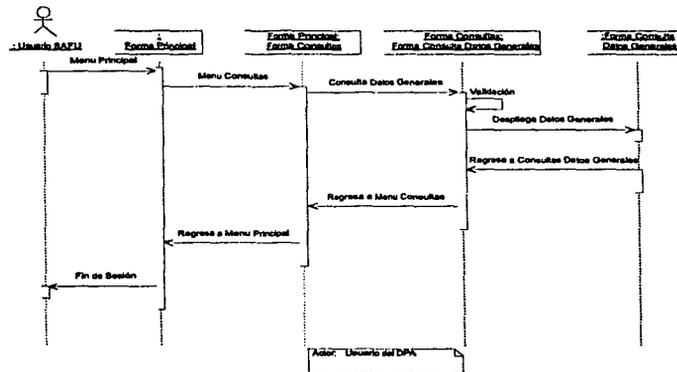


Figura. B.36: Diagrama de Secuencia F4.1 Seguridad Media

DIAGRAMAS DE SECUENCIA CONSULTA DE AVANZADA DE DATOS GENERALES

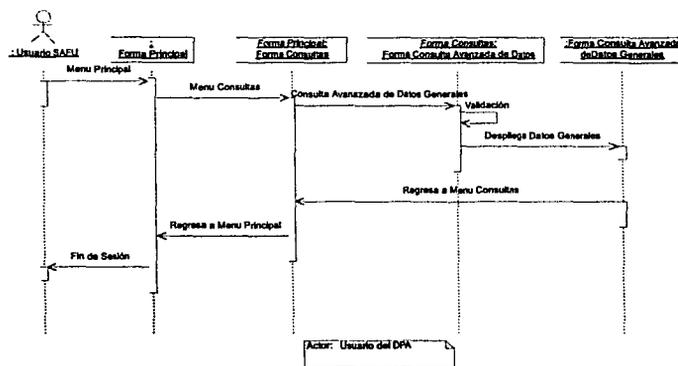


Figura. B.37: Diagrama de Secuencia F4.2 Seguridad Media

DIAGRAMAS DE SECUENCIA CONSULTA DE CONTRATOS

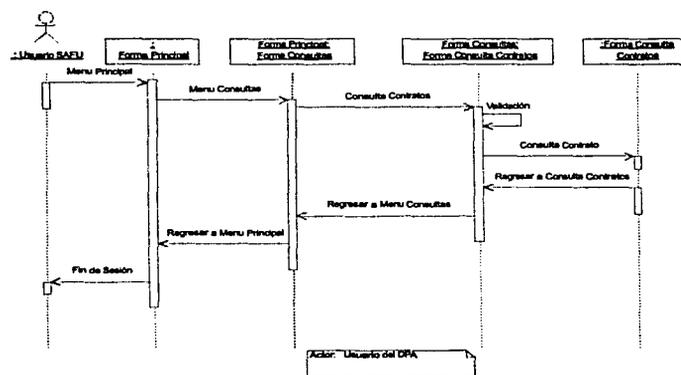


Figura. B.38: Diagrama de Secuencia F4.3 Seguridad Media

DIAGRAMAS DE SECUENCIA CONSULTAS ESPECIALES

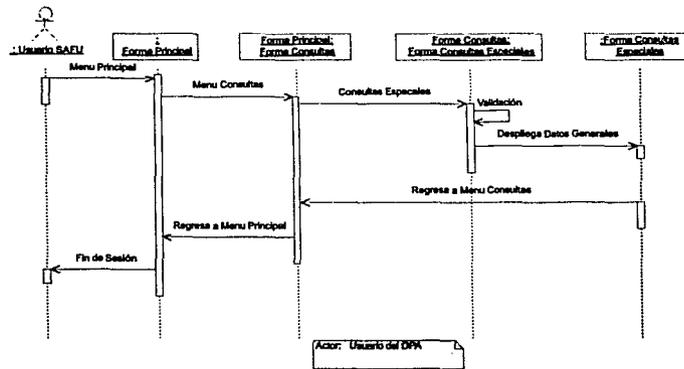


Figura. B.39: Diagrama de Secuencia F4.4 Seguridad Media

DIAGRAMAS DE SECUENCIA INFORMACIÓN

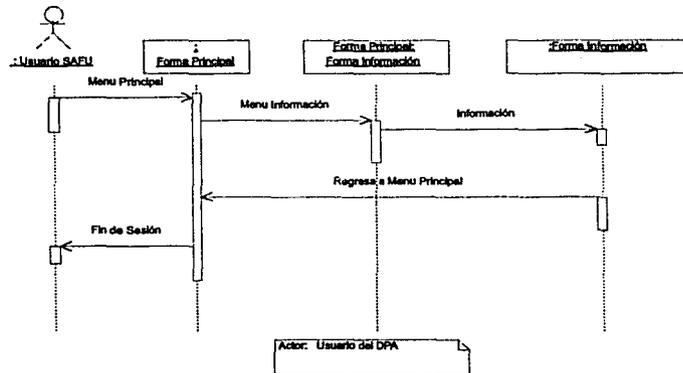


Figura. B.40: Diagrama de Secuencia F5 Seguridad Media

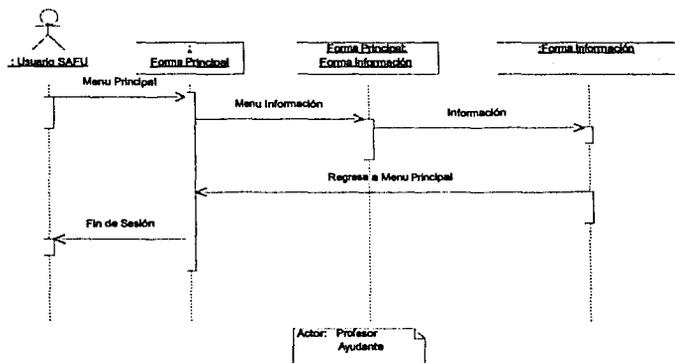


Figura. B.41: Diagrama de Secuencia F5 Seguridad Final

Bibliografía

- [1] *Enciclopedia del Estudiante* (CD-ROM). Salvat Editores S.A., Barcelona (España), última edición, 2000, ISBN 84-345-8277-5.
- [2] Albert GYR. *Nuevo Diccionario de la Computación e Internet* (Edición: bilingüe inglés-español), Trillas, México, segunda edición, 1999, ISBN 9682457823.
- [3] John P. van Gigch. *Teoría General de Sistemas*, Trillas, México, segunda edición, 1987, ISBN 968-24-2023-7.
- [4] Pankoj Jalote. *An Integrated Approach to Software Engineering*, Spring-Verlag, USA, primera edición, 1991, ISBN 0-387-97561-6.
- [5] *Enciclopedia Microsoft Encarta 2002* (CD-ROM). Microsoft Corporation, USA, 2001, última edición, ASIN B00005MF7S.
- [6] Abraham Silberschatz, Henry F. Korth, S. Sudarshan. *Database System Concepts*, McGraw-Hill, USA, tercera edición, 1997, ISBN 0-07-044756-X.
- [7] Angel Lucas Gómez. *Diseño y Gestión de Sistemas de Bases de Datos*, Paraninfo, Madrid (España), primera edición, 1993, ISBN 84-283-2011-X.
- [8] Anneliese von Mayrhauser. *Software Engineering: Methods and Management*, Academic Press Inc., USA, primera edición, 1990, ISBN 0-12-727320-4.
- [9] David Budgen. *Software Design*, Addison-Wesley, USA, primera edición, 1993, ISBN 0-201-54403-2.
- [10] Carlo Ghezzi. *Programming Language Concepts*, John Wiley & Sons, USA, tercera edición, 1997, ISBN 0-471-10426-4.
- [11] C.J. Date. *Introducción a los Sistemas de Bases de Datos*, Pearson Educación, México, séptima edición, 2001, ISBN 968-444-419-2.

- [12] Grady Booch, James Rumbaugh, Ivan Jacobson. *The Unified Modeling Language. User Guide*, Addison-Wesley, USA, primera edición, 1998, ISBN 0-201-57168-4.
- [13] Murray Cantor. *Object-Oriented Project Management with UML*, Addison-Wesley, USA, primera edición, 1998, ISBN 0-471-25303-0.
- [14] Desmond F. De'Souza, Alan Cameron Wills. *Objects, Components, and Frameworks with UML*, Addison-Wesley, USA, primera edición, 1999, ISBN 0-201-31012-0.
- [15] Davis Garlan, Mary Shaw. *An Introduction to Software Architecture*, Carnegie Mellon University, USA, reporte técnico, 1994, CMU/SEI-94-TR-21.
- [16] Bill Wood, Richard Pethia, Lauren Roberts Gold, Robert Firth. *A Guide to the Assessment of Software Development Methods*, Carnegie Mellon University, USA, reporte preliminar, 1998, CMU/SEI-88-TR-8.
- [17] Thomas B. Hilburn, Iraj Hirmanpour, Soheil Khajenoori, Richard Turner, Abir Qasem. *A Software Engineering Body of Knowledge*, Carnegie Mellon University, USA, reporte técnico, 1999, CMU/SEI-99-TR-004.