



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMA DE VALIDACIÓN Y REGISTRO DE
OPERACIONES DE LAS TORRES DE CONTROL

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN
P R E S E N T A :
MARTÍN RODRÍGUEZ VALDEZ

DIRECTOR DE TESIS:
M.I. JORGE VALERIANO ASSEM



MÉXICO 2005

m 341348



Universidad Nacional
Autónoma de México



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.

Agradecimientos

Con tu ejemplo y cariño, me has sabido conducir al otro lado del camino, me has enseñado entre muchas otras cosas, a levantarme y a enfrentarme a la vida dignamente. Sin ti, nada de esto hubiese sido posible. Gracias Papá.

Agradezco a Dios el haberme dado la bendición de tenerte como madre y de recibir tu inmenso amor, mismo que guardo con tanto cariño, tanto como tu anhelo de verme prosperar. ¡Este momento te lo dedico a ti Madre! Donde quiera que te encuentres.

En tu honor comparto Hermanita, la felicidad de haber finalizado mis estudios, que tu sabes, no puede ser completa pero si muy grande. Gracias Paty, me haces tanta falta.

Ana Gabriela, tenerte a mi lado ha sido la dicha más grande que me pudo haber ocurrido. Juntos hemos podido alcanzar éste hermoso sueño. Por tu valiosa ayuda, por tu inmenso amor, por nuestros hijos y por todo lo que me has ayudado a alcanzar. Muchas Gracias Gordita.

Santiago y Paulina, mis dos más grandes tesoros. Son el motor que me impulsa a dar lo mejor, han sido la inspiración que me ha llevado hasta aquí. Soy muy afortunado de ser su padre.

Auda y Fede, por que la vida nos ha dado la oportunidad de convivir, crecer y alcanzar, juntos nuestras metas. Por que han estado ahí cada vez que los he necesitado, son una parte muy pero muy importante en mi vida. Les agradezco cada detalle que han tenido para conmigo y por todo el amor que me han entregado. Los quiero tanto.

Ser tu hermano ha sido un verdadero placer, hemos pasado tantos buenos momentos juntos, nos hemos divertido como enanos, ¡sobre todo tu!, no podría imaginar mi vida sin ti. Gracias Mike, me siento muy orgulloso de ser tu hermano.

A mi familia Querida que ha estado permanentemente ahí, al pendiente y preocupada por mí bienestar, que me ha abrazado y me ha dado una palabra de aliento en momentos difíciles. Vito[†], Nonavita, Tita Lulú, Ticho, Ismael, Carmen, Chetito, Sergio, Lulucita, Milo, Lety, Golly, Emilio y a los Kobeh que tanto quiero.

Iván, tu más que nadie conoce el gran esfuerzo que ha significado todo esto, tu valiosísima ayuda ha sido clave fundamental para el logro de éste trabajo, pero lo mejor de todo es tu apoyo incondicional y tu amistad. Gracias Hermano...¡¡¡Un punto!!!

Mi sangre, Güerito y Beto, han sabido ser más que amigos, hermanos. En mi corazón estarán, como siempre lo ha estado mi cariño, admiración y respeto.

Quisiera hacer un reconocimiento especial a Adelita Castillejos, quien me dio la oportunidad más importante que he tenido y que gracias a sus consejos y sobre todo a la confianza ofrecida, he podido concretar mis estudios. Gracias Adelita.

A mi querida empresa y a todos aquellos que contribuyeron y me apoyaron para llevar a cabo este proyecto que hoy, ya es una realidad. De corazón, Gracias.

Por último quiero agradecer a mi muy querida Facultad y a todos mis queridos maestros que con sus conocimientos y dedicación, me formaron.



Introducción

A raíz de la urgente necesidad de establecer un método, en el cual se pueda mejorar en tiempo y forma las funciones, que hasta no hace mucho, eran llevadas a cabo manualmente. Se presenta esta obra con un enfoque adoptado a las necesidades del área de ingresos, las oficinas de las torres de control y a los requerimientos dictados por la Secretaría de Hacienda y Crédito Público. Dicho trabajo contiene el desarrollo del proyecto de un sistema de validación y registro de operaciones que ocurren en las torres de control y que son administradas por el Gobierno Federal, a través del Órgano desconcentrado encargado de dichos servicios. Este sistema o módulo, como parte de un plan bien establecido, tiene como objetivo interconectarse con otros sistemas ya establecidos y en un futuro con nuevos sistemas que se están desarrollando con el firme propósito de mejorar el servicio.

El trabajo consta de cuatro partes, las cuales abarcan un pequeño resumen de los antecedentes históricos de los servicios a la navegación en el espacio de la República Mexicana, sin dejar de mencionar la filosofía, visión, misión y objetivos que persigue dicho organismo. Se abre un paréntesis para mostrar la estructura orgánica, funciones y facultades de las áreas involucradas, comentando la participación de las distintas áreas en el proceso. Más adelante proporciono al lector los fundamentos teóricos que anteceden al desarrollo del sistema y presento enseguida la administración del proyecto, el análisis y diseño para la construcción del programa. Por último se describe el desarrollo y la metodología empleada.

Se considera que este trabajo puede ser de gran utilidad para todo aquel que desee adquirir conocimientos sobre los servicios que se ofrecen a la navegación aérea y en particular para el sistema de validación y registro de operaciones en las torres de control.

Conociendo que este documento podría abarcar más temas dentro de su contenido, utilizo solamente aquellos, que, a mi juicio, cumplen con lo requerido y mínimo necesario para poder concluir satisfactoriamente con dicho trabajo.



Universidad Nacional
Autónoma de México



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.



Índice Temático

Antecedentes.....	2
1.1. MARCO HISTÓRICO	2
1.2. FILOSOFÍA, MISIÓN, VISIÓN Y OBJETIVOS	3
1.2.1. Filosofía.....	3
1.2.2. Misión.....	3
1.2.3. Visión	3
1.2.4. Objetivos	4
1.3. ESTRUCTURA ORGÁNICA.....	4
1.4. SERVICIOS QUE OFRECE UNA EMPRESA FEDERAL DE SERVICIOS A LA NAVEGACIÓN AÉREA.....	4
1.4.1. Tránsito Aéreo.....	5
1.4.2. Navegación Aérea.....	5
1.4.3. Telecomunicaciones.....	6
1.4.4. Meteorología Aeronáutica.....	6
1.4.5. Despacho y Control de Vuelos.....	6
1.5. FACULTADES Y DIRECCIONES INVOLUCRADAS.....	7
1.5.1. Dirección General.....	7
1.5.2. Dirección de Operación.....	8
1.5.3. Dirección General de Tránsito Aéreo.....	9
1.5.4. Dirección General Técnica.....	9
1.5.5. Dirección General de Finanzas.....	10
1.5.6. Dirección General de Administración.....	10
1.6. PROCESO DE PARTICIPACIÓN DE LAS ÁREAS INVOLUCRADAS.....	11
Marco Teórico.....	15



Universidad Nacional
Autónoma de México



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.

2.1. BASES DE DATOS RELACIONALES.....	15
2.1.1. Introducción.....	15
2.1.2. Historia.....	16
2.1.3. Ventajas y Desventajas de un Sistema de Bases de Datos.....	18
2.1.4. Arquitectura de los Sistemas de Bases de Datos.....	19
2.1.5. El Sistema de Gestión de Bases de Datos, SGBD.....	21
2.1.6. Características de las Bases de Datos.....	21
2.1.7. Modelo Relacional.....	23
2.1.7.1. Introducción.....	23
2.1.7.2. Estructura del Modelo Relacional.....	25
2.1.7.3. Proceso de Normalización.....	26
2.1.7.4. Primera Forma Normal (1NF).....	26
2.1.7.5. Segunda Forma Normal (2NF).....	26
2.1.7.6. Tercera Forma Normal (3NF).....	26
2.1.8. Interrelaciones.....	27
2.1.8.1. Relaciones Uno A Uno.....	27
2.1.8.2. Relaciones Uno a Muchos.....	27
2.1.8.3. Relaciones Muchos A Muchos.....	27
2.1.9. Álgebra Relacional.....	27
2.1.10. Cálculo Relacional.....	28
2.2. INGENIERÍA DE SOFTWARE.....	29
2.2.1. Introducción.....	29
2.2.2. Antecedentes y Evolución.....	30
2.2.3. Definición de Programa.....	30
2.2.4. Metas de la Ingeniería de Software.....	31
2.2.5. Modelos de la Ingeniería de Software.....	31
2.2.6. Modelo Clásico.....	31
2.2.6.1. Definición del Modelo Clásico.....	31
2.2.6.2. El Modelo en Cascada (Clásico).....	32
2.2.7. Construcción de Prototipos.....	36
2.2.8. El Modelo Incremental.....	37
2.2.9. Modelo Espiral.....	37
2.2.10. Uso de Técnicas de Cuarta Generación (T4G).....	38
2.2.11. Combinación de Prototipos.....	40
2.2.12. Aseguramiento de Calidad.....	42
2.1.12.1. Calidad del Software y Garantía de Calidad.....	42
2.2.12.2. Factores de Calidad del Software.....	42
2.2.12.3. Revisión de las Técnicas Formales.....	43
2.2.12.4. Inspección del Software.....	44
2.2.12.5. El Proceso de Inspección.....	44
2.2.12.6. Métricas de Calidad.....	45
2.2.12.7. Enfoque Formal de Aseguramiento de Calidad en el Software, SQA.....	46
2.2.13. Fiabilidad en el Software.....	47
2.2.14. Enfoque para la Garantía de Calidad del Software.....	48
2.3. REDES DE COMPUTADORAS.....	49
2.3.1. Introducción a las Redes.....	49
2.3.2. El Modelo OSI.....	50
2.3.3. El Modelo IEEE.....	52
2.3.4. Topología de Red.....	53

2.3.5. Topología Física.....	53
2.3.6. Topología Lógica.....	53
2.3.7. Topología en Bus.....	54
2.3.8. Topología en Anillo.....	54
2.3.9. Topología en Estrella.....	55
2.3.10. Interconexión de Redes.....	56
2.3.11. Arquitectura Ethernet.....	59
2.3.12. Arquitectura Token-Ring.....	60
2.3.13. Protocolo TCP/IP.....	61
2.3.14. Máscara de Subred.....	63
2.3.15. Clases de Redes.....	63
2.3.16. Relación entre Direcciones IP y Direcciones Físicas.....	64
2.3.17. Sistemas Operativos de Red.....	65
2.3.18. Acceso Remoto a la Red Local.....	65
Administración del Proyecto.....	68
3.1. JUSTIFICACIÓN DEL PROYECTO.....	68
3.2. OBJETIVOS DEL PROYECTO.....	70
3.3. DIAGNÓSTICO DE LOS REQUERIMIENTOS.....	71
3.3.1 Requerimiento de Recursos Materiales.....	72
3.3.1.1 Software.....	72
3.3.1.2 Hardware.....	80
3.3.2 Requerimientos de Recursos Humanos.....	80
3.3.2.1 Funciones del personal asignado.....	80
3.3.2.2 Plantilla asignada al proyecto.....	81
3.2.3 Plan de trabajo.....	81
3.3.3 Requerimientos de Recursos Financieros.....	85
3.3.3.4 Gestión del Proyecto.....	88
Desarrollo del Proyecto.....	91
4.1. ANÁLISIS.....	91
4.1.1. Análisis de Requerimientos Técnicos.....	91
4.1.2. Análisis de Requerimientos del Usuario.....	93
4.1.2.1 Procedimiento Previo.....	93
4.1.2.2. Procedimiento de Recolección de Archivos de las Torres de Control.....	94
4.1.3. Objetivos del Sistema.....	94
4.1.4. Análisis.....	94
4.1.4.1. Entrada a la Aplicación.....	94
4.1.4.2. Interfaz principal.....	95
4.1.4.3. Módulo de captura y consulta de operaciones.....	95
4.1.4.4. SubMódulo Operaciones (1.1).....	96
4.1.4.5. Procedimiento de Corrección de Operaciones Diarias (1.11).....	96
4.1.4.6. Relación de Errores.....	98
4.1.4.7. Módulo de Reportes (1.12).....	99
4.1.4.8. Módulo de Correcciones (1.13).....	100
4.1.4.9. Módulo de Estación OACI (1.2).....	100
4.1.4.10. SubMódulo de Administración de Usuarios (1.3).....	101
4.1.4.12. Alta de Usuarios.....	102
4.1.4.13. Formulario de Remesa (1.32).....	103
4.1.4.14. Formulario de Remesa 20 (1.23).....	104

4.1.4.15. Módulo 2: Módulo de Transferencia.	104
4.1.4.16. SubMódulo de Exportaciones (2.1).	104
4.1.4.17. SubMódulo de Importaciones (2.22).	105
4.1.4.18. Formulario de Importaciones de Equipo (2.21).	105
4.1.4.19. Formulario de Importación de Estación IATA (2.22).	106
4.1.4.20. Importación de Matrículas (2.23).	106
4.1.4.21. Importación de Usuarios (2.25).	106
4.1.4.22. Importación de Operaciones (2.24).	106
4.1.5.24. Módulo 3: Módulo de Utilerías.	107
4.1.4.24. Registrar la Base de Datos (3.4).	108
4.1.4.25. Propiedades de la Base de Datos (3.5).	108
4.1.4.26. Soportar Operaciones (3.6).	108
4.1.4.28. Nivel de Usuario (5.1).	109
4.1.4.29. Cambio de Nivel de Usuario (5.2).	109
4.2. DISEÑO.	110
4.2.1. Diseño de Arquitectura de Datos.	110
4.2.1.1. Normalización.	110
4.2.1.2. Grados de Normalización.	110
4.2.1.3. Primera Forma Normal 1FN.	110
4.2.1.4. Segunda Forma Normal 2FN.	111
4.2.1.5. Tercera Forma Normal 3FN.	111
4.2.1.6. Tercera Forma Normal.	114
4.2.1.7. Diccionario de Datos.	119
4.2.1.8. Diagrama Entidad Relación.	122
4.2.2. Diseño de Arquitectura de Negocio.	122
4.2.2.1. Diseño de Casos de Uso.	123
4.2.2.2. Actores.	123
4.2.2.3. Casos de Usos.	123
4.2.2.4. Extensión.	124
4.2.2.5. Inclusión.	124
4.2.2.4. Documentación de las Tablas de Casos de Uso.	124
4.2.2.6. Diagrama de Actividad.	128
4.2.2.7. Identificación de Clases.	131
4.2.2.8. Clases Entidad.	131
4.2.2.8. Clases de Tipo Control.	132
4.2.2.9. Selección de Clases.	132
4.2.2.10. Diagrama de Clases.	132
4.2.2.11. Diagramas de secuencia.	133
4.2.3. Diseño de la Infraestructura Informática.	135
4.2.3.1. Recursos de Hardware.	135
4.2.3.2. Recursos de Software.	135
4.2.4. Codificación.	136
4.2.4.2. Pseudocódigo de la Clase Usuario.	139
4.2.4.3. Pseudocódigo de la Clase Operaciones Diarias.	139
4.2.4.4. Pseudocódigo de la Clase Error.	141
4.2.4.5. Pseudocódigo de la Clase Reportes.	142
4.2.5. Diseño de Pruebas.	142
4.2.5.1. Tipos de Pruebas.	143
4.2.5.2. Pruebas Unitarias de Integración.	143
4.2.5.3. Desarrollo de Integración Incremental.	143

4.2.5.4. Estrategias de Integración.....	144
4.2.5.5. Pruebas Globales.....	144
4.2.5.7. Los Procesos del Desarrollo como Fuente de Pruebas..	145
4.2.5.8. Plan de pruebas.....	145
4.2.5.9. Modelo de Casos de Uso Como Fuente de Pruebas.....	148
4.2.5.10. Modelo de Clases Como Fuente de Pruebas.....	148
4.2.5.11. Diagramas de Colaboración Como Fuente de Pruebas.	149
4.2.5.12. Plan de pruebas para el Sistema de Validación y Registro de Operaciones de las Torres de Control, (SIVAR).....	149
4.2.6. Implementación.....	156
4.2.7. Capacitación.....	156
Conclusiones.....	158
Apéndice A: Instalación del Sistema Operativo.....	159
A.1. Microsoft Windows NT 4.0 WS, Introducción.....	159
A.2 Reconocimiento del Sistema.....	160
Bibliografía.....	164

Capítulo

1



Universidad Nacional
Autónoma de México



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.

Capítulo

1

Antecedentes

1.1. MARCO HISTÓRICO

En México la evolución de los servicios de ayuda a la navegación aérea tiene su origen en una pequeña oficina de navegación aérea de la vieja Secretaría de Comunicaciones y Obras Públicas ubicada en la calle de Tacuba a principios del siglo pasado. El esfuerzo de un visionario y forjador de México del Siglo XX, intuyó la magnitud del impacto de la nueva modalidad de transporte, en la naciente aviación civil. Este hombre fue el Ing. Juan Guillermo Villasana, quien allá por el año 1920 hizo los primeros intentos de regulación para normar la nueva industria que empezaba a despuntar en el país. Comenzaba el despegue de la forma más joven y dinámica de trasladar personas y bienes abatiendo tiempos y distancias.

A lo largo del periodo de la transformación de la oficina de navegación aérea en la dirección general, los servicios de ayuda a la navegación aérea fueron creciendo en facilidades y apertura de estaciones, conformándose poco a poco la red de radioguías que es el antecedente inmediato a la generación de radiofaros actualmente en operación.

Para 1944 el Gobierno Federal concesiona dichos servicios a particulares, creándose la empresa Radio Aeronáutica Mexicana, S. A. (**RAMSA**). En junio del siguiente año RAMSA inicia la operación de su primera torre de control en el antiguo Aeropuerto de la Ciudad de México.

Al extenderse simultáneamente la aviación,, así como sus procedimientos técnicos y legales, la pequeña oficina de navegación aérea ascendió en importancia convirtiéndose en el año de 1928 en Departamento y hacia

1952 en Dirección de Aeronáutica Civil. En 1956 alcanza el rango de Dirección General de Aeronáutica Civil.

Para 1952, la entonces Secretaría de Comunicaciones y Obras Públicas amplía a RAMSA la concesión por 30 años para proporcionar los servicios radioaeronáuticos y de meteorología en todo el país. Al entrar en funciones las Comisiones de Estudios y Proyectos Técnicos Aeronáuticos y de Planeación de Aeropuertos en 1964, existían 42 aeropuertos que contaban con servicios de control aéreo. El control de aeródromo era operado en 18 torres de control a cargo de Aeronaves de México, 19 torres de control por Mexicana de Aviación, y 5 por RAMSA. El Control de Aproximación sólo se proporcionaba en el aeropuerto de la Ciudad de México por RAMSA, así como el naciente Centro de Control de Área.

Los servicios meteorológicos mantenían una infraestructura con la central meteorológica del aeropuerto internacional de la Ciudad de México así como 53 oficinas meteorológicas a lo largo de la nación

El 15 de noviembre de 1966, por decreto presidencial, RAMSA se convierte en empresa de participación estatal mayoritaria al adquirir el gobierno federal el 51% de su capital social.

Esta empresa operó durante 12 años y en 1978 el Ejecutivo Federal asume la prestación de los servicios mediante la creación de un Órgano desconcentrado, dependiente de la Secretaría de Comunicaciones y Transportes.

1.2. FILOSOFÍA, MISIÓN, VISIÓN Y OBJETIVOS

1.2.1. Filosofía.

Trabajar con orientación a resultados. Creatividad, innovación, respeto y lealtad institucional. Desarrollo humano, responsabilidad, confianza y compromiso hacia nuestra institución y al trabajo de nuestros compañeros. Disposición para asimilar el impacto tecnológico optimizando la calidad del servicio.

1.2.2. Misión.

Garantizar a través de servicios a la navegación, un transporte seguro y eficiente de personas y bienes en el espacio aéreo.

1.2.3. Visión.

Adoptar y establecer oportuna y eficientemente la infraestructura y técnicas innovadoras en la prestación de servicios a la navegación aérea, acorde al desarrollo de la aviación.

1.2.4. Objetivos.

Proporcionar con calidad los servicios de control de tránsito aéreo, radioayudas a la navegación aérea, telecomunicaciones, meteorología e información aeronáutica.

1.3. ESTRUCTURA ORGÁNICA.

En la siguiente figura (1.3.1) se muestra la estructura orgánica de la Oficina de Servicios a la Navegación. La encabeza la Dirección General y esta dividida por 5 Direcciones Generales Adjuntas.

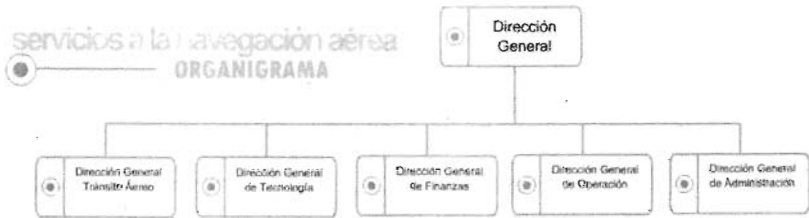


Figura 1.3.1 Estructura Orgánica.

1.4. SERVICIOS QUE OFRECE UNA EMPRESA FEDERAL DE SERVICIOS A LA NAVEGACIÓN AÉREA.

La Organización de Aviación Civil Internacional (OACI) dependiente de la Organización de Naciones Unidas (ONU) tiene la misión de normar, regular y emitir recomendaciones inherentes al transporte aéreo mundial en la parte civil.

México como Estado signatario del Convenio de Chicago, es responsable del espacio aéreo sobre el territorio nacional, así como aquellas partes sobre alta mar del Océano Pacífico y Golfo de México en donde el Convenio obliga a prestar los servicios a la navegación aérea. Esta responsabilidad dentro del Poder Ejecutivo Federal esta a cargo de la Secretaría de Comunicaciones y Transportes, la cual ha encomendado Al órgano desconcentrado proporcionar los servicios a la navegación aérea en esta porción del espacio.

1.4.1. Tránsito Aéreo.

Para proporcionar este servicio, se ha dividido imaginariamente el espacio aéreo en cuatro principales áreas y una región de información de vuelo oceánica que son México, Monterrey, Mazatlán, Mérida y área oceánica de Mazatlán. En cada uno de los aeropuertos de las ciudades antes citadas, se ubica un Centro de Control de Área y un centro de información de vuelo en Mérida.

El servicio de control de tránsito aéreo se proporciona desde los Centros de Control mencionados, además de las Torres de Control situadas en cada uno de los aeropuertos de la red. Algunas de ellas están dotadas con equipos monitores de radar para la vigilancia de las aeronaves en las inmediaciones de los aeropuertos y con base en estos datos se controla y se da instrucciones al tránsito que sale y llega a cada uno de los mismos. En los espacios donde no se cuenta con cobertura radar, pero que están bajo la jurisdicción de los centros y torres de control, el desplazamiento de aeronaves se organiza mediante la aplicación de procedimientos para el vuelo por Instrumentos, que hace posible a las aeronaves operar en ese espacio con la separación reglamentaria una de otra así como del terreno.

Como apoyo a la prestación del control de tránsito aéreo se obtiene información de los sistemas de radar que se encuentran instalados a lo largo de la república mexicana: Los hay de largo alcance (370 Km.), para la vigilancia de las aeronaves en ruta y de menor alcance (130km.), para el vuelo en las áreas terminales o de aproximación e inmediaciones de los aeropuertos.

La comunicación entre pilotos y controladores se efectúa mediante un sistema que permite el flujo de información en forma eficiente e instantánea y forma un circuito entre las aeronaves, la torre de control, el centro de control que esta suministrando el servicio y los centros adyacentes, mediante una coordinación precisa que asegura realizar las operaciones aeronáuticas eficazmente.

1.4.2. Navegación Aérea.

Para hacer posible la navegación dentro del espacio bajo la jurisdicción de México se cuenta con una infraestructura de facilidades en tierra, con las que se establecen rutas aéreas, que facilitan al piloto de una aeronave conducirla de un punto a otro, aún en condiciones meteorológicas desfavorables, de manera segura y precisa. Esta red de rutas llamadas aerovías, permite a los pilotos trasladarse a cualquier aeropuerto del sistema que cuente con este tipo de guías de navegación. El sistema de rutas internacionales, permite la navegación entre cualquier aeropuerto de la República a los aeropuertos en otros países.

Este sistema de navegación está constituido básicamente por radioayudas direccionales que le indican al piloto en todo momento la dirección y

distancia a la que se encuentra de dichas radioayudas; distancia medida electrónicamente que se le presenta al piloto, en instrumentos de la cabina, en millas y en fracciones de millas náuticas. Además, se cuenta en los aeropuertos más importantes o que presentan condiciones especiales con sistemas de aterrizaje de precisión. Estas operaciones se efectúan en forma segura aún en condiciones de tiempo adversas.

1.4.3. Telecomunicaciones.

El servicio de Telecomunicaciones Aeronáuticas aire/tierra es un apoyo fundamental para el tránsito aéreo. Está basado en equipos transmisores y receptores, instalados estratégicamente en estaciones terrestres locales y remotas que enlazan a los centros de control y las torres de control a través de canales de la Red Nacional de Microondas y estaciones terrenas satelitales, estableciendo la comunicación entre el piloto y controlador. El piloto informa su posición, solicitando instrucciones al controlador responsable de aplicar la separación entre aeronaves dentro del espacio bajo su control, El controlador da instrucciones y mantiene el tránsito fluido, seguro y ordenado. También las señales de radar son enviadas mediante canales de microondas y estaciones terrenas a los centros y torres de control.

1.4.4. Meteorología Aeronáutica.

Otro servicio de importancia vital lo constituye el de Meteorología Aeronáutica. Por este servicio la empresa de aviación y el piloto reciben información referente al tiempo que hay en las áreas terminales que circundan a los aeropuertos de origen y destino del vuelo, así como las condiciones meteorológicas que hay o se prevén a lo largo de toda la aerovía o rutas que se va a recorrer. Con esta información se aprovechan óptimamente las condiciones meteorológicas para seguir la mejor ruta que garantice la seguridad del vuelo o en algunos casos la ruta más corta, para poder conducir con eficiencia el vuelo de la aeronave.

1.4.5. Despacho y Control de Vuelos.

También existe el Servicio de Despacho y Control de Vuelos consistente en asesorar a las tripulaciones de la aviación en general, para la elaboración de su plan de vuelo con información meteorológica respecto a: vientos superiores, reportes horarios, pronósticos meteorológicos de los aeropuertos de destino y alternos en la ruta que se seguirá, e información de NOTAM¹.

El plan de vuelo es asesorado por el Despachador quien coordina y notifica dicho plan a las dependencias de tránsito aéreo comunicando a la estación de destino la salida de la aeronave o de un aterrizaje a la estación de Origen. Entre otros de los datos que se proporciona a los pilotos son los

¹ (Notice to Airman). Aviso al piloto.

relativos a las condiciones de ceniza volcánica en el ambiente, avisos de depresión, tormentas o huracanes. En algunos aeropuertos se cuenta con la facilidad para el piloto de notificar su plan de vuelo telefónicamente o por una frecuencia dedicada sin tener la necesidad de realizar este trámite personalmente en las oficinas de despacho las cuales casi siempre están ubicadas en la planta baja de las torres de control.

1.5. FACULTADES Y DIRECCIONES INVOLUCRADAS.

A continuación se detalla brevemente las responsabilidades propias de las áreas involucradas en éste proceso.

1.5.1. Dirección General.

- Programar, organizar, dirigir, controlar y evaluar el desarrollo de las actividades encomendadas a las unidades que integren la dirección general o unidad a su cargo.
- Formular los anteproyectos de programas, presupuestos y los demás que les corresponda, conforme a las normas establecidas;
- Formular los anteproyectos de iniciativa de leyes, reglamentos, decretos, acuerdos y órdenes en los asuntos de su competencia que deban remitirse a la Dirección General de Asuntos Jurídicos para su trámite correspondiente;
- Adoptar las medidas necesarias para la debida observancia de las leyes, reglamentos, decretos, acuerdos y demás disposiciones relacionadas con el funcionamiento y los servicios encomendados a la dirección general o unidad a su cargo, así como para prevenir incumplimiento y proponer la aplicación de las sanciones que procedan y, cuando resulte necesario recomendar se modifiquen las normas vigentes o se adopten nuevas, de conformidad con las disposiciones legales aplicables y, en su caso, con las Condiciones Generales de Trabajo;
- Establecer normas y lineamientos para el cumplimiento de los programas de la dirección general o unidad a su cargo en los Centros SCT, previa opinión de la Coordinación General de Planeación y Centros SCT;
- Recibir, calificar, aceptar o rechazar, cancelar u ordenar en su caso, la sustitución, ampliación o ejecución de las garantías que constituyan los particulares para el cumplimiento de las obligaciones o trámites de concesiones, permisos, autorizaciones, contratos o convenios que se deban otorgar para operar servicios relacionados con vías generales de comunicación, sus

servicios conexos y auxiliares, así como para cualquier tipo de contratos;

- Autorizar por escrito, conforme a las necesidades del servicio y de acuerdo con su superior jerárquico, a los servidores públicos subalternos para que firmen documentos o intervengan en determinados asuntos, relacionados con la competencia de la dirección general o unidad a su cargo, y Las demás facultades que les confieren las disposiciones legales y administrativas aplicables y sus superiores.

1.5.2. Dirección de Operación.

- Administrar la oportuna y correcta prestación de los servicios a la navegación aérea en las Gerencias Regionales; en el Control de Tránsito Aéreo, Meteorología Aeronáutica, Sistemas de Ayuda a la Navegación Aérea, Telecomunicaciones Aeronáuticas, Servicios de Despacho y Control de Vuelos y otros que sean necesarios en la República Mexicana.
- Operar y conservar los diferentes sistemas que integran los servicios a cargo del órgano descritos y hacer directamente las instalaciones correspondientes a ampliaciones de estos sistemas que se aprueben al órgano en su programa de inversiones.
- Asegurar que las Gerencias Regionales proporcionen los servicios de mensajes operacionales, administrativos y de cualquier índole que requieran las empresas, dependencias o particulares, en sus actividades de transporte aéreo nacional e internacional.
- Dirigir la planeación, organización, funcionamiento y desarrollo de los servicios de meteorología, control de tránsito aéreo, telecomunicaciones aeronáuticas, radioayudas a la navegación y de procedimientos operacionales, servicios de despacho e información de vuelo, a través de las Gerencias Regionales.
- Acordar con la Dirección General el tratamiento de los asuntos y propuestas de orden operativo.
- Coordinar la realización de estudios y estadísticas para determinar la eficiencia en las operaciones, servicios y actividades del área operativa.
- Administrar la correcta difusión y aplicación de los documentos, manuales, boletines de información, circulares, enmiendas y demás información de carácter internacional que se reciban de la Organización de Aviación Civil Internacional (OACI) y de la Dirección General de Aeronáutica Civil.

- Representar a la Dirección General en todas las funciones que le encomiende nacional o internacionalmente, para el sano funcionamiento de este órgano.
- Recopilar y conservar actualizada toda la información relativa a los servicios a la navegación aérea.

1.5.3. Dirección General de Tránsito Aéreo.

- Administrar los servicios de navegación, información aeronáutica, meteorología y telecomunicaciones.
- Establecer la normatividad para el funcionamiento eficaz y eficiente de los servicios de tránsito aéreo.
- Evaluar la calidad de la prestación del servicio de tránsito aéreo.
- Coordinar con las dependencias oficiales y organismos privados nacionales e internacionales en cuanto a técnicas y procedimientos correspondientes a las áreas de su responsabilidad.
- Dirigir la realización de estudios y proyectos aeronáuticos relacionados con las áreas de tránsito aéreo, navegación, meteorología, telecomunicaciones y navegación e información aeronáutica.
- Evaluar los planes y programas de las áreas a su cargo, así como asegurar su cumplimiento.
- Establecer la normatividad derivada de los manuales, circulares, enmiendas y demás información de carácter internacional que se reciban de la OACI.
- Coordinar con las Direcciones Generales y Gerencias Regionales para la toma de decisiones.
- Coordinar con la Dirección General el tratamiento de los asuntos y propuestas según se requiera.
- Asegurar la participación en la integración del presupuesto.
- Atender los asuntos que la Dirección General encomiende.

1.5.4. Dirección General Técnica.

- Coordinar la planeación, organización y desarrollo de los sistemas, estableciendo los controles necesarios a través de las Direcciones y la Subdirección que la conforman.
- Controlar la instalación de los sistemas y equipos necesarios para la prestación de los servicios y en su caso, verificar el buen funcionamiento de trabajos realizados por terceros.
- Coordinar y verificar la elaboración, difusión de normas y procedimientos de operación y mantenimiento de los equipos a través de manuales, boletines o cualquier otro medio de información, que tiendan al desarrollo de las actividades técnicas del órgano.

- Controlar la distribución y actualización de los manuales técnicos del órgano, en coordinación con la Dirección de Sistemas Organizacionales.
- Coordinar la elaboración de programas y temarios para la capacitación especializada del personal del órgano, con las Direcciones Generales y Gerencias Regionales.

1.5.5. Dirección General de Finanzas.

- Coordinar con las dependencias del Sector Público, en lo referente a la aplicación de políticas, normas, sistemas y procedimientos financieros a los que se sujeta el órgano.
- Proporcionar la información presupuestal, contable y financiera que le requieran las unidades administrativas de la SCT, SHCP, SFP, TESOFE y Contaduría Mayor de Hacienda.
- Consolidar el programa anual de recaudación por derechos, aprovechamientos y productos, así como el presupuesto de egreso de gasto corriente e inversión de acuerdo a las disposiciones jurídico-administrativas aplicables vigentes.
- Controlar que el ejercicio del presupuesto de egreso de gasto corriente e inversión, así como la recaudación por derechos, aprovechamientos y productos por los servicios prestados se realice de acuerdo a la legislación en la materia y los sistemas internos que se establezcan.
- Evaluar el avance de los programas establecidos bajo su ámbito de competencia.
- Autorizar los procedimientos de carácter financiero que se aplicarán en cada área de responsabilidad.
- Comunicar y coordinar, con las Direcciones Generales y Gerencias Regionales para tomar decisiones integrales.
- Representar al órgano por instrucciones de la Dirección General, en aquellos casos que se traten asuntos de su competencia.
- Aprobar el programa anual de trabajo de las áreas de responsabilidad que integran a la Dirección General.
- Proporcionar información financiera periódica a la Dirección General.
- Las demás funciones que le encomiende la Dirección General y que sean del ámbito de su competencia.

1.5.6. Dirección General de Administración.

- Planear, desarrollar y coordinar las actividades relacionadas con la administración de los recursos humanos y materiales, proporcionando los apoyos necesarios a todas las áreas del órgano.

- Integrar el programa anual de contratación de personal, de acuerdo a los planes de desarrollo del órgano conforme a los presupuestos aprobados.
- Disponer la elaboración de sistemas de control y prestaciones económico-sociales al personal del órgano con oportunidad.
- Proporcionar la información inherente a su área, a las unidades administrativas de la SCT, SHCP y SFP, cuando así lo requieran.
- Dirigir la formulación del Plan Anual de Adquisiciones de acuerdo a los presupuestos aprobados.
- Coordinar la elaboración del estudio de preinversión del órgano.
- Verificar que los suministros y existencias en los almacenes, satisfagan los máximos y mínimos requeridos.
- Evaluar cumplimiento de las disposiciones sobre el control de los bienes que el órgano tiene bajo su custodia.
- Representar al órgano, por instrucciones de la Dirección General en los asuntos de su competencia.
- Dirigir y controlar los programas de motivación, comunicación y desarrollo integral del personal.
- Asegurar la comunicación y coordinación con las Direcciones Generales y Gerencias Regionales para la toma de decisiones integrales.
- Participar en la integración del presupuesto anual de inversiones y de gasto corriente.
- Aprobar el programa anual de trabajo de las áreas de responsabilidad que integran a la Dirección General.
- Asegurar las relaciones laborales con las organizaciones sindicales en el órgano.

1.6. PROCESO DE PARTICIPACIÓN DE LAS ÁREAS INVOLUCRADAS.

Las áreas que participan en el proceso de validación de operaciones de la TWR son las siguientes, tal y como se observa en la figura 1.5.7.1.

La Dirección General debe programar, organizar, dirigir, controlar y evaluar el desarrollo de las actividades encomendadas a las unidades que integren la dirección general o unidad a su cargo.

Recaudar, controlar y enterar a la unidad administrativa correspondiente el cobro de los derechos, productos y aprovechamiento de los servicios que proporciona;

Proponer las modificaciones, en cuotas y tarifas, en los derechos y productos y aprovechamientos de los servicios que proporciona;

SISTEMA DE VALIDACIÓN Y REGISTRO DE OPERACIONES DE LAS TORRES DE CONTROL

La Dirección General de Operación debe administrar la oportuna y correcta prestación de los servicios a la navegación aérea en las Gerencias Regionales; en el Control de Tránsito Aéreo, Meteorología Aeronáutica, Sistemas de Ayuda a la Navegación Aérea, Telecomunicaciones Aeronáuticas, Servicios de Despacho y Control de Vuelos y otros que sean necesarios en la República Mexicana.

Operar y conservar los diferentes sistemas que integran los servicios a cargo del órgano descritos y hacer directamente las instalaciones correspondientes a ampliaciones de estos sistemas que se aprueben al órgano en su programa de inversiones.

La Dirección General de Tránsito Aéreo debe administrar los servicios de navegación, información aeronáutica, meteorología y telecomunicaciones.

Establecer la normatividad para el funcionamiento eficaz y eficiente de los servicios de tránsito aéreo.

La Dirección General Técnica debe coordinar la planeación, organización y desarrollo de los sistemas, estableciendo los controles necesarios a través de las Direcciones y la Subdirección que la conforman.

Controlar la instalación de los sistemas y equipos necesarios para la prestación de los servicios y en su caso, verificar el buen funcionamiento de trabajos realizados por terceros.

Y por último la Dirección General de Finanzas debe Coordinar con las dependencias del Sector Público, en lo referente a la aplicación de políticas, normas, sistemas y procedimientos financieros a los que se sujeta el órgano.

Proporcionar la información presupuestal, contable y financiera que le requieran las unidades administrativas de la SCT, SHCP, SFP, TESOFE y Contaduría Mayor de Hacienda.

Consolidar el programa anual de recaudación por derechos, aprovechamientos y productos, así como el presupuesto de egreso de gasto corriente e inversión de acuerdo a las disposiciones jurídico-administrativas aplicables vigentes.

En la figura 1.5.7.1 se muestra la interrelación existente entre las áreas involucradas.

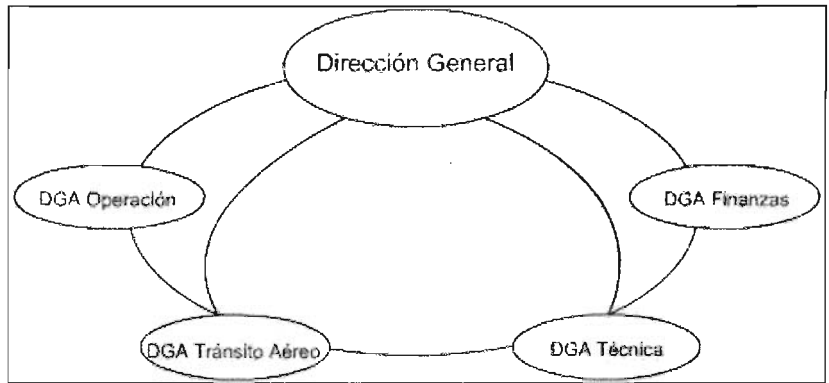


Figura 1.5.7.1. Relación entre las áreas involucradas.

La dirección general al mando de la organización, dirige uno de tantos procesos que se llevan a cabo con el intercambio de actividades entre las áreas involucradas en este proceso.

Capítulo

2



Universidad Nacional
Autónoma de México



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.

Capítulo

2

Marco Teórico

2.1. BASES DE DATOS RELACIONALES.

2.1.1. Introducción.

La tecnología de las bases de datos es una de las áreas computacionales de información de más rápido desarrollo.

Antes de las bases de datos, existieron los sistemas de procesamiento de archivos, basados en registros guardados, creándose respectivamente programas para su manipulación (obtener y guardar información), sin embargo, con estos sistemas de almacenamiento existía redundancia (Duplicación de Datos en la información), inconsistencia (Datos con diferentes valores), no manejaban la concurrencia (no permitían que diferentes usuarios accedieran a la misma información en el mismo tiempo) y se limitaban generalmente a programas con archivo.

La idea de base de datos surge como una necesidad de mantener datos relacionados. Las definiciones de base de datos son numerosas. Todas coinciden en que es un conjunto de datos almacenados con la capacidad de acceder directamente a ellos. Los datos están interrelacionados y estructurados de acuerdo a un modelo que sea capaz de recoger el máximo contenido semántico.

El propósito general de una base de datos es registrar y mantener la información² relacionada y significativa para una organización donde el

² Los términos "datos" e "información" se utilizan como sinónimos. Algunos autores hacen una distinción entre ellos, pues emplean "datos" para referirse a los valores registrados

sistema es operado. En otras palabras, es cualquier dato necesario para los procesos de toma de decisiones inherentes a la administración de la organización.

Una Base de datos se define de la siguiente manera:

*"Colección integrada y generalizada de datos, estructurada atendiendo a las relaciones naturales de modo que suministre todos los caminos de acceso que son necesarios a cada unidad de datos con objeto de poder atender todas las necesidades de los diferentes usuarios"*³ o también como:

"Colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es servir a una o más aplicaciones de la mejor forma posible; los datos se almacenan de modo que resulten independientes de los programas que los usan; se emplean métodos bien determinados para incluir nuevos datos y para modificar o extraer los datos almacenados".⁴

2.1.2. Historia.

Los predecesores de los sistemas de bases de datos fueron los sistemas de archivos. No hay un momento concreto en que los sistemas de archivos hayan cesado y hayan dado comienzo los sistemas de bases de datos. De hecho, todavía existen sistemas de archivos en uso.

Se dice que los sistemas de bases de datos tienen sus raíces en el proyecto estadounidense Apolo de mandar al hombre a la luna. En aquella época, no había ningún sistema que permitiera gestionar la inmensa cantidad de información que requería el proyecto. La primera empresa encargada del proyecto, NAA (North American Aviation)⁵, desarrolló un *software* denominado GUAM (General Update Access Method)⁶ que estaba basado en el concepto de que varias piezas pequeñas se unen para formar una pieza más grande, y así sucesivamente hasta que el producto final está ensamblado. Esta estructura, que tiene la forma de un árbol, es lo que se denomina una *estructura jerárquica*. A mediados de los sesenta, IBM se unió a NAA para desarrollar GUAM en lo que ahora se conoce como IMS (Information Management System)⁷. El motivo por el cual IBM restringió IMS al manejo de jerarquías de registros fue el de permitir el uso de dispositivos de almacenamiento serie, más exactamente las cintas magnéticas, ya que era un requisito del mercado por aquella época.

físicamente en la base de datos, e "información" para aludir al significado de esos valores según el sentido que les dé el usuario. Esta distinción es muy importante, hasta el punto de que parece preferible hacerla explícita donde sea pertinente, en lugar de atenerse a una diferenciación algo arbitraria entre dos términos en esencia similares. (C. J. Date)

³ Deen, 1985.

⁴ Martin, 1975.

⁵ NAA, (Aviación Norteamericana).

⁶ GUAM, (Método General de Acceso Actualizado).

⁷ IMS, (Sistema de Administración de Información).

A mitad de los sesenta, se desarrolló IDS (Integrated Data Store)⁸, de General Electric. Este trabajo fue dirigido por uno de los pioneros en los sistemas de bases de datos, Charles Bachmann. IDS era un nuevo tipo de sistema de bases de datos conocido como *sistema de red*, que produjo un gran efecto sobre los sistemas de información de aquella generación. El sistema de red se desarrolló, en parte, para satisfacer la necesidad de representar relaciones más complejas entre datos, que las que se podían modelar con los sistemas jerárquicos y en parte, para imponer un estándar de bases de datos. Para ayudar a establecer dicho estándar, CODASYL (Conference on Data Systems Languages)⁹, formado por representantes del gobierno de EEUU y representantes del mundo empresarial, formaron un grupo denominado DBTG (Data Base Task Group)¹⁰, cuyo objetivo era definir unas especificaciones estándar que permitieran la creación de bases de datos y el manejo de los datos. El DBTG presentó su informe final en 1971 y aunque éste no fue formalmente aceptado por ANSI (American National Standards Institute)¹¹, muchos sistemas se desarrollaron siguiendo la propuesta del DBTG. Estos sistemas son los que se conocen como sistemas de red, o sistemas CODASYL o DBTG.

Los sistemas jerárquico y de red constituyen la primera generación de los, Sistemas de Gestión de Bases de Datos, **SGBD**. Pero estos sistemas presentan algunos inconvenientes: Es necesario escribir complejos programas de aplicación para responder a cualquier tipo de consulta de datos, por simple que ésta sea. La independencia de datos es mínima y no tienen un fundamento teórico.

En 1970 Codd, de los laboratorios de investigación de IBM, escribió un artículo presentando el *modelo relacional*. En este artículo, presentaba también los inconvenientes de los sistemas previos, el jerárquico y el de red. Entonces, se comenzaron a desarrollar muchos sistemas relacionales, apareciendo los primeros a finales de los setenta y principios de los ochenta. Uno de los primeros es el Sistema R de IBM, que se desarrolló para probar la funcionalidad del modelo relacional, proporcionando una implementación de sus estructuras de datos y sus operaciones. Esto condujo a dos grandes desarrollos:

El desarrollo de un lenguaje de consultas estructurado denominado SQL, que se ha convertido en el lenguaje estándar de los sistemas relacionales.

La producción de varios SGBD relacionales durante los años ochenta, como DB2 y SLQ/DS de IBM y ORACLE.

Hoy en día, existen cientos de SGBD relacionales, tanto para microcomputadoras como para sistemas multiusuario, aunque muchos no son completamente fieles al modelo relacional.

⁸ IDS, (Almacenamiento Integrado de Datos).

⁹ CODASYL, (Conferencia sobre Lenguajes de Sistemas de Datos).

¹⁰ DBTG, (Grupo de Tarea de Bases de Datos).

¹¹ ANSI, (Instituto Americano Nacional de Estándares.)

Los SGBD relacionales constituyen la segunda generación de los SGBD. Sin embargo, el modelo relacional también tiene sus fallos, siendo uno de ellos su limitada capacidad al modelar los datos. Se ha hecho mucha investigación desde entonces tratando de resolver este problema. En 1976, Chen presentó el modelo entidad-relación, que es la técnica más utilizada en el diseño de bases de datos. En 1979, Codd intentó subsanar algunas de las deficiencias de su modelo relacional con una versión extendida denominada RM/T (1979) y más recientemente RM/V2 (1990). Los intentos de proporcionar un modelo de datos que represente al mundo real de un modo más fiel han dado lugar a los modelos de datos semánticos.

Como respuesta a la creciente complejidad de las aplicaciones que requieren bases de datos, han surgido dos nuevos modelos: el modelo de datos orientado a objetos y el modelo relacional extendido. Sin embargo, a diferencia de los modelos que los preceden, la composición de estos modelos no está clara. Esta evolución representa la tercera generación de los SGBD.

2.1.3. Ventajas y Desventajas de un Sistema de Bases de Datos.

Los sistemas de bases de datos presentan numerosas ventajas que se pueden dividir en dos grupos: las que se deben a la integración de datos y las que se deben a la interfaz común que proporciona el SGBD.

Ventajas por la integración de datos:

- Control sobre la redundancia de datos.
- Consistencia de datos.
- Más información sobre la misma cantidad de datos.
- Compartición de datos.
- Mantenimiento de estándares.

Ventajas Obtenidas por el Uso de SGBD:

- Mejora en la integridad de datos.
- Mejora en la seguridad.
- Mejora en la accesibilidad a los datos.
- Mejora en la productividad.
- Mejora en el mantenimiento gracias a la independencia de datos.
- Aumento de la concurrencia.
- Mejora en los servicios de copias de seguridad y de recuperación ante fallos.

Desventajas de los Sistemas de Bases de Datos:

- Complejidad.
- Tamaño.

- Costo económico del SGBD.
- Costo del equipamiento adicional.
- Costo de la conversión.
- Prestaciones. Un sistema de archivos está escrito para una aplicación específica, por lo que sus prestaciones suelen ser muy buenas. Sin embargo, los SGBD están escritos para ser más generales y ser útiles en muchas aplicaciones, lo que puede hacer que algunas de ellas no sean tan rápidas como antes.
- Vulnerable a los fallos.

2.1.4. Arquitectura de los Sistemas de Bases de Datos.

En la Figura 2.1.4.1 se detalla un esquema general de la arquitectura de una base de datos, las relaciones entre ellos y la base de datos lógica.

Una Vista Externa es una visión particular de un usuario o un grupo de usuarios de la Base de Datos. El Esquema Externo representa una forma de definición o formalización de esta vista externa.

La Vista Conceptual pretende ser la representación, total y abstracta de los datos que componen la Base; la formalización de esta se logra mediante el Esquema Conceptual.

La Vista Interna es de un nivel muy bajo y corresponde al almacenamiento físico de los datos de la Base como los tipos de registros almacenados.

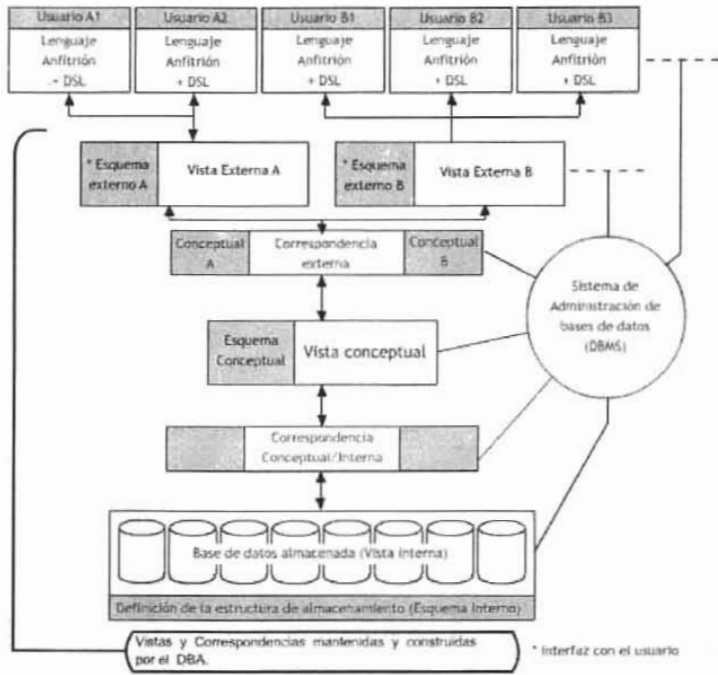


Figura 2.1.4.1. Esquema General de la Arquitectura de una Base de Datos.

Las correspondencias se pueden definir como una asociación de distintas representaciones para un mismo dato.

Un *DSL* es un sub-lenguaje de datos, es una combinación de dos lenguajes: un lenguaje de definición de datos (DDL) y un lenguaje de manipulación de Datos (DML). Este lenguaje representa un intermediario entre el Sistema de Base de Datos y algún lenguaje como C, FORTRAN etc. El DSL provee herramientas a los lenguajes tradicionales para que se integren al Sistema de Base de Datos. Puede haber distintos tipos de DSL para un mismo sistema.

SGBD. (Sistema de Administración de Bases de Datos) es el Software que maneja todos los accesos a la Base de Datos

La interfaz con el Usuario (Front-End) es el límite de acceso que tiene un Usuario común a la Base, todo lo que está bajo este límite es transparente y desconocido para él (Back-End).

El Administrador de Bases de Datos (DBA) es la persona o grupo de personas encargada del control general de la base de datos. Sus funciones son:

- Decidir el contenido de la Base de Datos.
- Decidir la estructura de almacenamiento y la estrategia de acceso.
- Vincularse con los usuarios.
- Definir los controles de autorización y procedimientos de validación.
- Definir una estrategia de respaldo y recuperación.
- Controles de desempeño y responder a los cambios de requerimientos.

Una base de datos se conforma de la siguiente manera:

- El Sistema de Gestión de Bases de Datos (SGBD/DBMS) o motor.
- Programas de aplicación.
- Usuarios (finales, administradores de bases de datos (DBA), programadores, entre otros).
- Computadoras.
- Programas de aplicación específica como generadores de informes, de interfaces, herramientas de desarrollo y de administración, etc.

2.1.5. El Sistema de Gestión de Bases de Datos, SGBD.

Un sistema de gestión de bases de datos consiste de una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. La colección de datos es la base de datos, y es la que contiene información por ejemplo acerca de una empresa determinada.

El objetivo principal de un SGBD es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información en la base de datos.

2.1.6. Características de las Bases de Datos.

Hay tres características importantes inherentes a los sistemas de bases de datos: La separación entre los programas de aplicación y los datos, el manejo de múltiples vistas por parte de los usuarios y el uso de un catálogo para almacenar el esquema de la base de datos. En 1975, el comité ANSI-SPARC (American National Standard Institute - Standards Planning and Requirements Committee)¹² propuso una arquitectura de tres niveles para los sistemas de bases de datos, que resulta muy útil a la hora de conseguir estas tres características.

El objetivo de la arquitectura de tres niveles es el de separar los programas de aplicación de la base de datos física. En esta arquitectura, el esquema de una base de datos se define en tres niveles de abstracción distintos:

¹² ANSI-SPARC. (American National Standard Institute - Standards Planning and Requirements Committee). Instituto Nacional Americano de Estándares - Comité de Planeación y Requerimientos Estándares.

En el *nivel interno* se describe la estructura física de la base de datos mediante un *esquema interno*. Este esquema se especifica mediante un modelo físico y describe todos los detalles para el almacenamiento de la base de datos, así como los métodos de acceso.

En el *nivel conceptual* se describe la estructura de toda la base de datos para una comunidad de usuarios (todos los de una empresa u organización), mediante un *esquema conceptual*. Este esquema oculta los detalles de las estructuras de almacenamiento y se concentra en describir entidades, atributos, relaciones, operaciones de los usuarios y restricciones. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar el esquema.

En el *nivel externo* se describen varios *esquemas externos* o *vistas de usuario*. Cada esquema externo describe la parte de la base de datos que interesa a un grupo de usuarios determinado, y oculta a ese grupo el resto de la base de datos. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar los esquemas.

La mayoría de los SGBD no distinguen del todo los tres niveles. Algunos incluyen detalles del nivel físico en el esquema conceptual. En casi todos los SGBD que se manejan vistas de usuario, los esquemas externos se especifican con el mismo modelo de datos que describe la información a nivel conceptual, aunque en algunos se pueden utilizar diferentes modelos de datos en el nivel conceptual y externo. Hay que destacar que los tres esquemas no son más que descripciones de los mismos datos pero con distintos niveles de abstracción. Los únicos datos que existen realmente están a nivel físico, almacenados en un dispositivo como puede ser un disco. En un SGBD basado en la arquitectura de tres niveles, cada grupo de usuarios hace referencia exclusivamente a su propio esquema externo. Por lo tanto, el SGBD debe transformar cualquier petición expresada en términos de un esquema externo a una petición expresada en términos del esquema conceptual, y luego, a una petición en el esquema interno, que se procesará sobre la base de datos almacenada. Si la petición es de una obtención (consulta) de datos, será preciso modificar el formato de la información extraída de la base de datos almacenada, para que coincida con la vista externa del usuario. El proceso de transformar peticiones y resultados de un nivel a otro se denomina *correspondencia* o *transformación*. Estas correspondencias pueden requerir bastante tiempo, por lo que algunos SGBD no cuentan con vistas externas. La arquitectura de tres niveles es útil para explicar el concepto de *independencia de datos* que podemos definir como la capacidad para modificar el esquema en un nivel del sistema sin tener que modificar el esquema del nivel inmediato superior. Se pueden definir dos tipos de independencia de datos:

La *independencia lógica* es la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación. Se puede modificar el esquema conceptual para ampliar la base de datos o para reducirla. Si, por ejemplo, se reduce la base de datos

eliminando una entidad, los esquemas externos que no se refieran a ella no deberán verse afectados.

La *independencia física* es la capacidad de modificar el esquema interno sin tener que alterar el esquema conceptual (o los externos). Por ejemplo, puede ser necesario reorganizar ciertos archivos físicos con el fin de mejorar el rendimiento de las operaciones de consulta o de actualización de datos. Dado que la independencia física se refiere sólo a la separación entre las aplicaciones y las estructuras físicas de almacenamiento, es más fácil de conseguir que la independencia lógica.

Por lo tanto, la arquitectura de tres niveles puede facilitar la obtención de la verdadera independencia de datos, tanto física como lógica. Sin embargo, los dos niveles de correspondencia implican un gasto extra durante la ejecución de una consulta o de un programa, lo cual reduce la eficiencia del SGBD. Es por esto, que muy pocos SGBD han implementado esta arquitectura completa, resumiendo las características más importantes de una base de datos son las siguientes:

- Considera interrelaciones entre datos y restricciones semánticas que deben estar presentes en una base de datos. Una base de datos no sólo debe almacenar entidades y atributos, sino que también debe almacenar interrelaciones entre datos.
- La redundancia de datos debe ser controlada, de forma que no existan duplicidades perjudiciales ni innecesarias. En las bases de datos no está permitida la redundancia lógica, pero si se admite cierta redundancia física por motivos de eficiencia.
- Las bases de datos pretenden servir a toda la organización, es decir; a múltiples usuarios y a diferentes aplicaciones.
- La independencia lógica y física de los datos.
- La definición y descripción del conjunto de datos contenido en la base debe ser única e integrada con los mismos datos.
- La actualización y recuperación de las bases de datos debe realizarse mediante procesos bien determinados, de modo que se mantenga la integridad, seguridad y confidencialidad de la base.

2.1.7. Modelo Relacional.

2.1.7.1. Introducción.

El modelo relacional se basa en dos ramas de las matemáticas: la teoría de conjuntos y la lógica de predicados de primer orden. El hecho de que el modelo relacional esté basado en la teoría de las matemáticas es lo que lo hace tan seguro y robusto. Al mismo tiempo, estas ramas de las

matemáticas proporcionan los elementos básicos necesarios para crear una base de datos relacional con una buena estructura, y proporcionan las líneas que se utilizan para formular buenas metodologías de diseño.

Hay quien ofrece una cierta resistencia a estudiar complicados conceptos matemáticos para tan solo llevar a cabo una tarea bastante concreta. Las matemáticas son básicas en el modelo relacional. La teoría matemática proporciona la base para el modelo relacional y, por lo tanto, hace que el modelo sea predecible, fiable y seguro. La teoría describe los elementos básicos que se utilizan para crear una base de datos relacional y proporciona las líneas a seguir para construirla. El organizar estos elementos para conseguir el resultado deseado es lo que se denomina diseño."

En 1970, el modo en que se veían las bases de datos cambió por completo cuando E. F. Codd introdujo el modelo relacional. En aquellos momentos, el enfoque existente para la estructura de las bases de datos utilizaba punteros físicos (direcciones de disco) para relacionar registros de distintos archivos. Si, por ejemplo, se quería relacionar un registro *A* con un registro *B*, se debía añadir al registro *A* un campo conteniendo la dirección en disco del registro *B*. Este campo añadido, un puntero físico, siempre señalaría desde el registro *A* al registro *B*. Codd demostró que estas bases de datos limitaban en gran medida los tipos de operaciones que los usuarios podían realizar sobre los datos. Además, estas bases de datos eran muy vulnerables a cambios en el entorno físico. Si se añadían los controladores de un nuevo disco al sistema y los datos se movían de una localización física a otra, se requería una conversión de los archivos de datos. Estos sistemas se basaban en el modelo de red y el modelo jerárquico, los dos modelos lógicos que constituyeron la primera generación de los SGBD.

El modelo relacional representa la segunda generación de los SGBD. En él, todos los datos están estructurados a nivel lógico como tablas formadas por filas y columnas, aunque a nivel físico pueden tener una estructura completamente distinta. Un punto fuerte del modelo relacional es la sencillez de su estructura lógica. Pero detrás de esa simple estructura hay un fundamento teórico importante del que carecen los SGBD de la primera generación, lo que constituye otro punto a su favor.

Dada la popularidad del modelo relacional, muchos sistemas de la primera generación se han modificado para proporcionar una interfaz de usuario relacional, con independencia del modelo lógico que soportan (de red o jerárquico). Por ejemplo, el sistema de red IDMS ha evolucionado a IDMS/R e IDMS/SQL, ofreciendo una visión relacional de los datos.

En los últimos años, se han propuesto algunas extensiones al modelo relacional para capturar mejor el significado de los datos, para disponer de los conceptos de la orientación a objetos y para disponer de capacidad deductiva.

El modelo relacional, como todo modelo de datos, tiene que ver con tres aspectos de los datos:

- Estructura de datos.
- Integridad de datos.
- Manejo de datos.

2.1.7.2. Estructura del Modelo Relacional.

El modelo relacional se basa en el concepto matemático de *relación*, que gráficamente se representa mediante una tabla. Codd, que era un experto matemático, utilizó una terminología perteneciente a las matemáticas, en concreto de la teoría de conjuntos y de la lógica de predicados.

El concepto de dominio es importante porque permite que el usuario defina, en un lugar común, el significado y la fuente de los valores que los atributos pueden tomar. Esto hace que haya más información disponible para el sistema cuando éste va a ejecutar una operación relacional, de modo que las operaciones que son semánticamente incorrectas, se pueden evitar. Por ejemplo, no tiene sentido comparar el nombre de una calle con un número de teléfono, aunque los dos atributos sean cadenas de caracteres. Sin embargo, el importe mensual del alquiler de un inmueble no estará definido sobre el mismo dominio que el número de meses que dura el alquiler, pero sí tiene sentido multiplicar los valores de ambos dominios para averiguar el importe total al que asciende el alquiler. Los SGBD relacionales no ofrecen un soporte completo de los dominios ya que su implementación es extremadamente compleja.

- Una tupla es una fila de una relación.
- El grado de una relación es el número de atributos que contiene.
- La cardinalidad de una relación es el número de tuplas que contiene.
- Una base de datos relacional es un conjunto de relaciones normalizadas.

Gráficamente se suelen representar las relaciones mediante tablas. Los nombres de las columnas corresponden a los nombres de los atributos y las filas son cada una de las tuplas de la relación. Los valores que aparecen en cada una de las columnas pertenecen al conjunto de valores del dominio sobre el que está definido el atributo correspondiente.

Una clave o llave candidata de una relación. Es un conjunto no vacío de atributos que identifican unívoca y mínimamente cada tupla. Toda relación siempre tendrá una llave candidata.

Clave o llave primaria. Es aquella llave candidata que el usuario elegirá, por consideraciones ajenas al modelo relacional, para identificar las tuplas

de la relación. El modelo relacional no incluye este concepto de elegir una llave como primaria, cuando hay varias candidatas.

Clave o llave alternativas. Son aquellas llaves candidatas que no han sido escogidas como llaves primarias.

Clave o llave ajena o foránea de una relación R2. Es un conjunto no vacío de atributos cuyos valores han de coincidir con los valores de la llave primaria de una relación R1 (R1 y R2 no son necesariamente distintas). Notar que la llave ajena y la correspondiente llave primaria han de estar definidas sobre los mismos dominios.

2.1.7.3. Proceso de Normalización.

El proceso de normalización es un estándar que consiste, básicamente, en un proceso de conversión de las relaciones entre las entidades, evitando:

- La redundancia de los datos: repetición de datos en un sistema.
- Anomalías de actualización: inconsistencias de los datos como resultado de datos redundantes y actualizaciones parciales.
- Anomalías de borrado: pérdidas no intencionadas de datos debido a que se han borrado otros datos.
- Anomalías de inserción: imposibilidad de adicionar datos en la base de datos debido a la ausencia de otros datos.

2.1.7.4. Primera Forma Normal (1NF).

Se dice que una tabla se encuentra en primera forma normal (1NF) si y sólo si cada uno de los campos contiene un único valor o valor atómico para un registro determinado.

2.1.7.5. Segunda Forma Normal (2NF).

La segunda forma normal compara todos y cada uno de los campos de la tabla con la llave definida. Si todos los campos dependen directamente de la llave se dice que la tabla está en segunda forma normal (2NF).

2.1.7.6. Tercera Forma Normal (3NF).

Se dice que una tabla está en tercera forma normal si y sólo si los campos de la tabla dependen únicamente de la llave, dicho en otras palabras los campos de las tablas no dependen unos de otros.

2.1.8. Interrelaciones.

2.1.8.1. Relaciones Uno A Uno.

Una interrelación es de uno a uno entre la tabla A y la tabla B cuando a cada elemento de la tabla A se le asigna un único elemento de la tabla B y para cada elemento de la tabla B contiene un único elemento en la tabla A.

2.1.8.2. Relaciones Uno a Muchos.

Una interrelación es de uno a varios entre las tablas A y B cuando un elemento de la tabla A posee varios elementos relacionados en la tabla B y cuando un elemento de la tabla B posee un único elemento relacionado en la tabla A.

2.1.8.3. Relaciones Muchos A Muchos.

Una interrelación es de varios a varios entre las tablas A y B cuando un elemento de la tabla A posee varios elementos relacionados en la tabla B y cuando un elemento de la tabla B posee varios elementos relacionados en la tabla A.

2.1.9. Álgebra Relacional.

Las operaciones de álgebra relacional manipulan relaciones. Esto significa que estas operaciones usan uno o dos relaciones existentes para crear una nueva relación. Esta nueva relación puede entonces usarse como entrada para una nueva operación. Este poderoso concepto - la creación de una nueva relación a partir de relaciones existentes hace considerablemente más fácil la solución de las consultas, debido a que se puede experimentar con soluciones parciales hasta encontrar la proposición con la que se trabajará.

El álgebra relacional consta de nueve operaciones:

Unión. La operación de unión permite combinar datos de varias relaciones. Para poder realizar esta operación es necesario e imprescindible que las tablas a unir tengan las mismas estructuras, es decir, que sus campos sean iguales.

Intersección. La operación de intersección permite identificar filas que son comunes en dos relaciones.

Diferencia. La operación diferencia permite identificar filas que están en una relación y no en otra.

Producto. La operación producto consiste en la realización de un producto cartesiano entre dos tablas dando como resultado todas las posibles

combinaciones entre los registros de la primera y los registros de la segunda.

Selección. La operación selección consiste en recuperar un conjunto de registros de una tabla o de una relación indicando las condiciones que deben cumplir los registros recuperados, de tal forma que los registros devueltos por la selección han de satisfacer todas las condiciones que se hayan establecido. Esta operación es la que normalmente se conoce como consulta.

Proyección. Una proyección es un caso concreto de la operación selección, esta última devuelve todos los campos de aquellos registros que cumplen la condición que he establecido. Una proyección es una selección en la que seleccionamos aquellos campos que deseamos recuperar.

Reunión. La reunión se utiliza para recuperar datos a través de varias tablas conectadas unas con otras mediante cláusulas JOIN, en cualquiera de sus tres variantes INNER¹³, LEFT¹⁴, RIGHT¹⁵. La operación reunión se puede combinar con las operaciones selección y proyección.

División. Esta operación algebraica consiste en asignar un valor a uno o varios campos de una tabla.

Asignación. Esta operación algebraica consiste en asignar un valor a uno o varios campos de una tabla.

Las cuatro primeras se toman de la teoría de conjunto de las matemáticas; las cuatro siguientes son operaciones propias del álgebra relacional y la última es la operación estándar de dar un valor a un elemento.

2.1.10. Cálculo Relacional.

El cálculo relacional usa un enfoque completamente diferente al álgebra relacional. No obstante, los dos lenguajes son lógicamente equivalentes. Esto significa que cualquier consulta que pueda resolverse en un lenguaje puede resolverse en el otro. La solución para toda consulta en este tipo de

¹³ INNER JOIN: Muestra de la primera tabla todos los registros que estén relacionados con registros de la tabla a la que se va a unir (segunda tabla), y de la tabla segunda todos aquellos registros que estén relacionados con registros de la primera tabla. Es decir, valores que no estén relacionados no me los va a mostrar.

¹⁴ LEFT JOIN: Muestra de la primera tabla todos los registros estén o no relacionados con registros de la tabla de la derecha, mientras que de la tabla de la derecha sólo cogerá aquellos registros que estén relacionados con registros de la tabla de la izquierda.

¹⁵ RIGHT JOIN: es el caso contrario del Left Join.

cálculo se define por una lista de resultados y una sentencia de calificación.

La lista de resultados está clara, son aquellos registros que cumplen las condiciones que deseamos. La sentencia de calificación contiene las condiciones que deseamos que cumplan los registros de la lista de resultados. La diferencia entre el cálculo y el álgebra radica en que el cálculo realiza la operación en un único paso, sin necesidad de tener que obtener tablas intermedias, el álgebra realiza las operaciones paso a paso. Normalmente el cálculo relacional se apoya en algún lenguaje de interrogación de bases de datos como puede ser el SQL y que se desarrolla más adelante.

El cálculo relacional incluye un concepto nuevo denominado cuantificador, los cuantificadores tratan de averiguar el número de registros afectados por una determinada operación, incluso antes de realizarla. Se dividen en:

Cuantificadores existenciales. Son aquellos que tratan de averiguar el número de registros que devolvería un tipo de consulta.

Cuantificadores universales. Son aquellos que indican que una condición se aplica a todas las filas de algún tipo. Se usa para brindar la misma capacidad que la operación división del álgebra relacional.

2.2. INGENIERÍA DE SOFTWARE.

2.2.1. Introducción.

Durante las tres primeras décadas, el principal desafío fue el desarrollo del hardware de las computadoras, de forma que se redujera el costo de procesamiento y almacenamiento de datos.

A lo largo de la década de los 80's, los avances en microelectrónica han dado como resultado una mayor potencia de cálculo a la vez que una reducción del costo. Hoy, el problema es diferente, el principal desafío es mejorar la calidad y reducir el costo de las soluciones basadas en computadoras; soluciones que se implementan con el software.

La potencia de las grandes computadoras de la era de los 80's está hoy disponible en una computadora personal. Las enormes capacidades de procesamiento y almacenamiento del hardware moderno representan un gran potencial de cálculo. El software es el mecanismo que nos facilita utilizar y explotar este potencial.³

2.2.2. Antecedentes y Evolución.

Con el acelerado avance tecnológico de la información, la cantidad y la complejidad de los productos de software se están incrementando considerablemente, así como también la exigencia en su funcionalidad y confiabilidad; por esto la calidad y la productividad se están constituyendo en las grandes preocupaciones tanto de gestores como para desarrolladores de software.

En los primeros años del software, las actividades de elaboración de programas eran realizadas por una sola persona utilizando lenguajes de bajo nivel y ajustándose a un computador en especial, que generaban programas difíciles de entender, aún hasta para su creador, después de algún tiempo de haberlo producido. Esto implicaba tener que repetir el mismo proceso para desarrollar el mismo programa para otras máquinas. Por consiguiente, la confiabilidad, facilidad de mantenimiento y cumplimiento no se garantizaban y la productividad era muy baja.

Posteriormente, con la aparición de técnicas estructuradas y con base en las experiencias de los programadores se mejoró la productividad del software. Sin embargo, este software seguía teniendo fallas, como por ejemplo: documentación inadecuada, dificultad para su correcto funcionamiento, y por su puesto, insatisfacción de cliente.

Conforme se incrementaba la tecnología de los computadores, también crecía la demanda de los productos de software, pero mucho más lentamente, tanto que hacia 1990 se decía que las posibilidades de software estaban retrasadas respecto a las del hardware en un mínimo de dos generaciones de procesadores y que la distancia continuaba aumentando.

En la actualidad muchos de estos problemas subsisten en el desarrollo de software, con una dificultad adicional relacionada con la incapacidad para satisfacer totalmente la gran demanda y exigencias por parte de los clientes.

2.2.3. Definición de Programa.

El elemento básico del software es el programa. Un programa es un grupo de instrucciones destinadas a cumplir una tarea en particular. Un programa puede estar conformado por varios programas más sencillos.

El software se puede clasificar en tres grupos: sistemas operativos, lenguajes de programación y aplicaciones.

2.2.4. Metas de la Ingeniería de Software.

Las metas primordiales de ésta disciplina tecnológica son mejorar la calidad de los productos de programación que son desarrollados y modificados a tiempo y dentro de un presupuesto definido. Aumentar la productividad y aumentar la satisfacción profesional de los ingenieros.

2.2.5. Modelos de la Ingeniería de Software.

Reconocer los problemas, causas y destruir los mitos del software son los primeros pasos a solucionar. Pero las propias soluciones tienen que proporcionar asistencia práctica a la persona que desarrolla un software, mejorar la calidad del software, por último permitir al mundo del software mantenerse en paz con el mundo del hardware.

No existe un único enfoque para solucionar el mal del software. Sin embargo, para solucionar este problema se tendrán que utilizar:

- Combinación de métodos completos para todas las fases del desarrollo del software.
- Mejores herramientas para automatizar estos métodos.
- Bloques de construcción más potentes para la implementación de software.
- Técnicas para la garantía de calidad del software y una filosofía predominante para la coordinación, control y gestión para conseguir una disciplina para el desarrollo del software.

Tipos de Modelos:

Modelo cascada, modelo prototipo, modelo espiral, técnicas de cuarta generación.

2.2.6. Modelo Clásico.

2.2.6.1. Definición del Modelo Clásico.

Es la sucesión de etapas por las que pasa el software desde que un nuevo proyecto es concebido hasta que se deja de usar. Cada una de estas etapas lleva asociada una serie de tareas que deben realizarse, y una serie de documentos (en sentido amplio: software) que serán la salida de cada una de estas fases y servirán de entrada en la fase siguiente.

Existen diversos ciclos de vida (Figura 2.2.6.1.1), es decir, diversas formas de ver el proceso de desarrollo de software, y cada uno de ellos va asociado a un paradigma de la ingeniería del software, es decir, a una serie de métodos, herramientas y procedimientos que debemos usar a lo largo de un proyecto. En este tema veremos algunos de los principales modelos de ciclo de vida clásico.

La elección de un modelo se realiza de acuerdo con la naturaleza del proyecto y de la aplicación, los métodos a usar y los controles y entregas requeridos.

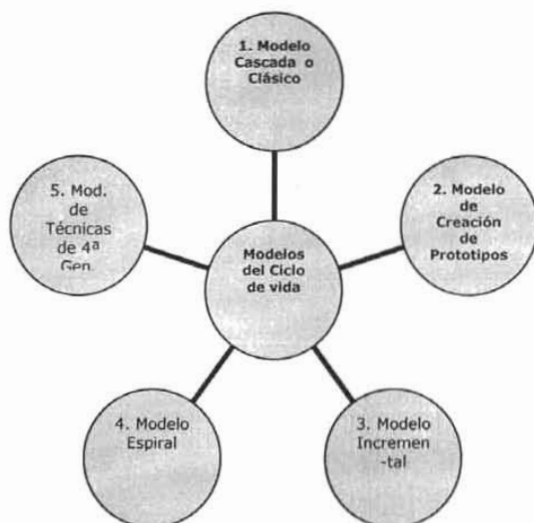


Figura 2.2.6.1.1. Modelos de Ciclos de Vida Clásico.

La función principal de un modelo es establecer el orden en el que se especifica, se realizan los propósitos, se diseña, implementa, revisa, prueba y se realizan otras actividades en un proyecto.

2.2.6.2. El Modelo en Cascada (Clásico).

El Modelo en cascada exige un enfoque sistemático y secuencial del desarrollo de software, que comienza en el nivel de la ingeniería de sistemas y avanza a través de fases secuenciales sucesivas (ver figura 2.2.6.1.2.) Estas fases son las siguientes:

Ingeniería y análisis del sistema. El software es siempre parte de un sistema mayor, por lo que siempre va a interrelacionarse con otros elementos, ya sea hardware, máquinas o personas. Por esto, el primer paso del Modelo de un proyecto consiste en un análisis de las características y el comportamiento del sistema del cual el software va a formar parte. En el caso de que queramos construir un sistema nuevo, por ejemplo un sistema de control, deberemos analizar cuáles son los requisitos y la función del sistema, y luego asignaremos un subconjunto de estos requisitos al software. En el caso de un sistema ya existente (supongamos, por ejemplo, que queremos informatizar una empresa) deberemos analizar el funcionamiento de la misma, - las operaciones que

se llevan a cabo en ella-, asignaremos al software aquellas funciones que vamos a automatizar.

La ingeniería de sistemas comprende, por tanto, los requisitos globales a nivel del sistema, así como una cierta cantidad de análisis y de diseño a nivel superior, es decir sin entrar en mucho detalle.

Este modelo es el más antiguo de los empleados en Ingeniería de Software y se desarrolló a partir del ciclo convencional de una ingeniería. No hay que olvidar que la Ingeniería de Software surgió como copia de otras ingenierías, especialmente de las ingenierías de hardware, para dar solución a los problemas más comunes que aparecían al desarrollar sistemas de software complejos, siendo sus objetivos los siguientes.

- Definir las actividades a llevarse a cabo en un proyecto de desarrollo de sistemas.
- Lograr congruencia entre la multitud de proyectos de desarrollo de sistemas en una misma organización.
- Proporcionar puntos de control y revisión administrativos de las decisiones sobre continuar o no con un proyecto.

Características:

- Tendencia a la implantación ascendente del sistema.
- La progresión lineal y secuencial de una fase a la siguiente.

Desventajas:

- Los proyectos reales raramente siguen el flujo secuencial que propone el modelo.
- Normalmente, es difícil para el cliente establecer explícitamente al principio todos los requerimientos.
- La dificultad para especificar claramente los requerimientos al comienzo del proyecto, antes de que se realice ningún trabajo de diseño y antes de escribir ningún código.
- No permite flexibilidad en los cambios y el cliente debe de tener paciencia.

Propósitos:

- Establece los criterios que se utilizan para determinar el paso de una actividad a otra.
- Está dirigido por documentos; con esto se dice que los productos del trabajo que se pasan de etapa a etapa son documentos.

- Ayuda a minimizar los gastos de la planificación ya que estos se realizan sin problemas.
- Este modelo funciona bien con proyectos complejos que se entienden correctamente ya que se puede obtener un gran beneficio atacando esta complejidad en forma ordenada.
- Evita una fuente común de errores ya que, elimina los cambios que se pueden producir a medio camino.
- Se puede contar con gente poco calificada o inexperta ya que presenta el proyecto con una estructura que ayuda a minimizar el esfuerzo inútil.

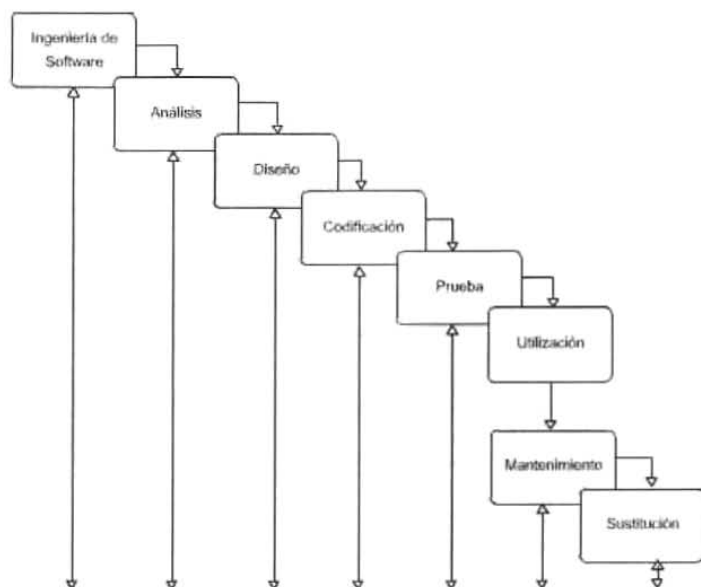


Figura 2.2.6.1.2. Modelo de Ciclos de Vida en Cascada.

Análisis de requisitos del software. El análisis de requisitos debe ser más detallado para aquellos componentes del sistema que vamos a implementar mediante software. El ingeniero del software debe comprender cuáles son los datos que se van a manejar, cuál va a ser la función que tiene que cumplir el software, cuáles son las interfaces requeridas y cuál es el rendimiento que se espera lograr. Los requisitos, tanto del sistema como del software deben documentarse y revisarse con el cliente.

Diseño. El diseño se aplica a cuatro diferentes características del software: la estructura de los datos, la arquitectura de las aplicaciones, la estructura interna de los programas y las interfaces. El diseño es el proceso que traduce los requisitos en una representación del software de forma que

pueda conocerse la arquitectura, funcionalidad e incluso la calidad del mismo antes de comenzar la codificación. Al igual que el análisis, el diseño debe documentarse y forma parte de la configuración del software (el control de configuraciones es lo que nos permite realizar cambios en el software de forma controlada y no traumática para el cliente).

Codificación. La codificación consiste en la traducción del diseño a un formato que sea legible para la máquina. Si el diseño es lo suficientemente detallado, la codificación es relativamente sencilla, y puede hacerse - al menos en parte - de forma automática, usando generadores de código. En el diseño, los requisitos del software se traducen a una serie de diagramas que representan la estructura del sistema software, de sus datos, de sus programas y de sus interfaces. Por último, en esta etapa de codificación se traducen estos diagramas de diseño a un lenguaje fuente, que luego se traduce, es decir, se compila.

Pruebas. Una vez que ya tenemos el programa ejecutable, comienza la fase de pruebas. El objetivo es comprobar que no se hayan producido errores en alguna de las fases de traducción anteriores, especialmente en la codificación. Para ello deben probarse todas las sentencias, no sólo los casos normales y todos los módulos que forman parte del sistema.

Utilización. Una vez superada la fase de pruebas, el software se entrega al cliente y comienza la vida útil del mismo. La fase de utilización se solapa con las posteriores -el mantenimiento y la sustitución- y dura hasta que el software, ya reemplazado por otro, deje de utilizarse.

Mantenimiento. El software sufrirá cambios a lo largo de su vida útil. Estos cambios pueden ser debidos a tres causas:

- Que, durante la utilización, el cliente detecte errores en el software: los errores latentes.
- Que se produzcan cambios en alguno de los componentes del sistema informático: por ejemplo cambios en la máquina, en el sistema operativo o en los periféricos.
- Que el cliente requiera modificaciones funcionales (normalmente ampliaciones) no contempladas en el proyecto.

En cualquier caso, el mantenimiento supone volver atrás en el Modelo, a las etapas de codificación, diseño o análisis dependiendo de la magnitud del cambio.

El modelo en cascada, a pesar de ser lineal, contiene flujos que permiten la vuelta atrás. Así, desde el mantenimiento se vuelve al análisis, el diseño o la codificación, y también desde cualquier fase se puede volver a la anterior si se detectan fallos. Estas vueltas atrás no son controladas, ni quedan explícitas en el modelo, y es uno de los problemas que presenta éste paradigma.

Sustitución. La vida del software no es ilimitada y cualquier aplicación, por buena que sea, acaba por ser sustituida por otra más amplia, más rápida o más bonita y fácil de usar.

La sustitución de un software que está funcionando por otro que acaba de ser desarrollado es una tarea que hay que planificar cuidadosamente y que hay que llevar a cabo de forma organizada. Es conveniente realizarla por fases, si esto es posible, no sustituyendo todas las aplicaciones de golpe, puesto que la sustitución conlleva normalmente un aumento de trabajo para los usuarios, que tienen que acostumbrarse a las nuevas aplicaciones, y también para los implementadores, que tienen que corregir los errores que aparecen. La sustitución implica el desarrollo de programas para la interconexión de ambos sistemas, el viejo y el nuevo, y para traspasar la información entre ambos, evitando la duplicación del trabajo de las personas encargadas del proceso de datos, durante el tiempo en que ambos sistemas funcionen en paralelo.

2.2.7. Construcción de Prototipos.

La construcción de prototipos es un proceso que facilita al programador la creación del modelo de software a construir (ver figura 2.2.7.1.) El modelo tomará una de las tres formas siguientes:

- Un prototipo en papel o un modelo basado en PC que describa la iteración hombre-máquina.
- Un prototipo que implemente algunos subconjuntos de la función requerida del programa deseado.
- Un programa existente que ejecute parte o toda la función deseada, pero que tenga otras características que deban ser mejoradas en el nuevo trabajo de desarrollo.

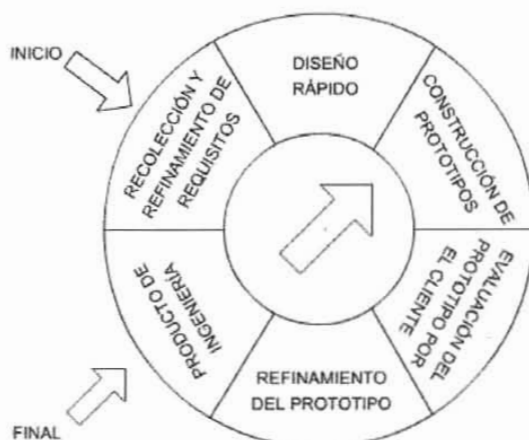


Figura 2.2.7.1. Construcción de Prototipos.

2.2.8. El Modelo Incremental.

Pertenciente a la familia de los Modelos de Procesos Evolutivos, el Modelo Incremental combina elementos del MLS (Modelo Lineal Secuencial) con la filosofía interactiva de construcción de prototipos.

En una visión genérica, el proceso se divide en 4 partes: Análisis, diseño, Código y Prueba. Sin embargo, para la producción del Software, se usa el principio de trabajo en cadena o "Pipeline", utilizado en muchas otras formas de programación. Con esto se mantiene al cliente en constante contacto con los resultados obtenidos en cada incremento. Es el mismo cliente el que incluye o desecha elementos al final de cada incremento a fin de que el software se adapte mejor a sus necesidades reales. El proceso se repite hasta que se elabore el producto completo. De esta forma el tiempo de entrega se reduce considerablemente. Al igual que los otros métodos de modelado, el Modelo Incremental es de naturaleza interactiva pero se diferencia de aquellos en que al final de cada incremento se entrega un producto completamente operacional.

El Modelo Incremental es particularmente útil cuando no se cuenta con una dotación de personal suficiente. Los primeros pasos los pueden realizar un grupo reducido de personas y en cada incremento se añade personal, de ser necesario. Por otro lado los incrementos se pueden planear para gestionar riesgos técnicos.

2.2.9. Modelo Espiral.

El modelo espiral, propuesto originalmente por Boehm, es un modelo de proceso de software evolutivo que conjuga la naturaleza iterativa de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial (ver figura 2.2.9.1).



Figura 2.2.9.1. Modelo en Espiral.

Proporciona el potencial para el desarrollo rápido de versiones incrementales del software. Durante las primeras iteraciones, la versión incremental podría ser: un modelo en papel o prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas del sistema diseñado.

El modelo en espiral se divide en un número de actividades de trabajo, también llamadas regiones de tareas. Generalmente existen entre tres y seis regiones de tareas:

Comunicación con el cliente. Las tareas requeridas para establecer comunicación entre el desarrollador y el cliente.

Planificación. Las tareas requeridas para definir recursos, tiempo y otra información relacionada con el proyecto.

Ingeniería. Las tareas requeridas para construir una o más representaciones de la aplicación.

Construcción y acción. Las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario.

Evaluación del cliente. Las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de la ingeniería e implementada durante la etapa de instalación.

2.2.10. Uso de Técnicas de Cuarta Generación (T4G).

Por técnicas de cuarta generación se entiende un conjunto muy diverso de Métodos y herramientas que tienen por objeto el facilitar el desarrollo de software. El Modelo T4G facilita al ingeniero de software, la especificación de algunas características del software de alto nivel. Las herramientas generan automáticamente el código fuente a partir de esta especificación. El paradigma T4G usa formas de leguaje especializado o notación gráficas que describan el problema que hay que resolver en términos que entienda el cliente, Un entorno para el desarrollo de software que soporte el paradigma T4G puede incluir las siguientes herramientas:

- Acceso a bases de datos utilizando lenguajes de conducta de alto nivel.
- Generación de código.
- Generación de pantallas.
- Gestión de entornos gráficos.
- Generación de informes.
- La generación automática de código permite reducir la duración de las fases del Modelo clásico. Las T4G comienzan con el paso de reunión de requisitos, después

son traducidos directamente a un prototipo, (observar la figura 2.2.10.1).

Ventajas:

- Permite al programador especificar características del software con un alto grado de abstracción y la herramienta genera el código basándose en esta especificación.
- Reducción del tiempo de desarrollo y mejoras de productividad para quienes hacen el software.
- Reducción del tiempo necesario para la producción de aplicaciones pequeñas e intermedias.
- Reducción de la cantidad de diseño y análisis necesario para aplicaciones pequeñas.
- Las t4g nos ayudan a resolver problemas de desarrollo de software ofreciendo una solución fiable porque permite al desarrollador centrarse en la representación de los resultados deseados.
- Utiliza generadores de código, eliminando la etapa de codificación.

Desventajas:

- No es tan fácil de usar como los lenguajes de programación.
- La capacidad de mantenimiento de grandes sistemas no está determinada.
- Se limita aplicaciones de negocios y trabajos con grandes bases de datos.
- Demanda un minucioso análisis, diseño y evaluación en aplicaciones grandes para lograr reducciones de tiempo.

TÉCNICAS DE 4ª GENERACIÓN

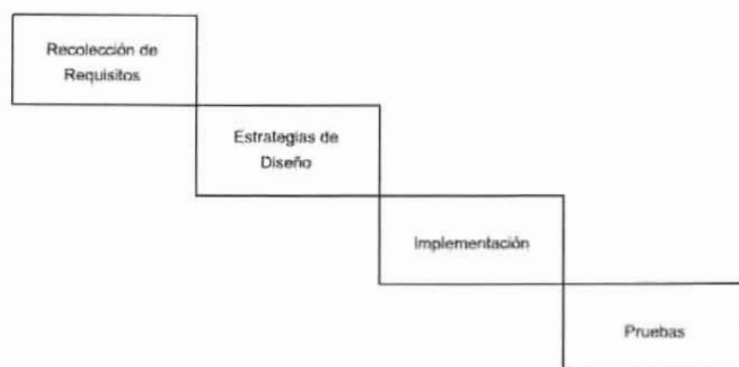


Figura 2.2.10.1. Técnicas de cuarta generación.

Recolección de Requisitos. El cliente describe requisitos, que son traducidos directamente a un prototipo operativo. Es muy difícil que se puedan establecer todos los requisitos desde el comienzo: El cliente puede no estar seguro de lo que necesita, o aunque lo sepa, puede ser difícil explicar la forma en como trabaja la herramienta de T4G para poder entenderla.

Estrategias de diseño. Para aplicaciones pequeñas, se puede ir directamente desde el paso de recolección de requisitos al paso de implementación, usando un lenguaje de cuarta generación no procedimental¹⁶ (L4G). Sin embargo es necesario un mayor esfuerzo para desarrollar una estrategia de diseño para el sistema, incluso si se utiliza un L4G. El uso de T4G sin diseño (para grandes proyectos) causara las mismas dificultades (poca calidad, mantenimiento pobre, mala aceptación para el cliente) que se encuentra cuando se desarrolla software mediante los enfoques convencionales.

Implementación en L4G. La implementación mediante un T4G permite, al que desarrolla el software, centrarse en la representación de los resultados deseados, que es lo que se traduce automáticamente en un código fuente que produce dichos resultados.

Pruebas o Evaluación. Para transformar una implementación t4g en un producto, el que lo desarrolla debe dirigir una prueba completa desarrollar una documentación con sentido y ejecutar el resto de las actividades, de "transición" requeridas en los otros paradigmas de ing. de software.

2.2.11. Combinación de Prototipos.

El trabajo comienza con la determinación de objetivos, alternativas y restricciones, paso que a veces se llama recolección preliminar de requisitos. Si los requisitos son inciertos, se puede usar un prototipo para definir mejor los requisitos usando el prototipo como guala, el que desarrolla puede después volver a los pasos del Modelo clásico (diseño, codificación, prueba). Alternativamente, el prototipo puede evolucionar hacia el sistema a producir, con una vuelta al ciclo de vida en el momento de la etapa de prueba. Se pueden usar las técnicas de cuarta generación para implementar el prototipo o para implementar el sistema durante el paso de codificación del Modelo clásico. También se puede usar un lenguaje de cuarta generación junto con el modelo en espiral en los pasos de creación de prototipos o de codificación.

No se necesita ser dogmático para elegir paradigmas para la ingeniería de software; la naturaleza de la aplicación debe dictar el método a elegir. Combinando prototipos, EL TODO puede ser mejor que LA SUMA de las partes.

¹⁶ Procedimental o Procesal. Del Inglés Procedural.

2.1.12. Análisis y Diseño Orientado a Objetos con UML.

Para el desarrollo de este sistema hemos usado el ciclo de vida clásico, apoyándonos en las herramientas UML, como por ejemplo, diagramas de casos de uso para generar entidades que luego son convertidas a clases. Estas últimas presentan mecanismos que se convierten en diagramas de secuencias y en diagramas de colaboraciones, lo cual nos proporciona el diseño listo para ser codificado.

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece nueve diagramas en los cuales modelar sistemas.

- Diagramas de Casos de Uso para modelar los procesos.
- Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- Diagramas de Colaboración para modelar interacciones entre objetos.
- Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- Diagramas de Componentes para modelar componentes.
- Diagramas de Implementación para modelar la distribución del sistema.

UML es una consolidación de muchas de las notaciones y conceptos más usados orientados a objetos. Empezó como una consolidación del trabajo de Grade Booch, James Rumbaugh, e Ivar Jacobson, creadores de tres de las metodologías orientadas a objetos más populares.

En 1996, el Object Management Group (OMG), un pilar estándar para la comunidad del diseño orientado a objetos, publicó una petición con propósito de un metamodelo orientado a objetos de semántica y notación estándares. UML, en su versión 1.0, fue propuesto como una respuesta a esta petición en enero de 1997. Hubo otras cinco propuestas rivales. Durante el transcurso de 1997, los seis promotores de las propuestas, unieron su trabajo y presentaron al OMG un documento revisado de UML, llamado UML versión 1.1. Este documento fue aprobado por el OMG en

Noviembre de 1997. El OMG llama a este documento OMG UML versión 1.1. El OMG está actualmente en proceso de mejorar una edición técnica de esta especificación, prevista su finalización para el 1 de abril de 1999.

UML prescribe una notación estándar y semánticas esenciales para el modelado de un sistema orientado a objetos. Previamente, un diseño orientado a objetos podría haber sido modelado con cualquiera de la docena de metodologías populares, causando a los revisores tener que aprender las semánticas y notaciones de la metodología empleada antes que intentar entender el diseño en sí. Ahora con UML, diseñadores diferentes modelando sistemas diferentes pueden sobradamente entender cada uno los diseños de los otros.

2.2.13. Aseguramiento de Calidad.

2.1.13.1. Calidad del Software y Garantía de Calidad.

Se define como la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente, por lo que se espera que deba hacer lo siguiente:

- Alcanzar las metas de negocios articuladas por el departamento solicitante (usuarios).
- Operar a un costo aceptable, dimensionalmente congruente con el valor producido para la empresa.
- Cumplir escrupulosamente, con las normas de desempeño definidas (como tiempo de respuesta y disponibilidad de sistema).
- Producir un resultado preciso y confiable.
- Ser fácil de aprender y utilizar.
- Ser flexible.

2.2.13.2. Factores de Calidad del Software.

Hewlett Packard ha desarrollado un conjunto de factores de calidad del software cuyas siglas son FURPS (en inglés). Éstas significan: funcionalidad, facilidad de uso, fiabilidad, rendimiento y capacidad de soporte.

Funcionalidad: Se obtiene mediante la evolución del conjunto de características y de posibilidades del programa, la generalidad de las funciones que se entregan y la seguridad de todo el sistema.

Facilidad de uso: Se calcula considerando los factores humanos, la estética global, la consistencia y la documentación.

Fiabilidad: Se calcula midiendo la frecuencia de fallos y su importancia, la eficacia de los resultados de salida, el tiempo medio entre fallos, la posibilidad de recuperarse a los fallos y la previsibilidad del programa.

Rendimiento: Se mide mediante la evaluación de la velocidad de proceso, tiempo de respuesta, el consumo de recursos, el rendimiento total de procesamiento y la eficiencia.

Capacidad de soporte: (También llamada facilidad de mantenimiento). Combina la posibilidad de ampliar el programa (extensibilidad), la adaptación y la utilidad, además de la factibilidad de prueba, la compatibilidad, la posibilidad de configuración (posibilidad de organizar y controlar elementos de la configuración del software), la facilidad con la que se puede instalar un sistema y la facilidad con la que se pueden localizar los problemas.

2.2.13.3. Revisión de las Técnicas Formales.

La revisión de técnicas formales (RTF), es una actividad de garantía de calidad del software, y es llevada a cabo por los profesionales de la ingeniería de software que tiene como objetivo:

- Describir los errores en la función, a la lógica o la implementación de cualquier representación del software.
- Verificar que el software bajo revisión alcanza los requisitos.
- Garantizar que el software ha sido representado de acuerdo con ciertos estándares predefinidos.
- Conseguir un software desarrollado de forma uniforme.
- Proyectos manejables.
- La revisión de técnicas formales sirve como campo de entrenamiento al permitir que los ingenieros más jóvenes puedan observar los diferentes enfoques de análisis, diseño e implementación, del software, además de promover la seguridad y continuidad.

La reunión de revisiones esta compuesta por las siguientes restricciones:

- Deben convocarse a la reunión (normalmente) entre 3 y 5 personas.
- Se debe preparar por adelantado (puede ser que se requiera más de 2 horas de trabajo de cada persona).
- La reunión debe durar menos de 2 horas.
- El centro de atención de la RTF es un producto (un componente de software). El productor (individuo que desarrolla el producto) informa al jefe del proyecto, que el producto esta terminado y requiere una revisión. El jefe de proyecto contacta con jefe de revisión se encarga de evaluar la disponibilidad del producto, genera copias del

material y las distribuye a dos o tres revisores para que se preparen por adelantado, estarán entre una y dos horas revisando el producto, tomando notas y familiarizándose con el trabajo.

- El jefe de revisión revisa el producto y establece una agenda para la reunión de revisión (entre los revisores y el producto). Uno de los revisores tomara el papel de registrador de los sucesos importantes que se produzca durante la revisión.

La RTF comienza con una explicación de la agenda y una breve introducción a cargo del productos con un "recorrido de inspección" del producto, mientras que los revisores exponen sus inconvenientes basándose en su preparación previa. Al final de la revisión, los participantes de la RTF deben decir si:

- Aceptan el producto sin posteriores modificaciones.
- Rechazan el producto debido a los serios errores encontrados.
- Aceptan el producto provisionalmente (se ha encontrado errores menores que deben ser corregidos, pero sin necesidad de una posterior revisión).

Hay diferentes tipos de revisiones con diferentes objetivos:

- Inspecciones para eliminar defectos (producto).
- Revisiones para estimación de progresos (procesos y producto).
- Revisiones de calidad (estándares y producto).

2.2.13.4. Inspección del Software.

Las inspecciones de software surgen a partir de la necesidad de producir software de alta calidad.

2.2.13.5. El Proceso de Inspección.

Podemos ver a la inspecciones de software como un repaso detallado y formal del trabajo en proceso. Pequeños grupos de trabajadores (4 ó 5) estudian el "producto de trabajo" independientemente y luego se reúnen para examinar el trabajo en detalle. El producto de trabajo representa 200 a 250 líneas de código, requerimientos, diseño y otros productos de trabajo son inspeccionados en un tamaño similar. Los productos de trabajo son considerados en progreso hasta que la inspección y las correcciones necesarias estén completas.

El tiempo de la inspección varía según cuan familiarizado esté el inspector con el material.

La inspección consiste en seis pasos (Fagan 1986):

Planificación: Cuando el desarrollador completa su "producto de trabajo", se forma un grupo de inspección y se designa un moderador. El moderador asegura que el producto de trabajo satisfaga el criterio de inspección. Se le asignan diferentes roles a las personas que integran el grupo de inspección, así como la planificación de tiempos y recursos necesarios.

Vista General (Overview): Si los inspectores no están familiarizados con el desarrollo del proyecto, una vista general es necesaria en este momento. Este es un paso opcional, pero no menos importante ya que en esta etapa se dará al grupo de inspección un "background" y el contexto a cubrir por la inspección.

Preparación: Los inspectores se preparan individualmente para la evaluación en la reunión, estudiando los productos de trabajo y el material relacionado. Aquí es aconsejable la utilización de listas de chequeos para ayudar a encontrar los defectos comunes. El tiempo que puede llevar esta etapa va a depender de cuán familiarizado esté el inspector con el trabajo que debe analizar.

Examen: En esta etapa, los inspectores se reúnen para analizar su trabajo individual en forma conjunta. El moderador deberá asegurarse que todos los inspectores están suficientemente preparados. La persona designada como lector presenta el producto de trabajo, interpretando o parafraseando el texto, mientras que cada participante observa en busca de defectos. Es recomendable que este examen no dure más de 2 horas ya que la atención en busca de defectos va disminuyendo con el tiempo. Al terminar con la reunión, el grupo determina si el producto es aceptado o debe ser reelaborado para una posterior inspección.

Retrabajo: El autor corrige todos los defectos encontrados por los inspectores.

Seguimiento: El moderador chequea las correcciones del autor. Si el moderador está satisfecho, la inspección está formalmente completa y el producto de trabajo es puesto bajo el control de configuración.

2.2.13.6. Métricas de Calidad.

La medición es el proceso por el cual los números o símbolos son asignados a atributos o entidades en el mundo real tal como sin describirlos de acuerdo a las reglas claramente definidas. La medida proporciona una medición cuantitativa de extensión, cantidad, dimensiones, capacidad y tamaño de algunos atributos de un proceso o producto. La métrica mide cuantitativamente el grado en que un sistema, componente o proceso posee un atributo dado.

Las métricas de software son la aplicación continua de mediciones basadas en técnicas para el proceso de desarrollo del software y sus productos para suministrar información relevante a tiempo, así el administrador junto con el empleo de estas técnicas mejorarán el proceso y sus productos. Las métricas de calidad sirven para:

- La evaluación de los modelos de análisis y diseño.
- Proporcionar una indicación de la complejidad de diseños procedimentales y de código fuente.
- Ayuda en el diseño de pruebas más efectivas.

Las métricas de software incluyen las siguientes actividades:

- Estimación de costo y el esfuerzo.
- Medición de la productividad.
- Acumulación de datos.
- Realización de modelos y mediciones de calidad.
- Elaboración de modelos de seguridad.
- Evaluación y modelos de desempeño.
- Valoración de las capacidades y de la madurez.
- Administración por métricas.
- Evaluación del método y herramientas.

2.2.13.7. Enfoque Formal de Aseguramiento de Calidad en el Software, SQA.

La garantía de la calidad del software es una "actividad de protección" que se aplica a lo largo de todo el proceso de ingeniería de software. La SQA (Software Quality Assurance)¹⁷ engloba:

- Métodos y herramientas de análisis, diseño, codificación y prueba.
- Revisiones técnicas formales, que se aplican durante cada paso de la ingeniería de software.
- Una estrategia de prueba multiescala.
- El control de la documentación del software y de los cambios realizados.
- Un procedimiento que asegure un ajuste a los estándares de desarrollo de software (cuando sea posible).
- Mecanismos de medida y de información.

La garantía de calidad de software comprende una gran variedad de tareas, asociadas con siete actividades principales:

- Aplicación de métodos técnicos.
- Realización de revisiones de técnicas formales.

¹⁷ SQA (Software Quality Assurance – Aseguramiento de la Calidad del Software.)

- Prueba del software.
- Ajuste a los estándares.
- Control de cambios.
- Mediciones.
- Registro y realización de informes.

La SQA comienza realmente con un conjunto de herramientas y métodos técnicos que ayudan a conseguir una especificación de alta calidad y un diseño de alta calidad.

Una vez que se ha creado una especificación (o prototipo) y un diseño, debe ser garantizada su calidad. La actividad central que permite garantizar la calidad es la revisión técnica formal. La revisión técnica formal es una especie de reunión del personal técnico con el propósito de descubrir problemas de calidad.

La prueba del software combina una estrategia de múltiples pasos con una serie de métodos de diseño de casos de prueba que ayudan asegurar una efectiva detección de errores.

El grado de aplicación de procedimientos y estándares en el proceso de la ingeniería de software varía de empresa a empresa.

Una de las principales amenazas para la calidad de software viene de una fuente aparentemente benigna: los cambios. Cada cambio realizado sobre el software en potencia puede introducir errores o crear efectos laterales que propaguen errores.

Para intentar mejorar la calidad del software se debe recolectar métricas del software. Las métricas del software engloban un amplio conjunto de medidas técnicas orientadas a la gestión.

El registro de la información y la generación de informes para las garantías de calidad del software dan procedimientos para la recolección y divulgación de información de SQA. Los resultados de las revisiones, auditorías, control de cambios, pruebas y otras actividades de SQA deben convertirse en una parte del registro histórico de un proyecto y deben ser divulgados a la plantilla de desarrollo para que tengan conocimiento de ellos.

2.2.14. Fiabilidad en el Software.

La fiabilidad del software se define en términos estadísticos como la probabilidad de operación libre de fallos de un programa de computadora es un entorno determinado y durante un tiempo específico.

Los fallos pueden ser simplemente desconcertantes o ser catastróficos. Puede que un fallo sea corregido en segundos mientras que otro lleve

semanas o incluso meses. Los modelos de fiabilidad del software entran en dos grandes categorías:

1. Modelos que predicen la fiabilidad como una función cronológica del tiempo (calendario).
2. Modelos que predicen la fiabilidad como una función del tiempo de procesamiento transcurrido (tiempo de ejecución de CPU).

Se han propuesto modelos estocásticos mucho más sofisticados para la fiabilidad del software:

Validez predicativa. La posibilidad de que el modelo prediga el comportamiento de fallo futuro basándose en los datos obtenidos de las fases de prueba y de operación.

Capacidad. La posibilidad de que el modelo genere datos que puedan ser fácilmente aplicados a los esfuerzos de desarrollo de software industriales.

Calidad de suposiciones. La plausibilidad de las suposiciones en las que se basan los fundamentos matemáticos del modelo cuando se llega a los límites de esas suposiciones.

Aplicabilidad. El grado en que se puede aplicar un modelo de fiabilidad en diferentes terrenos y tipos de aplicación del software.

Simplicidad. El grado en que el conjunto de datos que soportan el modelo es directo; el grado en que el enfoque y las matemáticas son intuitivos; el grado en que se puede automatizar el enfoque general.

2.2.15. Enfoque para la Garantía de Calidad del Software.

La garantía de calidad del software es una "actividad de protección que se aplica a cada paso del proceso de ingeniería del software. En una organización de desarrollo de software se lleva un mecanismo de garantía del software según los niveles de la organización.

En el nivel inferior la calidad es responsabilidad únicamente del individuo que deba crear, revisar y probar el software a cualquier nivel de conformismo.

En el nivel superior de la escala, existe un grupo de SQA que carga con la responsabilidad de establecer estándares y procedimientos para conseguir la calidad del software y asegurar que se sigue cada uno de ellos.

El estado actual de la garantía de calidad del software y de la gestión de configuraciones del software se evalúa examinando los siguientes puntos:

- Principios.
- Organización.
- Interfaces funcionales.
- Beneficios SQA.

Ventajas:

- El software tendrá menos defectos latentes, resultando un menor esfuerzo y un menor tiempo durante la prueba y el mantenimiento.
- Se dará una mayor fiabilidad y, por tanto, una mayor satisfacción del cliente.
- Se podrán reducir los costos de mantenimiento, un porcentaje sustancial de los costos totales del software.
- El costo del Modelo clásico total del software disminuirá.

Las Desventajas de SQA es que es difícil de institucionalizar en organizaciones pequeñas, en las que no están disponibles los recursos necesarios para llevar a cabo esas actividades y representa un cambio cultural.

2.3. REDES DE COMPUTADORAS.

2.3.1. Introducción a las Redes.

Una red de computadoras consiste en una o más computadoras conectadas por un medio físico y que ejecutan un software que permite a las computadoras comunicarse unas con las otras y compartir recursos.

En los primeros años las grandes compañías, incluyendo IBM, Honeywell y Digital Equipment Corporation, crearon su propio estándar de cómo las computadoras debían conectarse. Estos estándares describían los mecanismos necesarios para mover datos de una computadora a otra. Estos primeros estándares, sin embargo, no eran completamente compatibles. Por ejemplo, las redes que se adherían al SNA (Systems Network Architecture) de IBM no podían comunicarse directamente con las redes usando el DNA (Digital Network Architecture) de DEC.

En años posteriores, organizaciones de estándares, incluyendo la Organización Internacional de Estandarización (ISO) y el instituto de Ingenieros Eléctricos y Electrónica (IEEE), desarrollaron modelos que llegaron a ser globalmente reconocidos y aceptados como estándares para el diseño de cualquier red de computadoras. Ambos modelos describen la red en términos de capas funcionales. Se puede pensar por un momento en el servicio de correos. Cuando alguien desea mandar una carta a otra persona, la escribe, la mete en un sobre con el formato impuesto por correos, le pone un sello y la introduce en un buzón; la carta es recogida

por el cartero, clasificada por el personal de correos, según su destino y enviada a través de medios de transporte hacia la ciudad destino; una vez allí otro cartero irá a llevarla a la dirección indicada en el sobre; si la dirección no existe, al cabo del tiempo la carta devolverá al origen por los mismos cauces que llegó al supuesto destino.

Más o menos, esta es la forma en que funciona una red: la carta escrita es la información que se quiere transmitir; el sobre y sello es el paquete con el formato impuesto por el protocolo que se utiliza en la transmisión; la dirección del destinatario es la dirección del nodo destino y la dirección del remitente, será la dirección del nodo origen, los medios de transporte que llevan la carta cerca del destino es el medio de transmisión (cable coaxial, fibra óptica, etc.); las normas del servicio de correos, carteros y demás personal son los protocolos de comunicaciones establecidos.

Si se supone que se está utilizando el modelo OSI de la ISO. Este modelo tiene 7 niveles, es como decir que la carta escrita pasa por 7 filtros diferentes (trabajadores con diferentes cargos) desde que la ponemos en el buzón hasta que llega al destino.

Cada nivel de esta torre se encarga de realizar funciones diferentes en la información a transmitir. Cada nivel por el que pasa la información a transmitir que se ha insertado en un paquete, añade información de control, que el mismo nivel en el nodo destino irá eliminando. Además se encarga de cosas muy distintas: desde el control de errores, hasta la reorganización de la información transmitida cuando esta se ha fragmentado en tramas. Si la información va dirigida a una red diferente (otra ciudad en el caso de la carta), la trama debe llegar a un dispositivo de interconexión de redes (Ruteadores, Puertas de Enlaces, Puentes), que decidirá, dependiendo de su capacidad, el camino que debe seguir la trama.

Por eso es imprescindible que el paquete lleve la dirección destino y que esta contenga, además de la dirección que identifica al nodo, la dirección que identifica la red a la que pertenece el nodo.

2.3.2. El Modelo OSI.

En 1984, la ISO¹⁸ desarrolló un modelo llamado OSI.¹⁹ El cual es usado para describir el uso de datos entre la conexión física de la red y la aplicación del usuario final. Éste modelo es el mejor conocido y el más usado para describir los entornos de red.

¹⁸ ISO. Organización Internacional de Estandarización.

¹⁹ OSI. (Open Systems Interconnection - Interconexión de sistemas abiertos).

Aplicación
Presentación
Sesión
Transporte
Red
Enlace
Físico

Figura 2.2.9.1. Modelo OSI.

Como se muestra en la figura 2.2.9.1, las funciones más básicas, como el poner los bits de datos en el cable de la red están en la parte de abajo, mientras las funciones que atienden los detalles de las aplicaciones del usuario están arriba.

En el modelo OSI el propósito de cada capa es proveer los servicios para la siguiente capa superior, resguardando la capa de los detalles de como los servicios son implementados realmente. Las capas son abstraídas de tal manera que cada capa cree que se está comunicando con la capa asociada en la otra computadora, cuando realmente cada capa se comunica sólo con las capas adyacentes de la misma computadora.

Con la siguiente figura 2.2.9.2., se puede apreciar que a excepción de la capa más baja del modelo OSI, ninguna capa puede pasar información directamente a su contraparte en la otra computadora. La información que envía una computadora debe de pasar por todas las capas inferiores, La información entonces se mueve a través del cable de red hacia la computadora que recibe y hacia arriba a través de las capas de esta misma computadora hasta que llega al mismo nivel de la capa que envió la información. Por ejemplo, si la capa de red envía información desde la computadora A, esta información se mueve hacia abajo a través de las capas de Enlace y Física del lado que envía, pasa por el cable de red, y sube por las capas de Física y Enlace del lado de el receptor hasta llegar a la capa de red de la computadora B.

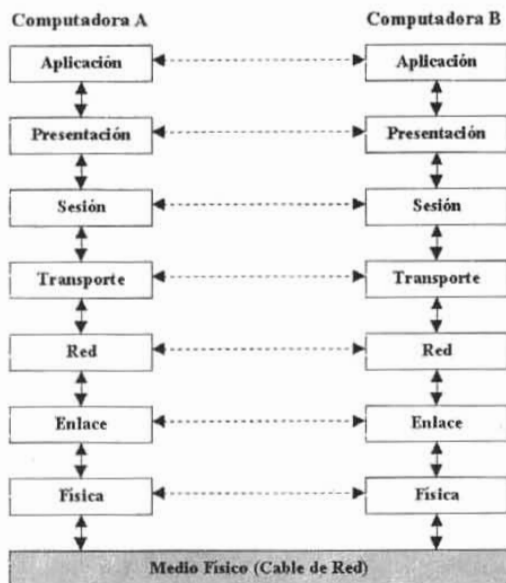


Figura 2.2.9.2. Medio Físico del Modelo OSI.

La interacción entre las diferentes capas adyacentes se llama interfaz. La interfaz define que servicios la capa inferior ofrece a su capa superior y como esos servicios son accedidos. Además, cada capa en una computadora actúa como si estuviera comunicándose directamente con la misma capa de la otra computadora. La serie de las reglas que se usan para la comunicación entre las capas se llama protocolo.

2.3.3. El Modelo IEEE.

Otro modelo de red fue desarrollado por el mismo instituto de Ingenieros Eléctricos y Electrónica (IEEE). Debido a la proliferación de Redes de Área Local (LAN) muchos productos aparecieron, y con ello la necesidad de una consistencia, entonces la IEEE empezó a definir estándares de red. El proyecto fue llamado 802, por el año y el mes en que empezó: Febrero de 1980.

Del proyecto 802 resultaron numerosos documentos, incluyendo los tres principales estándares para topologías de red.

- 802.3 define estándares para redes de bus, tales como Ethernet, que usa un mecanismo llamado CSMA/CD (Carrier Sense Multiple Access with Collision Detection).
- 802.4 define estándares para redes de "token" en bus. (La arquitectura de ArcNet es similar a este estándar en muchas maneras).
- 802.5 define estándar para redes de "token-ring".

2.3.4. Topología de Red.

La topología de red define únicamente la distribución del cable que interconecta las diferentes computadoras, es decir, es el mapa de distribución del cable que forma la red local. Define cómo se organiza el cable de las estaciones de trabajo. A la hora de instalar una red, es importante seleccionar la topología más adecuada a las necesidades existentes. Hay una serie de factores a tener en cuenta a la hora de decidirse por una topología de red concreta y son:

- La distribución de los equipos a interconectar.
- El tipo de aplicaciones que se van a ejecutar.
- La inversión que se quiere hacer.
- El costo que se quiere dedicar al mantenimiento y actualización de la red local.
- El tráfico que va a soportar la red local.
- La capacidad de expansión. Se debe diseñar una red local teniendo en cuenta la escalabilidad.

No se debe confundir el término topología con el de arquitectura. La arquitectura de una red engloba:

- La topología.
- El método de acceso al cable.
- Protocolos de comunicaciones.

Actualmente la topología está directamente relacionada con el método de acceso al cable, puesto que éste depende casi directamente de la tarjeta de red y ésta depende de la topología elegida.

2.3.5. Topología Física.

Es lo que hasta ahora se ha venido definiendo; la forma en la que el cableado se realiza en una red. Existen tres topologías físicas puras: Topología en anillo, topología en bus y topología en estrella.

Existen mezclas de topologías físicas, dando lugar a redes que están compuestas por más de una topología física.

2.3.6. Topología Lógica.

Es la forma de conseguir el funcionamiento de una topología física cableando la red de una forma más eficiente. Existen topologías lógicas definidas:

- Topología anillo-estrella: implementa un anillo a través de una topología de estrella física.
- Topología bus-estrella: implementa una topología en bus a través de una estrella física.

2.3.7. Topología en Bus.

Consta de un único cable que se extiende de un computadora al siguiente de un modo serie. Los extremos del cable se terminan con una resistencia denominada terminadora, que además de indicar que no existen más computadoras en el extremo, permiten cerrar el bus, como se muestra en la figura 2.3.7.1.

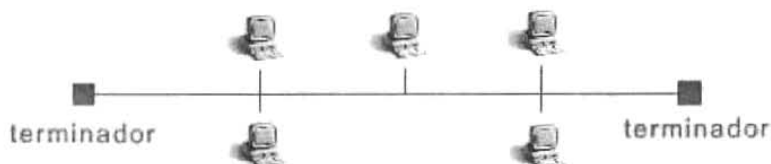


Figura 2.3.7.1. Topología en forma de bus.

Sus principales ventajas son:

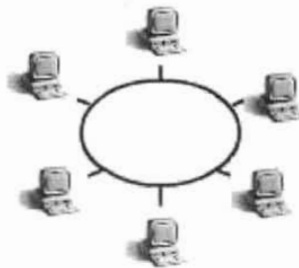
- Fácil de instalar y mantener.
- No existen elementos centrales del que dependa toda la red, cuyo fallo dejaría sin operar a todas las estaciones.

Sus principales inconvenientes son:

- Si se rompe el cable en algún punto, la red quedará fuera por completo.
- Cuando se decide instalar una red de este tipo en un edificio con varias plantas, lo que se hace es instalar una red por planta y después unir las todas a través de un bus troncal.

2.3.8. Topología en Anillo o Circuito.

El cable forma un bucle cerrado formando un anillo. Todas las computadoras que forman parte de la red se conectan a ese anillo. Habitualmente las redes en anillo utilizan como método de acceso al medio el modelo "paso de testigo". Los principales inconvenientes serían. Si se rompe el cable que forma el anillo se paraliza toda la red. Es difícil de instalar y requiere mantenimiento, figura 2.3.8.2.



Topología de Anillo

Figura 2.3.8.2. Topología en forma de anillo.

2.3.9. Topología en Estrella.

Todas las estaciones de trabajo están conectadas a un punto central (concentrador), formando una estrella física. Habitualmente sobre este tipo de topología se utiliza como método de acceso al medio "pooling", siendo el nodo central el que se encarga de implementarlo. Cada vez que se quiere establecer comunicación entre dos computadoras, la información transferida de uno hacia el otro debe pasar por el punto central. Existen algunas redes con esta topología que utilizan como punto central una estación de trabajo que gobierna la red. La velocidad suele ser alta para comunicaciones entre el nodo central y los nodos extremos, pero es baja cuando se establece entre nodos extremos. Este tipo de topología se utiliza cuando el traspase de información se va a realizar preferentemente entre el nodo central y el resto de los nodos, y no cuando la comunicación se hace entre nodos extremos. Si se rompe un cable sólo se pierde la conexión del nodo que interconectaba, es fácil de detectar y de localizar un problema en la red, tal y como se muestra en la siguiente figura. 2.3.9.1.



Topología de Estrella

Figura. 2.3.9.1. Topología Anillo Estrella

Uno de los inconvenientes de la topología en anillo era que si el cable se rompía toda la red quedaba fuera; con la topología mixta anillo-estrella, éste y otros problemas quedan resueltos. Las principales características son:

- Cuando se instala una configuración en anillo, el anillo se establece de forma lógica únicamente, ya que de forma física se utiliza una configuración en estrella.
- Se utiliza un concentrador, o incluso un servidor de red (uno de los nodos de la red, aunque esto es el menor número de ocasiones) como dispositivo central, de esta forma, si se rompe algún cable sólo queda fuera el nodo que conectaba, y los demás pueden seguir funcionando.
- El concentrador utilizado cuando se está utilizando esta topología se denomina MAU (Unidad de Acceso Multiestación), que consiste en un dispositivo que proporciona el punto de conexión para múltiples nodos. Contiene un anillo interno que se extiende a un anillo externo.
- A simple vista, la red parece una estrella, aunque internamente funciona como un anillo.
- Cuando la MAU detecta que un nodo se ha desconectado (por haberse roto el cable, por ejemplo), puentea su entrada y su salida para así cerrar el anillo.

2.3.10. Interconexión de Redes.

Hace algunos años era impredecible la evolución que las comunicaciones, en el mundo de la informática, iban a tener: no podía prever que fuese necesaria la interconexión ya no sólo de varias computadoras sino de cientos de ellos. No basta con tener los computadoras en una sala conectados, es necesario conectarlos a su vez con los computadoras del resto de las salas de una empresa, y con el resto de las sucursales de una empresa situadas en distintos puntos geográficos.

La interconexión de redes permite, si se puede decir así, ampliar el tamaño de una red local Sin embargo el término interconexión se utiliza para unir redes independientes, no para ampliar el tamaño de una.

El número de computadoras que componen una red local es limitado, depende de la topología elegida, (recuérdese que en la topología se define el cable a utilizar) aunque si lo único que se quisiera fuera sobrepasar el número de computadoras conectadas, podría pensarse en simplemente segmentar la red local. Sin embargo existen otros factores a tener en cuenta.

Segmento. Un segmento es una sección de cable que da servicio y que puede tener varios computadores conectados, el ancho de banda del mismo se reparte entre dichas computadores. Las redes de área local mayores están compuestas por varios segmentos interconectados. Cuando

se tiene una red grande se divide en subconjuntos, llamados segmentos. Para interconectar varios segmentos se utilizan puentes o ruteadores. El rendimiento de una red aumenta al dividirla en segmentos a cada segmento junto a las estaciones conectadas a éste se las llama subred.

Segmentación. Segmentar una red local consiste en dividirla en subredes para así poder aumentar el número de computadoras conectadas a ella y/o el rendimiento de la misma. Cuando se segmenta una red local, lo que se está haciendo es crear subredes pequeñas que, por decirlo de alguna manera, se autogestionan, de forma que la comunicación entre segmentos se realiza cuando es necesario, es decir, cuando un nodo de un segmento quiere comunicarse con un nodo del otro segmento; mientras tanto cada segmento de la red local está trabajando de forma independiente por lo que en una misma red local se están produciendo varias comunicaciones de forma simultánea; evidentemente esto mejora el rendimiento de la red local. La tabla 2.3.10.1 refleja las longitudes máximas de los segmentos dependiendo de las diferentes topologías de red.

Topologías	Longitud
Ethernet Gruesa	500 Metros
Ethernet Fina	185 Metros
Ethernet De Par Trenzado	100 Metros
Ethernet De Fibra Óptica	2.000 Metros
Token-Ring De Par Trenzado	100 Metros

Figura 2.3.10.1. Longitudes máximas de los segmentos de acuerdo a las topologías de red.

El dispositivo que se utiliza para segmentar una red debe ser inteligente ya que debe ser capaz de decidir hacia qué segmento debe enviar la información llegado a él: si hacia el mismo segmento desde el que la recibió o hacia otro segmento diferente.

La interconexión de redes locales se puede establecer a varios niveles: desde el nivel físico, a través de un dispositivo llamado concentrador hasta niveles más altos (niveles del modelo OSI) a través de dispositivos como un puente o un ruteador. La tabla 2.3.10.2., muestra el nivel en el que trabajan los diferentes dispositivos.

Dispositivo	Nivel
Repetidor	Físico
Concentrador	Físico
Puente	Enlace
Router	Red
Puerta De Enlace	Aplicación

Figura 2.3.10.2. Nivel de Operación de acuerdo al Modelo OSI.

Para la segmentación de redes locales, y teniendo en cuenta que uno de los motivos por el que se realiza esta operación es mejorar el rendimiento

de la red, es necesario emplear dispositivos inteligentes, como pueden ser un ruteador o un puente.

De forma genérica existen varias maneras de ampliar las redes locales:

- **Concentradores:** Para unir equipos dentro de una red.
- **Repetidores:** conexión a nivel físico, en el mismo segmento.
- **Puentes:** Conexión a nivel de enlace entre dos segmentos (iguales o distintos).
- **Ruteadores:** Conexión a nivel de red.
- **Puertas de enlace:** Conexión a nivel de presentación, entre dos redes distintas.

Hubs o Concentradores. Dispositivo que interconecta equipos dentro de una red. Es el dispositivo de interconexión más simple que existe.

Repetidores. Conectan a nivel físico dos redes locales, o dos segmentos de red local. Hay que tener en cuenta que cuando la distancia entre dos hosts es grande, la señal que viaja por la línea se atenúa y hay que regenerarla, ver siguiente diagrama 2.3.10.3.

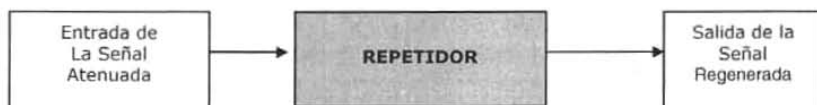


Figura 2.3.10.3. Repetidor de una Red.

Bridges o Puentes. Son dispositivos que ayudan a resolver el problema de limitación de distancias, junto con el problema de limitación del número de nodos de una red. Trabajan al nivel de enlace del modelo OSI, por lo que pueden interconectar redes que cumplan las normas del modelo 802 (3, 4 y 5). Si los protocolos por encima de estos niveles son diferentes en ambas redes, el puente no es consciente, y por tanto no puede resolver los problemas que puedan presentarsele.

Ruteador. Es como un puente incorporando características avanzadas. Trabajan a nivel de red del modelo OSI, por tanto trabajan con direcciones IP. Habitualmente se utilizan para conectar una red de área local a una red de área extensa. Son capaces de elegir la ruta más eficiente que debe seguir un paquete en el momento de recibirlo.

Gateways o Puerta de Enlace. Se trata de una computadora u otro dispositivo que interconecta redes radicalmente distintas. Trabaja al nivel de aplicación del modelo OSI. Cuando se habla de pasarelas a nivel de redes de área local, en realidad se está hablando de ruteadores.

Existen dos modos de transmisión:

Banda base. Es la transmisión digital de datos a través de un cable. La codificación utilizada es normalmente de tipo Manchester, que permite combinar una señal de reloj con los datos. La transmisión en banda base implica que solo puede haber una comunicación en el cable en un momento dado.

Banda ancha. Es la transmisión analógica de los datos. Para ello se utilizan módems que operan a altas frecuencias. Cada módem tiene una portadora diferente, de forma que es posible realizar varias comunicaciones simultáneas en el cable.

2.3.11. Arquitectura Ethernet.

A finales de 1960, la universidad de Hawaii desarrolló una red de área amplia (WAN, Red que se extiende a través de un área geográfica mayor a una LAN). La universidad necesitaba conectar varias computadoras que estaban esparcidas a través del campus. La pieza principal en el diseño de la red fue llamado Carrier-Sense Multiple Access with Collision Detection (CSMA/CD). Carrier-Sense significa que la computadora escucha el cable de la red y espera hasta un periodo de silencio para poder mandar su mensaje. Multiple Access se refiere a que múltiples computadoras pueden estar conectadas en el mismo cable de red. Collision Detection es una protección contra mensajes chocando en el tránsito.

Este temprano diseño de red fue la fundación de lo que hoy es Ethernet. En 1972, Xerox Corporation creó el Ethernet experimental, y en 1975 introdujo el primer producto Ethernet. La versión original de este producto de red fue diseñado como un sistema de 2.94 mbps (Megabits por segundos) conectando hasta 100 computadoras en un cable de un kilómetro.

El Ethernet de Xerox fue tan exitoso que Xerox, Intel y Digital crearon un estándar para Ethernet de 10mbps. Este diseño fue la base de la especificación IEEE 802.3. El producto Ethernet se apega en la mayoría de las partes del estándar 802.3.

El CSMA/CD funciona de la siguiente manera, cuando una computadora desea mandar información primero escucha el cable de la red para revisar que no se este usando en ese precioso momento (Carrier-Sense). Esto se oye muy sencillo, pero el problema reside en que dos o más computadoras al escuchar que no se esta usando el cable puede mandar el exacto mismo momento su información (Acceso Múltiple), y como solamente puede haber uno y sólo un mensaje en tránsito en el cable, se produce una colisión. Entonces las computadoras detectan la colisión y deciden reenviar su información a un intervalo al azar, es importante que sea al azar ya que si ambas computadoras tuvieran el mismo intervalo fijo se produciría un ciclo vicioso de colisiones y reenvíos (Collision Detection). Así por ejemplo al detectar la colisión una computadora se espera tres milisegundos y la otra

cinco milisegundos, siendo obvio que una computadora reenviara en primer lugar y la otra esperará a que el cable este de nuevo sin tránsito.

Evidentemente que en una misma red Ethernet al haber muchas computadoras tratando de enviar datos al mismo tiempo y/o al haber una transferencia masiva de datos se crea un gran porcentaje de colisiones y utilización. Si se pasa del 1% de colisiones y/o 15% de utilización de cable ya se dice que la red está saturada. Además, las señales de este tipo de red tienden a degradarse con la distancia debido a la resistencia, la capacidad u otros factores. Inclusive la señal todavía se puede distorsionar por las interferencias eléctricas exteriores generadas por los motores, las luces fluorescentes y otros dispositivos eléctricos. Cuanto más se aumenta la velocidad de transmisión de los datos. Más susceptible es la señal a degradarse. Por esta razón las normas de Ethernet especifican los tipos de cables, los protectores y las distancias del mismo, la velocidad de transmisión y otros detalles para trabajar y proporcionar un servicio relativamente libre de errores en la mayoría de los entornos.

Las redes Ethernet pueden utilizar diferentes tipos de cableado, cada uno con sus beneficios y problemas. Los tres cableados más comunes son Thinnet, Thicknet, y Twisted Pair (Par trenzado).

Thinnet o 10Base2 puede transmitir datos a 10mbps por Banda Base (señales digitales), pudiendo llegar el cableado hasta 185 metros. Se utiliza cable coaxial RG-58 el cual es bastante barato por lo que a esta red también se le conoce como CheapNet. Un mismo segmento de cable puede soportar hasta 30 computadoras. Es el más utilizado y recomendado para redes pequeñas. Utiliza la topología local bus, donde un mismo cable recorre todas y cada una de las computadoras.

Thicknet o 10Base5 transmite datos a 10mbps por Banda Base en un cableado que puede alcanzar 500 metros. El cableado es grueso y es utilizado principalmente para largas oficinas o hasta todas las computadoras de un edificio. Del cable principal (backbone) salen cables usualmente Par Trenzado que se conectan a directamente a cada una de las computadoras. Se pueden conectar hasta 100 computadoras con este cableado en un mismo segmento.

Twisted Pair o 10BaseT transmite datos a 10mbps por Banda Base y utiliza un Hub (concentrador) desde el cual con cable Par Trenzado se conecta cada una de las computadoras quedando en forma similar a estrella. El Hub queda en el centro de la estrella y funciona como "repetidor". El cable desde el Hub hasta la computadora no debe de medir más de 100 metros.

2.3.12. Arquitectura Token-Ring

La red Token-Ring es una implementación del estándar IEEE 802.5, en el cual se distingue más por su método de transmitir la información que por la forma en que se conectan las computadoras.

El primer diseño de una red de Token-Ring es atribuido a E. E. Newhall en 1969. IBM publicó por primera vez su topología de Token-Ring en marzo de 1982, cuando esta compañía presentó los papeles para el proyecto 802 del IEEE. IBM anunció un producto Token-Ring en 1984, y en 1985 éste llegó a ser un estándar de ANSI/IEEE.

A diferencia del Ethernet, aquí un Token (Ficha Virtual) es pasado de computadora a computadora como si fuera una papa caliente. Cuando una computadora desea mandar información debe de esperar a que le llegue el Token vacío, cuando le llega utiliza el Token para mandar la información a otra computadora, entonces cuando la otra computadora recibe la información regresa el Token a la computadora que envió con el mensaje de que fue recibida la información. Así se libera el Token para volver a ser usado por cualquiera otra computadora. Aquí debido a que una computadora requiere el Token para enviar información no hay colisiones, el problema reside en el tiempo que debe esperar una computadora para obtener el Token sin utilizar.

Los datos en Token-Ring se transmiten a 4 ó 16mbps, dependiendo de la implementación que se haga. Todas las estaciones se deben de configurar con la misma velocidad para que funcione la red. Cada computadora se conecta a través de cable Par Trenzado ya sea blindado o no a un concentrador llamado MAU (Media Access Unit), y aunque la red queda físicamente en forma de estrella, lógicamente funciona en forma de anillo por el cual da vueltas el Token. En realidad es el MAU es que contiene internamente el anillo y si falla una conexión automáticamente la ignora para mantener cerrado el anillo.

Un MAU puede soportar hasta 72 computadoras conectadas y el cable del MAU a la computadora puede ser hasta de 100 metros utilizando Par Trenzado Blindado, ó 45 metros sin blindaje. El Token-Ring es eficiente para mover datos a través de la red. En redes pequeñas a medianas con tráfico de datos pesado el Token Ring es más eficiente que Ethernet. Por el otro lado, el ruteo directo de datos en Ethernet tiende a ser un poco mejor en redes que incluyen un gran número de computadoras con tráfico bajo o moderado.

2.3.13. Protocolo TCP/IP.

Se trata de un conjunto de protocolos, aunque los mas conocidos sean TCP (nivel de transporte) e IP (nivel de red). Las aplicaciones que corren sobre TCP/IP no tienen que conocer las características físicas de la red en la que se encuentran; con esto, se evita el tener que modificarlas o reconstruirlas para cada tipo de red. Esta familia de protocolos genera un modelo llamado INTERNET cuya correspondencia con el modelo OSI queda reflejada en la tabla siguiente 2.3.13.1.

INTERNET		OSI/ISO
Aplicaciones		Aplicación
		Presentación
		Sesión
TCP	UDP	Transporte
IP		Red
ARP	RARP	Enlace
Red física (Ethernet)		Físico

Figura 2.3.13.1. Operación del Protocolo TCP/IP en el Modelo OSI.

Características de TCP/IP:

- Utiliza conmutación de paquetes.
- Proporciona una conexión fiable entre dos máquinas en cualquier punto de la red.
- Ofrece la posibilidad de interconectar redes de diferentes arquitecturas y con diferentes sistemas operativos.
- Se apoya en los protocolos de más bajo nivel para acceder a la red física (Ethernet, Token-Ring).

Una red TCP/IP transfiere datos mediante el ensamblaje de bloques de datos en paquetes conteniendo:

- La información a transmitir.
- La dirección IP del destinatario.
- La dirección IP del remitente.
- Otros datos de control.

Protocolo IP. Se trata de un protocolo a nivel de red cuyas principales características son:

- Ofrece un servicio no orientado a la conexión; esto significa que cada trama en la que ha sido dividido un paquete, es tratado por independiente. Las tramas que componen un paquete pueden ser enviadas por caminos distintos e incluso llegar desordenadas.
- Ofrece un servicio no muy fiable porque a veces los paquetes se pierden, duplican o estropean y este nivel no informa de ello pues no es consciente del problema.

Direccionamiento IP. Cada máquina con TCP/IP tiene asociado un número de 32 bits al que se llama dirección IP, y que está dividido en dos partes:

- Una parte que identifica la dirección de la red (NETID). Esta parte es asignada por el NIC (Network Information Center). Si la red local no va a conectarse con otras redes, no es necesario solicitar a ese organismo una dirección. El número de bits que ocupa esta parte depende del tamaño de la red y puede ser 8, 16 ó 24.

- Una parte que identifica la dirección de la máquina dentro de la red (HOSTID). Las direcciones de los hosts son asignadas por el administrador de la red.

Una dirección se representa por cuatro valores decimales separados por puntos, para que sea más fácil su escritura y memorización.

[0..255] . [0..255] . [0..255] . [0..255]

2.3.14. Máscara de Subred.

Cuando una red aparece segmentada (dividida en subredes), se debe utilizar un dispositivo que interconecte los segmentos y se hace necesario identificar de algún modo cada uno de los segmentos. Si todos los segmentos tienen la misma dirección IP, se hace necesaria la existencia de algún mecanismo que diferencie los segmentos. Este mecanismo es la máscara de la subred.

A cada dirección IP de red, es decir, a cada red física, se le asocia una máscara que tiene 32 bits. La máscara sirve para dividir la parte de la dirección IP destinada a identificar el host en dos partes: la primera identificará el segmento, y la segunda el host dentro de este segmento. En esta máscara los bits a 1 significan que el bit correspondiente de la dirección IP será tratado como bit correspondiente a la dirección de la subred, mientras que los bits a 0 en la máscara, indican que los bits correspondientes de la dirección IP serán interpretados como identificadores del host. Así con una misma dirección de red se pueden direccionar muchas subredes.

2.3.15. Clases de Redes.

El tipo depende del número de máquinas que forman la red; atendiendo esto se pueden distinguir tres clases de redes:

Redes de clase A: Se tratan de redes de mayor tamaño, redes que tengan más de 2^{16} hosts. El espacio reservado para la dirección de red es más pequeño por dos motivos: Porque existen menos redes de este tipo. Porque al tener más host necesitamos dejar más espacios para direccionar a estos. La parte que identifica la red consta de un cero (0) y 7 bits más.

Se podrán direccionar por tanto 2^7 redes que hace un total de 128 redes diferentes. Cada una de estas redes podrá tener 2^{24} posibles hosts. La dirección 127 no se utiliza (Tabla 2.3.15.1).

1.....7	8.....32
Dirección de la red 0...	Identificador de la máquina

Tabla 2.3.15.1. Clases de Red tipo A.

Redes de clase B: Son redes de tamaño mediano que tienen entre 2^8 y 2^{16} hosts. La parte que identifica la red consta de la secuencia uno-cero (10) y 14 bits con cualquier valor. Por tanto, el rango de valores para el primer byte de los dos asignados a la red es de 128-191.

Estas redes pueden tener $2^{16}=65536$ hosts cada una de ellas. El formato de las direcciones es como lo muestra la siguiente tabla 2.3.15.2.

1.....16	17.....32
Dirección de la red 10.....	Identificador de máquina

Tabla 2.3.15.2. Clases de Red tipo B.

Redes de clase C: Son redes menor tamaño que pueden tener hasta 2^8 hosts. La parte que identifica la red consta de la secuencia uno-uno-cero (110) y 21 bits con cualquier valor. Por tanto, el rango de valores para el primer byte de los dos asignados a la red es de 192-223. (Tabla 2.3.15.3.)

Estas redes pueden tener $2^8=256$ hosts cada una de ellas. El formato de las direcciones es, ver tabla 2.3.15.3.

0.....23	24.....31
Dirección de la red 110.....	Identificador de máquina.

Tabla 2.3.15.3. Clases de Red tipo C.

Tabla Esquemática de los Formatos de Direcciones, tabla 2.3.15.4.

	Byte 1	Byte 2	Byte 3	Byte 4
Clase A	0...126	0...255	0...255	0...255
Clase B	128...191	0...255	0...255	0...255
Clase C	192...223	0...255	0...255	0...255

Tabla 2.3.15.4. Tabla Esquemática de los Formatos de Direcciones.

Existen más clases de redes, como la D, E y F. Este tipo de redes son experimentales o se reservan para un uso futuro.

2.3.16. Relación entre Direcciones IP y Direcciones Físicas.

Se debe relacionar la dirección IP con suministrada con una dirección física. Situándose en la jerarquía de niveles utilizada por Internet, se observa que por debajo del protocolo IP existe el nivel de enlace, en el se asientan protocolos como ARP o RARP. Estos protocolos resuelven problemas relacionados con las direcciones.

ARP. Convierte una dirección IP en una dirección física.

RARP. Convierte una dirección física en una dirección IP.

En cada host debe existir una tabla de encaminamiento, que está limitada a la red que pertenece. Si la dirección IP no pertenece a la red, entonces hace dirigir los paquetes IP hacia la puerta de enlace o ruteador que esté conectado a esa red, el cual ya poseen unas tablas que referencias las redes que conocen. El contenido de estas tablas puede variar dinámicamente.

2.3.17. Sistemas Operativos de Red.

Los sistemas operativos de red, además de incorporar herramientas propias de un sistema operativo como son por ejemplo las herramientas para manejo de ficheros y directorios, incluyen otras para el uso, gestión y mantenimiento de la red, así como herramientas destinadas a correo electrónico, envío de mensajes, copia de ficheros entre nodos, ejecución de aplicaciones contenidas en otras máquinas, compartición de recursos hardware etc. Existen muchos sistemas operativos capaces de gestionar una red dependiente de las arquitecturas de las máquinas que se utilicen. Los más comunes son: Novell, Lantastic, Windows 3.11 para trabajo en grupo, Unix, Linux, Windows 95, Windows NT, Windows 98, Windows 2k, OS/2... Cada sistema operativo ofrece una forma diferente de manejar la red y utiliza diferentes protocolos para la comunicación.

2.3.18. Acceso Remoto a la Red Local.

Las redes locales actuales pueden extenderse más allá de los límites del propio lugar de trabajo. Con la informática móvil y la proliferación de las redes locales, es necesario que cuando un usuario se encuentre fuera de su lugar de trabajo exista alguna posibilidad de conectar con la red local de la empresa, ya sea para consultar correo electrónico, para enviar datos o imprimir un informe en un dispositivo de la propia empresa para que lo puedan ver otras personas de la compañía.

El acceso remoto a redes ofrece una función principal: permite acceder a los recursos de la red de la compañía, luego se permite acceder a ficheros que se encuentran en el servidor de red de la empresa, y se garantiza que todos los usuarios puedan acceder a una misma copia de un fichero, de forma que cualquier modificación realizada por un usuario queda disponible para todos los demás que tengan permisos para consultarlo.

Si la red local de la compañía posee acceso permanente a Internet los usuarios que conectan de forma remota pueden utilizar dicho recurso. De este modo, la empresa se convierte en un proveedor de Internet que proporciona acceso a sus propios empleados.

Todo este acceso lo facilita la red telefónica tanto la fija como la móvil (GSM). El aspecto de la telefonía móvil resulta muy interesante, ya que en la actualidad un teléfono GSM se puede conectar a una computadora (normalmente un portátil). El problema es el elevado precio de las llamadas, aunque no lo es tanto. Si se observan las tarifas de llamadas

telefónicas móviles a teléfonos fijos en un horario determinado por cada compañía telefónica (que suelen denominar súper reducido), veremos que son inferiores a las llamadas nacionales, por lo que una llamada desde Guadalajara a León por GSM puede resultar más barata que el teléfono fijo. Pero generalmente el horario súper reducido no coincide con las necesidades de comunicación de los usuarios y se hace necesario disponer de otra fuente de comunicación más barata. Por ello, si la red local tiene acceso a Internet, mediante un servicio como: Infovía, que proporciona la compañía Telefónica podemos conectar con la red de la empresa al precio de una llamada local.

Capítulo

3



Universidad Nacional
Autónoma de México



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.

Capítulo

3

Administración del Proyecto

3.1. JUSTIFICACIÓN DEL PROYECTO.

El área de ingresos debe consolidar el programa anual de recaudación por derechos, aprovechamientos y productos de acuerdo a las disposiciones jurídico-administrativas aplicables vigentes. De tal forma que es prioritario establecer y normalizar las actividades propias con el fin de alcanzar dicho objetivo. Se pretende para ello desglosar, analizar y emprender las acciones necesarias para un alto desempeño laboral.

El manejo de la información que se lleva a cabo en el área de ingresos, tiene un tratamiento manual que permite que se introduzcan factores no deseados, tales como un procesamiento lento y con posibles errores en los cálculos de facturación; y como consecuencia de esto las aerolíneas y aviones no comerciales pierden tiempo revisando los cargos y tramitando quejas por inconformidades en las facturas, ocasionando un ciclo vicioso con todas las áreas involucradas y sobretodo perdiendo tiempo valioso en la corrección. Además, la captura que se lleva a cabo en las torres de control no ha podido ser retroalimentada con sistemas sólidos de información y el capturista en las torres de control tampoco recibe instrucciones concretas ya que su captura no ha sido cotejada con estándares establecidos, permitiendo que se introduzca el mismo error muchas veces.

Un análisis preliminar, nos permite emitir los siguientes comentarios. "Es curioso mencionar que por falta de tiempo, -el cual es utilizado en el proceso-, no se pueda establecer un modelo sistematizado, sin embargo, en el análisis posterior quedará sustentado que el sistema propuesto dotará al área de ingresos de un amplio margen de tiempo útil, si no que además contribuirá a un mejor control en otros procedimientos".

Las Torres de Control envían decenalmente a la Administración Central información de los vuelos realizados en cada torre de control. Para ello se cuenta con programas de captura de operaciones en las torres de control. Cada decena es recibida en un archivo plano con la información de cada estación a la Dirección de Informática y ésta a su vez la entrega a la Dirección de Ingresos, integrada en una hoja de cálculo, quién a su vez la vuelve a procesar en otras hojas de cálculo, ordenada por nombre del cliente y la distribuye a los empleados asignados. Una vez dividida la carga de trabajo, los empleados revisan manualmente la información, provocando con ello gastos innecesarios, lentitud en el servicio y muchas horas/hombre involucradas en el proceso. Observando la siguiente figura 3.1.1 podemos darnos cuenta del flujo de procesos involucrados en el ciclo. El flujo se mueve de izquierda a derecha y de arriba abajo.



Figura 3.1.1 Flujo de Procesos.

En forma breve detallaremos el procedimiento que se lleva a cabo en las torres de control para adentrarnos en el proceso. Cada operación es registrada en una tira de papel que contiene la información particular del evento ocurrido en el aeropuerto. Cientos e incluso miles de operaciones diarias a nivel nacional ocurren y se registran en esas tiras de control. Ahí mismo se capturan en un pequeño sistema. El programa de captura entre otras cosas, genera un archivo de texto en donde se incluyen todos los datos pertinentes de cada operación, así por ejemplo la torre de control, registra el número de vuelo, la hora de salida o de llegada, el equipo o tipo

de aviación y la matrícula de la aeronave. El sistema guarda las operaciones y las ordena por día. Cuando se tienen capturados los eventos por decena, se crea el archivo con la información de la torre control y demás datos correspondientes.

Informática recibe cada uno de los archivos por decena y los ordena por aeropuerto, nombre de usuario, fecha y lo distribuye por remesas a cada empleado asignado para la revisión en el área de ingresos.

En ingresos cada encargado, recibe el documento y le aplica las formulas y tarifas de cobro y analiza las operaciones que son factibles o no para cobro. Una vez que se determina que vuelos fueron los efectivos para el cobro se suman todas y cada una de las operaciones efectuadas decenalmente por cliente y se elabora la factura.

3.2. OBJETIVOS DEL PROYECTO.

El objetivo principal del sistema es el de registrar, validar y homologar la información proveniente de las torres de control con el propósito de llevar, primero, un control de las operaciones en el sistema en una base de datos estándar y segundo, el de filtrar con el sistema todas las operaciones erróneas en los centros de captura, validando, detectando y corrigiendo los errores en las operaciones que fueron registradas en el proceso para que puedan ser conciliadas con los clientes y con los usuarios de la Dirección de Ingresos, según la política de cobros de la Dirección General de Finanzas. El sistema además deberá presentar al usuario todas las estadísticas de las operaciones a los altos mandos, para su posterior análisis.

Este módulo deberá ser capaz de reducir el tiempo de recolección, validación y registro de operaciones, además de que tendrá que eliminar los errores en las operaciones de los clientes de manera definitiva. Lo cual contribuirá a disminuir las quejas por parte de los clientes en cálculos erróneos o que ocasionan una menor recaudación para la Federación.

La interfaz gráfica del sistema será de fácil manejo y accesible para el usuario permitiendo de forma amigable procesar la información contando también con niveles y control de acceso por cuestiones de seguridad e integridad en los datos. Por ello será desarrollado en una base de datos relacional, lo cual nos dotará de información actualizada, consistente, organizada y confiable.

En resumen, se deberá contar al finalizar este trabajo con un sistema que controle, valide y actualice la información generada en las torres de control de las operaciones de tránsito aéreo de los aeropuertos de la República Mexicana, con el propósito de conciliar los vuelos con los usuarios y proporcionar estadísticas confiables y actualizadas de las mismas por medio de:

- Un sistema de fácil empleo.
- Control de Usuarios.
- Mantener un registro permanente de accesos.
- Detección y manejo de la información mal capturada.
- Generación de reportes.

Para ser utilizado por el personal administrativo, solucionando así el problema de seguridad, integridad y consistencia de la información. Finalmente, estudiar la posibilidad de emplear la información, interactuando con otros módulos y otros sistemas de bases de datos para formar un sistema de información efectivo y muy valioso.

3.3. DIAGNÓSTICO DE LOS REQUERIMIENTOS.

Una tarea básica en la planificación del desarrollo de un proyecto es la estimación de recursos requeridos para acometer el esfuerzo del desarrollo. La figura 3.3.1 ilustra los recursos de desarrollo en forma de pirámide. En la base se encuentran las herramientas o recursos materiales existentes que darán soporte al desarrollo. En el nivel más alto se encuentra el recurso primario más importante, los recursos humanos. Cada recurso queda especificado mediante algunas características que ahí se describen.²⁰



1: Especificar Habilidades requeridas, disponibilidad, duración de las tareas, número de personas involucradas

2: Especificar descripción de herramientas, disponibilidad, duración de uso, fecha de distribución.

Figura 3.3.1, Recursos de Desarrollo Piramidal.

²⁰ Roger S. Pressman, Ingeniería de Software: Un Enfoque Práctico. Tercera Edición, pág. 75.

3.3.1 Requerimiento de Recursos Materiales.

Para el análisis de los requerimientos de recursos materiales, empezaremos por analizar la plataforma o ambiente en el que será desarrollado el sistema, posteriormente revisaremos el manejador de bases de datos y por último las herramientas o aplicaciones de desarrollo.

3.3.1.1 Software.

Microsoft Windows NT 4.0 Server

El Área de Informática cuenta con un sistema operativo Windows NT Server 4.0 el cual permite trabajar hasta cierto punto con independencia del equipo y de la ubicación del propio lugar de trabajo, acceder en cualquier momento a recursos (discos, memoria, impresora, etc.) seleccionados en una red e iniciar, en otras computadoras, aplicaciones que no están instaladas en la misma computadora.

Para administrar direcciones de usuarios Internet, Domain Name Server (DNS) de Microsoft incluye una utilidad gráfica de administración e integración con Windows Internet Naming Services (WINS) para actualizaciones dinámicas para nombres y direcciones de hosts. Junto con Dynamic Host Configuration Protocol (DHCP), estos servicios ofrecen los medios para administrar gráficamente una infraestructura TCP/IP, un componente crítico de funcionalidad intranet.

Requisitos para su instalación, (Observar la siguiente tabla 3.3.1.1.1):

Computador: /Procesador	Computadora Pentium 133 (MHz.) o más; Pentium III se recomienda
Memoria	Los requerimientos de memoria RAM para Access 2003 Dependen del sistema operativo: Microsoft Windows NT 32 MB de RAM además de 8 MB de RAM para la aplicación de Access
Disco Duro	El espacio en el disco duro podría variar dependiendo de la configuración; La instalación personalizada podría requerir más o menos lo mismo que la instalación mínima. 170 MB de espacio en disco. Espacio adicional de 115 MB se requiere donde se instaló el sistema Operativo.

Sistema Operativo	Windows NT 4.0 con Service Pack 6 (SP6) o posterior.*
Dispositivos	CD-ROM
Monitor	Super VGA (800 × 600) o monitor con 256 colores
Periféricos	Microsoft Mouse, Microsoft IntelliMouse®, o compatible pointing device

Tabla 3.1.1.1. Requisitos de Instalación para Microsoft Windows NT 4.0 Server.

Microsoft NT 4.0 Workstation.

Windows NT Workstation es un sistema operativo diseñado para los usuarios avanzados que desean obtener las máximas características de protección y seguridad en sus ordenadores, como se puede ver en la tabla 3.3.1.1.2.

NT Workstation se puede instalar en máquinas con múltiples procesadores. Además, no sólo trabaja en las computadoras con microprocesadores Intel x86, sino también en plataformas con microprocesadores RISC (tales como MIPS R4x00, Digital Alpha o PowerPC). Esta es una característica importante, pues ofrece grandes ventajas a los usuarios (entre otras cosas la seguridad de que su inversión en software y recursos se garantiza con independencia de la plataforma), pero también es una de las limitaciones para hacer un sistema más optimizado y más abierto al hardware de los ordenadores Intel. Sí se instala en un ordenador x86, NT Workstation necesita como mínimo una Pentium a 90 MHz con 16 MB de RAM y 115 MB de disco duro, libres.

Principales características: Windows NT Workstation puede ejecutar las aplicaciones para Windows 32 bits, Windows 16 bits y MS-DOS. Las características de red están muy bien cuidadas. Integra soporte para 15 protocolos de red, entre ellos Servicios Cliente para Netware, NetBEUI, IPX/SPX, DCE RPC, DLC y, por supuesto, TCP/IP (en sus variantes DHCP, WINS, PPP y SLIP). Además, NT Workstation es por sí mismo un sistema operativo entre pares (peer-to-peer) con capacidad para crear redes con servidores no dedicados. El cliente de servicios Netware soporta script de conexión y gestión de archivos e impresoras.

Seguridad y protección: Windows NT Workstation está concebido para usuarios avanzados que exigen características de protección y seguridad (posee el certificado de seguridad C2). Así, es posible especificar permisos de acceso, lectura y ejecución tanto a nivel de archivos como a nivel de directorios. Para poder llevar a cabo estos permisos se tiene que haber formateado el disco como un sistema de archivos NTFS (no con el sistema de archivos FAT habitual en DOS y Windows 95). Las particiones NTFS ofrecen también compresión a nivel de archivos, mantienen un registro de actividades para restaurar el disco en caso de fallo de energía u otros problemas y permiten un tamaño de archivo máximo entre 4 GB y 64 GB.

Otras características de seguridad son administrador de cuentas, bloqueo de cuentas según el número de reintentos incorrectos y seguridad de red mediante credenciales.

Computador: /Procesador	Procesador Pentium II a 300 MHz o superior.
Memoria	Los requerimientos de memoria RAM son de 64 MB. RAM.
Disco Duro	Disco Duro Ultra Wide SCSI 4 GB 11ms de acceso.
Dispositivos	CD ROM.
Monitor	Súper VGA (800 × 600) o monitor con 256 colores
Otros	Tarjeta de Vídeo 2 MB. Monitor 14" SVGA Mouse Serial Teclado Windows 95

Tabla 3.1.1.2. Requisitos de Instalación para Microsoft Windows NT 4.0. Workstation.

Microsoft Access 2003.

Microsoft Access es un manejador de bases de datos muy accesible y fácil de usar, el cual permite aplicar el modelo relacional de datos a través de su sencilla interfaz y permite a distintas herramientas de desarrollo interactuar ya sea de manera local o de manera remota.

Motor de Bases de Datos. Los Motores de Bases de Datos, gestionan la manipulación física de los datos, almacenándolos en disco y proporcionándolos cuando sean solicitados. El motor de la base de datos que utiliza Access es Microsoft Jet y con la introducción del soporte para ODBC (Conectividad Abierta de Bases de Datos), estos motores pueden vincularse con el origen de los datos.

Modelo de objeto de acceso a datos. Tanto Access como VisualBasic proporcionan mecanismos sencillos para vincular controles de formularios directamente con un ODBC, haciendo que no resulte necesario tratar directamente con el motor de Datos. Un modelo de objetos de datos proporciona un conjunto de objetos con propiedades y métodos que se pueden manipular por medio del código. Microsoft proporciona tres modelos de objetos de acceso a datos: Objetos de acceso a datos (DAO); Objetos de datos remoto (RDO), utilizado normalmente para el acceso a orígenes de datos ODBC; y objetos de datos Activex de Microsoft ADO.²¹ (Ver tabla 3.3.1.1.3)

²¹ Rebeca M. Riordan, Diseño de Bases de Datos Relacionales con Access y SQL. Pág. 7.

Requerimientos de Instalación para el Servidor:

Computador: /Procesador	Procesador Pentium II Xeon a 400 MHz.
Memoria	Los requerimientos de memoria RAM son de 128 Mb RAM.
Disco Duro	Disco Duro Ultra Wide SCSI 9 GB 8ms de acceso.
Dispositivos	CD ROM SCSI 20X o superior interno.
Monitor	Súper VGA (800 × 600) o monitor con 256 colores
Otros	Unidad de Cinta 4Gb SCSI externa (Para la realización de Respaldos). CD ROM SCSI 20X o superior interno. Tarjeta de Video AGP 4Mb. Monitor 15" UVGA. Mouse Serial. Teclado Windows 95.

Tabla 3.1.1.3. Requisitos de Instalación para Microsoft Access.

Microsoft Visual Basic 6.0.

Debido a la integración existente entre Windows NT 4.0, Microsoft Access, la herramienta debe también ser parte integral de la plataforma de desarrollo, por esto, Visual Basic 6.0 se apunta como la mejor opción para el desarrollo de los programas cliente. Visual Basic combina los elementos de las herramientas de desarrollo visuales con la flexibilidad necesaria para el desarrollo de aplicaciones de Base de Datos complejas. Otra ventaja que presenta Visual Basic es el soporte a nivel mundial que tiene, ha sido elegido como la herramienta de desarrollo estándar en el mercado.

Requerimientos para su instalación, tal y como se presenta a continuación en la siguiente tabla (3.3.1.1.4.):

Computador: /Procesador	Computadora Pentium 90 (MHz.) o más; Pentium III se recomienda
Memoria	Los requerimientos de memoria RAM para Access 2003 Dependen del sistema operativo: Microsoft Windows NT 32 MB de RAM además de 8 MB de RAM para la aplicación de Access
Disco Duro	El espacio en el disco duro podría variar dependiendo de la configuración; La instalación personalizada podría requerir más o menos lo mismo que la instalación mínima.54 MB de espacio en disco.

	Espacio adicional de 94 MB para la instalación completa.
Librería MSD i	Se requiere de 57 MB a 493 MB dependiendo del tipo de instalación.
Dispositivos	CD-ROM.
Monitor	Súper VGA (800 × 600) o monitor con 256 colores.
Periféricos	Microsoft Mouse, Microsoft IntelliMouse®, o compatible pointing device

Tabla 3.1.1.4. Requisitos de Instalación para Microsoft Visual Basic 6.0.

Microsoft Visio 2003.

Visio 2003 es un programa para la creación de diagramas que puede crear diagramas empresariales y técnicos en los que se documenten y organicen ideas, procesos y sistemas complejos. Los diagramas creados en Visio 2003 permiten visualizar y comunicar información. También automatiza la visualización de los datos al sincronizarse directamente con los orígenes de datos para proporcionar diagramas actualizados, y se puede personalizar para cubrir las necesidades de la organización, algunos de los diagramas que contempla son los siguientes.

- Planos de espacios.
- Diagramas de flujo.
- Diagrama de Lluvia de ideas.
- Diagramas de bloques.
- Vincular dibujos a bases de datos.
- Cuenta con una amplia galería de dibujos.
- Genera formularios.
- Diagramas UML.

En la siguiente tabla 3.3.1.1.5 podemos observar los requerimientos para su instalación:

Computadora/Procesador	Procesador Pentium III a 700 MHz o superior.
Memoria	Los requerimientos de memoria RAM mínimos son de 128 Mb RAM.
Disco Duro	Se requieren 254 MB para la Instalación y un área libre de trabajo de 210 MB.
Administración/Configuración	Microsoft Visual SourceSafe 6.0c.
IDE²² Soportado	Microsoft Visual Studio .NET 7.0, 7.1
Sistemas Operativos	Windows 2000, NT 4.0 y Windows 2003

Tabla 3.1.1.5. Requisitos de Instalación para Microsoft Visio 2003.

²² Integrated Development Environment

Crystal Reports 8.0.

Crystal Reports es el generador de reporte por excelencia de Visual Basic desde versiones anteriores a .NET. Es un producto creado en su esencia orientado al usuario final; es decir que un ejecutivo pueda crear sus propios informes sin necesidad de asistencia de un desarrollador.

En la siguiente tabla se muestra las características que debe tener la tabla siguiente (tabla 3.3.1.1.6)

A pesar de su sencillez de uso, el software se puede integrar con multitud de entornos y lenguajes de desarrollo (ediciones developer y advanced). Crystal Reports es utilizado tanto en entornos de desarrollo Microsoft como Borland (Visual Basic, Interdev, Delphi, C++, Visual C++), y en cualquier lenguaje que soporte, por ejemplo, controles ActiveX o llamadas a APIs. Conviene mencionar también la creciente utilización de Crystal Reports para realizar informes a partir de aplicaciones estándar como ERPs (Baan, SAP, Navision, PeopleSoft), CRMs, y otras aplicaciones departamentales. En algunos casos existen drivers específicos para facilitar la integración, desarrollados conjuntamente por Crystal y el correspondiente fabricante. En otros casos, las aplicaciones incorporan Crystal Reports de manera nativa, por ejemplo Dimoni y Adonix, sin pretender ser exhaustivos.

Conexión con bases de datos: Crystal Reports permite realizar informes a partir de muy diversos orígenes de datos: Archivos de PC (incluyendo archivos de eventos), OLAP, ODBC, OLE DB, Dbase, FoxPro, Access, Paradox, Oracle, SQL Server, Informix, Lotus Notes, Exchange, Sybase, XML, y así hasta más de 30 bases de datos diferentes. Esta versión reconoce además el Dataset de .Net como origen de datos. También es posible mezclar en un mismo informe diversos orígenes de datos, algo muy valorado por los usuarios que trabajan en entornos heterogéneos. Conviene mencionar que las versiones 8 y superiores han mejorado especialmente la velocidad de acceso a los datos.

Diseño visual: Ésta es la característica que ha cautivado a los usuarios desde su lanzamiento, cuando las herramientas visuales no eran tan comunes, y que lo ha convertido en un estándar. La herramienta incorpora diferentes asistentes para facilitar la tarea a los nuevos usuarios, y también un diccionario de datos.

Distribución de informes: Crystal Reports permite exportar los resultados a múltiples formatos, incluyendo Excel, Word, XML, publicación web como HTML o DHTML, y PDF.

Computador: /Procesador	Procesador Pentium II o superior.
Memoria	Los requerimientos de memoria RAM mínimos son de 128 Mb RAM.
Disco Duro	Se requieren 350 MB.
IDE Soportado	Microsoft Visual Studio.
Sistemas Operativos	Windows 2000, NT 4.0 y Windows 2003

Tabla 3.1.1.6. Requisitos de Instalación para Cristal Report 8.0 de Seagate.

Microsoft Visual SourceSafe 6.0.

El control de versiones es un aspecto crítico del desarrollo de software y sitios Web en equipo, previniendo a los usuarios de una pérdida accidental de archivos, la recuperación de versiones previas, desarrollo en paralelo, unir y administrar versiones. Visual SourceSafe proporciona esta robusta funcionalidad de administración y configuración junto con capacidades de seguridad y funcionalidad para auditar.

Características de Visual SourceSafe:

- Administrar el proceso por el que el control de elementos pasa de una persona a otra. Los proveedores de control de código fuente admiten acceso a archivos tanto compartido como exclusivo. Si el acceso a los archivos de proyecto es exclusivo, el proveedor de control de código fuente permite a un solo usuario cada vez desproteger y modificar los archivos. Si el acceso es compartido, varios usuarios pueden desproteger el archivo y el proveedor de control de código fuente proporciona un mecanismo para combinar las versiones a medida que se protegen.
- Almacenar versiones sucesivas de elementos controlados mediante código fuente. Un proveedor de control de código fuente almacena los datos que diferencian entre sí versiones de un elemento controlado mediante código fuente. Por consiguiente, puede recuperarse cualquier versión de un elemento controlado mediante código fuente. También puede designarse cualquier versión como la última de ese elemento.
- Mantener información detallada de historial y versión acerca de los elementos controlados mediante código fuente. El control de código fuente almacena información detallada sobre cada versión de un elemento: la fecha y hora en que se creó, cuándo se ha desprotegido o protegido y el usuario que realizó la acción.

ESTA TESIS NO SALE DE LA BIBLIOTECA

- Colaborar entre proyectos. La acción de compartir archivos permite que varios proyectos compartan elementos controlados mediante código fuente. Los cambios efectuados en un elemento compartido se reflejan en todos los proyectos que lo comparten.
- Automatizar operaciones de control de código fuente que se repiten con frecuencia. Un proveedor de control de código fuente puede definir una interfaz de línea de comandos que admita las características clave del control de código fuente. Puede utilizarse esta interfaz en archivos por lotes para automatizar las tareas de control de código fuente que se realizan con regularidad.
- Recuperar información tras la eliminación accidental. Se puede restaurar la última versión de archivo protegida en control de código fuente.
- Conservar espacio en disco tanto en el cliente como en el servidor de control de código fuente. Admite conservar espacio en disco en el servidor mediante la utilización de la técnica de delta inverso para mantener el seguimiento de los cambios efectuados en archivos y proyectos. Esta técnica controla las versiones almacenando la última versión de un archivo y las diferencias entre cada versión y la versión anterior o siguiente. En el cliente, Visual SourceSafe admite conservar espacio en disco. Se pueden esconder carpetas y archivos para que no se descarguen al disco local.

Distribución del Software.

En la siguiente tabla se muestra la forma en que deberá instalarse el software entre los participantes en el proyecto, tal y como se muestra en la siguiente tabla 3.3.1.1.7.

Microsoft Visio 2000	Líder de Proyecto
Microsoft Visual SourceSafe 6.0	Todos
Microsoft Visual Basic 6.0	Todos
Crystal Reports 8.0	2 Analistas/Programadores
NT 4.0 SP 6.0 WS	Todos
Microsoft Access 2000	DBA y 2 Analistas/Programadores
Microsoft Office 2000	Todos
Librería MSDN	Líder de Proyecto y 2 Analistas/Programadores

Tabla 3.1.1.7. Requisitos de Instalación de las herramientas del software entre los participantes al proyecto.

3.3.1.2 Hardware.

En base a los requerimientos anteriores de software se ha decidido dotar el siguiente equipo de cómputo para el desarrollo del sistema. Habrá cuatro computadoras con las siguientes características (tabla 3.3.1.2.1):

Características	Computadora Pentium III @ 1.2 GB
RAM	256 MB
Disco Duro	20 GB
Monitor	Súper VGA (800 × 600) o monitor con 256 colores.
S. Operativo	NT 4.0 SP6.0 Workstation
Periféricos y otro:	Microsoft Mouse, Impresora de red, Tarjeta de Red, cuenta de correo, cuenta de usuario.

Tabla 3.1.2.1. Requisitos mínimos para el equipo de cómputo propuesto.

3.3.2 Requerimientos de Recursos Humanos.

Debido a que en el área de informática, se llevan a cabo diferentes proyectos, se ha dividido al personal en grupos de trabajo. La Dirección ha decidido asignar para este proyecto en particular cuatro especialistas: Un líder de proyecto, un administrador de bases de datos y 2 analistas/programadores. Quienes deberán conformar el equipo responsable para este proyecto.

3.3.2.1 Funciones del personal asignado.

Líder de proyecto: El líder de proyecto será la persona responsable de llevar a cabo la administración del proyecto. Determinando tiempos, suministrando los equipos, programas y demás insumos involucrados en el desarrollo. Revisando y analizando los productos obtenidos. Dará seguimiento a las actividades programadas y resolverá cualquier situación que impida el progreso del sistema. También diseñará la estructura del sistema y de los componentes, así como de sus interfaces, conexiones e iteraciones. Construyendo los esquemas, subsistemas, clases, asignaciones de responsabilidades y colaboraciones entre objetos que cumplen con las funciones del sistema en base a un análisis detallado de las necesidades del cliente.

Administrador de Bases de Datos: Es el encargado de analizar, diseñar y construir y dar mantenimiento a la base de datos.

Analista/Programador: Es la persona responsable de completar el diseño (bajo nivel), escribir el código necesario para el desarrollo del sistema y

ejecutar las pruebas de integración y de volumen, en base a las tareas especificadas.

3.3.2.2. Plantilla asignada al proyecto.

La dirección de informática asignó a este proyecto el siguiente personal para el desarrollo del sistema, en base a la carga de trabajo del área y al personal capacitado necesario, tal y como se puede ver en la siguiente tabla (tabla 3.3.2.2.1)

Cantidad	Puesto
1	Líder de Proyecto
1	Administrador de Bases de Datos
2	Analista/Programador
4	Total

Tabla 3.1.2.2.1. Requisitos mínimos para el equipo de cómputo propuesto.

3.2.3. Plan de trabajo.

A continuación se presenta el diagrama de Gantt desarrollado, así como el planteamiento y distribución de las tareas de acuerdo al plan de trabajo, tabla 3.2.3.1. En la tabla se puede observar el nombre de la tarea, la duración esperada, la fecha de comienzo y la fecha de terminación.

Diagrama de Gantt.



Tabla 3.2.3.1. Diagrama de Gantt.

Del diagrama anterior se desprenden las actividades que se muestran en la tabla 3.2.3.2.

ID	Nombre	Duración	Comienzo	Fin
1	Sistema de validación y registro de operaciones en las torres de control	47.d	08/11/2004 09:00	11/01/2005 19:00
2	Análisis	11.125d	08/11/2004 09:00	23/11/2004 10:00
3	Análisis de requerimientos del usuario	5.d	08/11/2004 09:00	15/11/2004 10:00
4	Análisis de requerimientos técnicos	6.d	15/11/2004 10:00	23/11/2004 10:00
5	Diseño	15.875d	15/11/2004 10:00	06/12/2004 19:00
6	Diseño de arquitectura de datos	7.d	15/11/2004 10:00	24/11/2004 10:00
7	Diseño de entidades y relaciones	1.d	15/11/2004 10:00	16/11/2004 10:00
8	Primera forma normal	1.d	15/11/2004 10:00	16/11/2004 10:00
9	Segunda forma normal	1.d	15/11/2004 10:00	16/11/2004 10:00
10	Tercera forma normal	1.d	15/11/2004 10:00	16/11/2004 10:00
11	Diseño de tablas, instancias de BD y dominios	4.d	16/11/2004 10:00	22/11/2004 10:00
12	Creación de scripts	2.d	22/11/2004 10:00	24/11/2004 10:00
13	Diseño de arquitectura de negocio	8.d	22/11/2004 09:00	01/12/2004 19:00
14	Diseño de casos de uso	3.d	22/11/2004 09:00	24/11/2004 19:00
15	Diseño de actividades, estados y clases	4.d	24/11/2004 09:00	29/11/2004 19:00
16	Diseño de secuencias	3.d	23/11/2004 09:00	25/11/2004 19:00
17	Diseño de algoritmos específicos	4.d	26/11/2004 09:00	01/12/2004 19:00
18	Diseño de infraestructura tecnológica	3.d	02/12/2004 09:00	06/12/2004 19:00
19	Diseño de ambiente de desarrollo	3.d	02/12/2004 09:00	06/12/2004 19:00
20	Diseño de ambiente de pruebas	1.d	06/12/2004 09:00	06/12/2004 19:00
21	Diseño de ambiente de producción	1.d	06/12/2004 09:00	06/12/2004 19:00
22	Codificación	17.d	06/12/2004 09:00	28/12/2004 19:00
23	Documentación del código	4.d	06/12/2004 09:00	09/12/2004 19:00

24	Declaración de datos	3.d	10/12/2004 09:00	14/12/2004 19:00
25	Construcción de sentencias	7.d	15/12/2004 09:00	23/12/2004 19:00
26	Estilo de entrada/salida	3.d	24/12/2004 09:00	28/12/2004 19:00
27	Pruebas	3.d	29/12/2004 09:00	31/12/2004 19:00
28	Pruebas unitarias	3.d	29/12/2004 09:00	31/12/2004 19:00
29	Pruebas de integración	3.d	29/12/2004 09:00	31/12/2004 19:00
30	Pruebas de volumen	2.d	29/12/2004 09:00	30/12/2004 19:00
31	Implementación	8.d	31/12/2004 09:00	11/01/2005 19:00
32	Manual de usuario	2.d	31/12/2004 09:00	03/01/2005 19:00
33	Manual de instalación	2.d	31/12/2004 09:00	03/01/2005 19:00
34	Manual de capacitación	2.d	31/12/2004 09:00	03/01/2005 19:00
35	Instalación	3.d	04/01/2005 09:00	06/01/2005 19:00
36	Capacitación	3.d	07/01/2005 09:00	11/01/2005 19:00

Tabla 3.3.2.2. Tabla de actividades asignadas respecto al tiempo.

3.3.2.4. Tabla de Sueldos por puesto.

La siguiente tabla nos muestra la tasa estándar por empleado respecto a su sueldo mensual, desglosado por hora (tabla 3.3.2.4.1).

ID	Nombre	Iniciales	Tipo	Tasa estándar	Sueldo Mensual
1	Líder de Proyecto	LP	Trabajo	\$143.95/hora	\$35,000.00
2	Analista/Programador 1	AP1	Trabajo	\$61.67/hora	\$15,000.00
3	Analista/Programador 2	AP2	Trabajo	\$61.67/hora	\$15,000.00
4	Administrador de Bases de Datos	ABD	Trabajo	\$115.13/hora	\$28,000.00

Tabla 3.3.2.4.1. Tabla estándar Sueldo/Empleado.

3.3.2.5. Tabla de Asignación de Tareas.

En la siguiente tabla 3.3.2.5.1., se pueden observar los nombres de las tareas, el personal asignado, las horas trabajadas y las unidades en por ciento que les fueron asignadas a cada persona.

Nombre de tarea	Personal Asignado	Trabajo	Unidades
Análisis	Líder de Proyecto	88 horas	100%
Análisis de requerimientos del usuario	Líder de Proyecto	40 horas	100%

SISTEMA DE VALIDACIÓN Y REGISTRO DE OPERACIONES DE LAS TORRES DE CONTROL

Análisis de requerimientos técnicos	Líder de Proyecto	48 horas	100%
Diseño	Administrador de Bases de Datos	127 horas	100%
Diseño de arquitectura de datos	Administrador de Bases de Datos	56 horas	100%
Diseño de entidades y relaciones	Administrador de Bases de Datos	8 horas	100%
Primera forma normal	Administrador de Bases de Datos	8 horas	100%
Segunda forma normal	Administrador de Bases de Datos	8 horas	100%
Tercera forma normal	Administrador de Bases de Datos	8 horas	100%
Diseño de tablas, instancias de bd y dominios	Administrador de Bases de Datos	32 horas	100%
Creación de scripts	Administrador de Bases de Datos	16 horas	100%
Diseño de arquitectura de negocio	Líder de Proyecto	64 horas	100%
Diseño de casos de uso	Analista/Programador 1	24 horas	100%
Diseño de actividades, estados y clases	Analista/Programador 1	32 horas	100%
Diseño de secuencias	Analista/Programador 1	24 horas	100%
Diseño de algoritmos específicos	Analista/Programador 1	32 horas	100%
Diseño de infraestructura tecnológica	Líder de Proyecto	24 horas	100%
Diseño de ambiente de desarrollo	Analista/Programador 2	24 horas	100%
Diseño de ambiente de pruebas	Analista/Programador 2	8 horas	100%
Diseño de ambiente de producción	Analista/Programador 2	8 horas	100%
Codificación	Líder de Proyecto	136 horas	100%
Documentación del código	Líder de Proyecto	32 horas	100%
Declaración de datos	Analista/Programador 1	24 horas	100%
Construcción de seriales	Analista/Programador 2	56 horas	100%
Estilo de entrada/salida	Analista/Programador 1	24 horas	100%
Pruebas	Líder de Proyecto	24 horas	100%
Pruebas unitarias	Analista/Programador 2	24 horas	100%
Pruebas de integración	Analista/Programador 1	24 horas	100%
Pruebas de volumen	Analista/Programador 2	16 horas	100%

Implementación	Líder de Proyecto	64 horas	100%
Manual de usuario	Analista/Programador 2	16 horas	100%
Manual de instalación	Analista/Programador 1	16 horas	100%
Manual de capacitación	Analista/Programador 1	16 horas	100%
Instalación	Analista/Programador 2	24 horas	100%
Capacitación	Analista/Programador 2	24 horas	100%

Tabla 3.3.2.5.1. Asignación de Tareas.

3.3.3 Requerimientos de Recursos Financieros.

De acuerdo a la norma de factibilidad y rentabilidad de bienes informáticos (Equipos de cómputo, periféricos, software, etc.) de la Dirección de Informática se estableció que la vida útil para el equipo de cómputo es de 5 a 7 años. Para las licencias de software oscila entre los 3 y 5 años y para los periféricos de 4 a 5 años. Esto obedece entre otros factores a la depreciación financiera y por supuesto al estado en que se guarden dichos activos. Por tal motivo se presentará el prorrato de cada recurso, indicando la vida actual del recurso del que se trate, así como de su expectativa de vida útil y el costo calculado en base al tiempo del proyecto. Cabe hacer mención que la mayoría de estos recursos han sido adquiridos con anterioridad.

3.3.3.2 Costos de Recursos Materiales.

En seguida se muestra la tabla de costos de los recursos materiales que se verán involucrados en el proceso de desarrollo del sistema (Tablas 3.3.3.2.1., 3.3.3.2.2., 3.3.3.2.3., y 3.3.3.2.4).

SISTEMA DE VALIDACIÓN Y REGISTRO DE OPERACIONES DE LAS TORRES DE CONTROL

Recurso	Antigüedad	Vida	Costo	Duración Proyecto	Prorrateo
Computadora Procesador Pentium III a 1 GB. Memoria 255 MB Disco Duro 4 GB Dispositivos CD ROM. Monitor Super VGA (1024 x 600) o monitor con 256 colores Otros Tarjeta de Video 2 Mb Monitor 14" S.VGA Mouse Serial Teclado Windows 95	2.5 años	3.5 años	15,000	47 días $\left(\frac{15,000}{5 \times 3.5}\right) = 8,750$ $\left(\frac{47}{365}\right) = 0.13$	(0.13 * 8,750.00) = \$1,137.50 por Computador por la duración del proyecto.
Impresora Laser 12 PPM	1	3 años	8,000	47 días $\left(\frac{8,000}{5 \times 3}\right) = 6,000.$ $\left(\frac{47}{365}\right) = 0.13$	(0.13 * 6,000) = \$780 por Impresora por la duración del proyecto.
Windows NT 4.0 Server	6	4 años	45,000	47 días $\left(\frac{45,000}{10 \times 4}\right) = 18,000.$ $\left(\frac{47}{365}\right) = 0.13$	(0.13 * 18,000) = \$93.6 por Licencia por la duración del proyecto.

Tabla 3.3.3.2.1. Costos Equipos y Sistema Operativo

SISTEMA DE VALIDACIÓN Y REGISTRO DE OPERACIONES DE LAS TORRES DE CONTROL

Microsoft Visio 2003	0	5 años	50,000 Edición Empresarial	47 días $((50'000/5 \times 5))=$ 50,000 $(47/365)$ =0.13	$(0.13$ $*50,000)$ = \$6271.5 por Licencia por la duración del proyecto.
Microsoft Visua Studio: Microsoft Visua Basic 6.0, Microsoft Visu: I Sourcesafe 6.0, Librería M iDN	6	2 años	11,500 licencia para 1	47 días $((11,500/4 \times 2))=$ 2,870. $(47/365)$ =0.13	$(0.13$ $*2,870)$ = \$374 por Licencia por la duración del proyecto.
Crystal Reports 8.0	4	3 años	6,000 licencia para 1	47 días $((6'000/10 \times 4))=$ 2,571. $(47/365)$ =0.13	$(0.13$ $*50,000)$ = \$335 por Licencia por la duración del proyecto.
Microsoft Office Professional 2003 con Microsoft Access 2003	6	4 años	5,750 Edición Profesional	47 días $((5'750/4 \times 4))=$ 5,750. $(47/365)$ =0.13	$(0.13$ $*5,750)$ = \$745.00 por Licencia por la duración del proyecto.

Tabla 3.3.3.2.2. Costos Software.

Recurso	Cantidad	Precio Unitario	Costo
Computadoras	4	\$1,137.50	\$4,550
Impresoras	1	\$780.00	\$780
Microsoft NT 4.0 Server	4	\$96.50	\$386
Microsoft Visio 2003	1	\$6,271.5	\$6,271.5
Microsoft Visual Studio 6.0	3	\$374	\$1,122
Crystal Reports 8.0	2	\$335	\$670
Microsoft Office Professional 2003	4	\$745	\$2,980

Tabla 3.3.3.2.3. Desglose de costos.

Insumos (Papel, tinta, etc.)	4	\$500 (12.5 pesos diarios aprox.)	\$2,000
Consumo de Energía: PC e impresora	1230 W	\$1.386 KWH	1,120 h ::: 1,120*1230= 1'377,600 WxH Costo = 1'377,600 W (1,377.6 KW) x \$1.386= 1,909.35
Subtotal	-	-	\$20,670
IVA	-	-	\$3,100
Total Costo 1	-	-	\$23,770

Tabla 3.3.3.2.4. Desglose de costos (continuación)

3.3.3.3. Costos de Recursos Humanos.

A continuación se muestra en la siguiente tabla el desglose de costos de recursos humanos. 3.3.3.2.1.

Nombre	Tasa estándar	Tiempo Producción (horas)	Total
Líder de Proyecto	\$143.95/hora	488	\$70,247.63
Analista/Programador 1	\$61.67/hora	216	\$13,320.72
Analista/Programador 2	\$61.67/hora	190	\$11,717.3
Administrador de Bases de Datos	\$115.13/hora	263	\$30,279.19
Total Costo 2		1,119 h	\$125,564.84

Tabla 3.3.3.2.1. Costos Equipos y Sistema Operativo

3.3.3.4 Gestión del Proyecto.

En base a los resultados arrojados por el análisis financiero de recursos materiales y recursos humanos, tenemos la siguiente información, como se puede observar en la siguiente tabla 3.3.3.4.1.

SISTEMA DE VALIDACIÓN Y REGISTRO DE OPERACIONES DE LAS TORRES DE CONTROL

Costo del Proyecto			Total
Costo 1 Recursos Materiales			\$23,770.00
Costo 2 Recursos Humanos			\$125,564.84
Gran Total			\$149,344.89

Tabla 3.3.3.4.1. Costos del Proyecto.

Concluimos por lo tanto que el presupuesto designado para la elaboración del Sistema de Validación y Registro de Operaciones de las Torres de Control es de casi \$150,000.00. Cantidad muy aceptable comparada con los miles de pesos que se pierden en el cálculo manual de las operaciones.

Capítulo

4



Universidad Nacional
Autónoma de México



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.

Capítulo

4

Desarrollo del Proyecto

4.1. ANÁLISIS.

Para llevar a cabo el análisis del proyecto consideramos dos grandes rubros en cuanto a los requerimientos. El primero de ellos es el rubro de los requerimientos técnicos, es decir, la infraestructura necesaria tanto para el desarrollo del proyecto como para su constante funcionamiento y explotación. El siguiente rubro que consideramos son los requerimientos funcionales que el área solicitante define para la elaboración del sistema. Una vez que se plantean los requerimientos de ambos rubros, se comienzan a analizar los requerimientos para ubicar los puntos que se deben atacar, las prioridades de desarrollo y el mejor enfoque que se deba utilizar, de modo tal, que una vez terminado el análisis, el diseño del proyecto defina cómo realizar las acciones previamente definidas.

4.1.1. Análisis de Requerimientos Técnicos.

El área encargada de suministrar el equipo será el área de soporte técnico y operación, ellos pondrán apunto las computadoras con las especificaciones solicitadas por el líder de proyecto. Las fechas para dicha petición han sido recibidas 10 días antes del inicio del proyecto y tomando en cuenta que la fecha de comienzo del sistema ha sido acordada el 8 de noviembre del 2004 se propone el siguiente planteamiento. (Ver tabla 4.1.1.1)

Equipos	Computadora Pentium III @ 1.2 GB, 256 MB, DD 20 GB, Monitor Súper VGA, SO, NT 4.0 SP6.0 Workstation, ratón, Impresora de Red.	Fecha de Inicio 01/11/2004 Fecha Límite 05/11/04
---------	---	---

Tabla 4.1.1.1 Propuesta de equipo de cómputo.

Las computadoras deberán de cargarse con el software determinado para cada empleado de acuerdo al plan de trabajo que se muestra en la tabla 4.1.1.2.

Equipo # 1 NT 4.0 SP 6.0 WS, Microsoft Visio 2003, Microsoft Visual Sourcesafe 6.0, Microsoft Visual Basic 6.0, Microsoft Office 2000, Librería MSDN, cuenta de correo, cuenta de usuario con permisos de administrador, cuenta de sourcesafe con permisos de administrador.	Líder de Proyecto	Fecha de Inicio 05/11/2004 Fecha Límite 8/11/05
Equipo # 2 NT 4.0 SP 6.0 WS, Microsoft Access 2003, Microsoft Visual Sourcesafe 6.0, Microsoft Visual Basic 6.0, Microsoft Office 2000, Librería MSDN, cuenta de correo, cuenta de usuario, cuenta de sourcesafe.	Administrador de la Base de Datos	Fecha de Inicio 09/11/2004 Fecha Límite 12/11/05
Equipo # 3 NT 4.0 SP 6.0 WS, Microsoft Access 2003, Microsoft Visual Sourcesafe 6.0, Microsoft Visual Basic 6.0, Microsoft Office 2000, Librería MSDN, cuenta de correo, cuenta de usuario, cuenta de sourcesafe y Crystal Reports 8.0.	Analista/Programador 1	Fecha de Inicio 15/11/2004 Fecha Límite 19/11/05
Equipo # 4 NT 4.0 SP 6.0 WS, Microsoft Access 2003, Microsoft Visual Sourcesafe 6.0, Microsoft Visual Basic 6.0, Microsoft Office 2000, Librería MSDN, cuenta de correo, cuenta de usuario, cuenta de sourcesafe y Crystal Reports 8.0.	Analista/Programador 2	Fecha de Inicio 22/11/2004 Fecha Límite 29/11/05

Tabla 4.1.1.2 Lista de Software que deberá instalarse en los equipos de cómputo.

Como podemos observar en la tabla 4.1.1.2 y de acuerdo a nuestro plan de trabajo, concluimos que se tiene un margen holgado para la preparación de las máquinas y la instalación del software, permitiendo al área de soporte, no descuidar sus actividades normales, sino además avanzar firmemente para la conclusión de las distintas actividades programadas.

4.1.2. Análisis de Requerimientos del Usuario.

Recordando el propósito fundamental del sistema, es el de validar y registrar las operaciones efectuadas en las torres de control a nivel nacional. Cada aeropuerto cuenta con una oficina de control y despacho de vuelos, las cuales registran todas las operaciones de aeronaves que se efectúan en cada torre, ya sea de salida o de llegada. Registrando en un sistema de captura todos los datos pertinentes de la operación llevada a cabo.

Posterior a las entrevistas con el usuario, se desarrolló un planteamiento paso por paso, enfocado a las solicitudes específicas de cada problema/solución presentado, durante el análisis. Se presenta a continuación el análisis obtenido.

4.1.2.1 Procedimiento Previo.

Estas actividades son realizadas decenalmente en el área de operación. Anualmente ingresos entrega a informática un calendario de recepción de las torres de control, enseguida se enumeran los siguientes pasos:

- En el área de operación se recibe la información vía Internet o Intranet, proveniente de las estaciones y de acuerdo al calendario establecido.
- El archivo de cada estación es guardado en la carpeta de la estación correspondiente, cada estación tiene su carpeta. El nombre del archivo esta conformado de la siguiente manera:

ACA041103

1. Siglas de la estación (3).
 2. Año del que son las operaciones (2).
 3. Mes del que son las operaciones (2).
 4. Decena del que son las operaciones (2).
- Se envía vía Intranet o Internet un acuse de recibo donde se les indica el estado en que la información fue recibida, correcta o incorrecta según sea el caso.
 - Mensualmente se pasa toda la información recolectada a una carpeta en el servidor para ser procesada en el sistema de validación y registro de operaciones en las torres de control.

4.1.2.2. Procedimiento de Recolección de Archivos de las Torres de Control.

De acuerdo al calendario de procesos que envía ingresos a la dirección de informática, la información debe estar disponible en la fecha solicitada con anterioridad.

Si por alguna razón queda pendiente de cargar alguna de las estaciones, en la tabla de observaciones se pondrá el nombre de la estación faltante junto con la decena perteneciente y cuando pueda cargarse se indicará también que se ya ha sido procesado.

Se les avisa vía mail o telefónicamente (sí existe algún problema con el correo) que ya pueden trabajar el mes correspondiente, si faltó alguna estación cuando es cargada esta estación también se les informa.

4.1.3. Objetivos del Sistema.

Los objetivos principales del sistema, se enmarcan en dos funciones primordiales; la primera es la importación o recolección y carga a la bases de datos de todas las operaciones en los aeropuertos federales, para ello el sistema deberá de ser capaz de validar la información recibida y calcular las operaciones que son viables de facturación; y la segunda es la de permitir al usuario de ingresos poder corregir aquellas operaciones que por omisión o error no fueron capturadas correctamente. Estableciendo un canal de retroalimentación de las operaciones con las torres de control, para un desempeño mejorado cada vez.

4.1.4. Análisis.

Es en este paso, donde detallaremos las características que deberá tener el modelo sugerido.

4.1.4.1. Entrada a la Aplicación.

La entrada al sistema deberá estar validada con una base de datos y ahí mismo deberá registrarse cada intento de ingreso al mismo. Para ello se propone lo siguiente:

La pantalla de ingreso deberá permitir la entrada al sistema, constará de 2 campos:

- 1) Número de Remesa: Permite 2 caracteres numéricos del 1 al 99.
- 2) Password: Admite 10 caracteres alfanuméricos.

El sistema estará restringido y sólo permitirá ingresar los datos hasta en 3 ocasiones, a la cuarta el sistema deberá cerrarse y tendrá que ser

ejecutado otra vez para proseguir con la carga del programa, se guardarán todos los intentos de ingreso al sistema.

Si la remesa y el password son correctos se podrá acceder al menú principal del sistema.

El programa debe ser diseñado para tener 3 niveles de acceso jerárquicos: Administrador, Supervisor y Capturista.

4.1.4.2. Interfaz principal.

El administrador tiene permisos de acceso total al sistema, el supervisor tiene permisos parciales y que iremos detallando conforme vayamos describiendo el uso de los módulos y por último el capturista sólo tendrá algunos permisos que describiremos más adelante.

Una vez que se haya ingresado al sistema, se mostrará la pantalla principal, la cual contiene el menú con todos los subsistemas del proyecto. A continuación describiremos cada uno de ellos. El menú principal deberá estar esta compuesto de 4 módulos, como se muestra en la tabla 4.5.1.2.1.

1	Captura/Consulta.
2	Transferencia.
3	Utilerías.
4	Herramientas.

Tabla 4.1.4.2.1 Menú Principal.

4.1.4.3. Módulo de captura y consulta de operaciones.

En este módulo se tendrán que realizar diferentes actividades como la edición, consulta y corrección desde un registro hasta un conjunto de datos en una interfaz diseñada para las operaciones diarias. Se podrán homologar las estaciones de la IATA²³ con las estaciones de OACI²⁴ y también editar, modificar y dar de baja remesas y usuarios en sus respectivos formularios.

Este módulo deberá contener los siguientes procedimientos y módulos tal como lo muestra el siguiente diagrama 4.1.4.3.1.

²³ Asociación Internacional de Transportación Aérea.

²⁴ Organización de Aviación Civil Internacional

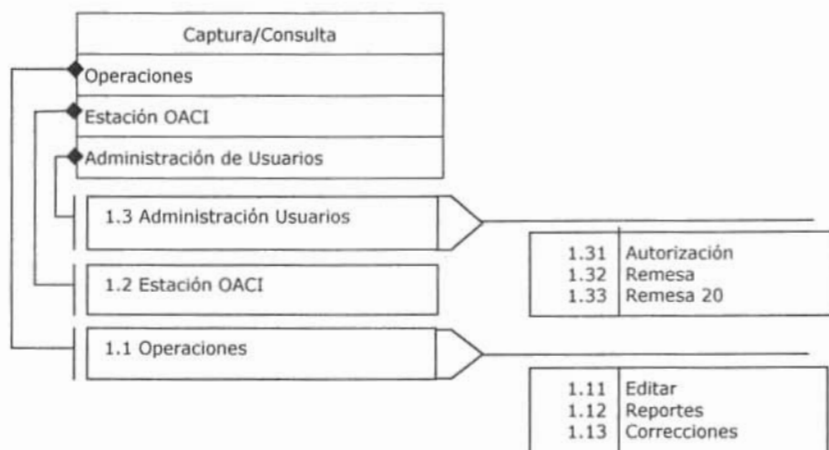


Diagrama 4.1.4.3.1. Módulo de Captura/Consulta.

4.1.4.4. SubMódulo Operaciones (1.1).

Empezaremos por analizar el formulario de Edición de Operaciones Diarias (1.1): Este submódulo deberá estar conformado por 3 pantallas: Edición (1.11), Reportes (1.12) y Correcciones (1.13).

4.1.4.5. Procedimiento de Corrección de Operaciones Diarias (1.11).

Una vez efectuada la carga de operaciones en los aeropuertos, este módulo deberá de ser capaz de presentar la información en una pantalla, en donde se muestren todos los registros, pero sobretodo aquellos que no tuvieron la información correcta o completa desde el lugar de captura en cada torre de control. A continuación se presentan los campos que son enviados desde las torres de control, definiendo a cada uno.

- Fecha de servicio: Fecha en la cual se llevó a cabo el servicio de navegación aérea (Salidas o Llegadas).
- Numerado: Índice o número de registro de cada operación.
- Consecutivo: Número que se incrementa de acuerdo con la fecha de servicio.
- Llegada/Salida: Sí es una llegada = L, y es una Salida es S.
- Número de Usuario: El número de usuario es obtenido de acuerdo a las Siglas
- Número de Vuelo: El numero de vuelo. Sí es aviación comercial, si es aviación general en forma de cobro por formula.
- Lugar de Origen.
- Lugar de Destino.

- Clave del Equipo o Aeronave.
- Clave de la Matrícula.
- Hora de Llegada/Salida.
- Fecha de Modificación.
- Fecha de Carga al sistema.
- Tipo de Error en la carga.
- Validación del registro correcto.
- Remesa.
- Siglas de la Aeronave.
- Siglas de IATA/OACI.
- Observaciones.

En el Formulario de Operaciones Diarias, se agrupan los datos importados de Operaciones Diarias en 19 registros. Cabe hacer mención que al momento de ejecutar la importación de operaciones de todos los aeropuertos, el sistema guardará en un campo, los registros incompletos o incorrectos de los campos de USUARIO, CLAVE_EQUIPO, CLAVE_MATRICULA, ORIGEN y DESTINO. Estos son los únicos que pueden ser modificados. Es importante mencionar que en este formulario estamos trabajando para una remesa en particular, determinada por el nivel de ingreso con el que entramos al sistema. Si olvidásemos el número de la remesa con el que estamos trabajando podremos obtener el dato que previamente se registro a la entrada en la base de datos con el nombre de **Remesa**.

Simulando una interfaz para las operaciones cargadas a la BD, debemos obtener un cuadro como el que se muestra en la tabla 4.1.4.5.1.

Fecha de servicio	Número	Consecutivo	Tipo_LS
Clave Equipo	Clave Matrícula	IdMm_LS	Fecha Modificación
Usuario	Vuelo	Origen	Destino
Fecha Carga	Errores	Marca Ok	Remesa
Siglas	Estiata	Ot servaciones	

Tabla. 4.1.4.5.1. Módulo de Captura/Consulta.

El campo de Usuario debe ser de tipo numérico, tendrá un tamaño de 5 caracteres, y acepta dígitos del 1 al 99,999. El campo de **Origen** y **Destino** son del tipo alfanumérico y acepta 4 caracteres. El campo de **Clave de Equipo**, acepta 10 caracteres alfanuméricos. El campo de **Clave de Matrícula** acepta 9 caracteres alfanuméricos.

En cada uno de los cinco casos, cuando ingresemos el dato, podemos verificar su validez al momento de escribirlo. En cuyo caso deberá indicarnos el resultado de dicho ingreso.

Por ejemplo, al escribir en Origen "abcd", nos debe mostrar el siguiente mensaje: "El origen es incorrecto" y activara la casilla de texto en rojo y si continuamos, sin modificar el error y grabamos, quedará en la casilla de error una "X" en su correspondiente ubicación dentro de la casilla de errores, como se muestra en la tabla siguiente.

4.1.4.6. Relación de Errores.

En la siguiente tabla 4.1.4.6.1 se muestra la relación en la posición de los errores al momento de la carga.

Clave Equipo	Posición 1	" X "
Origen	Posición 2	" X "
Destino	Posición 3	" X "
Clave Matricula	Posición 4	" X "

Tabla 4.1.4.6.1. Relación de Errores.

Y así sucesivamente, es decir, antes de guardar los cambios en el formulario de edición, podemos revisar que los datos que deseamos ingresar sean los correctos primero por el color del fondo de la casilla y/o por la posición de "X" en errores.

Con un sistema de búsqueda podemos movernos sobre los datos de la pantalla al primer registro con el botón de "Inicio" o al "Final", o ir de uno en uno con el botón de "Siguiente" o regresamos igualmente con el botón de "Anterior", sobre los registros de la base de datos.

Ahora, explicaremos las funciones o comandos necesarios para esta Interfaz. Deberán asociarse cada uno a un botón. El comando **Cancelar** como su nombre lo indica cerrará la aplicación. El comando **Guardar** efectúa la grabación de los movimientos actualizados en pantalla siempre y cuando hayan sufrido algún cambio. Por ejemplo, si nosotros corregimos la clave del equipo, primero lo ingresamos y si nuestro equipo esta correcto emitirá un mensaje "Clave de Equipo Correcto" y entonces sabremos que podemos realizar la grabación. Esto puede aplicarse en la misma forma a los otros 4 campos (Matricula, Usuario, Origen y Destino) referidos con anterioridad.

Al guardar, el sistema deberá efectuar una serie de validaciones -las cuales se explicaran con detalle más adelante- con el objeto de almacenar correctamente la información; si alguno de los registros no cumple con estas directrices el campo no será guardado. Uno de los casos podría ser cuando se trate de grabar un registro vacío, como matrícula. Si esto es cierto no se guardará la información. A pesar de ello, el sistema nos indicará en sus mensajes el problema y con ello podremos resolver más

rápidamente los errores; Conforme se realiza la validación se mostrarán seguidos de un aviso los mensajes correspondientes.

Otro de los comandos importantes que deberá tener este formulario es la búsqueda individual de información o en conjunto.

Deberá contar con 3 características:

1. Buscar un dato Alfabético o numérico. Si nosotros decidimos buscar un dato alfanumérico este nos da la posibilidad de buscar un dato haciendo una búsqueda general.
2. Si seleccionamos No. de Vuelo, ira única y exclusivamente a buscar el número de vuelo.
3. Si seleccionamos "Usuario" dentro del marco numérico, consultará sobre toda la base de datos dentro de los números de usuarios, aquellos que coincidan con el dato proporcionado. Aquí en esta consulta, podremos decidir hacer una Búsqueda General o una búsqueda por Error que en cuyo caso nos traerá los errores que pertenezcan a este número de usuario.

También podemos buscar solamente los errores de la toda la remesa, para ello no será necesario ingresar ningún dato. La búsqueda la hará automática.

Al seleccionar alguna de las opciones anteriores nos mostrará los botones de acuerdo a la selección indicada.

Por último, en caso de encontrar en la base de datos el dato ingresado nos habilitará el botón de Mostrar Resultados y nos dirá cuantos registros con ese dato se encontraron en la búsqueda.

Al pedir que nos muestre los resultados nos regresará a la pantalla de Edición con los datos encontrados, de esta manera podremos trabajar con datos más específicos.

4.1.4.7. Módulo de Reportes (1.12).

Es en este procedimiento es donde llevaremos a cabo la presentación de nuestras operaciones. El sistema deberá contar con dos tipos de reportes.

- Reporte de Remesa Por Usuario.
- Reporte de Operaciones.

Deberá presentarse en el menú principal en el módulo de Captura/Consulta, Operaciones, Reportes.

En el reporte deberá presentarse el nombre del usuario y de él se enlistan cada una de las remesas que le pertenecen. En la parte de la derecha, aparecerán 10 números ascendentes en forma vertical. Cada uno de ellos es una hoja con datos.

El reporte deberá permitirnos navegar por las páginas con solo colocar el cursor en el número que queramos o bien, utilizar las flechas para generar este movimiento.

4.1.4.8. Módulo de Correcciones (1.13).

Como su nombre lo indica, aquí se realizarán correcciones de una manera muy amplia. En el editor de operaciones diarias, podremos realizar la corrección de dato por dato. En cambio en el submódulo podremos realizar en un solo movimiento cambios a un campo y este se reflejará a toda la base de datos, sin tener que hacerlo uno por uno.

Las correcciones dependerán del número de usuarios. Con ello se pretende homologar los datos en la base. Los registros que pueden ser corregidos son la clave del equipo, el origen, el destino, y la clave de la matrícula. Su uso será muy sencillo, primero introduciremos el número de usuario con el cual trabajaremos, después seleccionaremos el registro que queremos corregir.

El submódulo cargará de la base de datos todas las claves de equipo correspondientes al usuario. Debemos poder seleccionar de la lista el campo que queremos corregir por un lado y por el otro lado escogeremos el campo correcto y que sustituirá el dato antiguo o incorrecto. Al "Actualizar", se convertirá la información desplegando un mensaje, el cual nos indicará si fueron realizadas o no las modificaciones.

4.1.4.9. Módulo de Estación OACI (1.2).

El propósito de este formulario es el de homologar las siglas de las estaciones IATA²⁵ con las siglas de las estaciones OACI²⁶.

Para ello diseñaremos una lista de estaciones IATA, en la cual podemos movernos con el ratón o con las flechas o bien escribiendo los 3 caracteres de IATA. Conforme vayamos escribiendo las estaciones, nuestra lista ira buscando letra por letra hasta llegar a la más cercana, es decir, si buscáramos la estación de Frankfurt, Alemania (FRA), colocaría el ratón sobre la lista, hasta seleccionarla, inmediatamente después escribiríamos el primer carácter (F), enseguida la lista se ubicará en la primera "F" de la lista, luego teclearíamos la letra (R), y la lista buscará entre las "F's", la

²⁵ IATA (International Air Traffic Association – Asociación Internacional de Tránsito Aéreo).

²⁶ OACI (Organización de Aviación Civil Internacional).

primera "R" y por último pulsáramos la letra (A) y aparecerá entonces las siglas FRA, seguidas del nombre.

Para poder realizar otra búsqueda, tendremos que pulsar la tecla de Escape. Cabe mencionar que la estación OACI debe ser de 4 caracteres alfabéticos.

Por último procederemos a guardar la operación, antes de guardar la operación nos preguntará si estamos seguros de realizar la operación, ya que una vez efectuada, no podrá deshacerse el cambio.

4.1.4.10. SubMódulo de Administración de Usuarios (1.3).

Este submódulo deberá contar con tres formularios: Autorización, Remesa y Remesa 20.

En estos tres formularios el acceso es permitido para los 3 niveles (Administrador, Supervisor y Capturista), pero en el caso de Autorización se restringen algunas de las acciones para supervisor y capturista.

4.1.4.11. Formulario de Autorización.

El propósito principal de este formulario es el de otorgar permisos de acceso al sistema, para ello ésta pantalla necesita de los siguientes datos para funcionar correctamente. Los datos que requiere son los siguientes: No. de Remesa, el password, el nombre del usuario y el nivel de acceso que se le dará.

El nivel de administrador tiene asignado el número 0, el supervisor 1 y el capturista el 2.

Una vez que conocemos estos datos podremos dar de alta a los usuarios de este sistema. Sin embargo es importante recalcar que un capturista no puede dar de alta a un supervisor o a un administrador y un supervisor no puede dar de alta a un administrador pero sí a un supervisor y a un capturista. El único que puede dar de alta a cualquier nivel es el administrador.

Este formulario constará de 3 casillas de texto y una lista desplegable de tres registros (0 Administrador, 1 Supervisor, 2 Capturista).

La primera casilla hace referencia al número de remesa y esta es de 2 caracteres numéricos, la segunda casilla es la del password, es de 10 caracteres alfanuméricos y su contenido no puede ser visto por razones de seguridad y la tercera es el nombre del usuario y consta de 50 caracteres alfanuméricos. La lista desplegable hace referencia al nivel de autorización y despliega 3 valores (0 Administrador, 1 Supervisor, 2 Capturista).

Se pueden emplear las flechas para buscar una remesa en particular, pero también se puede escribir el número de remesa en la correspondiente

casilla. Por ejemplo, si escribimos 2 en la casilla de remesa nos desplegará un mensaje diciéndonos que esa remesa no existe, pero si escribimos 3 en remesa nos actualizará la pantalla con los datos de esa remesa. Al movernos con las flechas nos mostrará el registro actual y el total de registros en la parte inferior derecha del formulario.

4.1.4.12. Alta de Usuarios.

Este procedimiento nos debe permitir limpiar los campos para ingresar los nuevos datos en el formulario. Si quisiéramos dar de alta una remesa que ya existe, nos lo indicará y si quisiéramos podemos sobrescribir los datos que hubiesen existido en la remesa por los nuevos datos.

Edición. Para editar alguna remesa, con el simple hecho de sobrescribir el dato que se requiera cambiar seguido de "Guardar", se realizará el cambio.

Borrar. También es posible eliminar una remesa con todo su contenido, sólo es cuestión de llamar a pantalla la remesa y solicitar borrar, en cuyo caso nos mostrará un aviso para que elimine toda la remesa en cuestión.

Cancelar. Con el botón de cancelar podemos cerrar la aplicación y regresar el menú principal.

Guardar. El botón "Guardar" permite ingresar un dato nuevo o actualizar uno existente, la diferencia consiste en que si encuentra el dato en la base de datos, sabrá de antemano que se trata de una actualización y preguntará si queremos continuar con la actualización. Si se trata de un registro nuevo simplemente hará el registro del dato nuevo sin preguntar nada, anunciándonos que se ha dado de alta un nuevo registro en nuestro centro de mensajes.

Es importante recalcar que para poder efectuar el cambio o el ingreso de un registro es necesario conocer las restricciones de este formulario. Para poder dar de alta o cambiar un registro que pertenezca a un administrador. Sólo podrá llevarse a cabo si el ingreso al formulario lo hizo un administrador.

Si quisiera hacerle un cambio o dar de alta a un administrador con un nivel de acceso de Supervisor o capturista, me mostraría el siguiente mensaje:

Por lo tanto la regla para poder efectuar los cambios, dar de baja o alta un registro es la siguiente:

- El Administrador puede dar de alta, baja y editar a otro Administrador, a un Supervisor y a un Capturista.
- El Supervisor puede dar de alta, baja y editar a Supervisor y Capturista, pero no a un Administrador.

- El Capturista puede dar de alta, baja y editar a un Capturista pero no al Administrador o Supervisor.

4.1.4.13. Formulario de Remesa (1.32).

El objetivo principal de este formulario estriba básicamente en establecer la remesa perteneciente a los usuarios.

Constará de 5 campos de texto y 6 botones de comando. El primer campo de texto tiene 2 caracteres numéricos (1-99) y pertenece a la remesa. El segundo campo de texto es el del Número de Usuario en el se pueden ingresar datos numéricos desde el 1 hasta el 30,000. Debajo de este campo de texto se encuentra el campo de Siglas y sólo nos muestra la información relacionada con el número de usuario y la remesa, en él, no se puede ingresar información. En los dos casos se deberá de validar la existencia del dato.

Por debajo del formulario se encontrarán dos campos para dar de baja a la remesa y al usuario respectivamente y tienen las mismas características que los mencionados anteriormente.

Los botones de comando que se utilizan en este formulario son los siguientes:

Habilitar Buscar. El primero habilita la búsqueda de datos a través de flechas, dándonos la opción de buscar por la remesa o por el usuario. Al igual que en los otros formularios nos mostrará los mensajes en la parte inferior del formulario.

Limpiar. El botón de limpiar como su nombre lo indica quitará del formulario los datos, dejando las casillas en blanco.

Borrar. También es posible eliminar el dato con todo su contenido, sólo es cuestión de llamar un dato a la pantalla y darle clic en borrar, en cuyo caso nos mostrará un aviso para que elimine la remesa/usuario en cuestión.

Guardar. El botón "Guardar" permitirá ingresar un nuevo dato siempre y cuando exista en la base de datos y no haya sido dado de alta en este formulario.

Baja de Remesas. En este botón se podrá dar de baja toda una remesa y sus respectivos usuarios. Este es un procedimiento muy delicado por que puede dar marcha atrás a mucho tiempo de captura, por ello hay que tener mucho cuidado al querer dar de baja toda la remesa y sus usuarios. Sin embargo siempre nos preguntará si queremos dar de baja la remesa y los números de usuarios que le pertenecen.

Baja de Usuarios. En este botón se podrá dar de baja a un usuario en particular.

4.1.4.14. Formulario de Remesa 20 (1.23).

En este formulario se le asigna a una remesa determinada la remesa 20, que fue cargada y por alguna omisión en la importación, no traía datos suficientes para determinar la remesa que pudiera corresponderle. El procedimiento consiste en seleccionar una nueva remesa y asignársela a algún usuario. Haremos esto hasta que todos los usuarios hayan sido relacionados con una nueva remesa. Al final, cuando se le han asignados a todos los números de usuario, las nuevas remesas, habrá que designar a una sola remesa el resto de las remesas 20.

4.1.4.15. Módulo 2: Módulo de Transferencia.

En este módulo se realizan actividades relacionadas con la transferencia de datos ya sea desde nuestro sistema "Exportaciones", creando un archivo o como transferencia a nuestro sistema "Importaciones" obteniendo los datos desde un archivo. Pueden acceder a este módulo el supervisor y el administrador únicamente, tal y como se muestra en el diagrama 4.1.4.15.1.

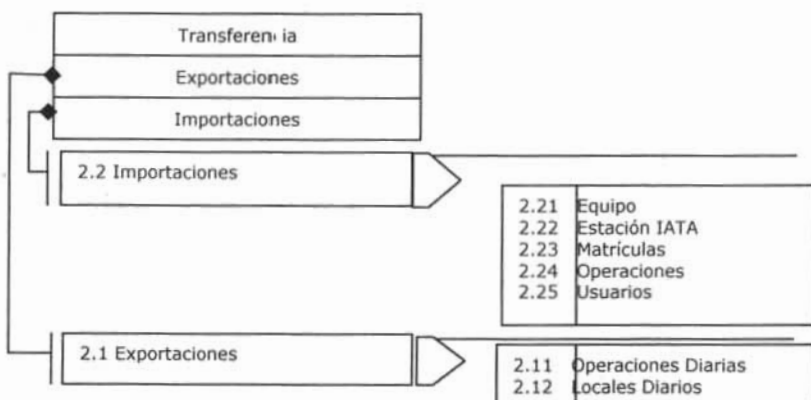


Diagrama del Módulo de Transferencia. 4.1.4.15.1.

4.1.4.16. SubMódulo de Exportaciones (2.1).

Formulario de Exportaciones de Operaciones Diarias (2.11) y Formulario de Exportaciones de Locales Diarios (2.12). Podemos exportar registros de operaciones diarias y locales diarios desde nuestra base de datos.

En este caso en particular, sólo necesitaremos generar el archivo y seleccionar la fecha en la cual estamos exportando los datos. Al accionar el comando para generar el documento, éste crea un archivo de texto llamado "VloPcTwr.txt" en el directorio del servidor "\\servreccenb4 \TWR1\" en donde se encuentra toda la información referente a operaciones diarias.

En la primera columna hará referencia a la fecha de servicio, la segunda al número de usuario, después al equipo, número de vuelo, origen, destino, fecha de exportación, número de remesa y errores respectivamente. Cada vez que se ejecute este formulario, borrará el contenido del archivo y hará de nueva cuenta la carga de los datos, siempre y cuando se trate de una nueva remesa.

4.1.4.17. SubMódulo de Importaciones (2.22).

Es aquí donde se llevará a cabo la importación de los datos relacionados con Matrículas, Equipos, Usuarios, operaciones diarias, Locales diarios y Estaciones IATA/OACI. Cada uno tendrá un archivo designado que es traído de la VAX (computadora central). En un archivo de texto formateado para poder traerlo a nuestro sistema. Tendrá 5 formularios de importaciones, todos tendrán el mismo funcionamiento, excepto el de Operaciones.

4.1.4.18. Formulario de Importaciones de Equipo (2.21).

Empezaremos por elegir el disco donde se encuentra nuestro archivo, es decir el servidor "Servreccenb4". Inmediatamente después nos toca seleccionar la carpeta donde se encuentra el archivo, que está dentro de la carpeta de TWR1 y se llama "catálogos".

Esta ruta será la misma para los otros tres formularios (Estación IATA 2.22, Matrículas 2.23 y Usuarios 2.25).

El nombre del archivo siempre será el mismo y cada formulario tiene su propio archivo de texto asignado, en este caso se trata de "FACEQUIPESOPC.TXT". El cual es un archivo de texto con los datos referentes a la tabla de Equipo. Una vez elegido el archivo correcto, el formulario estará listo para ejecutarse y traer todos los datos contenidos en él hacia nuestra base de datos.

Una vez elegido el archivo se procederá a ejecutar el formulario. En la importación de Equipos, se puede observar el tiempo de carga y el número de registros del archivo. La importación de Equipo trae la siguiente información (Tabla 4.1.4.18.1):

Número de Usuario	Cve_Equipo	PesoPromTon
Número <= 30,000	Texto(10)	Número

Tabla 4.1.4.18.1 Importación de Equipo.

Para cada uno de los otros tres formularios los archivos que se tendrán que seleccionar son los siguientes:

4.1.4.19. Formulario de Importación de Estación IATA (2.22).

Para el formulario de importación de Estación IATA, el archivo a seleccionar deberá ser "FACESTFACSPC.TXT". La importación de Estación IATA, traerá la siguiente información:

4.1.4.20. Importación de Matrículas (2.23).

Para el formulario de importación de matrículas, el archivo a seleccionar deberá ser "FACMATRICPC.TXT" y la información que contendrá es la siguiente, ver tabla 4.1.4.20.1.

Cve_Matricula	Siglas	Número de Usuario	Cve_Equipo
Texto(9)	Texto(6)	Numérico<=30,000	Texto(10)

Tabla 4.1.4.20.1 Importación de Matrículas.

4.1.4.21. Importación de Usuarios (2.25).

Para el formulario de importación de usuarios, el archivo a seleccionar deberá ser "FACUSUAPC.TXT" y la información que contiene es la siguiente, ver tabla 4.1.4.21.1.

C e_Matricula	Siglas	Tipo_cliente	Nombre
Numérico<=30,000	Texto(6)	Texto(3)	Texto(40)

Tabla 4.1.4.21.1 Importación de Usuarios.

4.1.4.22. Importación de Operaciones (2.24).

Este formulario será distinto a los anteriores, ya que en lugar de que le demos la ruta para obtener el archivo de texto, éste automáticamente va y busca los archivos de la remesa y los va cargando uno por uno. Indicándonos el nombre y la cantidad de archivos que importó cuando seleccionemos el comando "ejecutar".

Por otro lado, tenemos que tomar en cuenta que una vez que se cargan los archivos, estos movimientos quedarán registrados en el sistema, de forma tal que no dejará cargar un archivo dos veces.

4.1.5.24. Módulo 3: Módulo de Utilerías.

En este módulo se realizarán actividades relacionadas con el mantenimiento del sistema, desde compactar, reparar y hacer copia de seguridad de la base de datos, como recuperar y soportar datos. En este módulo se le restringe totalmente el acceso a los capturistas y al Supervisor no podrá registrar ni ver las propiedades de la base de datos, tal y como se puede observar en el diagrama 4.1.4.23.1.

3.1	Compactar Base de Datos
3.2	Reparar Base de Datos
3.3	Registrar Base de Datos
3.4	Propiedades Base de Datos
3.55	Soportar Operaciones

3.61	Locales Diarias
3.62	Operaciones Diarias
3.63	Recuperación de Operaciones Diarias y Locales

Diagrama 4.1.4.23.1 Importación de Matrículas.

Las siguientes acciones: Compactar Base de Datos (3.1), Reparar la Base de Datos (3.2) y Crear Copia de Seguridad (3.3), se describen a continuación.

Nuestra base de datos se le ha dado el nombre de "Facopertwr.mdb", contiene todos los datos que necesita nuestro sistema para funcionar correctamente, si se llegará a vaciar o perder la información contenida en la base de datos, nuestro sistema no funcionará adecuadamente, hasta que se vuelvan a cargar todos los datos de nueva cuenta.

El uso continuo de la base de datos provoca que su tamaño crezca demasiado. Por ello es importante compactar la base de datos continuamente. La frecuencia con que debe hacerse depende del uso que se le dé. Se aconseja hacerlo mínimo antes de iniciar cualquier proceso. Esta acción sólo puede realizarla el Supervisor y el Administrador.

La ruta que habrá que seguir deberá ser la siguiente:

\\Servrecursosb4\ingresos\facopertwr.mdb

En donde el primer nombre se refiere al servidor donde se encuentra alojada la Base de Datos. Luego de elegir el archivo de la base de datos, habrá que seleccionar la versión correspondiente.

Antes de empezar con la compactación, nos preguntará si deseamos continuar. Si es afirmativa nuestra respuesta, iniciará con el proceso. Una vez que concluya el proceso de compactar, mostrará en la pantalla un aviso indicándonos, cuando eso suceda. Este proceso puede tomar un par de minutos, dependiendo del tamaño de la base de datos.

Se aconseja realizar esta actividad cada semana como mínimo antes de efectuar cualquier otro procedimiento. Si tratáramos de hacer estas actividades luego de realizar alguna actividad en nuestro sistema, nos mostrará un mensaje indicando que tendremos que cerrar nuestra base de datos y reiniciar nuestro sistema (Salir-Entrar).

4.1.4.24. Registrar la Base de Datos (3.4).

Este procedimiento es de uso exclusivo de los programadores del sistema. Básicamente se trata de un procedimiento para migrar en un futuro nuestra base de datos a SQL Server.

4.1.4.25. Propiedades de la Base de Datos (3.5).

Esta función nos mostrará las características propias de la base de datos.

4.1.4.26. Soportar Operaciones (3.6).

En este procedimiento tenemos tres opciones:

Soportar Locales Diarios (3.61). En esta situación pasaremos nuestros datos de locales diarios a una tabla que se llama HisOperDiarias. Ahí quedarán registrados los movimientos. Cada decena habremos de hacer lo mismo una vez que haya sido importada toda la información de operaciones y locales diarios. Al hacer esto, limpiaremos las operaciones efectuadas, para posteriores importaciones de datos.

Desde el menú de herramientas podremos, seleccionar a "Locales Diarias". Y con ello estaremos iniciando el soporte o resguardo de nuestra información. Al finalizar de hacer el respaldo, la tabla quedará vacía, para aceptar una nueva carga de información.

Soportar Operaciones Diarias (3.62). Una vez que hemos soportado las operaciones de locales diarias, tendremos que soportar las operaciones diarias. Se trata del mismo procedimiento que el anterior. Sólo que en lugar de soportar los registros de locales diarios, soportaremos los registros de las operaciones diarias. Una vez que hemos soportado estos registros, el procedimiento adecuado para seguir soportando será el de no borrar el contenido de la tabla de HisOperDiarias a menos que necesitemos limpiar la tabla. Pero antes tenemos que recuperar la información de los registros de (Locales y Operaciones Diarias).

Recuperar Operaciones Diarias y Locales Diarios (3.63). Como se ha dicho en repetidas ocasiones, la acción de recuperar datos de Operaciones y Locales Diarias, implica que los datos hayan sido trasladados a la tabla histórica. No obstante, puede suceder que las 3 tablas (Operaciones, Locales y la Histórica) se encuentren sin datos, en cuyo caso podría tratarse de un error en la base de datos o bien, el sistema aun no ha importado la información. Pero aparte de estas, lo común será soportar sin borrar el contenido de la tabla histórica (HisOperDiarias).

En el caso de que por algún motivo tengamos que utilizar el procedimiento de recuperación, - como se ha dicho - es importante cerciorarnos de que la información que será recuperada, puede traernos datos de otras remesas o de todas las remesas soportadas y no solamente de una remesa, a menos que se trate de la primera importación.

4.1.4.27. Módulo 4: Módulo de Datos del Sistema.

El último módulo contendrá 3 aspectos que se enlistan a continuación en el diagrama 4.1.4.27.1.

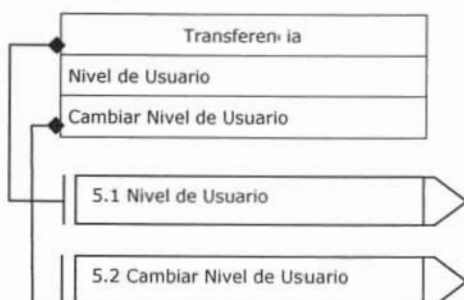


Diagrama 4.1.4.20.1 Módulo de Datos del Sistema.

4.1.4.28. Nivel de Usuario (5.1).

Este procedimiento nos despliega un mensaje con el nivel del usuario con el que se ingresó al sistema. Si llegáramos a olvidar el nivel de usuario, podremos saberlo consultando este procedimiento.

4.1.4.29. Cambio de Nivel de Usuario (5.2).

Aquí podremos reingresar al sistema con otro nivel de acceso, para lo cual nos desplegará el formulario de ingreso tal y como lo hacemos cada vez que iniciamos nuestro sistema. Con la única diferencia de que no se cierra totalmente el sistema, sino que sólo nos da la oportunidad de cambiar el nivel de usuario (Administrador, Supervisor o Capturista).

4.2. DISEÑO.

Partiendo del análisis previamente realizado, es necesario ahora definir tanto la arquitectura en la que serán distribuidos los datos para su óptima utilización y explotación. Esta arquitectura se define a partir de un modelo lógico de entidades y relaciones en las que matemáticamente se definen tanto los datos que formarán parte de la aplicación como su forma de interactuar. Por otro lado es necesario definir la arquitectura de los elementos a través de los cuales la aplicación cumplirá con los requerimientos funcionales y accederá a los datos. En esta etapa del proyecto es necesario definir qué clases formarán parte del sistema y con qué métodos contarán para llevar a cabo todas las funcionalidades esquematizadas en los casos de uso, que a su vez definirán también el procedimiento a seguir para llevar a cabo las pruebas en posteriores etapas del proyecto.

4.2.1. Diseño de Arquitectura de Datos.

4.2.1.1. Proceso de Normalización al Sistema.

En la siguiente sección se presenta el proceso de normalización del sistema, en donde se detalla la normalización de la base de datos.

4.2.1.2. Reglas de Aplicación.

En la tabla siguiente, a manera de resumen, se describe brevemente en que consiste cada una de las reglas, tabla 4.2.1.2.1.

Regla	Descripción
1FN	Incluye la eliminación de todos los grupos repetidos.
2FN	Asegura que todas las columnas que no son llave sean completamente dependientes de la llave primaria (PK).
3FN	Elimina cualquier dependencia transitiva. Una dependencia transitiva es aquella en la cual las columnas que no son llave son dependientes de otras columnas que tampoco son llave.

Tabla 4.2.1.2.1. Formas Normales.

4.2.1.3. Aplicación de la Primera Forma Normal, (1FN).

La regla de la Primera Forma Normal establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas.

Poner la base de datos en la Primera Forma Normal resuelve el problema

de los encabezados de columna múltiples. Muy a menudo, los diseñadores de bases de datos inexpertos harán algo similar a la tabla no normalizada. Una y otra vez, crearán columnas que representen los mismos datos. La normalización ayuda a clarificar la base de datos y a organizarla en partes más pequeñas y más fáciles de entender. En lugar de tener que entender una tabla gigantesca y monolítica que tiene muchos diferentes aspectos, sólo tenemos que entender los objetos pequeños y más tangibles, así como las relaciones que guardan con otros objetos también pequeños.

4.2.1.4. Aplicación de la Segunda Forma Normal, (2FN).

La regla de la Segunda Forma Normal establece que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas. Una dependencia parcial es un término que describe a aquellos datos que no dependen de la llave primaria de la tabla para identificarlos.

Una vez alcanzado el nivel de la Segunda Forma Normal, se controlan la mayoría de los problemas de lógica. Podemos insertar un registro sin un exceso de datos en la mayoría de las tablas.

4.2.1.5. Aplicación de la Tercera Forma Normal, (3FN).

Una tabla está normalizada en esta forma si todas las columnas que no son llave son funcionalmente dependientes por completo de la llave primaria y no hay dependencias transitivas. Comentamos anteriormente que una dependencia transitiva es aquella en la cual existen columnas que no son llave que dependen de otras columnas que tampoco son llave.

Cuando las tablas están en la Tercera Forma Normal se previenen errores de lógica cuando se insertan o borran registros. Cada columna en una tabla está identificada de manera única por la llave primaria, y no debe haber datos repetidos. Esto provee un esquema limpio y elegante, que es fácil de trabajar y expandir.

Un dato sin normalizar no cumple con ninguna regla de normalización. Vamos a explicar en que consiste cada una de las reglas, considerando los datos de la siguiente tabla 4.2.1.5.1.

Remesa	Passw	NomUsuSen	Nivel Autor.	NumUsu	Siglas	Tipo Cliente	Nombre	CveEquipo	PesoProm
1	123	Pedro	1	9700	MXA	FOR	Mexicana aviación	B727200	85
1	123	Pedro	1	9704	AMX	FOR	Aerolíneas de México	MD80	65
1	123	Pedro	1	9704	AMX	FOR	Aerolíneas de México	MD88	47
2	456	Juan	1	11	COA	COM	Compañía	DC9	50
3	456	Luis	3	97	CHP	FOR	Aerolínea	B727276	65
	789	Laura	2	97	CHP	FOR	Aerolínea	CD9	50

Tabla 4.2.1.5.1 Ejemplo de un conjunto de datos sin normalizar.

Al examinar estos registros, podemos darnos cuenta que contienen un dato repetido para todos los grupos. La 1FN prohíbe los grupos repetidos, por lo tanto tenemos que convertir a la primera forma normal. Los pasos a seguir son:

- Tenemos que eliminar los grupos repetidos.
- Tenemos que crear una nueva tabla con la PK²⁷ de la tabla base y el grupo repetido.

Primero separamos los registros que tienen algo en común, como son los de la Remesa con los del Password de Usuarios, y al último; los niveles de autorización, tal y como se aprecia en la tabla 4.2.1.5.2.

REMESAS-AUTORIZACION

ID_Remesa	Pass	NomUsuSen	NivelAutorizac
1	123	Pedro	1
2	456	Juan	1
3	456	Luis	3
3	789	Laura	2

Tabla 4.2.1.5.2. Tabla Remesas-Autorización.

Hacemos lo mismo para los Usuarios o compañías con los equipos (los cuales son comunes a sí mismos), ver tabla 4.2.1.5.3.

USUARIOS-EQUIPOS

NumUsu	Siglas	TipoCli	Nombre	CveEquipo	PesoProm
9700	MXA	FOR	Mexicana de aviación	B727200	85
9704	AMX	FOR	Aerolíneas de México	MD80	65
11	COA	COM	Compañía	DC9	50
97	CHP	FOR	Aerolínea	B727276	65
97	CHP	FOR	Aerolínea	DC9	50

Tabla 4.2.1.5.3. Tabla Usuarios-Equipos.

²⁷ PK. Primary Key (llave primaria)

Ahora procederemos a aplicar la segunda forma normal, es decir, tenemos que eliminar cualquier columna no llave que no dependa de la llave primaria de la tabla. Los pasos a seguir son:

- Determinar cuáles columnas que no son llave no dependen de la llave primaria de la tabla.
- Eliminar esas columnas de la tabla base.
- Crear una segunda tabla con esas columnas y la(s) columna(s) de la PK de la cual dependen.

La tabla REMESAS-AUTORIZACION está en 2FN. Cualquier valor único de NumUsu determina un sólo valor para cada columna. Por lo tanto, todas las columnas son dependientes de la llave primaria NumUsu.

Por su parte, la tabla USUARIOS-EQUIPOS no se encuentra en 2FN ya que la columna PesoProm es dependiente de CveEquipo, pero no es dependiente directamente de NumUsuario. Lo que haremos a continuación es eliminar estas columnas de la tabla USUARIOS-EQUIPOS y crear una tabla EQUIPO-PESO con dichas columnas y la llave primaria de la que dependen.

Las tablas quedan ahora de la siguiente manera. Pero aún existen grupos repetidos, por lo que hay que seguir eliminando grupos repetidos. Quedando finalmente 5 tablas a las que llamaremos:

- Remesas
- Autorización
- Permisos
- Usuarios
- EquipoPeso

La AUTORIZACIÓN (tabla 4.2.1.5.4) depende del Password que se da y únicamente se puede acceder a la remesa que tenga asignado dicho password. No se puede entrar a Remesas ajenas.

AUTORIZACIÓN

Remesa	Passw	NomUsuSen	NivelAutorizac
1	123	Pedro	1
2	456	Juan	1
3	456	Luis	3
3	789	Laura	2

Tabla 4.2.1.5.4. Tabla Autorización.

La tabla USUARIOS debe estar separada de los equipos (Tabla 4.2.1.5.5.)

USUARIOS

NumUsu	Siglas	TipoCli	Nombre
9700	MXA	FOR	Mexicana de Aviación
9704	AMX	FOR	Aerolíneas de México
11	COA	COM	Compañía
97	CHP	FOR	Aerolínea

Tabla 4.2.1.5.5. Tabla Usuarios.

La tabla REMESAS (Figura 4.2.1.5.6.) contiene los Números de Usuario o compañías. Así; una remesa tiene varios usuarios, además; ésta nace de la tabla de USUARIOS y de la tabla de AUTORIZACION.

REMESAS

Remesa	NumUsu
1	9700
1	9704
2	11
3	97

Tabla 4.2.1.5.6. Tabla Remesas.

De la tabla de EQUIPOS-USUARIOS (la cual convertimos a USUARIOS), obtenemos como nueva tabla a EQUIPO-PESO (ver tabla 4.2.1.5.7.), en la que colocamos únicamente a los equipos y su peso promedio.

EQUIPO-PESO

NumUsu	CveEquipo	PesoPromTonel
9700	B727200	85
9704	MD80	65
11	DC9	50
97	B727276	65

Tabla 4.2.1.5.7. Tabla Equipeso.

4.2.1.6. Desarrollo a la Tercera Forma Normal.

La tercera forma normal nos dice que tenemos que eliminar cualquier columna no llave que sea dependiente de otra columna no llave. Los pasos a seguir son:

- Determinar las columnas que son dependientes de otra columna no llave.
- Eliminar esas columnas de la tabla base.
- Crear una segunda tabla con esas columnas y con la columna no llave de la cual son dependientes.

Al observar las tablas que hemos creado, nos damos cuenta que todas las tablas ya se encuentran en 3FN, pues ya se hicieron todas las divisiones posibles, lo único que falta ver es cuál es la llave primaria. Por ejemplo; la

tabla AUTORIZACION tiene a las columnas **Passw**, **NomUsuSen**, **Nivel Autorización** como dependientes de la columna **Remesa**. USUARIOS tiene a las columnas **Siglas**, **TipoCliente**, y **Nombre** como dependientes de la columna **NumUsu**. EQUIPO-PESO tiene a la columna **PesoPromTonel** como dependiente de las columnas **NumUsu** y **CveEquipo**. Obsérvese que aquí se tienen dos columnas como llave primaria (PK).

En la tabla OPERACIONES DIARIAS, se van a tener todos los registros que están cambiando (Tabla de trabajo 4.2.1.6.1). De estos registros nos interesa saber: Fecha de servicio, si es una llegada o salida, número de usuario, número de vuelo, origen, destino, equipo con que vuela, matrícula del equipo, hora de llegada o salida.

OPERDIARIAS

Fecha Servicio	Numerado	Consecutivo	Tip_LS	NumUsu	NumVuelo
2004/07/18	01	56	L	9709	463
2004/07/19	02	57	S	9710	605
2004/07/20	03	58	S	9711	424

Origen	Destino	CveEquipo	CveMatricula	HhMm_LS	FechaModif
MEX	COL	AT45	XATAH	03:28	08/08/2004
TIJ	LAP	DC91	XARXG	20:35	08/08/2004
GDL	TIJ	B737	XATWG	04:03	08/08/2004

FechaCarga	Errores	MarcaOK	Remesa	Observaciones	EstIata	Siglas
08/08/2004		1	2		COL	TAO
08/08/2004		1	7		LAP	SER
08/08/2004		1	3		TIJ	LCD

Tabla 4.2.1.6.1 Tabla OperDiarias.

Analicemos la tabla Operaciones Diarias. La **1FN** prohíbe los grupos repetidos, por lo tanto tenemos que convertir a la primera forma normal. Los pasos a seguir son:

- Tenemos que eliminar los grupos repetidos.
- Tenemos que crear una nueva tabla con la PK de la tabla base y el grupo repetido.

No hay nada que eliminar, no hay nada que crear

Ahora procederemos a aplicar la segunda forma normal. Tenemos que eliminar cualquier columna no llave que no dependa de la llave primaria de la tabla. Los pasos a seguir son:

- Determinar cuáles columnas que no son llave no dependen de la llave primaria de la tabla.
- Eliminar esas columnas de la tabla base.

- Crear una segunda tabla con esas columnas y la(s) columna(s) de la PK de la cual dependen.

Las columnas no llave dependen de la PK. Ya se encuentra normalizada bajo la 2FN

La tercera forma normal nos dice que hay que eliminar cualquier columna no llave que sea dependiente de otra columna no llave. Los pasos a seguir son:

- Determinar las columnas que son dependientes de otra columna no llave.
- Eliminar esas columnas de la tabla base.
- Crear una segunda tabla con esas columnas y con la columna no llave de la cual son dependientes.

Todas las columnas dependen de una única columna llave, por lo que se encuentra normalizada en la 3FN.

ESTFACS

EstIata	EstOasi	Grupo	NombreEst	Región
MEX	MMM	5	Mexico	Nal
VSA	MMVA	3	Villahermosa	Nal
VIE	LOWW	0	Viena Austria	Ext

Tabla 4.2.1.6.2. Tabla Estfacs.

Esta tabla (Tabla 4.2.1.6.2.) es necesaria para validar los registros que se introduzcan en la tabla de OPERDIARIAS

Ahora hay que examinar los registros que se encuentran en MATRICULAS.

Tenemos que se repite el NumUsu. Esto se debe a que hay un vínculo con la tabla de USUARIOS. La 1FN prohíbe los grupos repetidos, por lo tanto tenemos que convertir a la primera forma normal. Los pasos a seguir son:

- Tenemos que eliminar los grupos repetidos.
- Tenemos que crear una nueva tabla con la PK de la tabla base y el grupo repetido.

El campo NumUsu debe existir, en esta tabla, pues es necesario para poder saber a qué usuario pertenece una matrícula. No hay nada que eliminar, no hay nada que crear.

Ahora procederemos a aplicar la segunda forma normal. Tenemos que eliminar cualquier columna no llave que no dependa de la llave primaria de la tabla. Los pasos a seguir son:

- Determinar cuáles columnas que no son llave no dependen de la llave primaria de la tabla.
- Eliminar esas columnas de la tabla base.

- Crear una segunda tabla con esas columnas y la(s) columna(s) de la PK de la cual dependen.

Las columnas no llave dependen de la PK. Ya se encuentra normalizada bajo la 2FN

La tercera forma normal nos dice que hay que eliminar cualquier columna no llave que sea dependiente de otra columna no llave. Los pasos a seguir son:

- Determinar las columnas que son dependientes de otra columna no llave.
- Eliminar esas columnas de la tabla base.
- Crear una segunda tabla con esas columnas y con la columna no llave de la cual son dependientes.

MATRÍCULAS

CveMatrícula	Siglas	Numusu	Equipo
XAMEE	MXA	9700	B722
XAMAL	GMT	9515	B732
XAMAM	000	0	FA20M
XAFPV	GRO	9506	MD81

Tabla 4.2.1.6.3. Tabla Matrículas.

La siguiente decisión es ¿qué tan lejos debe llevar la normalización? La normalización es una ciencia subjetiva. Determinar las necesidades de simplificación depende de nosotros. Si nuestra base de datos va a proveer información a un solo usuario para un propósito simple y existen pocas posibilidades de expansión, normalizar los datos hasta la 3FN quizá sea algo exagerado. Las reglas de normalización existen como guías para crear tablas que sean fáciles de manejar, así como flexibles y eficientes. A veces puede ocurrir que normalizar los datos hasta el nivel más alto no tenga sentido.

¿Se están dividiendo tablas sólo para seguir las reglas o estas divisiones son en verdad prácticas? Éstas son el tipo de cosas que nosotros como diseñadores de la base de datos, necesitamos decidir, y la experiencia y el sentido común nos pueden auxiliar para tomar la decisión correcta. La normalización no es una ciencia exacta, más bien subjetiva.

Existen seis niveles más de normalización que no se han discutido aquí. Ellos son Forma Normal Boyce-Codd, Cuarta Forma Normal (4NF), Quinta Forma Normal (5NF) o Forma Normal de Proyección-Unión, Forma Normal de Proyección-Unión Fuerte, Forma Normal de Proyección-Unión Extra Fuerte y Forma Normal de Clave de Dominio. Estas formas de normalización pueden llevar las cosas más allá de lo que necesitamos. Éstas existen para hacer una base de datos realmente relacional. Tienen que ver principalmente con dependencias múltiples y claves relacionales.

En resumen, la Normalización es una técnica que se utiliza para crear relaciones lógicas apropiadas entre tablas de una base de datos. Ayuda a prevenir errores lógicos en la manipulación de datos, facilita también agregar nuevas columnas sin romper el esquema actual ni las relaciones.

Se discutieron las primeras tres formas. Éstas proveen suficiente nivel de normalización para cumplir con las necesidades de la mayoría de las bases de datos. Normalizar demasiado puede conducir a tener una base de datos ineficiente y hacer a su esquema demasiado complejo para trabajar. Un balance apropiado de sentido común y práctico puede ayudarnos a decidir cuándo normalizar, como se puede observar en las tablas normalizadas que a continuación se presentan (tablas 4.2.1.6.4, 4.2.1.6.5, 4.2.1.6.6, 4.2.1.6.7, 4.2.1.6.8, 4.2.1.6.9 y 4.2.1.6.10).

Tabla AUTORIZACION

Remesa	Password	NomUsuSen	NivelAutorizac
1	123	Pedro	1
2	456	Juan	1
3	456	Luis	3
3	789	Laura	2

Tabla 4.2.1.6.4. Tabla Autorización.

Tabla USUARIOS

NumUsu	Siglas	TipoCli	Nombre
9700	MXA	FOR	Cía Mexicana de Aviación
9704	AMX	FOR	Aerolíneas de México
11	COA	COM	Compañía
97	CHP	FOR	Aerolínea

Tabla 4.2.1.6.5. Tabla Usuarios.

Tabla REMESAS

Remesa	NumUsu
1	9700
1	9704
2	11
3	97

Tabla 4.2.1.6.6. Tabla Remesas.

Tabla EQUIPO-PESO

NumUsu	CveEquipo	PesoPromTonel
9700	B727200	85
9704	MD80	65
11	DC9	50
97	B727276	65

Tabla 4.2.1.6.7. Tabla Equipeso.

Tabla OPERDIARIAS

Fecha Servicio	Numerado	Consecutivo	Tip_LS	NumUsu	NumVuelo
2004/07/18	01	56	L	9709	463
2004/07/19	02	57	S	9710	605
2004/07/20	03	58	S	9711	424

Origen	Destino	CveEquipo	CveMatricula	HhMm_LS	FechaModif
MEX	COL	AT45	XATAH	03:28	08/08/2004
TIJ	LAP	DC91	XARXG	20:35	08/08/2004
GDL	TIJ	B737	XATWG	04:03	08/08/2004

FechaCarga	Errores	MarcaOK	Remesa	Observac	EstIata	Siglas
08/08/2004		1	2		COL	TAO
08/08/2004		1	7		LAP	SER
08/08/2004		1	3		TIJ	LCD

Tabla 4.2.1.6.8. Tabla Operdiarias.

Tabla ESTFACS

EstIata	EstOasi	Grupo	NombreEst	Región
MEX	MMMX	5	Mexico	Nal
VSA	MMVA	3	Villahermosa	Nal
VIE	LOWW	0	Viena Austria	Ext

Tabla 4.2.1.6.9. Tabla Estfacs.

Tabla MATRÍCULAS

CveMatricula	Siglas	Numusu	Equipo
XAMEE	MXA	9700	B722
XAMAL	GMT	9515	B732
XAMAM	000	0	FA20M
XAFPV	GRO	9506	MD81

Tabla 4.2.1.6.10. Tabla Matrículas.

4.2.1.7. Diccionario de Datos.

A continuación se enlista el diccionario de datos (Tabla 4.2.1.7.1) de las tablas obtenidas luego de haber realizado la normalización.

Tabla: Autorización	Acronimo	Llave	Tipo	Tamaño	Descripción	Valor
	Remesa	SI	Entero	2	Número asignado a personal de Ingreso	1 al 99
	Password	NO	Texto	10	Contraseña sistema	Alfanumérico
	NombredelUSuario	NO	Texto	50	Nombre del Usuario	Nombre del Usuario

SISTEMA DE VALIDACIÓN Y REGISTRO DE OPERACIONES DE LAS TORRES DE CONTROL

	NivelAutorizacion	NO	Entero	2	Nivel de Autorización que se le da al ingresante Administrador, 0 Supervisor, 1 Capturista, 2	0 al 2
Tabla: EquiPeso	NumUsuario	SI	Entero largo	4	Número asignado a los Clientes Aeronáuticos	1 al 9999
	Cve_Equipo	SI	Texto	10	Clave asignada al tipo de aeronave	Alfanumérico
	PesoPromTonel	NO	Doble	8	Peso Promedio en toneladas de las aeronaves	Númerico
Tabla: EstFacs	EstIata	SI	Texto	3	Clave de la estación en formato Iata	A64
	Grupo	NO	Entero	2	cuando gpo= 0, es un aeropuerto no controlado; de 1-4 son aeropuertos controlados	0 al 4
	NombreEst	NO	Texto	30	Nombre de la estación	ARGENTINA CHIS.
	Region	NO	Texto	3	Nal= Nacional, Ext= Extranjero	NAL/EXT
Tabla: OperDiarias	FechaServicio	SI	Texto	10	Fecha en la que se llevo a cabo el servicio de control de tránsito	DD/MM/AAAA
	numerado	SI	Doble	8	Número generado por el sistema de las operaciones registradas	1 a 99,999,999
	Consecutivo	NO	Entero	2	Número consecutivo de operaciones registradas	1 a 99
	Tip_LS	NO	Texto	1	Tipo de operación de llegada o salida	L/S
	NumUsuario	NO	Entero largo	4	El numero de usuario es de acuerdo a las Siglas	1 al 9999
	NumVuelo	NO	Texto	10	El numero de vuelo si es aviación comercial, si es aviación general en forma de cobro por formula	Númerico
	Origen	NO	Texto	4	Estación de donde despega la Aeronave	Alfabético
	Destino	NO	Texto	4	Estación a donde aterriza la aeronave	Alfabético
	Cve_Equipo	PK	Texto	10	Clave del equipo, se valida contra el catalogo de equipo	Alfanumérico
	CveMatricula	PK	Texto	9	Clave de la matrícula se valida contra el catalogo de matriculas	Alfabético
	HhMm_LS	NO	Texto	5	Horas y minutos de despegue o aterrizaje	HH:MM
	FechaModif	NO	Texto	11	Fecha en que se modifiko el registro	DD/MM/AAAA

SISTEMA DE VALIDACIÓN Y REGISTRO DE OPERACIONES DE LAS TORRES DE CONTROL

	FechaCarga	NO	Texto	11	Fecha en que se realizó la carga de la información a la tabla	DD/MM/AAAA
	Errores	NO	Texto	4	Los errores se manejan por posición de 1 caracter; 1 equipo, 2 estOri, 3 estDes, Matricula	XXXX
	MarcaOk	NO	Entero	2	1 = ya fue procesado 0= sin procesar 5 =salidas incobrables, 2=erroneos	0 a 5
	Remesa	PK	Entero	2	Número asignado a personal de Ingreso	1 al 99
	Observaciones	NO	Texto	30	Observación	Campo de texto
	EstIata	NO	Texto	3	Clave de la estación en formato Iata	
	Siglas	NO	Texto	3	Siglas de la Aeronave	Alfanumérico
Tabla: Remesa	Remesa	SI	Entero	2	Número asignado al personal de Ingresos	1 al 99
	NumUsuario	NO	Entero largo	4	El numero de usuario es de acuerdo a las Siglas	1 al 9999
	Siglas	NO	Texto	5	Siglas de la Aeronave	Alfanumérico
Tabla: Usuario	NumUsuario	SI	Entero largo	4	El numero de usuario es de acuerdo a las Siglas	1 al 9999
	Siglas	NO	Texto	5	Siglas de la Aeronave	Alfanumérico
	TipoCliente	NO	Texto	3	Tipo de Cliente	Alfabético
	Nombre	NO	Texto	40	Nombre del Cliente	Alfanumérico
	UsuarioAsa	NO	Entero largo	4	Número de Usuario ASA	1 al 9999

Tabla 4.2.1.7.1. Tabla Diccionario de Datos.

4.2.1.8. Diagrama Entidad Relación.

Se presenta en seguida el DER²⁸ obtenido de acuerdo al análisis anterior, diagrama 4.2.1.8.1.

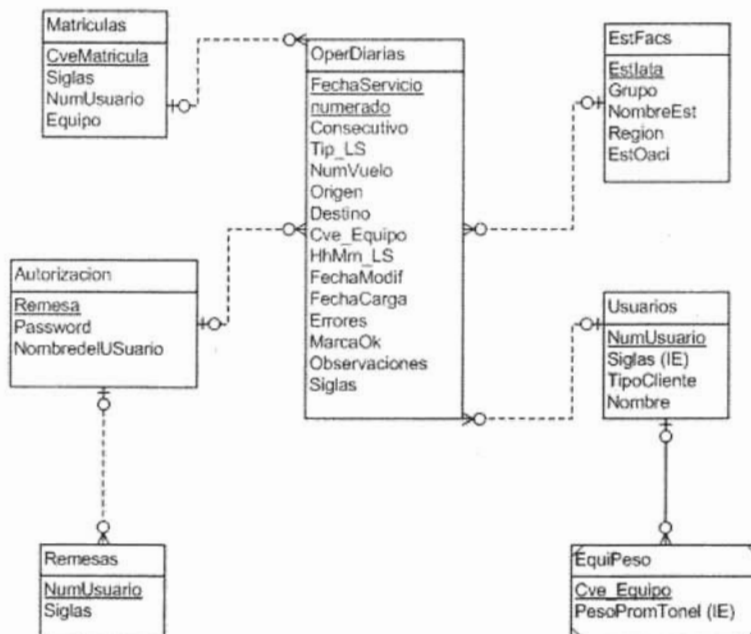


Diagrama Entidad Relación	Fecha: 22 de noviembre del 2004.	
SISTEMA DE VALIDACIÓN Y REGISTRO DE OPERACIONES DE LAS TORRES DE CONTROL		
Microsoft Access 2003	Rev: 1	Autor: Martín Rodríguez
Nombre del Archivo: derokfin.vsd	Compañía: SERNA	

Tabla 4.2.1.8.1. Diagrama Entidad-Relación.

4.2.2. Diseño de Arquitectura de Negocio.

Una arquitectura común de los sistemas de información que abarca una interfaz para el usuario y el almacenamiento persistente de datos se conoce con el nombre de Arquitectura de 3 capas²⁹, aquí se presenta una breve descripción de cada una de ellas:

²⁸ DER, Diagrama Entidad-Relación.

²⁹ Gartner Página 95.

1. Presentación: Consta de ventanas, reportes, interfaces, etc.
2. Lógica de aplicaciones: Tareas y Reglas que rigen el proceso.
3. Almacenamiento: Mecanismo de Almacenamiento persistente.

La calidad tan especial de la arquitectura de 3 capas consiste en aislar la lógica de la aplicación y en convertirla en una capa intermedia bien definida y lógica del software. En la capa de presentación se realiza relativamente poco procesamiento de la aplicación; las ventanas envían a la capa intermedia peticiones de trabajo. Y éste se comunica con la capa de almacenamiento.

4.2.2.1. Diseño de Casos de Uso.

El Modelo de Casos de Uso describe un sistema en término de sus distintas formas de utilización, cada una de estas formas es conocida como un caso de uso. Cada caso de uso o flujo se compone de una secuencia de eventos iniciada por el usuario. Para ello se define el concepto de Actor, correspondiente al tipo de usuario que está involucrado en la utilización de un sistema, siendo el actor una entidad externa al propio sistema. Juntos, el actor y el caso de uso representan los dos elementos básicos de este modelo, ver la siguiente figura 4.2.2.1.1.

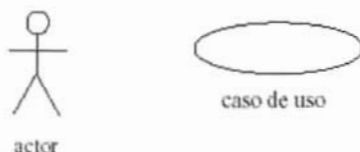


Figura 4.2.2.1. Elementos Básicos del Diseño de Casos de Uso.

4.2.2.2. Actores.

Los actores son entidades distintas a los usuarios, en el sentido que los usuarios son las personas reales que utilizan el sistema, mientras que los actores representan un cierto papel que una persona real puede jugar. Utilizando terminología orientada a objetos, se considera al actor como un tipo de usuario, mientras que a los usuarios se les considera como un tipo de objeto o instancia de una clase. Incluso, una misma persona puede aparecer como diferentes instancias de diferentes actores.

4.2.2.3. Casos de Usos.

Después de haber definido los actores del sistema, se define la funcionalidad propia del sistema por medio de los casos de uso. Utilizando

terminología orientada a objetos, cada caso de uso define una clase o forma particular de usar el sistema mientras que cada ejecución del caso de uso se puede ver como una instancia del caso de uso, o sea, un objeto, con estado y comportamiento. Cada caso de uso constituye un flujo completo de eventos especificando la interacción que toma lugar entre el actor y el sistema.

Cuando los casos de uso se dividen en casos de uso separados, se utilizan principalmente las relaciones de extensión e inclusión como se describe a continuación.

4.2.2.4. Extensión.

Un concepto importante que se utiliza para estructurar y relacionar casos de uso es la extensión. La extensión especifica cómo un caso de uso puede insertarse en otro para extender la funcionalidad del anterior. El caso de uso donde se va a insertar la nueva funcionalidad debe ser un flujo completo, por lo cual éste es completamente independiente del caso de uso a ser insertado. De esta manera, el caso de uso inicial no requiere consideraciones adicionales en relación con los casos de uso que puedan ser insertados, evitando complejidad adicional.

4.2.2.5. Inclusión.

Una relación adicional entre casos de uso es el de inclusión la cual apoya la reutilización de los casos de uso.

Como las asociaciones extensión e inclusión entre casos de uso son ambas asociaciones de clases, se puede uno preguntar cuando se debe escoger entre ellas. De hecho, en la mayoría de los casos, la selección es bastante obvia, y no causa problemas. Un criterio importante es ver qué tanto se acoplan las funcionalidades de los casos de uso. Si el caso de uso a ser extendido es útil por sí mismo, la relación debe ser descrita utilizando extensión. Si los casos de uso son fuertemente acoplados, la inserción debe tomar lugar para obtener un curso completo, la relación debe ser descrita utilizando inclusión.

En el Sistema de Validación y Registro de Operaciones de las Torres de Control se utilizan los actores ya identificados como punto de partida. Dado que el Usuario es el actor primario se comienza con él. El sistema tiene que poder dar ciertos servicios al usuario, como consultas e ingreso de datos propios. De aquí podemos definir nuestros casos de uso principales, nótese que los nombres de los casos de uso deben corresponder a acciones y no tanto a sustantivos.

4.2.2.4. Documentación de las Tablas de Casos de Uso.

Parte fundamental del modelo de casos de uso es una descripción textual detallada de cada uno de los casos de uso identificados. Estos

documentos son sumamente críticos ya que a partir de estos se desarrollará el sistema completo.

Las descripciones de los casos de uso representan todas las posibles interacciones de los actores con el sistema únicamente en base a eventos enviados o recibidos por los actores. En esta etapa no se incluyen eventos internos al propio sistema ya que esto será tratado durante el análisis y únicamente agregaría complejidad innecesaria en esta etapa. El formato del documento que utilizaremos consiste de una primera columna con los aspectos a describirse y una segunda columna con las descripciones correspondientes.

4.2.2.5. Diagrama de Casos de Uso.

En el diagrama que a continuación observamos se presentan la interacción que forman los actores con los casos de uso. Diagrama 4.2.2.5.1.

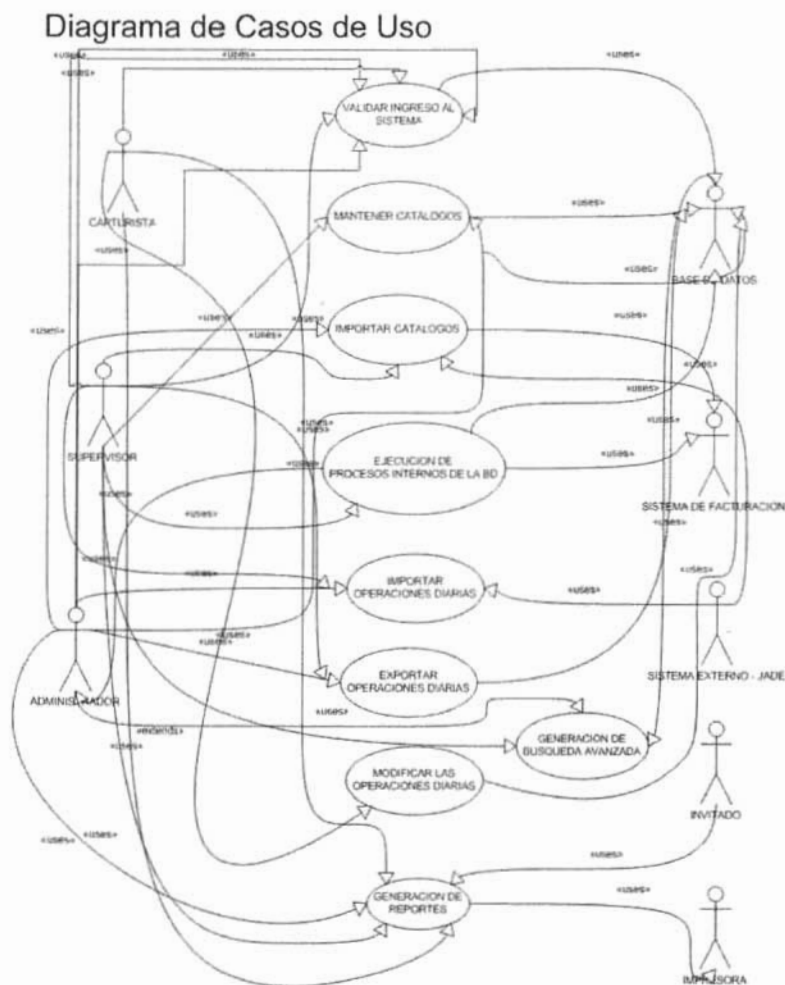


Diagrama 4.2.2.5.1. Diagrama de Casos de Uso.

En la gráfica anterior Diagrama 4.2.2.5.1 se muestran los actores que participan en el diagrama y los casos de uso con los cuales interactúan. De este diagrama obtendremos las tablas siguientes (4.2.2.5.2., 4.2.2.5.3 y 4.2.2.5.4) con la descripción de los casos de uso.

Caso de uso:	Exportación de Operaciones Diarias
Actores:	Base de datos, Administrador y Supervisor.
Propósito:	Realizar la exportación de la base de datos a un archivo de texto de todas las operaciones diarias procesadas por el sistema en un periodo de tiempo determinado.
Resumen:	Este caso de uso es iniciado por el front end cuando el usuario envía un formulario y realiza una búsqueda en la base de datos.
Precondiciones:	Opción del Menú principal
Flujo Principal:	De front-end hacia la base de datos. Este proceso involucra algunos elementos del back end.
Subflujos:	N/A
PostCondiciones:	N/A
Excepciones:	E1: No poder crear archivo de exportacion. E2: No rellenar la información correctamente. E3: No poder leer la base de datos. E4: Si no existen registros de la búsqueda se le informa al actor Front-End.

Tabla 4.2.2.5.2. Caso de Uso, exportación de OperDiarias.

Caso de uso:	Búsqueda avanzada
Actores:	Base de datos, Administrador y Supervisor.
Propósito:	Realizar una búsqueda en base a un determinado conjunto de datos de entrada.
Resumen:	Este caso de uso es iniciado por el front-end cuando el usuario envía un formulario y realiza una búsqueda en la base de datos
Precondiciones:	Opción del Menú principal.
Flujo Principal:	De front-end hacia la base de datos. Este proceso involucra algunos elementos del back end.
Subflujos:	N/A
PostCondiciones:	N/A
Excepciones:	E1: No poder crear formularios. E2: No rellenar la información correctamente. E3: No poder leer la base de datos. E4: Si no existen registros de la búsqueda se le informa al actor Front-End.

Tabla 4.2.2.5.3. Caso de Uso, exportación de Búsqueda Avanzada.

Caso de uso:	Generación de Reportes
Actores:	Capturista, administrador, supervisor, invitado.
Propósito:	Realizar una consulta a la base de datos para obtener como respuesta reportes sobre los usuarios.
Resumen:	Este caso de uso es iniciado por el operador, cuyo inicio se efectúa al presionar el botón de generar reporte en la pantalla correspondiente después de haber seleccionado los campos requeridos.

Precondiciones:	Pantalla principal, Menú del operador.
Flujo principal:	De operador hacia la base de datos. Además involucra una consulta al registro.
Subflujos:	N/A
PostCondiciones:	N/A
Excepciones:	E1: No poder crear el reporte. E2: No rellenar la información correctamente. E3: No poder leer la base de datos. E4: Si no existen registros de la búsqueda.

Tabla 4.2.2.5.4. Caso de Uso. Generación de Reportes.

4.2.2.6. Diagrama de Actividad.

El diagrama de actividad es un diagrama de flujo del proceso multi-propósito que se usa para modelar el comportamiento del sistema. Los diagramas de actividad se pueden usar para modelar un caso de uso, o una clase, o un método complicado. Un diagrama de actividad es parecido a un diagrama de flujo; la diferencia clave es que los diagramas de actividad pueden mostrar procesamiento paralelo (parallel processing). Esto es importante cuando se usan diagramas de actividad para modelar procesos de negocio algunos de los cuales pueden actuar en paralelo, y para modelar varios hilos en los programas concurrentes.

De acuerdo a lo conceptos mencionados, a continuación se muestran algunos de los diagramas de actividad (Diagramas 4.2.2.6.1 y 4.2.2.6.2.) correspondientes a los casos de uso utilizados para nuestro sistema.

DIAGRAMA DE ACTIVIDADES (ALTAS, BAJAS, CAMBIOS EN LOS REGISTROS DE CATALOGOS)

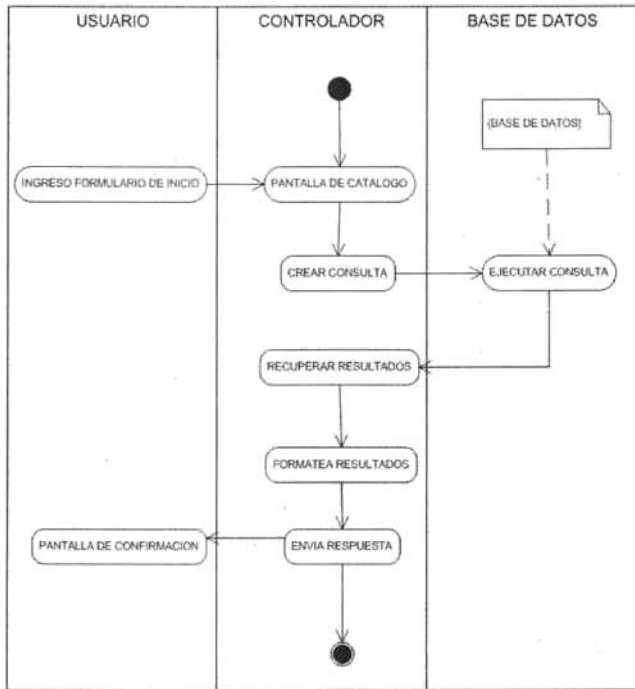


Diagrama 4.2.2.6.1. Diagrama de Actividades. Altas, Bajas y Cambios en los Registros de Catálogos.

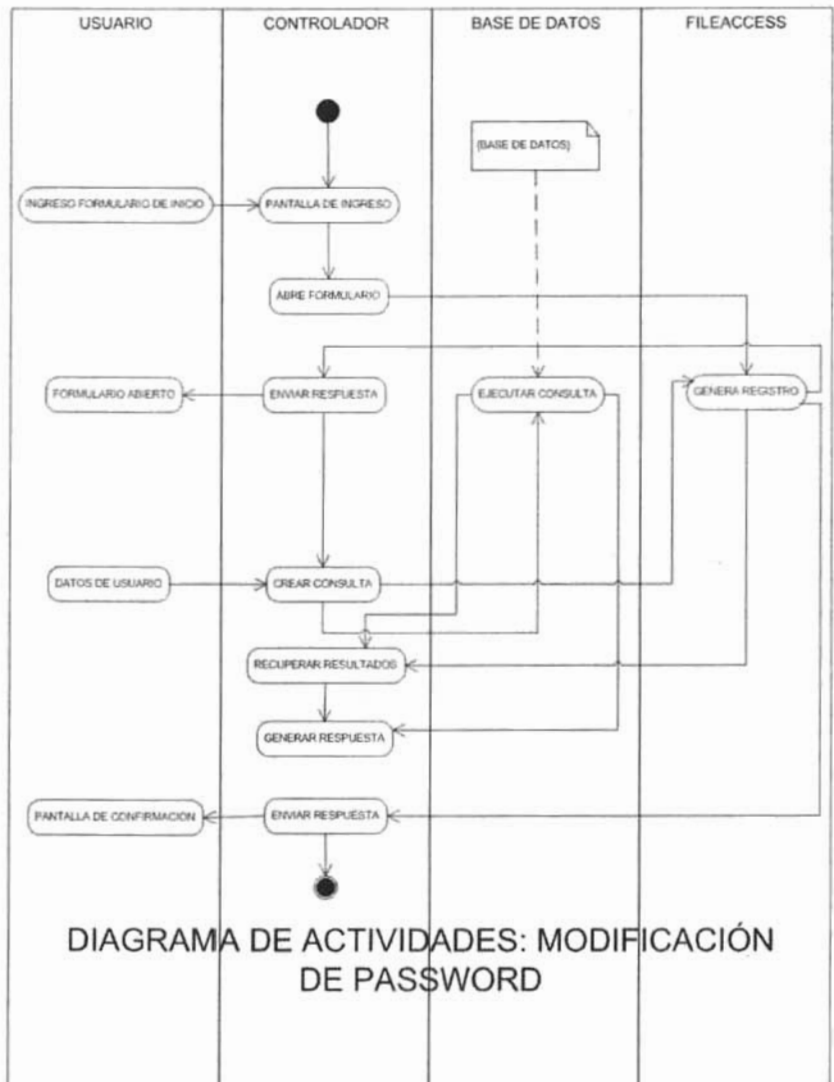


Diagrama 4.2.2.6.1. Diagrama de Actividades para la Modificación de Password.

El objetivo del modelo de análisis es comprender y generar una arquitectura de objetos para el sistema con base a lo especificado en el modelo de requisitos. Durante esta etapa no se considera el ambiente de implantación, lo cual incluye al lenguaje de programación, manejador de base de datos, distribución o configuración de hardware, etc.

4.2.2.7. Identificación de Clases.

Para llevar a cabo la transición del modelo de requisitos al modelo de análisis se deben identificar los objetos necesarios para implementar todos los casos de uso.

La arquitectura de objetos debe considerar los tres tipos de estereotipos de objetos. Para lograr esto se deben identificar primero las clases de interfaz, las de entidad y finalmente las de control. Por otro lado, los objetos entidad e interfaz deben contener funcionalidad más bien local, limitando su efecto en los demás objetos.

Las clases interfaz son bastante fáciles de identificar, donde se cuenta con al menos tres estrategias:

- Se pueden identificar con base a los actores.
- Se pueden identificar con base a las descripciones de las interfaces del sistema que acompañan al modelo de requisitos.
- Se pueden identificar con base a las descripciones de los casos de uso y extraer la funcionalidad que es específica a las interfaces.

4.2.2.8. Clases Entidad.

Se utilizan clases entidad para modelar la información que el sistema debe manejar a corto y largo plazo. La información a corto plazo existe por lo general durante la ejecución del caso de uso, mientras que la información a largo plazo sobrevive a los casos de uso, por lo cual es necesario guardar esta información en alguna base de datos.

Las clases entidad se identifican en los casos de uso, donde la mayoría se identifican del modelo del dominio del problema en el modelo de requisitos. Es por lo tanto esencial trabajar de forma organizada cuando se modelan clases de tipo entidad.

La siguiente es una lista de las operaciones típicas que deben ser ofrecidas por una clase de tipo entidad:

- Guardar y traer información.
- Comportamiento que debe modificarse si el objeto entidad cambia.
- Crear y remover el objeto entidad.

4.2.2.8. Clases de Tipo Control.

Las clases de tipo control típicamente actúan como “pegamento” entre los otros tipos de clases y por lo tanto proveen la comunicación entre los demás tipos de clases. Son típicamente las más efímeras de todos los tipos de clases, dependiendo de la existencia del propio caso de uso.

Las clases de tipo control se identifican directamente de los casos de uso. Como primera aproximación, se asigna una clase de control a cada caso de uso, concreto y abstracto. Dado que se asigna inicialmente el comportamiento a las clases interfaz y entidad para cada caso de uso, el comportamiento restante se asigna a las clases de control.

4.2.2.9. Selección de Clases.

A partir de los tipos de clases que se deben identificar, se realizó la selección de clases que intervienen en el Sistema de Validación y Registro de Operaciones de las Torres de Control, ver tabla 4.2.2.9.1.

CASO DE USO	ACTORES	CLASE INTERFAZ
Alta catálogos	Usuario, Base de datos	Usuario, Base de Datos
Baja catálogos	Usuario, Base de datos	Usuario, Base de Datos
Modificación catálogos	Usuario, Base de datos	Usuario, Base de Datos
Reactivar usuarios	Usuario, Base de datos	Usuario, Base de Datos
CASO DE USO	ACTORES	CLASE ENTIDAD
Alta catálogos	Usuario, Base de datos	Catálogo
Baja catálogos	Usuario, Base de datos	Catálogo
Modificación catálogos	Usuario, Base de datos	Catálogo
Reactivar usuarios	Usuario, Base de datos	Persona
CASO DE USO	ACTORES	CLASE CONTROL
Alta catálogos	Usuario, Base de datos	Manejador de Catálogo
Baja catálogos	Usuario, Base de datos	Manejador de Catálogo
Modificación catálogos	Usuario, Base de datos	Manejador de Catálogo
Reactivar usuarios	Usuario, Base de datos	Persona

Tabla 4.2.2.9.1. Selección de Clases a Verificar.

4.2.2.10. Diagrama de Clases.

Después de haber identificado y seleccionado las clases, se debe construir el diagrama de clases (Diagrama 4.2.2.10.1) para el dominio del problema. Este diagrama se muestra en la figura siguiente es el que se aplica para el, puede ayudar a identificar clases adicionales, y servirá de base para encontrar los atributos y asociaciones entre ellas.

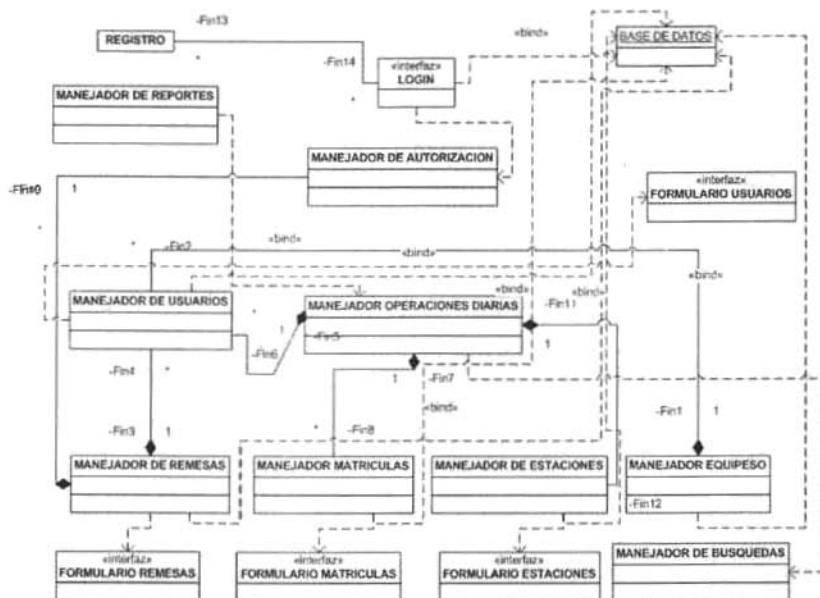


Diagrama 4.2.2.10.1. Diagrama de Clases.

4.2.2.11. Diagramas de secuencia.

Los diagramas de secuencia pueden generarse a partir de las propias clases o incluso a partir de la interacción entre subsistemas. Esto es a menudo una técnica muy útil ya que se puede diseñar un subsistema mostrando sólo las interfaces de los subsistemas relacionados. A un nivel más detallado se pueden mostrar las interacciones internas del subsistema entre los propios objetos. Normalmente, los *eventos* en el diagrama corresponden a protocolos y se especifican exactamente como se verían en el código final.

En este punto se muestran algunos diagramas de secuencia o dinámicos utilizados para nuestro sistema de los cuales serán los de Login.

En el siguiente dibujo (Figura 4.2.2.11.1) muestro el diagrama de secuencia del Login, se puede observar como el usuario empieza el recorrido enviando datos a la clase Login, posteriormente en la clase Manejador Usuario se hará el movimiento de la validación del usuario a la clase Base de datos, la que a su vez regresará la respuesta al usuario para que pueda ingresar a la clase Manejador Autorización estando ahí se desplegará la clase Autorización. También en Autorización se utiliza la clase Manejador de Mensajes en la que se hará envío y recepción de éstos junto con la clase Base de datos, además de hacer uso del registro de usuarios; todo lo anterior con sus respectivas respuestas.

SISTEMA DE VALIDACION Y REGISTRO DE OPERACIONES DE LAS TORRES DE CONTROL

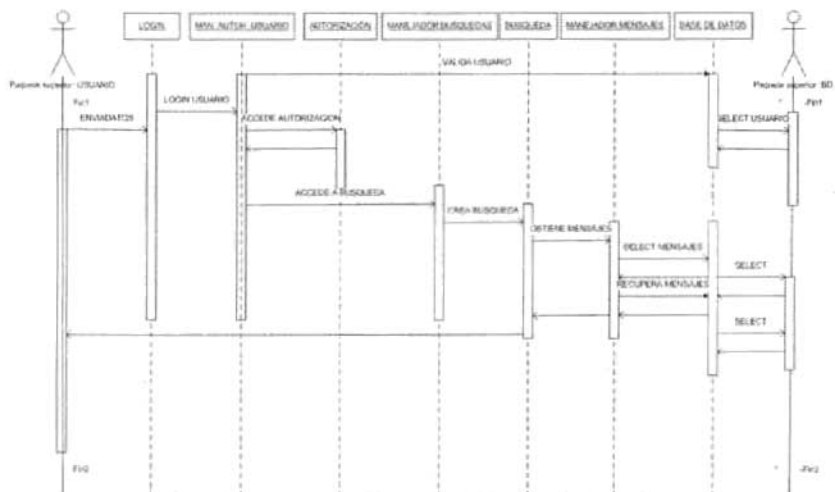


Diagrama 4.2.2.11.1. Diagrama de Secuencia de Login.

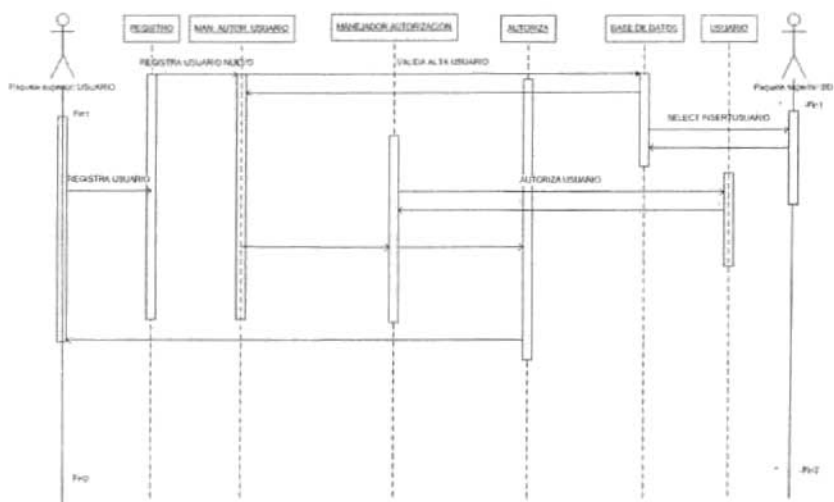


Diagrama 4.2.2.11.2. Diagrama de Registro de Datos.

En la figura anterior 4.2.2.11.2 se muestra el diagrama de secuencia de registro de datos, en el momento en que el usuario se registra a través de la clase Registro, posteriormente la clase Manejador Autorización Usuario realizará el movimiento correspondiente hacia la clase Base de Datos, obteniendo la sesión obteniendo en respuesta para poder enviar al usuario a la clase Manejador Autorización en donde visualizará en Autorización.

4.2.3. Diseño de la Infraestructura Informática.

4.2.3.1. Recursos de Hardware.

A continuación se describen los recursos de hardware necesarios para que las herramientas de software que SERNA utilizará para su funcionamiento, trabajen adecuadamente en el ambiente de desarrollo del sistema:

Los requerimientos de hardware para el sistema operativo de Microsoft que se instalará en los equipos de trabajo son equivalentes para Visual Basic 6.0, y el manejador de bases de datos Microsoft Access 2003, etc., todas las herramientas estarán conviviendo físicamente en el mismo servidor.

En la tabla que a continuación se presenta (Tabla 4.2.3.1.1), se muestra el desglose de las especificaciones que se requieren para la instalación en los equipos para el desarrollo.

Características	Computadora Pentium III @ 1.2 GB
RAM	256 MB
Disco Duro	20 GB
Monitor	Súper VGA (800 x 600) o monitor con 256 colores.
Periféricos y otros	Microsoft Mouse, Impresora de red, Tarjeta de Red, CDROM.

Tabla 4.2.3.1.1. Características de los equipos de cómputo.

En la gráfica podemos observar que los requisitos para la memoria no deben ser menores a 256 MB para un desempeño adecuado. Así mismo se puede ver las otras características que se requieren, como lo son el disco duro, el monitor, el ratón, tarjeta de red, etc.

4.2.3.2. Recursos de Software.

En el análisis de requerimientos de software se hizo un análisis detallado de las características necesarias para el desarrollo del sistema. De tal forma que en este punto sólo mostraremos las generalidades de dichas especificaciones. En el Apéndice A se da una explicación más profunda sobre las características más particulares de algunos programas descritos aquí y se presentará una guía detallada para su instalación.

Aplicación	Requisitos en RAM/Disco Duro
Microsoft Visio 2003	256 MB RAM/400 MB DD
Microsoft Visual Sourcesafe 6.0	128 MB RAM/80 MB DD
Microsoft Visual Basic 6.0	64 MB RAM/100 MB DD
Crystal Reports 8.0	128 MB RAM/350 MB DD

NT 4.0 SP 6.0 WS	32 MB RAM/115 MB I D
Microsoft Access 2013	128 MB RAM/200 MB DD
Microsoft Office 2003	128 MB RAM/300 MB DD
Librería MSDN	64 MB RAM/200 MB I D
Promedio Más Alto/ Total	256 MB RAM/1.75 TB DD

Tabla 4.2.3.2.1. Requerimiento Mínimos.

En base a la tabla 4.2.3.2.1 se observan las características deseables para la instalación de dichos programas.

4.2.4. Codificación.

Hemos llegado a uno de los puntos más importantes del proyecto, en donde plasmaremos a través del código todo el análisis y desarrollo de nuestro sistema desarrollado hasta este punto, presentaré varios ejemplos de Pseudocódigo con una de las actividades más representativas, con el fin de ejemplificar el desarrollo del mismo.

Basado, en nuestro diagrama de clases, obtuvimos la siguiente tabla 4.2.4.1 que contienen las clases y métodos utilizados en el proyecto.

Clase	Métodos	Descripción	Eventos
Autorización	Mtd_busca_autoriza	Select de T Autorizacion todo	txt_n_numRemesa_KeyPress
Autorización	Mtd_busca_autorizacion	Select de T Autorizacion y lo compara con la remesa	txt_n_numRemesa_KeyPress
Autorización	Mtd_busca_autorizacion1	Select de T Autorizacion y lo compara con todas las Variables de entradas	Function Checa()
Autorización	Mtd_busca_nivelautoriza2	Select de T Autorizacion y lo ordena por Remesa	remesa_txt_KeyPress
Autorización	Mtd_busca_nivelpermiso	Select de T Permisos	mtd_llena_cboautorperm()
Autorización	Mtd_deletereg	Delete de T Autorizacion donde coincide con la remesa	Borrar_Click()
Autorización	Mtd_editreg	Update de T Autorizacion en Password, NombreUsuario, NivelAutoriza	Public Sub Actualiza()
Autorización	Mtd_newreg	Insert en T Autorizacion	Cmd_Guardar_Click()
Buscador	Mtd_busca_hist	Select de T Histoperdiarias	Soportar_tabla()
Buscador	Mtd_busca_operaciones	Select de T Operdiarias	Soportar_tabla()
Conecta	Conecta	Se crea una instancia de la conexión	
Conecta	Connstring	Crea la cadena para la conexión	
Equipo	Mtd_delete_equi	Delete de T Equipeso	Abrir_Click()

SISTEMA DE VALIDACIÓN Y REGISTRO DE OPERACIONES DE LAS TORRES DE CONTROL

	peso		
Equipo	Mtd_import_equipo	Insert en T Equipos	Abrir_Click()
Equipo	Mtd_Insert_Terroresequipo	Insert en T TerroresEquipo	Abrir_Click()
Error	Mtd_delete_err	Delete de T TerroresMatriculas	Abrir_Click()
Error	Mtd_delete_err_equipo	Delete de T TerroresEquipo	Abrir_Click()
Error	Mtd_reg_usuario_err	Select de T TerroresUsuario y ordena por reg_error	Datos_err()
Error	Mtd_Sel_Terroresequipo	Select de T Terroresequipo y ordena por reg_error	Datos_err()
Error	Mtd_sel_TerroresMatriculas	Select de T Terroresmatriculas y ordena por reg_error	Datos_err()
Estaciones	mtd_delete_Terrores_Estiata	Delete en T TerroresEstiata	Abrir_Click()
Estaciones	Mtd_LeftJoinEstfacs	Leftjoin de Estfacs y EstiataOaci	cbo_estfacs_Key Up, Refresca()
Estaciones	Mtd_update_estoaci	Update de T EstiataOaci	Actualiza ()
Login	Mtd_login	Select de T Autorizacion y lo compara con la remesa y el password	cmdOK_Click()
Matriculas	Mtd_delete_matriculas	Delete de T Matriculas	Abrir_Click()
Matriculas	Mtd_import_matricula	Insert en T Matricula	Abrir_Click()
Matriculas	Mtd_Insert_Terroresmatriculas	Insert en T TerroresMatricula	Abrir_Click()
Operaciones Diarias	Mtd_edit_histoper	Update Hisoperdiarias de T Operdiarias donde marca Ok=01	Soportar_tabla_Click()
Operaciones Diarias	Mtd_insert_histoper	Insert en T HisOperdiarias de un Select de Operdiarias	Soportar_tabla_Click()
Remesa	Mtd_busca_laremesa	Select de T Autorizacion y lo compara con la remesa	Cmd_nuevo_click
Remesa	Mtd_busca_nivelautoriza1	Select de T Autorizacion y lo ordena por Remesa	remesa_txt_Key Press
Remesa	Mtd_busca_remesa1	Select de T Remesas y lo compara con la remesa	remesa_txt_Key Press
Remesa	Mtd_busca_remesas	Select de T Remesas y lo ordena por Remesa y NumUsuario	mtd_llena_campos_remesa()
Remesa	Mtd_delete_remesa	Delete de T Remesas donde coincide con la remesa	Borrar_Click()
Remesa	Mtd_delete_usuario_remesa	Delete de T Remesas donde borra todos los registros de un Usuario	Borrar_Click()
Remesa	Mtd_edit_remesa	Update de T Remesas en Siglas, Remesas y NumUsuario	Actualiza ()
Remesa	Mtd_find_remesa	Select de T Autorizacion donde coincide con la remesa y el usuario dados	cmd_nuevo () "guardar"

Remesa	Mtd_newreg_remesa	Insert en T Remesas si no existe pregunta por insertar en T Autorizacion	inserta()
Usuario	Mtd_busca_elusuario	Select de T Usuarios y lo compara con el Num. de Usuario	Cmd_nuevo_click()
Usuario	Mtd_busca_numusuario	Select de T Remesas y lo compara con el Num. de Usuario	Function Usuario()
Usuario	Mtd_busca_numusuario2	Select de T Remesas y lo ordena por NumUsuario	Function_Llenita()
Usuario	Mtd_delete_err_usuarios	Delete de T TErroresUsuarios	Abrir_Click()
Usuario	Mtd_delete_usuarios	Delete de T Usuarios	Abrir_Click()
Usuario	Mtd_import_usuarios	Insert en T Usuarios	Abrir_Click()
Usuario	Mtd_reg_error_usuarios	Insert en T TErroresUsuarios	Abrir_Click()

Tabla 4.2.4.1. Tabla de Métodos y Clases.

Como se puede observar, cada clase tiene varios métodos que a su vez contienen eventos que efectúan las acciones pertinentes de acuerdo a lo planteado.

De ahí desglosamos algunas de las actividades de desarrollo para su codificación y presentamos el Pseudocódigo empleado para dichas operaciones.

4.2.4.2. Pseudocódigo de la Clase Usuario.

INICIO

Crear un método que se llame mtd_usuaFec(byval usuario as string, byval fechaoper as date) as integer.

Utilizar la clase Usuario.

Usuarios, Usuario_TipoCliente.

Seleccionar el registro donde sea igual al usuario y a la fecha se encuentre entre fecha de inicio y termino, tipo de cliente sea "for".

Si no existe el registro entonces regresa 0.

Si Existe el registro entonces regresa 1

En el formulario frmimportaroperaciones, frmimportaroperaciones40 la misma modificación al método de learchivo.

Antes de donde pregunta lo de exonerado.

Realizar la pregunta que si ls_ok_for = "S" entonces realizar lo siguiente.

Verificar que si la variable que regreso el método es 1

Entonces poner la variable ls_ok_for = "S" en caso contrario poner ls_ok_for = "N"

FIN

4.2.4.3. Pseudocódigo de la Clase Operaciones Diarias.

Actividad para restaurar información a operdiarias.

INICIO

Temporaloperdiarias → Operdiarias.

De la tabla tbl_usuarios

Leer la tabla tbl_usuarios solo aquellos que campo de estado sean igual 0

Con el dato de usuario, mes, año buscar en la tabla de temporaloperdiarias las operaciones que se encuentren de ese usuario con esos parámetros, la comparación mes y año va ser con el campo de fecha de operación, otra condición será que el campo de temporaloperdiarias.edoresta = 0

Graba las operaciones seleccionadas en la tabla operdiarias

Inserta en operaciones (Todos los registros).

Cambia:

En el campo de remesa se va a poner el valor que trae ese usuario en la tabla de remesa.

Actualizar:

Temporaloperdiarias.fecharesta = now, el registro que esta insertando en operaciones.

Temporaloperdiarias.edoresta =1

Trabaja con transacciones este cambio para todos los registros.

Después de insertar todos los movimientos en operaciones si sale correctamente la operación, actualizar en la tabla de tbl_usua_Resta.fecharesta =now

Tbl_usua_rest.edoresta = 1

Si la inserción o actualización en bloque fallo entonces realizar un rollback.

Campos a insertar y actualizar:

Temporaloperdiarias	Operdiarias	
FechaServicio	FechaServicio	
	numerado	X
Consecutivo	Consecutivo	
Tip_LS	Tip_LS	
NumUsuario	NumUsuario	
NumVuelo	NumVuelo	
Origen	Origen	
Destino	Destino	
Cve_Equipo	Cve_Equipo	
CveMatricula	CveMatricula	
HhMm_LS	HhMm_LS	
FechaModif	FechaModif	
FechaCarga	FechaCarga	Now
errores	Errores	
MarcaOk	MarcaOk	
Remesa	Remesa	2
Observaciones	Observaciones	
Exonerado		
EstIata	EstIata	
Siglas	Siglas	
Tipocia	fechatransf	
FechaResta	FechaResta	Now

EdoResta		Y
NumUsuario		
Mesote		
AñoOpe		
EdoResta		Y
FechaResta		now

X=max(numerado) donde temporaloperdiarias.fechaservicio = operdiarias.fechaservicio y temporaloperdiarias.estiat = operdiarias.estiat

Y = traer el valor de la tabla de remesas que tiene ese usuario

Now = fecha en que se esta ingresando un nuevo registro o actualizando

Edorest = cambiar el valor que tenga por 1

Valida la estación origen y destino, equipo, matricula para lo cual se puede utilizar el método mtd_llena_cadena_errores_oper.

Trae número de remesa:

A que remesa se va ingresar la información para lo cual puedes usar el método li_numremesa = cls_ins_facsicotwr1.MTD_BUSCA_REMESA(Numusuario).

Para grabar las operaciones llamar al método.

mtd_seloper_graba(ByVal li_gpo As Integer, ByVal ls_tipo_ls As String, ByVal ls_formula As String, ls_cadena2 As String) As Integer y si regresa 1 entonces grabar .

FIN

4.2.4.4. Pseudocódigo de la Clase Error.

Modifica el formulario de correcciones.

INICIO

Agregar una nueva opción en el combo de seleccione campo a modificar que diga matrícula-equipo.

En la parte de Cambios de datos: Antiguo, Nuevo, Mostrar todas las matriculas y Mostrar todos los equipos del usuario

Sólo actualizar el nuevo equipo donde la matricula y usuario sea igual a lo seleccionado en los diferentes combos.

FIN

Nota: En cuanto a la seguridad podrán trabajarlo todos los usuarios del sistema.

4.2.4.5. Pseudocódigo de la Clase Reportes.

INICIA

Objetivo: Que maneje dos opciones de tablas.

Temporaloperdiarias y Operdiarias

Maneje opciones de usuarios

0 = todos los usuarios de esa remesa.

Número de usuario en específico.

Usuario origen destino totaloperaciones.

Total de operaciones por usuario y salto de hoja.

FIN

4.2.5. Diseño de Pruebas.

El siguiente paso en el diseño del sistema es el diseño de pruebas, para ello se desarrollará un programa de pruebas para probar que no existen errores en el programa, es decir hacer que el programa falle.

La confiabilidad es un aspecto del diseño, por lo que debe estar dentro del sistema, más adelante se indican las estrategias específicas de pruebas.

La prueba del equipo lógico es el método más usado para determinar si éste funciona como debe. El proceso de pruebas es uno de los componentes de un conjunto de actividades que permiten asegurar la calidad del producto (equipo lógico).

Uno de los principios básicos en la realización de pruebas es que éstas han de ser llevadas a cabo por personas distintas a los diseñadores de los programas, tanto para evitar una simple verificación de que el programa funcione correctamente, como para probar que ese programa ha sido concebido e interpretado correctamente.

Los casos de prueba deben ser escritos tanto para condiciones de entradas inválidas o inesperadas, como para condiciones válidas y esperadas. Deducido de la experiencia y observación de pruebas de diferentes programas, es que la probabilidad de encontrar errores adicionales en una sección del programa, es proporcional al número de errores ya encontradas en la misma sección.

4.2.5.1. Tipos de Pruebas.

Pruebas Unitarias: Se aplica a cada uno de los componentes del sistema que se desarrollen individualmente para comprobar su correcto funcionamiento.

Pruebas de integración: Se prueba la integración entre los componentes del sistema para demostrar que se puede incluir correctamente en el sistema.

Pruebas de sistemas: Se prueba el sistema globalmente.

Pruebas del tipo CAJA BLANCA, que permite examinar la estructura interna del programa.

Pruebas del tipo CAJA NEGRA, donde los casos de prueba se diseñan considerando exclusivamente las entradas y salidas del sistema, sin preocuparse por la estructura interna del mismo.

4.2.5.2. Pruebas Unitarias de Integración.

Las pruebas unitarias se realizan al crear cada módulo o componente individual de un sistema. Las pruebas de integración son realizadas sobre estos componentes individuales son llamados cuando es necesario y que los datos que se transmiten entre dichos componentes son los requeridos.

4.2.5.3. Desarrollo de Integración Incremental.

Al probar cada módulo individualmente es necesario crear módulos auxiliares que simulen las acciones de los módulos indicados por el módulo que se está probando.

Así mismo se han de crear módulos conductores para establecer las precondiciones necesarias, al módulo objeto de la prueba y examinar los resultados de la prueba. Debido a todo esto, a menudo se combinan ambos tipos de prueba unitarias y de integración.

El tipo de prueba incremental consiste en agregar cada módulo o componente individual al conjunto de componentes existentes y el conjunto resultante de prueba. Cuando las pruebas unitarias y de integración se realizan separadamente, es difícil examinar los componentes individuales o módulos que causan resultados incorrectos, por el contrario, con el tipo de prueba incremental es probable que surjan al incorporar un nuevo componente a un grupo previamente probado, sean debidos precisamente a éste último o a las interfaces entre él y los otros componentes.

4.2.5.4. Estrategias de Integración.

Es importante determinar la secuencia en que van a producir e integrar los componentes. Para una estructura de programas organizada jerárquicamente, se podrá enfocar el problema de la prueba utilizando varias estrategias diferentes:

- Estrategias de arriba a abajo (Top-Down)
- Estrategias de abajo a arriba (Bottom-Up)
- Estrategias combinadas
- Comparación de estrategias.

4.2.5.5. Pruebas Globales.

Una vez que se han probado los componentes individuales y se han integrado, se ha de probar el sistema global. En ésta etapa pueden distinguirse los siguientes tipos de prueba, cada uno con un objetivo claramente definido:

- Pruebas funcionales
- Pruebas de comunicación
- Pruebas de rendimiento
- Pruebas de volumen
- Pruebas de sobrecarga
- Pruebas de disponibilidad de datos
- Pruebas de facilidad de uso
- Pruebas de operación
- Pruebas de entorno
- Pruebas de seguridad

Las pruebas de aceptación, son aquellas que realiza el usuario con el objeto de comprobar si el sistema es aceptable para él. Estas pruebas son del mismo tipo que las mencionadas anteriormente, pero son determinadas por el usuario, en lugar de serlo por el equipo de desarrollo.

4.2.5.6. Planificación de las Pruebas.

La fase de pruebas, por su envergadura e importancia necesita una organización seria y fiable.

Ante una fase de pruebas, se debe tomar como axioma que se van a encontrar errores.

Los componentes de una planificación serán:

- Objetivos, definir los objetivos de cada fase de las pruebas.
- Criterios de terminación, Especificar cuándo se deben acabar las pruebas.

- Cronología, fijar los tiempos necesarios para cada fase (diseño, escritura, ejecución).
- Especificar a los responsables de cada fase, así como quién corregir los errores detectados.

Con estos puntos estaremos próximos a depurar el sistema en cuestión. Utilizando las herramientas enunciadas en tiempos y formas especificadas. Se debe hacer una distinción técnica entre las pruebas del software completo (pruebas de verificación de requerimientos) y las pruebas del sistema que incorpora el software (pruebas del sistema o pruebas de validación del sistema). El objetivo de las primeras es buscar y encontrar discrepancias entre los requerimientos y la ejecución del software.

El proceso de verificación de los requerimientos comienza con el análisis de esos requerimientos y una inspección en la cual se busca evaluar la consistencia, conclusión y factibilidad de los requerimientos, tanto individualmente como en conjunto. Es conveniente especificar claramente qué se va a hacer para determinar que el sistema satisface sus requerimientos.

Estas especificaciones son cruciales a la hora de diseñar las pruebas de verificación. El diseño de estas pruebas requiere los siguientes pasos:

- Revisar la capacidad de verificación del requerimiento.
- Especificar el criterio de verificación.
- Hacer visibles las propiedades o elementos del software necesarios para verificar el cumplimiento del requerimiento.
- Controlar los elementos del software necesarios para llevar a cabo las pruebas.
- Elaborar el plan de pruebas.
- Ejecutar el plan de pruebas y reportar sus resultados.

4.2.5.7. Los Procesos del Desarrollo como Fuente de Pruebas.

Los procesos o diagramas de desarrollo son una rica fuente de pruebas para el software desarrollado. En particular, podemos crear pruebas de verificación de los casos de uso, pruebas que busquen errores en el manejo de las restricciones de integridad de las clases del sistema, pruebas de verificación de contratos y pruebas de verificación de diagramas de colaboración.

4.2.5.8. Plan de pruebas.

El estándar IEEE 829-1983³⁰ (Tabla 4.2.5.8.1) describe los tipos de documentos que pueden producirse durante el proceso de prueba. Puede resultar interesante comparar nuestra propuesta con el estándar.

³⁰ ANSI/IEEE Standard 829-1983 for Software Test Documentation. IEEE Press, 1983. Este estándar se incluye en la colección *Software Engineering Standards* publicado por IEEE Press.

Proyecto SIVAR	IEEE 829-1983
Plan de pruebas de un requerimiento	Plan de prueba
(opcional) Tabla de decisiones asociada al requerimiento	
Análisis de verificabilidad del requerimiento	
Criterio de verificación	
Requisitos de observabilidad	
Requisitos de controlabilidad	
Pistas	
Catálogo de modelos de defectos	
Requerimientos de prueba	Especificación de los requerimientos para el diseño de los casos de prueba
Suite de prueba	
Caso de prueba	Caso de prueba
	Descripción del procedimiento de prueba
	Descripción del elemento a probar
Ambiente de prueba	
Bitácora de pruebas	Bitácora de pruebas
	Reporte de incidentes de prueba
Análisis de resultados y acciones recomendadas	
Resumen gerencial del proceso	Resumen de pruebas

Tabla 4.2.5.8.1. Estándar IEEE 829-1983

El propósito del plan de pruebas es explicar el alcance, enfoque, recursos requeridos, calendario, responsables y manejo de riesgos de un proceso de pruebas.

Puede haber un plan global que explique el énfasis a realizar sobre los distintos tipos de pruebas (verificación e integración).

Un plan de pruebas incluye:

Identificador del plan. Preferiblemente de alguna forma mnemónica que permita relacionarlo con su alcance, por ej. TP-Global (plan global del proceso de pruebas), TP-Req-Seguridad1 (plan de verificación del requerimiento 1 de seguridad), TP-Contr-X (plan de verificación del contrato asociado al evento de sistema X), TP-Unit-Despachador_iniciar (plan de prueba unitario para el método iniciar de la clase Despachador). Como todo proceso de desarrollo, está sujeto a control de configuración, por lo que debe distinguirse adicionalmente la versión y fecha del plan.

Alcance. Indica el tipo de prueba y las propiedades/elementos del software a ser probado.

Elementos a probar. Indica la configuración a probar y las condiciones mínimas que debe cumplir para comenzar a aplicarle el plan. Por un lado, es difícil y riesgoso probar una configuración que aún reporta fallas; por otro lado, si esperamos a que todos los módulos estén perfectos, puede que detectemos fallas graves demasiado tarde.

Estrategia. Describe la técnica, patrón y/o herramientas a utilizarse en el diseño de los casos de prueba. Por ejemplo, en el caso de pruebas unitarias de un procedimiento, ésta sección podría indicar: "Se aplicará la estrategia caja-negra de fronteras de la precondition" o "Ejercicio de los caminos ciclomáticos válidos". En lo posible la estrategia debe precisar el número mínimo de casos de prueba a diseñar, por ej. 100% de las fronteras, 60% de los caminos ciclomáticos. La estrategia también hace explícito el grado de automatización que se exigirá, tanto para la generación de casos de prueba como para su ejecución.

Categorización de la configuración. Explica las condiciones bajo las cuales, el plan debe ser: Suspendido, repetido y/o culminado. En algunas circunstancias el proceso de prueba debe suspenderse en vista de los defectos o fallas que se han detectado. Al corregirse los defectos, el proceso de prueba previsto por el plan puede continuar, pero debe explicarse a partir de qué punto, ya que puede ser necesario repetir algunas pruebas. Los criterios de culminación pueden ser tan simples como aprobar el número mínimo de casos de prueba diseñados o tan complejos como tomar en cuenta no sólo el número mínimo, sino también el tiempo previsto para las pruebas y la tasa de detección de fallas.

Tangibles. Explica los documentos a entregarse al culminar el proceso previsto por el plan p. ej. Subplanes, especificación de pruebas, casos de prueba, resumen gerencial del proceso y bitácora de pruebas.

Procedimientos especiales. Identifica el diagrama de las tareas necesarias para preparar y ejecutar las pruebas, así como cualquier habilidad especial que se requiere.

Recursos. Especifica las propiedades necesarias y deseables del ambiente de pruebas, incluyendo las características del hardware, el software de sistemas (p. ej. el sistema de operación), cualquier otro software necesario para llevar a cabo las pruebas, así como la colocación específica del software a probar (p. ej. qué módulos se colocan en qué máquinas de una red local) y la configuración del software de apoyo. La sección incluye un estimado de los recursos humanos necesarios para el proceso. También se indica cualquier requerimiento especial del proceso: actualización de licencias, espacio de oficina, tiempo en la máquina de producción, seguridad, etc.

Calendario. Esta sección describe los hitos del proceso de prueba y el diagrama de dependencia en el tiempo de las tareas a realizar.

Manejo de riesgos. Explica los riesgos del plan, las acciones mitigantes y de contingencia.

Responsables. Especifica quién es el responsable de cada una de las tareas previstas en el plan.

4.2.5.9. Modelo de Casos de Uso Como Fuente de Pruebas.

Cada caso de uso describe una tarea que debe llevar a cabo un actor; por lo que pueden generarse casos de prueba para validar que las tareas se llevan a cabo según lo indica el caso de uso. Adicionalmente el caso de uso describe un flujo de control que describe qué pasos pueden repetirse, cuáles son opcionales, qué cursos alternos hay, flujos que deben probarse:

- Casos correspondientes al curso normal del caso.
- Casos correspondientes a cursos excepcionales.
- Casos que surgen de requerimientos específicos a un elemento del caso de uso.
- Casos asociados a la prueba de características descritas en documentos asociados al caso de uso.

4.2.5.10. Modelo de Clases Como Fuente de Pruebas.

Los defectos a los que está propensa la implementación de un modelo de clases, incluyen:

- Errores en la multiplicidad de la asociación.
- Falta o sobra una asociación.
- Actualización anómala, típicamente en la actualización de información replicada.
- Eliminación anómala, como por ejemplo:
- En la eliminación de un objeto que ayuda a definir una entidad débil.
- En la eliminación de un objeto que pertenece a una asociación 1 a N (por el lado del 1).
- En el manejo de nulificaciones (nullifies), cascadas y restricciones;
- Asociaciones incorrectas entre objetos.
- Inserciones incorrectas.

En general, no se satisfacen las restricciones de integridad.

4.2.5.11. Diagramas de Colaboración Como Fuente de Pruebas.

La implementación de un evento de sistema debe cumplir con su diagrama de colaboración. Para revisar si efectivamente cumple, debería ejecutarse la implementación, revisando que los mensajes enviados corresponden a lo indicado por el diagrama de colaboración. Idealmente se deberían determinar las secuencias de mensajes esperados y luego comparar estas secuencias contra las trazas o secuencias generadas por la implementación. Cabe mencionar que hay herramientas que generan estas trazas, indicando no sólo el mensaje sino el objeto que envía y el objeto que recibe.

Generar manualmente las trazas esperadas a partir de los diagramas de colaboración, no necesariamente es factible. El problema radica en que los diagramas de colaboración no especifican cómo cambia el estado de un objeto al recibir un mensaje *m*. Por ende, si ese objeto envía mensajes *m1*, *m2* como parte de la reacción a *m*, pero sólo si se satisface una guardición que depende del estado del objeto, es muy probable que sea imposible determinar el estado: en consecuencia no es posible determinar si *m1*, *m2* deben agregarse a la traza esperada. Este problema sólo se resuelve si se realiza alguna de las siguientes acciones:

Se especifican los cambios al estado ocurrido como reacción a la recepción de un mensaje y previo a cada envío de un mensaje de reacción. Adicionalmente hace falta especificar los parámetros y respuestas asociadas a los mensajes, ya que las guardiciones pueden depender de sus valores. Se puedan ejecutar los diagramas de colaboración a la par que la implementación, tomando valores de la implementación cuando sean necesarios.

Hasta ahora se ha sugerido generar casos de prueba a partir de la estructura de los parámetros del evento del sistema, es decir la generación de los casos utilizan una estrategia *caja negra* respecto al diagrama de colaboración. Sin embargo, el diagrama de colaboración puede verse como un diagrama de flujo de alto nivel, donde las guardiciones corresponden a puntos de decisión. Por ende podemos generar casos de prueba *caja blanca* para lograr una cierta *cobertura* del diagrama asociado al diagrama de colaboración.

4.2.5.12. Plan de pruebas para el Sistema de Validación y Registro de Operaciones de las Torres de Control, (SIVAR).

Se tienen identificados más de 20 casos de uso, de los cuales se utilizarán 6 para ejemplificar la ejecución del plan de pruebas utilizando los procesos de UML como fuentes de prueba (diagramas y tablas de casos de uso, diagramas de estados y diagramas de clases. La tabla 4.2.5.13.1 muestra

el documento de plan de pruebas para todo el sistema con el mnemónico TP_SIVAR_v.0.0.³¹

Proyecto SIVAR	Plan de pruebas – TP_SIVAR_v.0.0
Alcance	El plan de pruebas debe verificar que el modelo de análisis y los procesos UML que de él se desprenden para el Sistema de Validación y Registro de Operaciones de las Torres de Control cumplen con los requerimientos de usuario.
Elementos probar	a (3 Diagramas de casos de uso, 3 diagramas de estados y 1 diagramas de clases)
Estrategia	Se utilizará la estrategia de caja negra para los casos de uso y diagramas de estados. Para el diagrama de clases y los contratos la estrategia será la prueba de caja blanca.
Tangibles	Debe entregarse un documento con el resultado de las pruebas dentro de tablas que contengan información concierne a cada caso de prueba, con requerimientos, elementos probados, datos E/S, defectos encontrados, posible solución, nivel de tolerancia para el fallo, prioridad, fecha y hora de prueba, responsables, comentarios, etc.
Procedimientos especiales	Las pruebas deben basarse en los diagramas realizados en Visio por lo que el usuario que tenga la función de tester (verificador) debe tener conocimiento de la metodología RUP, así como de la nomenclatura de Visio.
Recursos	Se requiere un equipo de tres personas como para realizar las pruebas de la aplicación.
Calendario	Se muestra el diagrama de Gantt que ilustra el plan general de desarrollo de este sistema y que está en el punto 3.2.3.1 de este trabajo.
Responsables	Verónica Salazar, Raúl Ortiz y Francisco Herrera.

Tabla 4.2.5.12.1. Plan de pruebas para nuestro sistema.

Las figuras 4.2.5.12.2 y 4.2.5.12.3 muestran respectivamente los diagramas de casos de uso y de estados que el tester³² debe tomar en cuenta para las pruebas a realizar.

³¹ TP_SIVAR_v.0.0 – Test Plan Sistema de Validación y Registro de Operaciones en las Torres de Control. Versión 0.0

³² Tester. Persona que ejecuta las pruebas.

DIAGRAMA DE CASOS DE USO PARA USUARIOS

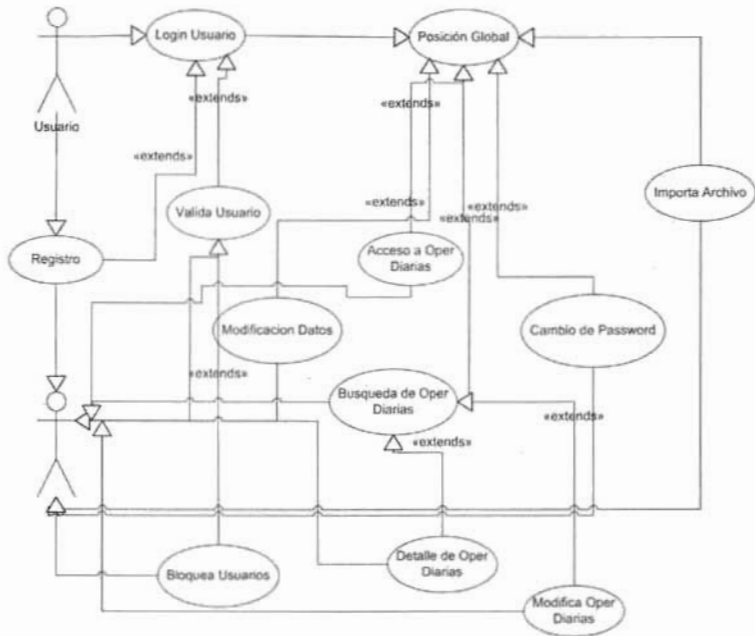


Fig. 4.4.1 Diagrama de casos de uso (usuario) desprendido del modelo del análisis previo.

DIAGRAMA DE ESTADOS PARA USUARIO EXISTENTE Y PARA NUEVOS

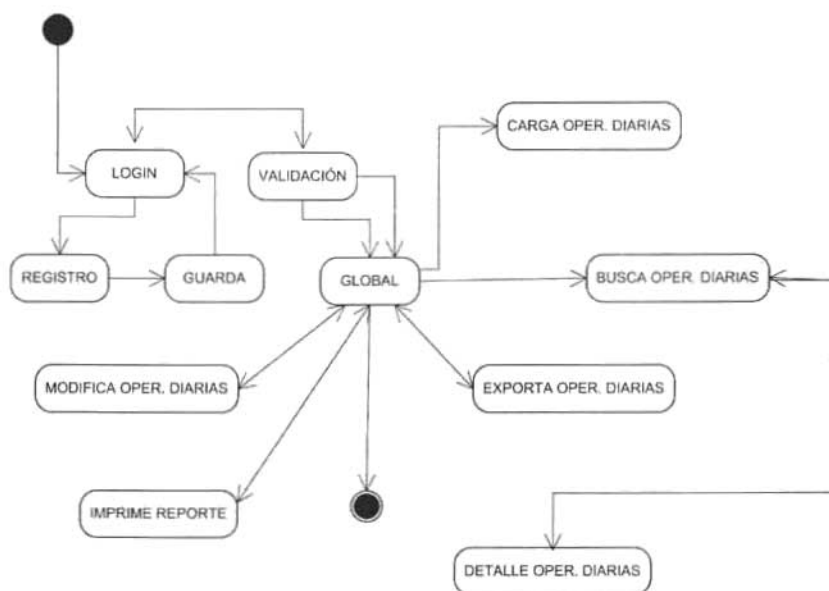


Diagrama 4.2.5.12.2 Diagrama de estados desprendido del modelo de análisis para el SIVAR.

La tabla 4.2.5.12.2 muestra el detalle para cinco casos de prueba de acceso al sistema que engloban el mismo número de casos de uso: Login, Registro, Validación, Bloqueo y Posición Global del usuario independientemente de su nivel.

Sistema	Caso de prueba TP_Acc_Usuario
Requerimiento(s)	<p>El acceso a la aplicación debe ser controlado por un nombre de usuario y contraseña.</p> <p>El nombre del usuario y su contraseña deben ser únicos. No puede haber dos usuarios del sistema compartiendo cualquiera de estos dos valores. Siempre que se genere un nombre de usuario o contraseña nuevos, deberá validarse que no existan en la base de datos. De existir tales valores, se solicitará al usuario que defina un nombre de usuario o contraseña diferentes.</p> <p>Los mensajes de error al no poder ingresar al sistema no deben especificar en qué se equivocó el usuario, deben ser genéricos. Ej. "Datos erróneos. verifique su nombre de</p>

	<p>usuario y/o contraseña e intente nuevamente”.</p> <p>Si el usuario fracasa en su intento por ingresar a la aplicación más de tres veces, es decir, proporciona una combinación de nombre de usuario y/o contraseña inválidos, su cuenta se deberá desactivar, debiendo notificar esta situación al administrador para que dicha cuenta pueda ser reactivada. El valor de intentos fallidos debe ser paramétrico.</p> <p>La contraseña debe poder ser modificada únicamente por el usuario dueño de la misma. En caso de olvido o pérdida, se deberá notificar al administrador para que éste asigne una nueva contraseña temporal, que deberá ser modificada por el usuario en cuestión a la brevedad posible.</p> <p>La aplicación debe mostrar en su página principal, dos opciones: La primera, para que los usuarios existentes puedan ingresar a su cuenta, y los usuarios que no estén registrados en el sistema, puedan hacerlo en línea utilizando la sección “Registro de usuarios nuevos”.</p> <p>Una vez que el usuario ha sido registrado podrá ingresar al sistema.</p> <p>La posición global de la aplicación debe mostrar el contenido de la misma de acuerdo al nivel del usuario y, además, debe ser capaz de reconocer al usuario que se ha conectado.</p>
Caso(s) de uso	LoginUsuario, RegistroUsuario, ValidaUsuario, BloqueoUsuarios, PosiciónGlobal (DCU_Usuario)
Diagrama de estados	Edos_Usuario
Datos de entrada	<p>Login Usuario: Nombre de usuario y contraseña</p> <p>Registro Usuario: Datos personales, datos académicos, experiencia laboral, nombre de usuario y contraseña</p> <p>Valida Usuario: Nombre de usuario y contraseña (proporcionados por el CU Login Usuario – transparentes al usuario)</p> <p>Bloqueo Usuario: Número de intentos fallidos (proporcionados por el CU ValidaUsuario – transparente al usuario)</p> <p>Posición Global: Nombre del usuario y nivel de acceso (proporcionados por el CU Valida Usuario – transparentes al usuario)</p>
Datos de salida	<p>Login Usuario: Si el nombre de usuario y contraseña son válidos no habrá datos de salida y se enviará al usuario a su pantalla principal (Posición Global). Por el contrario, si éstos son inválidos, debe visualizarse un mensaje que diga: “Datos erróneos. verifique su nombre de usuario v/o</p>

	<p><i>contraseña e intente nuevamente</i>". Si alguno(s) de los campos no lleva(n) información, debe observarse una validación de tipo Alert indicando esta situación. Si alguno(s) de los campos lleva caracteres inválidos debe observarse una validación de tipo Alert.</p> <p>Registro usuario: Si los datos ingresados por el usuario son correctos se debe observar el mensaje de confirmación: <i>"El proceso de registro ha concluido con éxito. Utilice su nombre de usuario y contraseña para ingresar al sistema"</i>. Si alguno(s) de los campos no lleva(n) información, debe observarse una validación de tipo Alert indicando esta situación. Si alguno(s) de los campos lleva caracteres inválidos debe observarse una validación de tipo Alert.</p> <p>Valida Usuario: No tiene datos de salida visibles al usuario. Si la validación es exitosa, se remite al caso de prueba Posición Global. En caso contrario regresa al caso de prueba Login Usuario e incrementa el contador de intentos. Al sobrepasar tres intentos fallidos se debe guardar un 1 en el campo Nivel de la tabla Personas, que será el indicador de bloqueo de cuenta.</p> <p>Bloqueo Usuario: En caso de sobrepasar tres intentos para ingresar, el mensaje de salida debe decir: <i>"Por motivos de seguridad, el Sistema se cerrará. Póngase en contacto con el administrador del sistema"</i>.</p> <p>Posición Global: Debe mostrar un mensaje de bienvenida con el nombre y primer apellido del usuario que ha ingresado exitosamente al sistema. Se deben desplegar las siguientes opciones en el menú principal: Modificación de datos propios, Búsqueda de cursos de operaciones y, como subinciso de éste último Detalle de Operaciones. Debe mostrarse, en caso de haber muestra el conjunto de datos localizados.</p>
Defecto encontrado	Al sobrepasar los tres intentos fallidos en el caso de prueba Login Usuario, la cuenta no se desactiva. Se puede seguir intentando sucesivamente.
Posible solución	Debe leerse el parámetro de intentos fallidos de la base de datos y tomarlo en cuenta para compararlo con el contador del caso de prueba Valida Usuario o debe verificarse si se está insertando correctamente un 1 en la tabla de Personas en el campo Nivel.
Nivel de tolerancia para la falla	Baja. Este caso de prueba es elemental para cumplir con los mecanismos básicos de seguridad y debe resolverse cuanto antes. No se está cumpliendo con el requerimiento de desactivar la cuenta al superar cierto número de intentos por ingresar al sistema.
Prioridad	0 - Cero. Esta incidencia tiene la más alta prioridad
Fecha y hora de prueba	Lunes 09 de junio de 2003 - 16:30 hs
Revisado por	Ángeles Barbosa Morán

Comentarios	<p>Todos los casos de prueba fueron exitosos con excepción del bloqueo de cuenta que es parte del caso de prueba Login Usuario. Se pudo observar que las validaciones correspondientes a VisualBasic dentro de los casos de prueba Login Usuario y Registro Usuario existen y funcionan correctamente. La validación del usuario se realiza de igual forma verificando la existencia de éste en la base de datos. Las opciones desplegadas en la posición global (Pantalla de Inicio) del usuario una vez que se firmó, se validó y logró ingresar son las correctas. Se debe comenzar a trabajar cuanto antes, en la resolución del bloqueo de usuarios debido a que representa una falla de seguridad en el sistema. Un ataque podría llevarse a cabo utilizando un programa que pruebe millones de combinaciones de letras y números tanto en el nombre de usuario como en la contraseña para intentar ingresar en el sistema de forma ilegal. El bloqueo de usuarios es la forma de evitar dicha situación.</p>
-------------	---

Tabla 4.2.5.12.3 Caso de prueba para el Login, registro, validación y posición global del usuario.

La tabla 4.2.5.12.4 muestra el detalle para el caso de prueba de modificación de datos propios de todos los usuarios del sistema.

Sistema	Caso de prueba TP_Mod_Oper_Diarias
Requerimiento(s)	<p>El usuario que originalmente solicitó algún tipo de operación, como pueden ser modificaciones en los datos de alguna operación en particular por ejemplo, recibirá la notificación de la decisión del operador en su propia posición global y la información habrá sido afectada siempre y cuando la decisión haya sido aprobada por el operador. Si éste último toma la decisión de rechazar el cambio, simplemente no se afectarán los datos.</p>
Caso(s) de uso	ModificacionDatosOperDiarias
Diagrama de estados	Edos_Usuario, Edos_Admin, Edos_Operador
Datos de entrada	Cualquier de los 4 campos que incluya los formularios de Operaciones Diarias, Matrícula, Usuario, Equipo, Origen o Destino.
Datos de salida	<p>Deberá visualizarse una pantalla que exhiba los datos con error que deberán modificarse en el(los) registro(s) de la base de datos junto con tres botones que deben tener las etiquetas: Guardar, Cancelar. Si se pulsa el botón Cancelar, el sistema deberá enviar al usuario a la pantalla de edición anterior. Si pulsa Guardar, deberá enviarlo a una pantalla que confirme que la operación fue exitosa, o a una pantalla de error si la operación terminó en alguna excepción controlada. La pantalla de operación exitosa debe desplegar el mensaje: <i>Su operación ha sido actualizada exitosamente.</i></p> <p>NOTA: Para que el cambio surta efecto en la base de datos el operador deberá aprobar la operación solicitada.</p>

Defecto encontrado	Ninguno
Posible solución	N/A
Nivel de tolerancia para la falla	N/A
Pricidad	N/A
Fecha y hora de prueba	Lunes 09 de junio de 2004 - 17:00 hrs.
Revisado por	Carlos De Lille
Comentarios	El caso de prueba no presentó fallas. Se comprobó que las pantallas desplegadas fueron las esperadas y tampoco se encontraron casos en los que el sistema haya caído dentro de una excepción controlada.

Tabla 4.2.5.12.4. Caso de prueba para la modificación de datos propios del usuario.

4.2.6. Implantación.

Una vez concluido el desarrollo del sistema, llegamos al punto crucial de nuestro trabajo. Es aquí donde el sistema ya probado y corregido, debe ser entregado al cliente para su implantación como herramienta de trabajo.

De conformidad con lo previsto en el análisis y la asignación de tareas en el capítulo anterior. Se requiere implementar el sistema en cada una de las computadoras de la Dirección de Ingresos. Para lo cual y siguiendo la solicitud de la Dirección de Ingreso, se llevo a cabo la instalación en las computadoras asignadas para tales propósitos.

4.2.7. Capacitación.

El programa de capacitación consta de dos puntos muy importantes para el aprendizaje efectivo: Teoría y práctica. La parte teórica consiste en presentar una introducción al programa, presentando en forma detallada cada una de las funcionalidades del mismo, así como el procedimiento para llevar a cabo las tareas específicas. Los antecedentes del proyecto y la incorporación de las necesidades y requerimientos de los usuarios al sistema. Es decir, la integración del programa de aquello que es necesario y urgente de implementar.

El área de capacitación consta de una sala de usos múltiples, en la cual se llevaron las computadoras de cada uno de los usuarios. Las cuales ya habían sido cargadas con el programa. La parte teórica tomó fuerza con la puesta en marcha del sistema a la vista de los usuarios. Fue una experiencia enriquecedora el observar como los usuarios participaban activamente en la capacitación. La curva de aprendizaje fue muy corta y al cabo de unas pocas de uso, pudimos resolver todas y cada una de las dudas de nuestros usuarios.

Conclusiones



Conclusiones

Se puede concluir que el proyecto elaborado en este documento ha alcanzado su objetivo satisfactoriamente, facilitando y agilizando el proceso de revisión de los cargos en las operaciones a las aerolíneas; también se ha podido retroalimentar la información con las torres de control con el fin de evitar redundancia y errores en las operaciones que han sido recabados por los controladores de tránsito aéreo al momento de registro de la operación. Ahora se lleva un estricto control de cargos que no afecta a los usuarios, sino que coincide con las líneas aéreas, logrando así quitar de las líneas productivas las sobrecargas por reclamos.

El sistema fue pensado desde su concepción para poder conectarse con otros sistemas u otras bases de datos. En la actualidad se están haciendo pruebas con el sistema de facturación y se ha comportado favorablemente.

Todo ello gracias a los conocimientos adquiridos en la carrera de Ingeniería en Computación. Los cuales me han forjado técnicamente con bases sólidas. Los fundamentos y herramientas proporcionadas por mi querida facultad me han permitido, eso y mucho más. El gran nivel y exigencia la hacen única. Para mi es un orgullo muy grande ser parte de ella, con la frente en alto puedo decir que he alcanzado, luego de mucho. La capacidad para enfrentarme a los retos que se me presenten en mi vida.



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.

Apéndice A: Instalación del Sistema Operativo

A.1. Microsoft Windows NT 4.0 WS, Introducción.

El sistema operativo se suministra en formato CD, por ello nuestra plataforma tiene que tener instalada un lector de CD-ROM compatible con NT (casi todos lo son) o estar conectado a una red que tenga uno compartido, también tendremos tres discos de inicio, si no es así tendríamos que generarlos.

Los discos de inicio pueden generarse desde cualquier PC que tenga CD-ROM, bastaría con introducir el CD de NT, ir al directorio I386 y ejecutar la instrucción WINNT /OX, nos pedirá tres disquetes formateados y vacíos.

El programa WINNT antes de seguir adelante tenemos que hacer un inciso y conocer el comando, de acuerdo a la tabla que a continuación se presenta Tabla. A.1.1.

PARAMETRO	EFECTO
/S[:] nombre de la ruta	Indica la ruta de origen de los archivos de Windows NT; también puede ser una unidad de RED.
/T[:] nombre del directorio	Indica el directorio temporal en el que se guardarán los archivos para la instalación, también puede ser una unidad de RED.
/I [:] nombre del archivo .INF	Indica el nombre del archivo de instalación, por defecto es el archivo DOSNET.INF.
/O	Solo crea los discos de inicio.
/OX	Crea los discos de inicio para la instalación desde CD-ROM.



Universidad Nacional
Autónoma de México



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.

/F	Desactiva la comprobación de los archivos disponibles en los diskettes de inicio.
/C	Pasa por alto los controles de espacio disponibles en los diskettes de inicio.
/B	Para hacer la instalación sin los diskettes de inicio. Mediante /S indicaremos la ruta de los archivos de instalación si no es desde el CD-ROM en el directorio I386.
/V	Permite la instalación automática. Mediante /S indicaremos la ruta de los archivos de instalación si no es desde el CD-ROM en el directorio I386

Tabla A.1.1. Comandos de Instalación.

Para todos los casos de instalación NT 4.0 necesita un volumen de disco local compatible con NT, con al menos 119 Mb de espacio libre, no comprimido y formateado con FAT o NTFS.

Podemos encontrarnos con distintas circunstancias al hacer la instalación. Si tenemos un sistema operativo instalado con acceso a un lector de CD, desde el directorio I386 del CD ejecutamos WINNT/B para que comience la instalación y no se generen los tres discos de inicio.

Para realizar una instalación a través de red necesitamos tener instalado el cliente para red de MS-DOS (viene incluido en el CD de NT Server o descargarse desde el web de Microsoft), Windows 3.11 o Windows 95, nos conectamos aun CD compartido en red o a un disco duro que contenga los archivos de instalación y ejecutamos WINNT/B desde el directorio I386. Si no tenemos nada instalado en el disco duro, introduciremos el CD-ROM en su lector, el **disco 1** en la disquetera y después encenderemos el sistema. Sí estamos instalando NT 4.0 en un equipo que pueda arrancar desde el lector de CDROM podemos cambiar en el BIOS la secuencia de arranque de manera que empiece por el CD, aunque el disco duro esté sin formatear y consecuentemente sin ningún tipo de sistema operativo instalado, el programa de instalación se inicia sólo y te permite hacer la instalación sin tener nada en el disco duro y sin tener que generar los disquetes de instalación.

Una vez iniciada la instalación todas las opciones anteriores siguen prácticamente el mismo esquema.

A.2 Reconocimiento del Sistema.

Lo primero que hace es reconocer el hardware indispensable para empezar a trabajar y comprobar que no exista una versión de NT, en este caso se detendrá la instalación y tendremos que realizarla desde ese sistema NT ya instalado (usando WINNT32) o eliminar la partición donde

estuviera ubicado. A continuación comenzará la carga de los archivos necesarios para la instalación y nos pedirá que introduzcamos el **disco 2** o en el caso de estar haciendo una instalación sin discos pasará a un menú, donde nos pregunta:

Si queremos ayuda sobre la instalación (*F1*)

Si queremos instalar NT (*ENTRAR*)

Si queremos reparar (*R*), este apartado lo veremos en un próximo documento.

Si queremos salir de la instalación (*F3*)

Presionamos "*ENTRAR*". La configuración de unidades de almacenamiento pasará a la fase de detección de los controladores ESDI/IDE, SCSI y unidades de CDROM conectadas, preguntándonos si queremos detectar controladoras SCSI (*ENTRAR*) o no detectarlas (*I*); éste sería el caso si no tuviéramos ningún dispositivo SCSI, en este caso no se detectaría el CDROM IDE, ya que NT lo trata como un dispositivo SCSI. Nosotros pulsaremos "*ENTRAR*" Nos pedirá el **disco 3**, aparece una pantalla con el resultado de la detección. Si no hubiera sido detectado alguno de nuestros discos duros o lectores de CDROM, tendríamos que instalar el controlador del fabricante presionando (*S*). Si los hubiera detectado le daremos ("*ENTRAR*"). Aparecerá en pantalla la licencia del producto la cual debemos de leer atentamente, dando al avance página hasta que nos permita dar a (*F8*). Seguidamente nos dará un listado de componentes instalados en el sistema, los cuales podremos cambiar en caso necesario, creo que sería mejor dejar las cosas como están y luego cambiar lo que queramos desde el sistema operativo ya instalado. Ahora pasaremos al gestor de particiones de disco y de ubicación de la instalación el cual nos pregunta: ¿Dónde queremos instalar NT? .Para ello nos moveremos con el cursor hasta la partición donde queremos instalarlo y luego presionaremos (*ENTRAR*).

Si tenemos espacio sin asignar nos moveremos con el cursor a ese espacio no particionado y pulsando la tecla (*C*) crearemos una nueva partición. Lo más importante es tener un espacio de aproximadamente 300 Mb para la instalación de NT 4.0. Si queremos borrar una partición moveremos el cursor a la partición existente y pulsaremos (*E*). En nuestro caso tenemos una partición FAT³³ con el tamaño necesario para la instalación del NT, por lo que moveremos el cursor hasta situarlo encima de dicha partición y pulsaremos (*ENTRAR*). Ahora nos preguntará si deseamos convertir la partición a NTFS o dejarlo como está, con el cursor nos moveremos a la opción que deseemos. La instalación es más rápida sobre FAT, y así lo haremos nosotros, recordando que cuando terminemos la instalación tendremos que ejecutar CONVERT C:/FS: NTFS para convertir a NTFS, siempre que queramos convertir el sistema de archivo a este tipo.
NTFS nos permite configurar permisos de seguridad sobre archivos y directorios; FAT es más rápido pero no tiene opciones de seguridad.

³³ FAT.- File allocation Table. Tabla de localización de Archivos.

También nos preguntará el directorio donde queremos ubicar el bloque de programas del NT, por defecto "WINNT" y pasará a examinar los discos para comprobar su integridad, para ello pulsaremos (*ENTRAR*); si consideramos que los discos están en perfecto estado pulsaremos (*ESC*). Llegado a este punto el sistema se pondrá a copiar los archivos necesarios para la instalación del sistema NT, cuando acabe este proceso retiraremos el disco de la unidad y del CD-ROM y presionaremos (*ENTRAR*). Con esto concluye la primera parte de la instalación, que sería igual para el NT Server y Workstation. La segunda fase, basada en el entorno gráfico de NT, es distinta para cada versión. Una vez pasada la primera parte de la instalación, se reinicia la computadora y comienza la instalación en entorno gráfico. Comienza el entorno gráfico, nos saldrá una pantalla donde nos indica los pasos que va a seguir la instalación, donde pulsaremos (*siguiente*), y pasaremos a otra donde nos indica el tipo de instalación que vamos a realizar:

Típica: Recomendada para la mayoría de los equipos.

Portátil: Se instalará con opciones útiles para equipos portátiles.

Compacta: Para ahorrar espacio en disco, no se instalará ninguno de los componentes opcionales.

Personalizada: Para usuarios más avanzados. Puede personalizar todas las opciones de instalación disponibles.

Nosotros explicaremos la instalación personalizada y pulsaremos (*siguiente*). En el paso siguiente pondremos el nombre y la organización a la que va a pertenecer la licencia (*siguiente*), e introduciremos la clave del CD de NT, la cual viene en la carátula del CD, (*siguiente*), pasaremos a poner el nombre que va a tener el equipo para su reconocimiento en red, (*siguiente*), y nos preguntará la contraseña del administrador, en caso de olvido tendremos que reinstalar el sistema operativo, (*siguiente*). Nos preguntará si queremos un disco de rescate. El disco de rescate es importante por si existe un bloqueo o un fallo en el arranque de NT, este disco se tendrá que acuatiza cada cierto tiempo, y siempre antes de hacer un cambio en el equipo, sobre todo si es un cambio de hardware. En éste caso la utilidad para generarlo es "RDISK" que será ampliada mas adelante, (*siguiente*), la pantalla que sigue es la selección de componentes en actuaremos eligiendo o descartando los componentes que creamos adecuados, una vez determinado que es lo que queremos instalara le daremos a (*siguiente*).

Configurando el acceso a red. Si el equipo está conectado a una red a través de RDSI (ISDN) o un adaptador de red pulsaremos como activo en el cuadro a tal efecto. Si a su vez va a tener control de acceso telefónico a redes también marcaremos el cuadro a tal efecto. Si nuestro equipo no va a tener nada de lo anterior pulsaremos el botón redondo que nos indica tal opción (No conectar este equipo a una red en este momento) Nuestro equipo esta conectado a una LAN a través de un adaptador de red

Seleccionaremos la búsqueda automática de adaptadores de red, en el caso que no lo detectase tendremos que utilizar el disco del fabricante o un

adaptador existente en la lista de adaptadores de NT. Una vez instalado el adaptador de red, pasaremos a la instalación de los protocolos con los que van a trabajar nuestro sistema, los cuales pueden ser TCP/IP, IPS/SPX, NetBEUI, pudiéndose seleccionar otros desde una lista o instalarlos desde un disco del fabricante. Nosotros instalaremos TCP/IP, si tuviéramos que instalar un cliente para NETWARE tendríamos que instalar IPX/SPX, y si lo queremos compatibilizar con antiguas redes de Microsoft o queremos una instalación sencilla hasta 8 equipos utilizaremos NetBEUI solamente. Nosotros instalaremos TCP/IP ya que será imprescindible para el acceso a Internet a través de acceso telefónico a redes, (*siguiente*).

Pantalla de los servicios de red. Sale un listado con los servicios mínimos de red que no se pueden tocar desde la instalación, en el caso que quisiéramos quitar algunos tendremos que esperar a que se acabe la instalación, podremos ampliarlo dependiendo de los que queramos, en el caso de una red Netware necesitaríamos instalar el servicio de Cliente para Netware.

Ya hemos acabado la instalación de red, le damos (*siguiente*), si tenemos alguna duda (*ATRÁS*). Comenzado la instalación de todo lo que le hemos indicado en los pasos previos. Pasando a la introducción de los datos para el protocolo TCP/IP. Si nuestro equipo adquiere la dirección IP desde un servidor DHCP automáticamente le daremos que (*SI*), en el caso de tener una dirección IP fija le daremos que (*NO*). Ahora seguiremos con la introducción de los datos del TCP/IP de nuestro equipo, si tenemos una dirección fija de red la pondremos una vez activada la casilla a para tal efecto, con la mascara de red adecuada, si no tenemos ningún ROUTER o GATEWAY para la solución de encaminamiento lo dejaremos en blanco, en caso de que existiera pondríamos la dirección de este.

Esta es la instalación mínima. Si utilizamos WINS para la resolución de nombres le indicaremos el servidor de WINS en caso contrario le diremos que (*SI*) queremos continuar. Nos indicará los niveles de enlace entre los servicios, los protocolos y los adaptadores de comunicaciones tanto de red como de RAS los cuales podemos habilitar o deshabilitar a nuestro gusto, (*siguiente*). NT esta listo para iniciar la red (*siguiente*). Ahora llega el momento de decirle si vamos a formar parte de un **dominio NT** o en un grupo de trabajo en el caso de trabajar en un dominio necesitamos la asistencia del administrador para que de alta la máquina. Nosotros en un principio no trabajamos en un dominio sino en un grupo de trabajo. Pondremos el Nombre, (*siguiente*), ya hemos terminado la instalación de la red. Pasaremos a poner la fecha y la hora (*CERRAR*), y las características de la tarjeta gráfica, aunque recomiendo dejar la que nos haya dado por defecto y continuar la instalación, posteriormente podremos cambiar el tipo. Si ha detectado una tarjeta tendremos que pulsar (*PRUEBA*), antes de (*ACEPTAR*). Copiará el resto de los archivos, guardará la configuración y nos pedirá que insertemos un disco que etiquetaremos como "Disco de reparación" (*ACEPTAR*), borrará los archivos temporales y nos pedirá que reiniciemos.



Bibliografía

C. J. Date.

Introducción a los Sistemas de Bases de Datos.

Vol. 1. 7ª Edición.

Prentice Hall, 2001.

Craig Larman.

**ULM Y Patrones: Introducción al Análisis y Diseño
Orientado a Objetos.**

1ª Edición, 1999.

Prentice Hall.

Rebeca M. Riordan.

Designing Relational Database Systems.

1ª Edición, 2000.

Microsoft / Mc Graw Hill.

Roger S. Pressman.

Ingeniería del Software, un enfoque Práctico.

3ª Edición, 1993.

Mc Graw Hill.

SENEAM, 25 Años.

**Historia de los Servicios a la Navegación en el Espacio
Aéreo Mexicano.**

Ing. Agustín Arellano Rodríguez.

1ª Edición, 2003.



Universidad Nacional
Autónoma de México



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.

**Tesis: Sistema de Administración de Recursos Humanos.
Iván José Márquez Larios, Joaquín Martínez Hernández y
Alberto Zygmunt Rodríguez y Niegolewska.**

Facultad de Ingeniería, UNAM.
México, D. F., 2004.

The Handbook of Software Quality Assurance (3rd Edition)

G. Gordon Schulmeyer, James I. Mcmanus.
Prentice Hall, 1999.

Bibliografías en Internet:

www.webcindario.com/somnus1654/clasifica.htm

www.utnc.edu.mx/informatica/EGETSU/1%20Modelo%20Relacional.doc

www.ciberia.ya.com/alexcasan/intsql.htm

www.gris.det.uvigo.es/~jose/doctorado/introduccion/tsld008.htm

www.vermic.com/art_inqr.htm

www.dcc.uchile.cl/~alvherre/charla/inicio.html

www.geocities.com/mustang51r/index.html

www.gris.det.uvigo.es/~jose/doctorado/introduccion/tsld004.htm

www.gris.det.uvigo.es/~jose/doctorado/introduccion/tsld005.htm

www.lisi.usb.ve/reportes/2000-HC-01.pdf

<http://ji.ehu.es/teaching/has/Traspas-html%5C1-CASE%5CApunesCASE.htm>

www.itcm.edu.mx/apuntes/sistemas/ingsoftware/

www.vermic.com/art_inqr.htm