



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Sistema de Información de Aeropuertos**

**TESIS**

**QUE PARA OBTENER EL TÍTULO DE  
INGENIERO EN COMPUTACIÓN  
PRESENTA:**

**Luis Arturo Dávalos Escobedo**

**DIRECTORA DE TESIS:**

**Ing. Liliana Josefina Muñiz Zafra**



**MÉXICO, D.F.**

**2004**



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.

ESTA TESIS NO SALE  
DE LA BIBLIOTECA



# Sistema de Información de Aeropuertos



## INDICE TEMÁTICO

<b>1. DESCRIPCIÓN DEL CASO DE ESTUDIO</b>	<b>7</b>
1.1 AEROPUERTOS Y SERVICIOS AUXILIARES	7
1.2 REQUERIMIENTOS DE INFORMACIÓN DE ASA	8
1.3 RECURSOS Y LIMITACIONES EXISTENTES	10
<b>2. ANTECEDENTES HISTÓRICOS</b>	<b>11</b>
2.1 METODOLOGÍAS EN EL DESARROLLO DE SISTEMAS	11
2.1.1 Antecedentes	11
2.1.2 Sistema de Información	12
2.1.3 Ciclo de vida de un sistema de información	12
2.1.4 Metodologías para el Desarrollo de Sistemas de Información	12
2.1.5 Metodología clásica, en cascada o secuencial	12
2.1.6 Modelo evolutivo	14
2.1.7 Modelo exploratorio	14
2.1.8 Metodología en espiral	15
2.1.9 Herramientas CASE	16
2.1.9.1 Antecedentes	16
2.1.9.2 Evolución	17
2.1.9.3 Elementos de una herramienta CASE	17
2.1.9.3.1 Repositorio de datos	17
2.1.9.3.2 Herramienta generación de prototipo	17
2.1.9.3.3 Generador de Bases de Datos	17
2.1.9.3.4 Generador de código	17
2.1.9.3.5 Generador de documentación	18
2.1.9.3.6 Módulo de administración de proyectos	18
2.1.9.3.7 Bibliotecas de clases de objeto	18
2.1.9.3.8 Herramientas de Servidor Web	18
2.1.9.3.9 Clasificación de las Herramientas CASE	19
2.1.9.10 Posibles inconvenientes en el uso de herramientas CASE	19
2.2 ANÁLISIS, DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE SISTEMAS	19
2.2.1 Análisis de Sistemas	19
2.2.2 Herramientas para el análisis de sistemas	21
2.2.2.1 Entrevistas	21
2.2.2.2 Diagrama Entidad Relación DER	21
2.2.2.3 Diagrama Jerárquico Funcional (DJF)	23
2.2.2.3.1 Función del Sistema	23
2.2.2.3.2 Convenciones para dibujar el diagrama jerárquico funcional	23
2.2.2.4 Diagramas De Flujo De Datos (DFD)	23
2.2.3 Planificación de sistemas	25
2.2.3.1 Alcance (ámbito) del software	25
2.2.3.2 Recursos	26
2.2.3.2.1 Recursos humanos	26
2.2.3.2.2 Recursos de hardware	26
2.2.3.2.3 Recursos de software	26
2.2.3.4 Reusabilidad	26
2.2.4 Estimación del proyecto de software	28
2.2.4.1 Técnicas de descomposición	28
2.2.4.1.1 Estimación del esfuerzo	28
2.2.4.1.2 COCOMO (CONstructive COSt MOdel)	28
2.2.4.1.2.1 Determinación de costos	29
2.2.4.1.3 modelo de estimación de Putnam	29
2.2.4.1.4 Juicio experto	29



## Sistema de Información de Aeropuertos



2.2.5 Cotización	30
2.3 DISEÑO DE SISTEMAS	30
2.3.1 Diseño de la Base De Datos	30
2.3.1 Diseño de la Interfaz de usuario	30
2.3.2.1 Diseño de las formas	31
2.3.2.2 Diseño de pantallas	31
2.3.2.3 Diseño de reportes	32
2.4 ETAPA DE DESARROLLO	32
2.5 IMPLANTACIÓN DEL SISTEMA	32
2.6 MANEJADORES DE BASES DE DATOS	32
2.6.1 Características de un manejador de base de datos relacional	33
2.6.2 Manejadores de bases de datos comerciales	34
2.6.2.1 Oracle	34
2.6.2.1.1 Características	34
2.6.2.2 Sybase	34
2.6.2.2.1 Características	34
2.6.2.3 Informix	34
2.6.2.3.1 Características	34
2.6.2.3.4 Microsoft SQL Server	35
2.6.2.3.4.1 Características	35
2.6.2.3.5 MySQL	36
2.6.2.3.5.1 Características	36
2.6.3.6 PostgreSQL	36
2.6.3.6.1 Características	36
2.7 HERRAMIENTAS DE DESARROLLO	36
2.7.1 Herramientas de desarrollo que existen en el mercado	37
2.7.1.1 Visual Basic	37
2.7.1.1.1 Características	37
2.7.1.2 Microsoft Access	37
2.7.1.2.1 Características	37
2.7.2.3 Power Builder	38
2.7.2.3.1 Características	38
2.7.2.4 Delphi	38
2.7.2.4.1 Características	38
2.8 COMUNICACIONES Y REDES	38
2.8.1 Proceso de comunicación y concepto de red	38
2.8.2 Medios de transmisión	39
2.8.2.1 Medios aéreos	39
2.8.2.1.1 Ondas de radio	39
2.8.2.1.2 Enlaces con microondas	39
2.8.2.1.3 Medios Infrarrojos	39
2.8.2.2 Medios Sólidos	39
2.8.2.2.1 Cable Coaxial	39
2.8.2.2.2 Par trenzado	40
2.8.2.2.3 Fibra óptica	40
2.8.3 Tipos de transmisión	40
2.8.4 Clasificación de las redes	41
2.8.4.1 Clasificación por tecnología de transmisión	41
2.8.4.2 Clasificación por escala	41
2.8.4.2.1 Redes LAN	41
2.8.4.2.1.1 Token Ring	42



## Sistema de Información de Aeropuertos

2.8.4.2.1.2 Estrella	42
2.8.4.2.1.3 Bus	42
2.8.4.2.2 Redes MAN	42
2.8.4.2.3 Redes WAN	42
2.8.5 El modelo OSI	43
2.8.5.1 Capa física	43
2.8.5.2 Capa de enlace	43
2.8.5.3 Capa de red	43
2.8.5.4 Capa de transporte	43
2.8.5.5 Capa de sesión	43
2.8.5.6 Capa de presentación	43
2.8.5.7 Capa de aplicación	44
2.8.6 Protocolos	44
2.8.6.1 Grupos de protocolos	44
2.8.6.2 TCP/IP (Transmission Control Protocol/Internet Protocol)	45
2.8.6.2.1 TCP Transmission Control Protocol	45
2.8.6.1.2 Internet Protocol IP	46
2.8.6.1.3 Dirección IP	46
2.8.6.1.4 Máscara de subred	46
2.8.6.1.6 Servidores Proxy.	46
2.8.6.1.7 DNS	46
2.8.6.1.7.1 Servidores DNS	46
2.8.7 Ethernet	46
2.8.8 Fast Ethernet	47
2.8.9 Gigabit Ethernet	47
2.8.10 Principales dispositivos que conforman una red	47
2.8.10.1 Tarjetas de red	47
2.8.10.2 Estaciones de trabajo	48
2.8.10.3 Servidores	48
2.8.10.4 Repetidores	48
2.8.10.5 Puentes (Bridges)	48
2.8.10.6 Ruteadores (Routers)	48
2.8.10.7 Brouters	48
2.8.10.8 MAU (Multistation Access Unit)	48
2.8.10.9 Concentradores Hubs	48
2.8.10.10 Switching Hub o Switch Ethernet	49
2.9 INTERNET	49
2.9.1 Direcciones Url	49
2.9.2 Servicios de Internet	50
2.10 PLATAFORMA CLIENTE/SERVIDOR	51
2.10.1 El Cliente	51
2.10.2 El Servidor	51
2.10.3 La Red	51
2.10.4 Ventajas de la plataforma cliente servidor	52
2.11 SEGURIDAD	52
2.11.1 Lineamientos de seguridad	52
2.11.1.1 Privacidad de la Información	52
2.11.1.2 Integridad de la Información	53
2.11.1.3 Confiabilidad del personal	53
2.11.2 Clasificación de la información y sistemas de control de acceso	53
2.11.3 Medidas de seguridad	53
2.11.4 Virus	54
2.11.5 CortaFuegos (Firewalls)	54



2.12	DESARROLLO DE APLICACIONES EN INTERNET	55
2.12.1	Lenguaje HTML (Hyper Text Markup Language)	55
2.12.2	Páginas estáticas y páginas dinámicas	55
2.12.3	Clasificación de las páginas dinámicas	55
2.12.3.1	Páginas dinámicas de cliente	56
2.12.3.2	Páginas dinámicas de servidor	56
2.12.3.3	Tecnologías de programación para Internet dinámico	56
2.12.3.3.1	Javascript	56
2.12.3.3.2	Visual Basic Script (VBScript)	57
2.12.3.3.3	DHTML	57
2.12.3.3.4	Cascading Style Sheets (CSS)	57
2.12.3.3.5	Applets de Java	58
2.12.3.3.6	CGI (Common Gateway Interface)	58
2.12.3.3.7	Perl(Practical Extracting and Reporting Language)	58
2.12.3.3.8	ASP (Active Server Pages)	58
2.12.3.3.9	PHP (Preprocesor Hipertext)	58
<b>3.</b>	<b>REQUERIMIENTOS DEL CLIENTE</b>	<b>59</b>
3.1	ENTREVISTAS CON LOS USUARIOS.	59
3.1.1	Lic. Rosario Díaz de Guzmán, Secretaria Técnica General de ASA	59
3.1.2	Lic. Juan Luis Díaz, Jefe de la Unidad de Nuevos Proyectos de ASA	60
3.1.3	Ing. Jorge Mandri, Jefe de la Unidad de Nuevos Proyectos de ASA	61
3.1.4	Ing. Federico Dovali, Jefe de la Unidad de Operación Aeroportuaria	62
3.1.5	Ing. Sergio Rivera Romero, Asesor de la Unidad de Informática	62
3.2	ANÁLISIS DE LA ENTREVISTAS	62
3.3	RECOPIACIÓN DE LOS REQUERIMIENTOS DEL SISTEMA	66
<b>4.</b>	<b>DESARROLLO DE LA BASE DE DATOS</b>	<b>67</b>
4.1	BASES DE DATOS RELACIONALES	67
4.1.1	Introducción a las Bases de Datos Relacionales	67
4.1.2	Ventajas de las Bases de Datos Relacionales	67
4.1.3	Modelo Entidad Relación	68
4.1.4	Relaciones entre entidades	70
4.1.5	Diagrama Entidad-Relación	70
4.1.6	Tipos de relaciones	70
4.1.7	Normalización	74
4.2	Lenguaje SQL	77
4.2.1	Antecedentes	77
4.2.2	Estructura del Lenguaje SQL	77
4.2.2.1	Comandos	78
4.2.2.2	Cláusulas	78
4.2.2.3	Operadores Lógicos	78
4.2.2.4	Operadores de Comparación	79
4.2.2.5	Funciones de Grupales	79
4.3	JUSTIFICACION DEL DBMS	79
4.3.1	Recursos Disponibles en ASA	79
4.3.2	Características de MySQL	80
4.3.3	Características de PostgreSQL	80
4.3.4	Justificación de la elección de MySQL	81
4.4	ANALISIS	81
4.4.1	Entidades	83
4.4.2	Relaciones	85
4.4.3	Atributos	85
4.4.4	Diseño	91



<b>5. ANALISIS, DISEÑO Y DESARROLLO DE LA APLICACIÓN</b>	<b>92</b>
5.1 PROGRAMACIÓN ESTRUCTURADA	92
5.2 JUSTIFICACIÓN DE LA PLATAFORMA DE DESARROLLO	92
5.2.1 Plataforma Linux	93
5.2.2 Distribuciones Linux	93
5.2.2.1 Distribución Red Hat	93
5.2.3 El Servidores Web	94
5.2.4 Tecnologías para el desarrollo de aplicaciones Web	95
5.2.4.1 CGI (Common Gateway Interface)	95
5.2.4.2 Perl	95
5.2.4.3 ASP (Active Server Pages)	95
5.2.4.4 JSP (Java Server Pages)	95
5.2.4.5 PHP (Hipertext Preprocesor)	96
5.2.5 Justificación de la plataforma de desarrollo RedHat-Apache-Php	96
5.3 ANÁLISIS	97
5.3.1 Diagrama Jerárquico Funcional	98
5.3.2 Diagrama de Contexto	99
5.3.3 Diagrama de Flujo de Datos Del Sistema	100
5.3.4 Mantenimiento a catálogos	103
5.4 DISEÑO Y DESARROLLO	104
5.4.1 Diseño de un Sitio Web	104
5.4.2 Diseño del Sitio Web en cuestión de estudio	105
5.4.3 Desarrollo	108
5.5 CONCLUSIONES	115
<b>6. SEGURIDAD EN LOS SISTEMAS DE INFORMACIÓN</b>	<b>116</b>
6.1 SEGURIDAD DEL SISTEMA OPERATIVO	116
6.2 SEGURIDAD DE LA BASE DE DATOS	119
<b>7 AFINACIÓN DEL SISTEMA DE INFORMACIÓN</b>	<b>121</b>
7.1 PRESENTACIÓN PRELIMINAR AL CLIENTE.	121
7.2 RECOPIACIÓN DE OBSERVACIONES	122
7.3 AFINACIÓN DEL SISTEMA DE INFORMACIÓN	123
<b>8. OPERACIÓN DEL SISTEMA</b>	<b>129</b>
8.1 DOCUMENTACIÓN	130
8.2 MANUAL DE USUARIO	130
8.3 ESQUEMA DE OPERACIÓN	130
8.4 PERSONAL TÉCNICO RESPONSABLE	130
8.5 REQUERIMIENTOS MINIMOS DE OPERACIÓN	130
8.6 AMBIENTE DE TRABAJO	130
8.6.1 Edición de datos	132
8.6.1.1 Agregar datos	132
8.6.1.2 Modificar datos	133
8.6.1.3 Eliminar registros	135
8.7 CATÁLOGOS DEL SISTEMA	136
8.8 EDICIÓN DE DATOS DE AEROPUERTOS	137
8.9 DEPENDENCIA DE CATÁLOGOS	139
8.10 CONSULTA DE DATOS POR PARTE DE LOS USUARIOS FINALES	139
8.11 RESPALDOS DE LA BASE DE DATOS	143
8.12 RESTAURACIÓN DE LA BASE DE DATOS	143
8.13 SALIDA DEL SISTEMA	144



## Sistema de Información de Aeropuertos

---



CONCLUSIONES DEL PROYECTO	145
BIBLIOGRAFÍA	146
APÉNDICE A - COMANDOS DDL DE LA BASE DE DATOS	147



## Sistema de Información de Aeropuertos



### 1. DESCRIPCIÓN DEL CASO DE ESTUDIO

#### 1.1 AEROPUERTOS Y SERVICIOS AUXILIARES (ASA)

El Aeropuerto de la Ciudad de México se construyó en 1929 y estaba conformado por dos pistas, este hecho representó el inicio del desarrollo aeroportuario en México. Para 1931, fueron agregados a dicho aeropuerto el edificio terminal de pasajeros y algunos hangares. Para el año de 1960, el país contaba ya con 31 aeropuertos, los cuales eran operados y administrados por diferentes organismos, tanto públicos como privados, sin embargo, algunos de los aeropuertos tenían serias deficiencias en cuanto a infraestructura y operación se refiere.

Era necesario crear un organismo especializado en la materia, con el objeto de modernizar a todos los aeropuertos de la República Mexicana, proporcionar el mantenimiento adecuado a los aeropuertos existentes, mejorar los servicios de navegación aérea, modernizar los sistemas de suministro de combustibles, incrementar el desarrollo de la aviación comercial y establecer nuevas rutas de aviones, entre otras tareas. Por lo tanto, el 10 de junio de 1965 se crea por decreto presidencial Aeropuertos y Servicios Auxiliares (ASA), obteniendo el carácter de organismo público descentralizado con patrimonio propio.

Para dicho año, ASA operaba los 34 aeropuertos existentes en el país, proporcionando los servicios relacionados al aterrizaje y despegue de las aeronaves. Cabe destacar que algunos de los servicios seguían siendo prestados por otras empresas, tal como el suministro de combustible. En 1978, la Empresa Nacional de Combustibles de Aviación pasa a formar parte de Aeropuertos y Servicios Auxiliares, con lo que ASA toma el control de todos los servicios que se les proporcionaban a los aeropuertos en el país. ASA tomó entonces el control total de todos los aeropuertos en México.

En 1995 se publica en el Diario Oficial de la Federación la Nueva Ley de Aeropuertos, en la que se observa la necesidad de abrir los aeropuertos a la iniciativa privada y en 1998 se concesionan al sector privado 35 de los 58 aeropuertos operados por ASA. Desde entonces, se definió una regionalización de los aeropuertos en grupos bajo el esquema de sociedades mercantiles. Así pues, se conformaron los siguientes grupos, para la operación de los aeropuertos:

Grupo Aeroportuario ASA Corporativo: Ciudad Obregón, Ciudad del Carmen, Colima, Comitán, Campeche, San Cristóbal de las Casas, Chetumal, Cuernavaca, Ciudad Victoria, Guaymas, Ixtepec, Loma Bonita, Loreto, Lázaro Cárdenas, Matamoros, Nuevo Laredo, Nogales, Poza Rica, Pachuca, Palenque, Puerto Escondido, Querétaro, Terán, Tehuacan, Tuxtla Gutiérrez, Toluca, Taquín, Tepic y Uruapan.

Grupo Sureste: Cancún, Cozumel, Huatulco, Mérida, Minatitlán, Oaxaca, Tapachula, Veracruz y Villahermosa.

Grupo Pacífico: Tijuana, Mexicali, Hermosillo, Los Mochis, La Paz, Bajío, San José del Cabo, Morelia, Puerto Vallarta, Aguascalientes, Guadalajara y Manzanillo.

Grupo Centro Norte: Ciudad Juárez, Chihuahua, Reynosa, Monterrey, Torreón, Culiacán, Durango, Mazatlán, San Luis Potosí, Tampico, Zacatecas, Zihuatanejo y Acapulco.

Grupo Ciudad de México: Aeropuerto Internacional de la Ciudad de México

Grupo Puebla: Aeropuerto de Puebla



## Sistema de Información de Aeropuertos



De esta manera, la operación y administración de los aeropuertos queda bajo la responsabilidad de los grupos aeroportuarios antes mencionados, incluyendo al grupo ASA.

Este fenómeno de privatización de los aeropuertos no es propio del país, pues una buena parte de los aeropuertos que operan en el mundo han recibido la participación de la iniciativa privada con el objeto de modernizar y mejorar la infraestructura y atender con seguridad, calidad y oportunidad al creciente número de pasajeros que hacen uso de los aeropuertos, así como a la mercancía que se transporta en los mismos.

ASA ofrece los servicios de Operación Aeroportuaria, Suministro de Combustibles y Consultoría a los aeropuertos que opera directamente y a los aeropuertos concesionados de los grupos aeroportuarios privados.

La Operación Aeroportuaria, está conformada por el conjunto de actividades relacionadas con el funcionamiento de equipos, instalaciones y facilidades de una terminal con el propósito de que el usuario se encuentre seguro dentro de la terminal, con lo que se optimiza el funcionamiento del aeropuerto. Dentro de las principales actividades de la operación aeroportuaria se encuentran: el control del tráfico aéreo, las ayudas tanto visuales como de radio, asignación de posiciones de estacionamiento en las plataformas para los aviones, las instalaciones, equipo y personal del CREI, supervisión del servicio que brindan las aerolíneas de pasajeros y carga, seguridad en equipos de vuelo, de servicio en plataforma, de las personas que hacen uso del aeropuerto así como de sus bienes, coordinación y vigilancia de aduanas, con respecto al control de sanidad, migración, armas, contrabando y drogas, control de acceso a zonas restringidas, mantenimiento a instalaciones y equipos, y la elaboración de manuales e instructivos.

La Unidad de Combustibles, lleva a cabo los procesos de recepción, almacenamiento, control de calidad y suministro de Turbosina y Gas avión a las aeronaves comerciales, nacionales y extranjeras en todos los aeropuertos del país.

La consultoría es el servicio que pretende aprovechar los conocimientos de los ingenieros y técnicos mexicanos, especializados en la industria aérea, así como los de especialistas en diseño, administración, sistemas de cómputo, finanzas y comercialización dentro y fuera de la Red de ASA, incluyendo aeropuertos internacionales ubicados en centro y Sudamérica. El servicio de consultoría comprende los siguientes tópicos:

**Estudios y Planeación:** Comprende la planificación para la remodelación de los edificios terminales, y en caso de ser necesario, la construcción de nuevos, estudios que permitan ubicar de manera eficiente las señalizaciones, tanto en el interior como en el exterior de los aeropuertos, estudios de factibilidad técnica en el desarrollo de nuevos aeropuertos, así como del impacto ambiental que su construcción implique, sistemas de automatización para edificios (edificios inteligentes), evaluación de costos y de proyectos, evaluación de la infraestructura aeroportuaria, estudios de impacto ambiental, entre otros.

**Proyectos:** Establecen las normas arquitectónicas y de servicios en los nuevos aeropuertos, determinan las especificaciones necesarias para las ampliaciones y/o remodelaciones de los edificios terminales, se llevan a cabo planes maestros para el desarrollo de nuevos aeropuertos y se desarrollan proyectos que permitan optimizar el consumo de energía.

**Licitaciones y Contratos:** La parte correspondiente a Licitaciones y Contratos permite integrar las bases de los concursos para el desarrollo de las obras por parte de empresas privadas, así como la integración de procedimientos operativos para las licitaciones, la supervisión de las obras realizadas y el pago de las mismas.



## Sistema de Información de Aeropuertos



Consultoría de operaciones: Comprende la elaboración de normas y procedimientos de operación de los aeropuertos, basados en los estándares internacionales, la elaboración de planes de seguridad, contingencia y emergencia, se desarrollan programas de capacitación y adiestramiento al personal y se imparten seminarios y cursos sobre seguridad operacional y aeroportuaria.

### 1.2 REQUERIMIENTOS DE INFORMACIÓN DE ASA.

Actualmente ASA presta los servicios de consultoría y suministro de combustibles a los aeropuertos nacionales que están a cargo de organismos privados. Se pretende expandir dichos servicios de ASA a otros aeropuertos en el extranjero, concretamente en América Latina aunque se tienen proyectos para el desarrollo aeroportuario en algunos países del continente africano. También se requiere tener a la mano la información fundamental de cualquier aeropuerto a nivel nacional, operado o no por ASA.

Es por eso que se necesita contar con un sistema de información que suministre datos referidos a la operación de los aeropuertos nacionales que están bajo la responsabilidad de ASA, de los aeropuertos nacionales que están bajo la responsabilidad de grupos privados, y de los aeropuertos internacionales de centro y Sudamérica, como de cualquier parte del mundo, debido a que se tiene contemplada la firma de convenios de intercambio tecnológico con aeropuertos de Europa.

Este sistema, deberá operar bajo ambiente cliente servidor para la captura, consulta e impresión de datos, y deberá ser capaz de enviar información de sólo lectura a páginas dinámicas de Internet. El usuario, alimentará los datos desde las terminales remotas en las oficinas de ASA, y podrá consultarlos, ya sea desde la red interna de ASA, o desde cualquier parte del mundo vía Internet con sus respectivas autenticaciones de usuario (clave y contraseña).

De los aeropuertos nacionales, es importante conocer los siguientes datos: siglas (cada aeropuerto tiene sus siglas de manera independiente, por ejemplo, MEX para el Aeropuerto de la Ciudad de México, HUX para el aeropuerto de Huatulco, etc.), nombre (por ejemplo Aeropuerto Internacional de la Ciudad de México), tipo (Metropolitano, Regional, Turístico, Fronterizo y Estación), clasificación (Internacional, Nacional y Regional), alcance (Largo, Mediano y Corto), año de inicio de operaciones (1965), localidad (Ciudad de México), el estado de la república al que pertenece (Distrito Federal). Con respecto a las pistas, se requieren los siguientes datos: número de pistas, tipo (asfálticas o de concreto) y longitud de las mismas (3846x45m), así como el tipo de aviones para los cuales fueron diseñadas (En el caso del aeropuerto de la Ciudad de México, el avión máximo operable es el Boeing 747).

Los aeropuertos, cuentan con plataformas de pernoctación para la aviación comercial y de carga. De los edificios terminales es importante conocer la superficie de la terminal para pasajeros comerciales y para aviación general. Es necesario conocer: Información referente a las ayudas de navegación (ayudas visuales y radio ayudas), sistemas de prevención de accidentes y calles de rodajes. El tipo de servicios que tiene cada aeropuerto: Bancos, restaurantes, taxis, reservaciones de hoteles, renta de vehículos, etc. La capacidad de operación del aeropuerto, es un factor muy importante que el sistema debe considerar: capacidad de operaciones anuales, y la capacidad de pasajeros de cada aeropuerto. El historial de obras realizadas en el aeropuerto, la fecha, el tipo de obra (ampliación o remodelación) y la descripción de la obra. Los aeropuertos, están clasificados en Nacionales, Internacionales, Locales y Regionales). Es importante tener a la mano algunas fotografías del aeropuerto, edificio terminal, pistas, etc.

Con respecto a los aeropuertos en el extranjero, los datos a recabar son muy similares: nombre, siglas, y ubicación geográfica (país, entidad, localidad). El tipo de administración del aeropuerto (gubernamental o privado), así como el nombre del organismo encargado de la administración.



## Sistema de Información de Aeropuertos

---



La capacidad de operación anual en pasajeros y operaciones, el número de plataformas, y el tipo de las mismas (desembarco, pernocta, aviación general), así como las dimensiones en metros cuadrados de las mismas.

El tipo de sistemas de navegación aérea, el tipo de sistemas de suministro de combustible, las instalaciones de prevención de accidentes y desastres, el historial de remodelaciones de los edificios terminales, el número de estacionamientos, el tamaño en metros cuadrados de los mismos, y la capacidad del número de vehículos, así como los servicios que ofrece el aeropuerto (taxis, casas de cambio, alquiler de vehículos, etc.). Es importante que el sistema despliegue las imágenes que se tengan disponibles de los aeropuertos, tales como torre de control, edificio principal, pistas, etc.

### 1.3 RECURSOS Y LIMITACIONES EXISTENTES

En las oficinas generales del Corporativo ASA, se cuenta con una red basada en plataforma Windows. Se tienen en los equipos diferentes versiones de éste sistema operativo: Windows 98, Windows Me, Windows XP, Windows 2000 profesional, y Windows 2000 Server y NT 4.0 en el caso de los servidores. Con respecto a los lenguajes de programación, se cuentan con licencias de Visual Basic y Power Builder.

Como manejadores de Bases de Datos relacionales, se cuenta con el programa Access, y la base de datos que acompaña a Power Builder, que es SQL Anywhere Studio, apta para trabajar en ambiente cliente servidor. La red es del tipo Ethernet, cuenta con varios servidores Windows NT 4.0 y 2000 Server, con direcciones IP fijas (similares a las red de DGSCA).

El sitio Web de ASA está soportado por un servidor Windows NT 4.0, con el Internet Information Server, puede manipular páginas ASP, y es el servidor en donde será instalado el Sistema de Información de Aeropuertos.

Por motivos de recortes en el presupuesto, ASA está adoptando las políticas de desarrollar sobre plataforma libre, es decir, sobre Linux. Los sistemas que entrarán en esta nueva etapa, estarán soportados por un servidor Web gratuito con alguna base de datos libre, y PHP como plataforma de desarrollo.



---

### 2. ANTECEDENTES HISTÓRICOS

#### 2.1 METODOLOGÍAS EN EL DESARROLLO DE SISTEMAS

##### 2.1.1 Antecedentes

Hoy en día los sistemas de información constituyen el medio fundamental para el manejo de las organizaciones. Durante los primeros años del desarrollo de la informática el hardware era el elemento más importante de los sistemas informáticos mientras que el software de aplicación no lo era tanto, existían pocas metodologías para la programación de aplicaciones ya que estas se realizaban prácticamente sin ninguna planificación.

Los costos de los equipos de cómputo fueron disminuyendo considerablemente mientras que las organizaciones fueron teniendo necesidades cada vez mayores de aplicaciones más potentes y robustas. Esto hizo que surgieran un conjunto de métodos, herramientas y procedimientos que facilitaron el desarrollo de sistemas de información, suministrando las bases para desarrollar software de alta calidad de una forma más productiva.

Considerar a dichos desarrollos (software) únicamente como los programas que están instalados en una computadora, es una visión errónea, debido a que el proceso de ingeniería de software va más allá de un conjunto de programas; implica la documentación necesaria para instalar los programas, la documentación necesaria para utilizarlos, el desarrollo y el mantenimiento de los mismos. El término Ingeniería de Software empezó a utilizarse a principios de los años 60's, las computadoras estaban en su tercera generación, y los programas que se implementaban en las mismas tenían ciertas deficiencias, debido a que las metodologías en el desarrollo del software no eran suficientes. Actualmente, se han implementado metodologías nuevas, con el propósito de tener nuevas herramientas para desarrollar software de manera más eficiente.

El proceso de ingeniería de software, implica un conjunto estructurado de etapas en el desarrollo de los programas; dichas etapas, varían en función del tipo de organización y del tipo de sistema que se va a implementar, es entonces necesario, disponer de un modelo de proceso para el desarrollo del software. La identificación de la llamada crisis del software en 1960 y la idea de que el desarrollo de software es una disciplina de la ingeniería, permitió ver al desarrollo de software como otro proceso de ingeniería.

Proceso de Ingeniería.

- Especificación
- Diseño
- Implementación
- Prueba
- Instalación
- Mantenimiento

Características de un proceso de software.

- Claridad
- Visibilidad
- Facilidad de soporte
- Aceptación
- Fiabilidad
- Robustez



### 2.1.2 Sistema de Información

Un sistema de información es la parte de una organización que proporciona, utiliza y distribuye la información, conjuntamente con los recursos que ésta tiene asociados (humanos, técnicos y financieros). En muchas ocasiones este sistema de información está automatizado parcial o totalmente.

A nivel de software, un sistema de información, está constituido por bases de datos y programas de aplicación que convierten los datos en información requerida por los usuarios, con el objeto de que la organización pueda disponer con rapidez y eficacia de información completa y confiable. Por lo tanto, el desarrollo de un sistema de información consiste en cualquier tipo de automatización o modificación (corrección, mejora, progreso, etc.) realizada sobre un sistema de información con el fin de satisfacer las necesidades cambiantes de una organización.

### 2.1.3 Ciclo de vida de un sistema de información

El ciclo de vida de un sistema de información, identifica y describe cada una de las fases en las que se puede descomponer el proceso de desarrollo de dicho sistema de información, la naturaleza y el contenido de las tareas a realizar, así como los resultados esperados en cada una de las fases.

### 2.1.4 Metodologías para el desarrollo de sistemas de información

Son métodos que indican cómo hacer más eficiente el desarrollo de sistemas de información. La vida de dichos sistemas suele estructurarse en fases con el fin de facilitar su planificación, desarrollo y mantenimiento. Las metodologías de desarrollo de sistemas deben definir: objetivos, fases, tareas, productos y responsables necesarios para la correcta realización del proceso y su seguimiento.

Los principales objetivos de una metodología de desarrollo son:

- Asegurar la calidad tanto del desarrollo como del sistema en sí.
- Satisfacer las necesidades de los usuarios.
- Conseguir un mayor nivel de rendimiento y eficiencia de los recursos asignados al desarrollo.
- Ajustarse a los plazos y costos previstos en la planificación.
- Generar de forma adecuada la documentación asociada.
- Facilitar el mantenimiento posterior de los sistemas.

### 2.1.5 Metodología clásica, en cascada o secuencial

La metodología clásica, conocida como metodología en cascada o metodología secuencial, se basa en una serie de etapas secuenciales: análisis, diseño, desarrollo, pruebas y mantenimiento. Cabe destacar que en ésta metodología no se continúa con la siguiente etapa hasta que no se hayan concluido las especificaciones de la etapa actual.

Ciclo de vida en cascada:

- Análisis y definición de requerimientos
- Diseño del sistema
- Implementación y prueba unitaria.
- Integración y prueba.
- Operación y mantenimiento.



### Análisis y Definición de Requerimientos

Por medio de entrevistas con el cliente, se define la funcionalidad, restricciones y objetivos del sistema.

### Diseño del sistema

Divide los requerimientos correspondientes al hardware y al software, y establece la arquitectura global del sistema. El diseño de software involucra la representación de las funciones del sistema, que pueden ser transformadas en uno o más programas ejecutables.

### Implementación y prueba unitaria

Durante esta etapa, el diseño del software es realizado como un conjunto de programas o unidades, verificando que cada una de esas unidades tenga una función específica dentro del sistema.

### Integración y prueba

Los programas individuales o unidades son integrados, y probados como un sólo programa. Es necesario verificar que los requerimientos son cumplidos al 100%. Después de la prueba, el software es liberado al cliente.

### Operación y mantenimiento

El sistema es instalado y puesto en marcha. El mantenimiento implica corregir los posibles errores que no hayan sido descubiertos, incluso, tomar nota de nuevos requerimientos de información que no se conocían.

### Limitaciones:

- Es el modelo más sencillo, más difundido y utilizado.
- Plantea una organización sistemática y secuencial del proceso de software.
- El único objeto formalizado es el código y aparece al final del proceso.
- Únicamente se prueba el código.
- El mantenimiento se realiza sobre el código fuente, que no refleja decisiones de diseño previas.

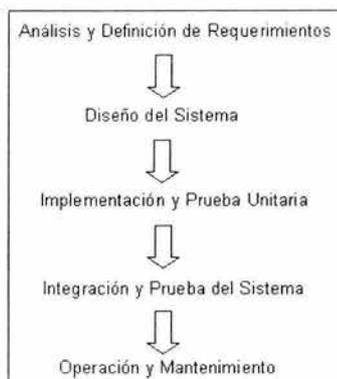


Figura 2.1 Metodología clásica



### 2.1.6 Modelo evolutivo.

El modelo evolutivo surge a raíz de las limitaciones de retroalimentación entre sus etapas que tiene el modelo de ciclo de vida en cascada. El objetivo es trabajar con el cliente desde el inicio hasta el final del desarrollo, y construir un sistema final utilizando especificaciones un tanto informales. Es importante tener un panorama amplio con respecto a las necesidades de información de los clientes. El sistema irá evolucionando en función de las nuevas características que el cliente vaya sugiriendo. Este modelo centra su atención en aquellas partes del sistema cuyo entendimiento sea más complicado.

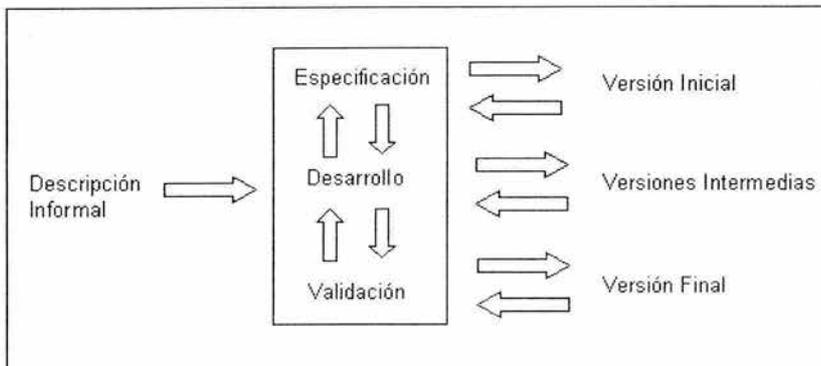


Figura 2.2 Modelo Evolutivo

### 2.1.7 Modelo exploratorio

La programación exploratoria está basada en la idea implementar un desarrollo inicial, y posteriormente refinar este desarrollo hasta que el sistema cumpla con las expectativas para las cuales fue diseñado. La programación exploratoria es un buen camino para realizar desarrollos en donde es difícil establecer una detallada especificación del sistema. La clave para completar un desarrollo basado en esta metodología, radica en utilizar técnicas que permitan iterar de manera rápida tantas veces como sea necesario hasta tener un desarrollo acorde a las necesidades del negocio. Esto implica que el desarrollador deberá tener un amplio dominio de lenguajes de programación como LISP o Prolog.

Las principales desventajas de esta metodología son:

- Deficiente visibilidad del proceso.
- Sistemas deficientemente estructurados.
- Puede ser necesario emplear técnicas especiales (lenguajes de prototipado).

Aplicación:

- Muchas veces el objetivo de este modelo es construir un prototipo usado para entender y validar especificaciones.
- Por tanto, puede ser parte de algún proceso de desarrollo (cascada).
- Sistemas interactivos de tamaño pequeño o medio.



## Sistema de Información de Aeropuertos

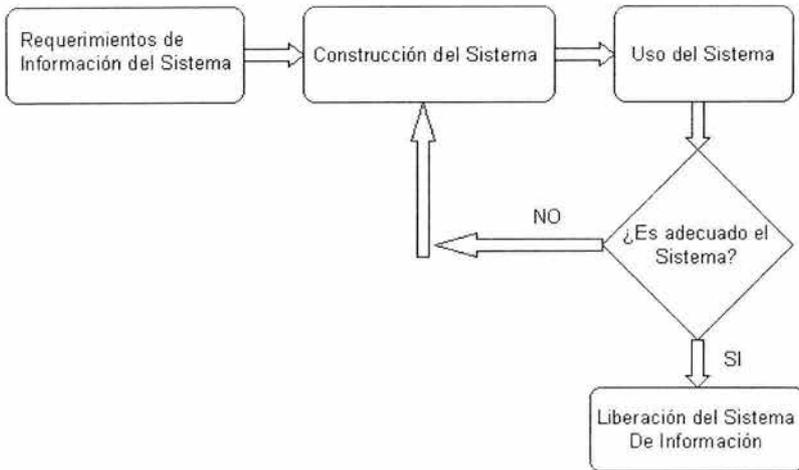


Figura 2.3 Modelo Exploratorio

### 2.1.8 Metodología en espiral

El proceso de Metodología en Espiral fue descrito por Boehm en 1988 haciendo un tratamiento explícito del proceso de riesgo. Se caracteriza por asociar a cada fase del proceso de desarrollo con un giro dentro de una espiral.

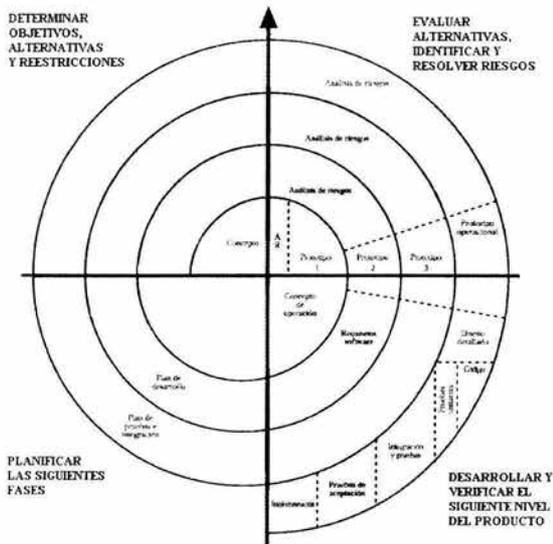


Fig. 2.4 Metodología en espiral



## Sistema de Información de Aeropuertos



La primera parte, inicia por determinar los alcances, las características de desempeño, funcionalidad y restricciones del sistema de información. En la siguiente etapa o giro de la espiral, se especifican los requisitos de información del sistema, y se lleva a cabo la validación de dichos requisitos. En la siguiente etapa se lleva a cabo el diseño del sistema, se integra y se realizan pruebas al diseño.

La última etapa consiste en evaluar el prototipo operacional, realizar un diseño detallado, codificar el sistema, hacer las evaluaciones tanto unitarias para cada parte del sistema como de manera integral para todo el sistema, así como las pruebas de aceptación por parte de los usuarios.

Cabe destacar que antes de iniciar / concluir cada ciclo de la espiral o etapa, se lleva a cabo un análisis de riesgo. El concepto de riesgo, es un tanto difícil de definir con precisión; en todos los casos interviene el criterio del desarrollador para emitir un fallo, por ejemplo, si la intención es utilizar un nuevo lenguaje de programación, el riesgo sería no tener un compilador apropiado disponible. El riesgo es consecuencia de tener información inadecuada o no disponible.

Cada vuelta en la espiral se divide en 4 sectores:

- Definición de objetivos: Identificación de los objetivos específicos de cada fase
- Evaluación y reducción del riesgo
- Desarrollo y validación: Elección de un modelo adecuado para la siguiente fase del desarrollo.
- Planificación: Se revisa el estado del proyecto y se planifica el siguiente giro.

Modelo en espiral.

- En cada giro se construye un nuevo modelo del sistema completo
- Este modelo puede combinarse con otros modelos del proceso de desarrollo (cascada, evolutivo).
- Es el mejor modelo para el desarrollo de grandes sistemas.
- Su elevada complejidad desaconseja su uso en pequeños sistemas.
- El análisis del riesgo requiere la participación de personal con alta calificación
- Ha sido utilizado con éxito en el desarrollo de grandes sistemas, consiguiendo duplicar la productividad.

### 2.1.9 HERRAMIENTAS CASE

#### 2.1.9.1 Antecedentes

Desde finales de los años sesenta, durante la llamada crisis del software, se han desarrollado numerosas herramientas para facilitar el desarrollo de sistemas de información. Existe un grupo de herramientas, conocidas como Herramientas CASE (Computer Aided Software Engineering: Ingeniería de Software Asistida por Computadora). Cabe destacar que las herramientas CASE pueden ser utilizadas en cualquier parte del ciclo de vida de un sistema de información: existen herramientas que proporcionan asistencia durante las fases de análisis de requerimientos de información y diseño de especificaciones, y también existen herramientas que permiten la generación automática de código a partir de las especificaciones del diseño.

Las herramientas CASE reducen e incluso eliminan, muchos de los problemas de diseño y desarrollo propios de los proyectos de mediano y gran tamaño. Uno de los mayores propósitos de la tecnología CASE, es el de separar el diseño de la implementación.



### 2.1.9.2 Evolución

La tecnología CASE surge cuando empiezan a aparecer las primeras metodologías estructuradas y las investigaciones sobre los ambientes de desarrollo alcanzan una etapa de madurez. A mediados de los años ochenta, el CASE se difunde de manera masiva y surgen las primeras herramientas de documentación y diagramación automática. También tiene un papel importante en este desarrollo la aparición de las estaciones de trabajo en ambiente gráfico, las cuales aportan una interfaz gráfica asociada a una gran capacidad de proceso y que constituyen la plataforma original de muchas herramientas CASE. A mediados de los años noventa, surge un conjunto de nuevas herramientas CASE, esta nueva generación de herramientas CASE suele aparecer bajo la clasificación de herramientas para el desarrollo de aplicaciones, con las que se pueden crear de manera asistida, aplicaciones de alto nivel con una muy buena interfaz gráfica. Estas herramientas para el desarrollo de aplicaciones, constan de varios elementos: manejadores de bases de datos relacionales de alta capacidad, herramientas de diagramación, lenguajes de cuarta generación, orientados a objetos, con el manejo de eventos, soporte para el desarrollo en arquitectura cliente/servidor, etc. Las herramientas actuales, aparte de contar con las características antes mencionadas, cuentan con soporte para el desarrollo en Internet (servidor Web), soporte para documentación automática, compatibilidad con diferentes plataformas (UNIX, Windows), etc.

### 2.1.9.3 Elementos de una herramienta CASE

#### 2.1.9.3.1 Repositorio de datos

El repositorio de datos, es la base de datos central, comúnmente denominada diccionario de recursos de información. El repositorio va más allá del diccionario de datos, ya que incluye toda la información que se va generando a lo largo del ciclo de vida del sistema, por ejemplo: componentes de análisis y diseño (diagramas de flujo de datos, diagramas entidad-relación, esquemas de bases de datos, diseños de pantallas), estructuras de programas, algoritmos, etc. No todas las herramientas CASE del mercado poseen un repositorio propio, algunas trabajan sobre un repositorio suministrado por otro fabricante o vendedor y otras carecen completamente de él.

#### 2.1.9.3.2 Herramienta generación de prototipo

El objetivo principal de esa herramienta es poder mostrar al usuario, desde los momentos iniciales del diseño, el aspecto que tendrá la aplicación una vez desarrollada. Ello facilitará la aplicación de los cambios que se consideren necesarios, todavía en la etapa de diseño.

#### 2.1.9.3.3 Generador de Bases de Datos

Muchas de las herramientas CASE del mercado, pueden generar, a partir del diccionario de datos definido en el repositorio, la base de datos de la aplicación a partir de un Diagrama Entidad Relación, por medio de comandos tipo DDL (Lenguaje de Definición de Datos) del lenguaje SQL. Algunas herramientas incorporan incluso la posibilidad de realizar ingeniería inversa, modificando el diccionario de datos cuando se dan modificaciones en la Base de Datos.

#### 2.1.9.3.4 Generador de código

Permite generar el código en algún lenguaje 4GL, con las siguientes características:

- Lenguaje estandarizado, por lo general C, C++ o Java.
- Portabilidad del código generado capacidad para poder ejecutarlo en diferentes plataformas.
- Generación del código asociado a las pantallas e informes de la aplicación. Mediante esta característica se obtendrá la interfaz de usuario de la aplicación.



### 2.1.9.3.5 Generador de documentación

El módulo generador de la documentación se alimenta del repositorio de datos para transcribir las especificaciones ahí contenidas.

Algunas características de los generadores de documentación son:

- Generación automática de documentos a partir de los datos del repositorio sin necesidad de un esfuerzo adicional.
- Combinación de información textual y gráfica para facilitar su comprensión.
- Facilidad para la introducción de textos complementarios a la documentación que se genera de forma automática.
- Interfaz con otras herramientas: procesadores de textos, editores gráficos, etc.

### 2.1.9.3.6 Módulo de administración de proyectos

Algunas herramientas CASE incorporan un módulo para la administración de los proyectos de desarrollo de sistemas. Las principales características de este módulo son:

- Personalización por medio de archivos .ini de las opciones del programa (opciones de cálculo, selección de datos a visualizar, etc.).
- Presentación de diferentes vistas del proyecto (por tareas, por recursos, por fechas, etc.)
- Resolución de conflictos de los recursos.
- Facilidades para la impresión de programas de trabajo.
- Conexión entre varios proyectos.
- Facilidades de importación / exportación hacia otras plataformas.
- Facilidad de comunicación con otras herramientas (hojas de cálculo, aplicaciones gráficas, correo electrónico, páginas Web, etc.).

### 2.1.9.3.7 Bibliotecas de clases de objetos

La función de estas bibliotecas es obtener de ellas objetos, módulos o partes del código que se puedan implementar directamente en la aplicación que se está desarrollando. Las bibliotecas de clases suelen ser específicas de un determinado lenguaje, sin embargo, se tiende a eliminar esta limitación. Hay bibliotecas de clases que se han diseñado para:

- La creación de interfaces gráficas de usuario (GUI).
- El acceso a bases de datos.
- La integración de funciones multimedia.
- El tratamiento de documentos.
- El intercambio electrónico de datos.
- El desarrollo de aplicaciones científicas, matemáticas o de ingeniería.

### 2.1.9.3.8 Herramientas de Servidor Web

Las principales características de estas herramientas son:

- Soporte de un determinado estándar de comunicaciones con el servidor central (*host*) a través de terminales por medio de un navegador de Internet.
- Componentes para la creación de las interfaces gráficas de usuario por medio de páginas Web.
- Conexión a la Base de Datos desde Páginas Web



### 2.1.9.3.9 Clasificación de las Herramientas CASE

Las herramientas CASE pueden clasificarse en función del ciclo de vida del sistema de información que soportan

- CASE frontales (Front-end) o superiores (Upper CASE), abarcan las primeras fases de análisis y diseño.
- CASE posteriores (Back-end) o inferiores (Lower CASE), abarcan el diseño a detalle y la generación de código.
- ICASE (Integrated CASE) abarcan ambos aspectos

### 2.1.9.4 Posibles inconvenientes en el uso de herramientas CASE

- Dificultad de aprendizaje en su uso
- Si se están utilizando por primera vez su uso parece ser laborioso y complicado.
- El cliente debe estar seguro de lo que necesita
- El costo de estas herramientas es elevado.

## 2.2 ANÁLISIS, DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE SISTEMAS

La etapa del análisis permite obtener información de los requerimientos de información del cliente como la descripción de los alcances, los objetivos y los requisitos que el sistema debe cubrir, así como las especificaciones formales que describan las funcionalidades del sistema para su aprobación por parte de los usuarios finales. En diseño, se realiza un boceto con las características que tendrá la aplicación una vez construida. La parte de implementación, contempla la construcción del sistema de información en base a la etapa de diseño y se realizan las pruebas preliminares antes de presentarle el producto al cliente. La parte de implementación del sistema comprende la instalación del sistema de información al cliente, la capacitación, el vaciado de datos en el caso de tener una migración de un sistema anterior, las pruebas de validación y el desempeño del sistema con datos reales y la corrección de los errores que no hayan sido encontrados en la etapa anterior.

### 2.2.1 ANÁLISIS DE SISTEMAS

La etapa del análisis debe contemplar los siguientes puntos:

1. El dominio de información de un problema debe ser representado y comprendido.
2. Las funciones que el software debe realizar deben ser definidas.
3. El comportamiento del software al responder a eventos externos debe ser representado.
4. El proceso de análisis debe ir desde la información fundamental hacia los detalles de implementación.

La tarea de análisis de requerimientos y especificaciones es un proceso de descubrir, refinar, modelar y especificar las necesidades del usuario o empresa. Es tarea del cliente intentar concretar, detallar y formular un concepto (a veces no muy claro) de las funciones que deberá realizar el sistema de información.

El análisis de requerimientos de software puede ser dividido en cinco áreas:

- 1) reconocimiento de problemas
- 2) evaluación y síntesis
- 3) modelación
- 4) especificación
- 5) revisión



## Sistema de Información de Aeropuertos



Al inicio, el analista estudia las especificaciones del sistema (si existen) y el plan de proyecto de software. Es importante entender el software dentro del contexto de sistemas y revisar el alcance de éste. La meta del analista es reconocer los elementos problemáticos básicos como los percibe el cliente (usuario). En esta primera fase se debe realizar una observación consciente de lo que sucede en el negocio. Luego se analiza la información recolectada y se identifican los problemas, las posibles mejoras al sistema actual en caso de que existan y los objetivos que el nuevo sistema deberá cubrir.

Las personas involucradas en esta primera fase son los analistas, usuarios principales y administradores del sistema. Las actividades consisten en hacer entrevistas a los usuarios principales y administradores de los sistemas actuales, a fin de obtener la información suficiente que permita estimar los alcances del proyecto y la documentación de los resultados. La salida de esta fase es un estudio de factibilidad que contiene una definición del problema y la concentración de los objetivos.

La siguiente fase es la de la determinación de los requerimientos de información para los demás usuarios finales involucrados. Entre las herramientas utilizadas se encuentran el muestreo y la investigación de los datos relevantes, entrevistas, cuestionarios, el comportamiento de las personas encargadas de la toma de decisiones y la elaboración de prototipos. Varios de los métodos para determinar los requerimientos de información involucran la interacción con los usuarios. Esta fase sirve para formar la imagen que el analista tiene de la organización y de sus objetivos.

Las personas involucradas en esta fase son los analistas, los administradores y todos los niveles de usuarios que tendrá el sistema. El analista de sistemas necesita conocer los detalles de las funciones actuales del sistema: quién (las personas involucradas), qué (la actividad del negocio), dónde (el ambiente dentro del cual se lleva a cabo el trabajo), cuándo (en que momento) y cómo (de qué manera se desarrollan los procedimientos actuales) del negocio bajo estudio. Puede haber muy buenas razones para desarrollar el negocio utilizando los métodos actuales, y deben ser considerados cuando se diseña un sistema nuevo. Sin embargo, si la razón de las operaciones actuales es que "siempre se ha hecho así", el analista puede desear la mejora de los procedimientos. Al término de esta fase, el analista debe comprender el por qué de las funciones del negocio y tener información completa sobre las personas, objetivos, datos y procedimientos involucrados.

La evaluación del problema y síntesis de solución es la siguiente área en la que se enfocan los esfuerzos de los analistas. Estos deben definir todos los elementos externos de datos observables, evaluar el flujo y contenido de la información, definir y elaborar todas las funciones del software, entender el comportamiento del software dentro del contexto de los eventos que afectan al sistema, establecer las características de la interfaz y descubrir restricciones adicionales del diseño. Cada una de estas tareas servirá para describir el problema a fin de que una solución o enfoque pueda ser sintetizado.

Entre las herramientas utilizadas en esta fase están: el uso de diagramas de flujo de datos para diagramar la entrada, proceso y salida de las funciones del negocio en una forma gráfica estructurada. A partir de los diagramas de flujo de datos se desarrolla un diccionario de datos, que lista todos los conceptos de datos utilizados, así como sus especificaciones, si son alfanuméricos, numéricos o de fechas, y el tanto espacio que ocupan cuando se imprimen. Durante esta fase, el analista de sistemas también analiza las decisiones estructuradas que se hacen. Las decisiones estructuradas son aquellas para las que pueden ser determinadas las condiciones como posibles alternativas, acciones a tomar y reglas de acción. Hay tres métodos principales para el análisis de decisiones estructurales: lenguaje estructurado, las tablas de decisión y los árboles de decisión.



## Sistema de Información de Aeropuertos



No todas las decisiones de la organización son estructuradas, pero es importante que el analista de sistemas las comprenda. Las decisiones semiestructuradas (decisiones tomadas bajo riesgo) son sustentadas frecuentemente por los sistemas de apoyo a decisiones. Cuando se analiza este tipo de decisiones, el analista las examina de acuerdo al grado de habilidad para la toma de decisiones requerida, el grado de complejidad del problema y la cantidad de criterios considerados cuando se toma la decisión.

Cada problema de sistema es único y nunca hay una única solución correcta. La manera en que se formula una solución o recomendación depende de la capacidad y entrenamiento profesional individual de cada analista.

### 2.2.2 HERRAMIENTAS PARA EL ANÁLISIS DE SISTEMAS

Se utilizan herramientas de análisis como:

- Entrevistas
- Diagrama Entidad/Relación (DER)
- Diagrama Jerárquico Funcional (DJF)
- Diagrama de flujo de datos (DFD)

#### 2.2.2.1 Entrevistas

Las entrevistas son una herramienta importante, a través de éstas se definen cuales son las necesidades del usuario. Estas pueden realizarse con la aplicación de un cuestionario, o preguntas dirigidas en una conversación normal, en la cual se toman notas y se detectan los requerimientos. Se realizan antes de tener un diseño de sus necesidades y después de este para afinar detalles en posibles enfoques diferentes de sus necesidades.

Generalmente se pregunta:

- ¿Existe un sistema de información propiamente dicho para sus requerimientos?
- ¿Cuáles son los principales problemas que ha detectado con el actual sistema?
- ¿Cuántos sistemas se tienen hasta este momento?
- ¿Qué desearía que haga el sistema?

#### 2.2.2.2 Diagrama Entidad-Relación (DER)

Fue introducido por Peter Chen en 1976. Representa la realidad a través de un esquema gráfico empleando entidades, atributos y relaciones. Una entidad es un objeto concreto o abstracto que representa información de interés para un caso de estudio en particular. Los atributos definen las características de las entidades, mientras que las relaciones son reglas de correspondencia entre entidades con el objeto de intercambiar información. Las entidades son representadas con rectángulos, los atributos de las entidades se representan con elipses y una etiqueta dentro de un rombo indica la relación que existe entre las entidades, destacando con líneas las uniones de éstas, la llave primaria de una entidad es aquel atributo que se encuentra subrayado.

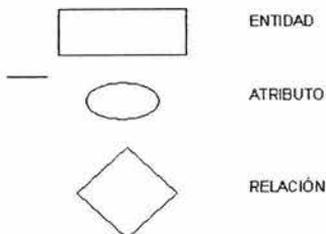
Existen los siguientes tipos de relaciones: relación uno a uno, uno a muchos y muchos a muchos. Estas últimas deben romperse con una entidad de intersección.

En la relación uno a uno, a cada elemento de una entidad A le corresponde uno y sólo un elemento de una entidad B y viceversa, mientras que en la relación uno a muchos, a cada elemento de una entidad A le corresponde uno y sólo un elemento de la entidad B, y a cada elemento de la entidad B le pueden corresponder uno o más elementos de la entidad A.



## Sistema de Información de Aeropuertos

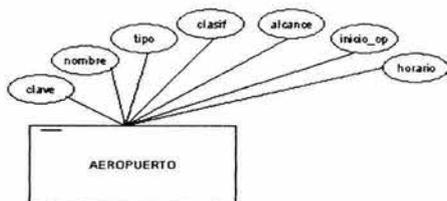
En la relación uno a muchos, a cada elemento de una entidad A le pueden corresponder uno o más elementos de la entidad B, y a cada elemento de la entidad B le pueden corresponder uno o más elementos de la entidad A.



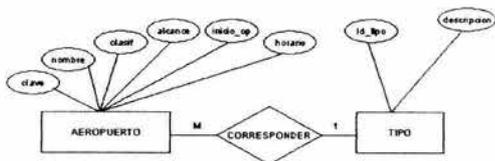
Por ejemplo: Se tienen los siguientes datos de un aeropuerto: Clave, Nombre, Tipo, Clasificación, Alcance y Año de inicio de operaciones:

<b>Clave:</b>	HUX
<b>Nombre:</b>	BAHIAS DE HUATULCO
<b>Tipo</b>	Turístico
<b>Clasificación:</b>	Internacional
<b>Alcance:</b>	Largo
<b>Año de Operación:</b>	1987
<b>Horario:</b>	07:00-18:00

La representación de los datos de un aeropuerto por medio de un por medio de una entidad con sus respectivos atributos sería:



La información se puede clasificar por medio de catálogos, por ejemplo, se puede crear una entidad referida a los tipos de aeropuertos y crear una relación uno a muchos con la entidad AEROPUERTO.





## Sistema de Información de Aeropuertos



De manera análoga, se pueden crear las entidades de CLASIFICACIÓN y ALCANCE, y relacionarlas con la entidad AEROPUERTO.

### 2.2.2.3 Diagrama Jerárquico Funcional (DJF)

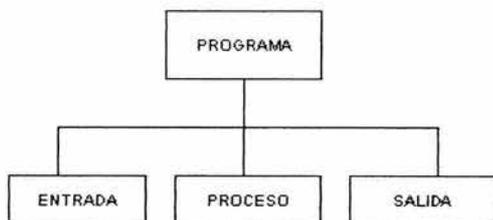
El diagrama jerárquico funcional tiene muchas similitudes con el modelo de Red para bases de datos. Una base de datos jerárquica se compone de un conjunto de funciones que se organizan formando conjuntos de árboles.

#### 2.2.2.3.1 Función del Sistema

Una función de un sistema de información es una actividad que será ejecutada para ayudar a cumplir la misión del negocio y de esta forma lograr sus objetivos. Una función puede ser dividida, es decir, descompuesta en una serie de funciones, cada una de las cuales es dividida de la misma manera. El resultado es una jerarquía de funciones. El principio de la descomposición de funciones consiste en definir una función general que muestre una visión global del sistema. Se pueden detectar funciones en las notas de entrevistas y otras fuentes, escribiendo una función general que describa al sistema como un todo y dividiéndolo en varias partes.

El producto es un modelo jerárquico de funciones el cual consiste de un diagrama con las funciones y la información necesaria para especificar cada función. Un diagrama jerárquico no muestra la secuencia en que las funciones son ejecutadas.

#### 2.2.2.3.2 Convenciones para dibujar el diagrama jerárquico funcional



- Dibujar cada función en una caja.
- Usar texto normal para la descripción de la función.
- Si se desea descomponer la función posteriormente, dibujar tres asteriscos en la parte superior derecha de la caja.
- Mostrar la referencia de la función sobre o al lado izquierdo de la caja.

- Dibujar las funciones comunes con una línea vertical a un lado de la descripción de la función dentro de la caja.
- Mostrar la etiqueta de identificación de la función maestra en la parte derecha superior de la caja en cada una de las funciones copia.

### 2.2.2.4 Diagramas De Flujo De Datos (DFD)

El diagrama de flujo de datos es un modelo que describe como fluyen los datos en el sistema de información, como los datos son transformados por medio de procesos en el sistema, las entidades externas que son fuente o destino de los datos (y en consecuencia los límites del sistema) y los lugares en donde se almacenan los datos a los cuales tiene acceso el sistema, permitiendo así describir el movimiento de los datos a través del sistema.



## Sistema de Información de Aeropuertos



En síntesis, el Diagrama de Flujo de Datos describe:

- Los lugares de origen y destino de los datos (los límites del sistema)
- Las transformaciones a las que son sometidos los datos (los procesos internos)
- Los lugares en los que se almacenan los datos dentro del sistema
- Los medios por donde fluyen los datos.

El diagrama de flujo de datos tiene un enfoque *top-down*, debido a que se construye de manera descendente partiendo de lo general a lo particular. Permite comunicar diferentes modelos, para facilitar el entendimiento entre el usuario y el analista de sistemas. La información que muestra está especificada a un nivel lógico ya que no identifica soporte físico.

Simbología:

Entidad Externa:



Entidades que no pertenecen al sistema pero que son fuentes de información.

Proceso:



Indican aquellos lugares dentro del sistema en donde la información (flujos de datos) que ingresan se procesan o transforman. Es decir, son las funciones o procesos que transforman entradas de datos en salidas de información.

Flujo de datos:



Representa la manera en la que los datos fluyen desde su origen hasta su destino, es decir, representa una estructura de datos en movimiento de una parte del sistema a otro.

Almacén o archivo:



Representa un archivo lógico en donde se agregan o de donde se extraen datos.

Para construir in DFD es necesario tener en cuenta los siguientes aspectos:

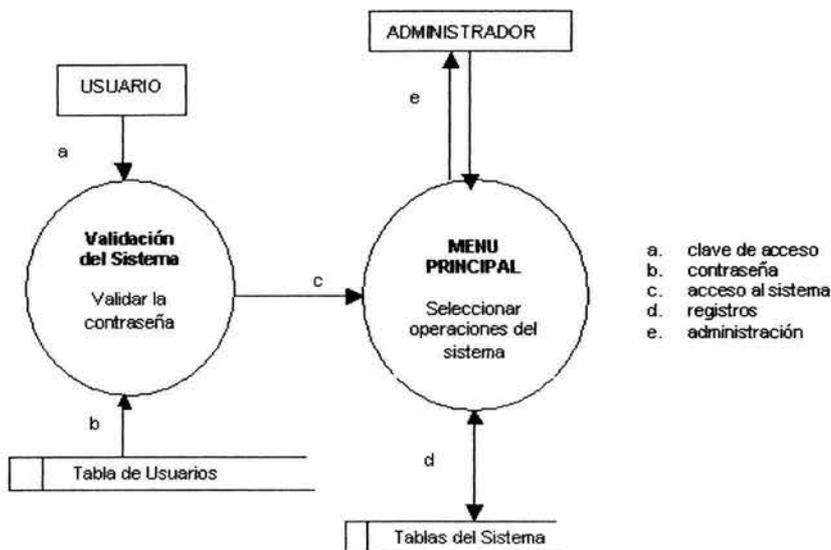
- Identificar las entidades externas ya que ello implica definir los límites del sistema.
- Elegir nombres con significado tanto para procesos como también para flujos de datos, almacenes y entidades externas. Si es posible a partir del vocabulario del usuario evitando terminologías técnicas.
- Identificar el papel del proceso del sistema, no quien lo realiza.
- Enumerar los procesos, mediante un esquema de numeración consistente que implique, para los lectores del DFD, una cierta secuencia de ejecución.
- Evitar los DFD excesivamente complejos. Deberán ser comprensibles, digeribles y agradables a la vista sin demasiados elementos.
- Todos los elementos se relacionan entre sí a través de flujos de datos.
  - Procesos: Se relacionarán con: Almacenes, Entidades externas y Otros procesos
  - Almacenes: Se relacionarán solamente con Procesos.
  - Entidades Externas: Se relacionarán solamente con Procesos.



## Sistema de Información de Aeropuertos



- En todos los niveles del Diagrama de Flujo de Datos deberá haber igual cantidad de Entradas y de Salidas.



### 2.2.3 Planificación de sistemas

El objetivo de la planificación de los proyectos de software es el de suministrar una estructura que permita al director hacer estimaciones razonables de recursos, costos y agendas. Estas estimaciones se hacen sin un marco de tiempo limitado, al principio de un proyecto de software y deben ser utilizadas normalmente a medida que progresa el proyecto.

#### 2.2.3.1 Alcance (ámbito) del software

El alcance del software, describe la función, el rendimiento, las restricciones, las interfaces y la fiabilidad del software. Las funciones descritas en la especificación del ámbito se evalúan y en algunos casos se refinan para dar más detalles antes del comienzo de la estimación. Dado que las estimaciones del costo y de las agendas están orientadas a la función del sistema, muchas veces es útil llegar a un cierto grado de descomposición. Las consideraciones de rendimiento abarcan los requisitos de tiempo de respuesta y de procesamiento. Las restricciones identifican límites del software debido al hardware externo, a la memoria disponible y a otros sistemas existentes. La función del sistema, el rendimiento y las restricciones están íntimamente relacionados.

El software interactúa con otros elementos del sistema informático. El planificador considera la naturaleza y la complejidad de cada interfaz para determinar cualquier efecto sobre los recursos, los costos y la agenda del desarrollo.

El concepto de interfaz abarca lo siguiente:

- Hardware
- Software
- Gente que hace uso del software por medio de terminales u otros medios de entrada/salida
- Procedimientos que preceden o suceden al software en una secuencia de operaciones.



### 2.2.3.2 Recursos

La segunda tarea de la planificación del desarrollo del software es la estimación de los recursos requeridos para llevar a cabo las tareas de desarrollo de software.

#### 2.2.3.2.1 Recursos humanos

El planificador comienza evaluando el ámbito y seleccionando las habilidades técnicas para llevar a cabo el desarrollo. Hay que especificar tanto la posición dentro de la organización como la especialidad. El número de personas requerido para un proyecto de software sólo puede ser determinado después de hacer una estimación del esfuerzo del desarrollo.

#### 2.2.3.2.2 Recursos de hardware

Durante la planificación del proyecto de software se deben considerar las categorías de hardware: el sistema de desarrollo, la máquina objetivo y los demás elementos del hardware del nuevo sistema. El sistema de desarrollo está compuesto por la computadora que se utilizará durante la fase de desarrollo del software y sus periféricos asociados.

#### 2.2.3.2.3 Recursos de software

Los ingenieros de software utilizan un conjunto de herramientas que son parecidas en muchos casos a las herramientas que usan los ingenieros del hardware en el diseño y la ingeniería asistidos por computadora. Por medio de las herramientas de planificación de proyectos, se pueden generar estimaciones útiles de esfuerzo, costos y duración de los proyectos de software, definir una estructura de descomposición del trabajo, y planificar una agenda factible para el proyecto. También son considerados como parte de los recursos de software, los lenguajes de programación, los manejadores de bases de datos relacionales, los generadores de reportes, etc., disponibles para la realización del proyecto

#### 2.2.3.4 Reusabilidad

Reusabilidad es la creación y reutilización de bloques constructivos de software. Cuando se especifica el software reusable como recurso, el planificador del software debe considerar dos "reglas":

- Si el software existente satisface los requisitos, hay que adquirirlo. El costo de la adquisición de software será siempre menor que el costo de desarrollo de un software equivalente.
- Si el software existente requiere de alguna modificación antes de poder integrarlo de forma adecuada en el sistema, hay que proceder con cuidado. El costo de modificación de software existente puede a veces ser mayor que el costo de desarrollo de un software equivalente.

### 2.2.4 Estimación del proyecto de software

El proceso de gestión del proyecto de software comienza con un conjunto de actividades que globalmente, se denominan planificación del proyecto. La primera de estas actividades es la estimación. La estimación es una actividad importante que no debe llevarse a cabo de forma descuidada. Existen técnicas útiles para la estimación de costos y de tiempos. El objetivo de la estimación es permitir al administrador del proyecto hacer estimaciones razonables de recursos, costo y tiempo.



## Sistema de Información de Aeropuertos



Dentro de la planeación existen las siguientes actividades:

Determinar el medio ambiente del software. Hay que evaluar:

- La función. ¿Qué hace?
- El rendimiento. Requisitos de tiempos de respuesta y de procesamiento.
- Las restricciones. Límites del software originados por el hardware externo, por la memoria disponible o por otros sistemas existentes.
- Las interfaces. Definir la comunicación hombre-máquina y máquina-máquina.
- La fiabilidad. Definir la tolerancia a fallas y a ataques de virus y/o hackers.

Para evaluar el entorno la técnica más usada es la entrevista con el cliente.

Estimación de recursos requeridos. Cada recurso queda especificado mediante cuatro características: descripción del recurso, informe de disponibilidad, fecha en la que se requiere el recurso, tiempo durante el que será aplicado el recurso.

Se puede pensar en tres clases de recursos:

- Recursos de hardware y software.
  - Recursos de software reutilizables. Existen cuatro categorías de recursos de software que se deben tomar en cuenta.
    - Componentes ya desarrollados. Se pueden comprar a un tercero o haber sido desarrollados para un proyecto anterior. Están validados totalmente y listos para usarse en este proyecto.
    - Componentes ya experimentados. Existen componentes parecidos a los que se requieren para el proyecto actual. El equipo tiene experiencia en el área de aplicación y los cambios son mínimos.
    - Componentes con experiencia parcial. Existen componentes que se relacionan con los que se requieren para el proyecto actual. El equipo sólo tiene experiencia en el área de aplicación anterior y los cambios serán sustanciales.
    - Componentes nuevos. Los componentes deben construirse específicamente para este proyecto.
- Hay que considerar lo siguiente:
- Si los componentes ya desarrollados no cumplen con los requisitos del proyecto, hay que comprarlos. El costo de compra y de integración será siempre menor que el costo de desarrollarlos. Además el riesgo es bajo.
  - Si se dispone de componentes ya experimentados, los riesgos asociados a la modificación e integración generalmente se aceptan.
  - Si se dispone de componentes con experiencia parcial, hay que analizar su uso con cuidado. Puede ser que el costo de modificarlos e integrarlos sea mayor que el de desarrollar componentes nuevos.
- Recursos humanos. Hay que especificar el esfuerzo requerido en hombres-mes u hombres-año, el número de personas dependiendo del tiempo de entrega y la posición de las personas dentro del equipo (jefe, programador, bibliotecario, especialista en comunicaciones, base de datos, etc.).

Para realizar estimaciones de costos y esfuerzos hay tres opciones:

1. Basar las estimaciones en proyectos similares ya terminados.

- Es razonable si el cliente, condiciones de administración, el medio ambiente, los requisitos, las fechas límites, son similares a proyectos anteriores.
- A pesar de eso la experiencia anterior no ha sido siempre un buen indicador de resultados futuros.



2. Utilizar técnicas de descomposición del problema.
  - Utilizan un enfoque de divide y vencerás.
  - Descomponen el proyecto en sus funciones principales y la estimación del costo y esfuerzo puede realizarse en base a métricas históricas de manera más fiable.
3. Desarrollar un modelo empírico de cálculo de costos y esfuerzos.

### 2.2.4.1 Técnicas de descomposición

La estimación de proyectos de software es una forma de resolución de problemas y, en la mayoría de los casos, el problema a resolver (esto es, desarrollar estimaciones de costo y esfuerzo para un proyecto de software) es demasiado complejo para considerarlo como una sola parte. Por esta razón, descomponemos el problema, reorganizándolo como un conjunto de pequeños problemas.

#### 2.2.4.1.1 Estimación del esfuerzo

La estimación del esfuerzo es la técnica más utilizada para calcular el costo de un proyecto de ingeniería de software. En la resolución de cada tarea del proyecto se aplica un número determinado de personas /día / mes o / año. Se asocia un costo a cada unidad de esfuerzo y se deriva el costo total estimado. Para cada función debe realizarse una serie de tareas de ingeniería del software; análisis de requisitos, diseño, implementación y pruebas.

#### 2.2.4.1.2 COCOMO (COConstructive COSt MOdel)

Barry Boehm, en su libro Software "Engineering Economics" (Prentice-Hall, 1981), introduce una jerarquía de modelos de estimación de Software con el nombre de COCOMO, por su nombre en Inglés *Constructive, Cost, Modelo* o Modelo Constructivo de Costos. Se tienen las siguientes clases de modelos:

Modelo I: El Modelo COCOMO básico calcula el esfuerzo y el costo del desarrollo de software en función del tamaño del programa, expresado en las líneas estimadas.

Modelo II: El Modelo COCOMO intermedio calcula el esfuerzo del desarrollo de software en función del tamaño del programa y de un conjunto de guías de costos que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.

Modelo III: El modelo COCOMO avanzado incorpora todas las características de la versión intermedia y lleva a cabo una evaluación del impacto de las guías de costos en cada caso (análisis, diseño, etc.) del proceso de ingeniería de software.

Los modelos COCOMO están definidos para tres tipos de proyectos de software:

- Orgánicos.
  - Proyectos pequeños y sencillos.
  - Equipos pequeños con experiencia en la aplicación.
  - Requisitos poco rígidos.
- Semiacoplados.
  - Proyectos de tamaño y complejidad intermedia.
  - Equipos con variados niveles de experiencia.
  - Requisitos poco o medio rígidos.
- Empotrados.
  - Proyectos que deben ser desarrollados con un conjunto de requisitos de hardware y software muy restringidos.



### 2.2.4.1.2.1 Determinación de costos

#### Modo orgánico

Para determinar el esfuerzo del personal y del tiempo de desarrollo, se utilizan dos ecuaciones:

El costo es

$$K_m = 2.4 S_k^{1.05}$$

donde  $K_m$  se expresa en personas-mes,  $S_k$  es el tamaño expresado en miles de líneas de código fuente.

El tiempo de desarrollo se da por

$$t_d = 2.5 K_m^{0.38}$$

donde  $K_m$  se obtiene de la ecuación anterior y  $t_d$  es el tiempo de desarrollo en meses. Estas ecuaciones se han obtenido por medio de ajustes de curvas realizado por Boehm sobre 63 proyectos.

#### Modo Empotrado.

En este modo, el proyecto tiene unas fuertes restricciones, que pueden estar relacionadas con el procesador y el interfaz de hardware. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla. Las estimaciones de tiempo y costo se basan en las mismas ecuaciones que en el modo orgánico, pero con diferentes constantes. Así, el costo se da por

$$K_m = 3.6 S_k^{1.20}$$

El tiempo de desarrollo por

$$t_d = 2.5 K_m^{0.32}$$

#### Modo semiacoplado

Dependiendo del problema, el grupo puede incluir una mezcla de personas experimentadas y no experimentadas.

Las ecuaciones son

$$K_m = 3.0 S_k^{1.12}$$

$$t_d = 2.5 K_m^{0.35}$$

### 2.2.4.1.3 modelo de estimación de Putnam

Es un modelo multivariable dinámico que asume una distribución específica del esfuerzo a lo largo de la vida de un proyecto de desarrollo de software. El modelo se ha obtenido a partir de distribuciones de mano de obra en grandes proyectos (esfuerzo total de 30 personas-año o más).

### 2.2.4.1.4 Juicio experto

Se basa en:

- Experiencia del consultor
- Tipo de proyecto "área en la cual vamos a trabajar"
- Tiempo de duración
- Número de personas involucradas "cuántas y calidades"
- Recursos de software
- Recursos de hardware
- Si se cotiza el hardware, el costo se pondrá por separado.



### 2.2.5 Cotización

Es una Propuesta de trabajo que incluye duración, costos y recursos.

Características principales de una cotización

- Se entregan personalmente
- Se realizan en hojas embretadas
- Incluye vigencia de la cotización (entre 15-30 días)
- Garantías
  - Modificaciones tolerables
  - Migraciones
  - Crédito por fallas (vigencia)
  - Mantenimiento sin/con carga adicional
- Descripción
- En su caso se presentan anexos con listas de productos y servicios disponibles
- Se establece la forma de pago

### 2.3 DISEÑO DE SISTEMAS

Durante la etapa de diseño se tomarán los requerimientos detallados del análisis para realizar el diseño lógico del sistema, el cual esta formado por el diseño de la base de datos, que muestra la forma como se almacenará la información, el diseño de la interfaz de usuario, que muestra el diseño de las pantallas, y el diagrama de estructuras, que se apoya en el diseño descendente para representar gráficamente los módulos que conforman el sistema.

#### 2.3.1 Diseño de la Base De Datos

Durante el diseño de la base de datos se planea como las entidades y los atributos del sistema llegarán a ser tablas y columnas. Empezando con un diseño conceptual normalizado, representado por un modelo de entidad relación, se puede producir un diseño lógico inicial de la base de datos.

- Producir un diseño involucra varios pasos interrelacionados:
- Seleccionar las llaves primarias
- Convertir entidades en tablas
- Convertir atributos en columnas
- Convertir relaciones en llaves foráneas
- Definir atributos candidatos a índices

Otros puntos que se deben considerar en el diseño son:

- Definición de campos
- Desnormalización
- Datos calculados

El objetivo de aplicar las reglas de normalización, es obtener un modelo de la base de datos sin redundancia de información. Con esto se evita desperdiciar recursos de almacenamiento y tener tiempos de respuesta grandes. El tiempo de respuesta, es el tiempo en la cual el sistema nos devuelve la información. Si se requiere mejorar este tiempo aplicaremos la desnormalización y tendremos información redundante sólo de aquello que sirva para agilizar las consultas, es decir, tendremos una "redundancia controlada de información".



### 2.3.2 Diseño de la interfaz del usuario

La interfaz es el medio por el cual el usuario se comunicará con la computadora y está formado por pantallas. Las pantallas muestran las formas de captura y los reportes, es decir, son el medio por el cual se desplegará la información.

El objetivo principal en el diseño de una interfaz es que ayude a los usuarios y a sus empresas a obtener o introducir información al sistema, la interfaz del usuario debe contar con las siguientes características:

- Eficacia, el usuario debe tener acceso al sistema, de tal forma que sea congruente con sus necesidades particulares.
- Eficiencia, al desarrollar interfaces que mejoren la velocidad de captura de los datos y reduzcan los errores.
- Consideración del usuario, es decir, debe existir una retroalimentación para saber la manera en que progresa su trabajo.

El diseño de la interfaz del usuario está formado por el diseño de formas, pantallas y reportes. Al cumplir con los lineamientos de cada uno se estará cumpliendo con los objetivos del diseño de la interfaz del usuario.

#### 2.3.2.1 Diseño de las formas

El analista debe contar con la habilidad para diseñar de manera integral una forma que resulte útil, para lograr tal objetivo debe seguir los cuatro lineamientos para el diseño de formas:

- Diseñar formas fáciles de llenar.
- Asegurarse que las formas cumplan con el propósito para el cual fueron diseñadas.
- Diseñar formas que aseguren un llenado preciso.
- Mantener las formas atractivas

El diseño de la forma con un flujo adecuado minimiza el tiempo invertido y el esfuerzo realizado por los usuarios en el llenado. Las formas deben seguir un flujo de izquierda a derecha, y de arriba hacia abajo. Las formas no deben aparecer amontonadas, sino más bien, deben dar una apariencia de organización y lógica, aún después de llenarse.

#### 2.3.2.2 Diseño de pantallas

- Para el buen diseño de pantallas existen cuatro lineamientos:
- Mantener la pantalla sencilla
- Mantener una presentación consistente
- Facilitar el movimiento del usuario entre pantallas
- Crear pantallas atractivas

La pantalla debe mostrar sólo lo que es necesario para la acción particular que se lleva a cabo. Ésta se divide en tres secciones: la primera de ellas se llama encabezado y se localiza en la parte superior, sirve para indicar al usuario donde se encuentra dentro de la aplicación; en la sección del cuerpo, es donde el usuario realizará la captura de los datos y finalmente la sección de comentarios, donde puede ir un menú de palabras clave acerca de las funciones principales del sistema.

Las pantallas deben atraer al usuario y mantener su atención. Esto se logra con el uso de espacios abiertos que rodeen los campos de captura de datos, de tal forma que la pantalla no se vea sobrecargada. Nunca se debe saturar una forma, así como nunca se debe saturar una pantalla.



## Sistema de Información de Aeropuertos



Siempre será mejor utilizar pantallas múltiples, que amontonar todo en una sola.

Para crear pantallas atractivas podemos hacer uso de diferentes tipos y tamaños de letras, usar imágenes y colores. Existen ciertos lineamientos para el uso de imágenes, una de ellas es utilizar las imágenes típicas que los usuarios puedan interpretar fácilmente.

### 2.3.2.3 Diseño de reportes

Los reportes son la información de salida que requieren los usuarios imprimir, por tal razón, juzgan al sistema dependiendo de los resultados que ofrezca. Cuando se diseña un reporte se deben cumplir los siguientes objetivos:

- Diseñar una reporte para satisfacer el objetivo planteado
- Diseñar un reporte que se adapte al usuario
- Proveer la cantidad adecuada de información
- Asegurar que el reporte este disponible donde se necesite
- Proporcionar oportunamente el reporte

Durante la fase de análisis de determinación de los requerimientos, el analista de sistemas identifica los propósitos a satisfacer y con base en tales propósitos se diseña el reporte. Éstos deben diseñarse con base a las necesidades de los usuarios y sólo deben proporcionar la cantidad de información adecuada. Al diseñar el reporte el analista de sistemas incorpora elementos tanto funcionales como estéticos, de tal forma que el informe proporcione al usuario la información que requiere en un formato legible. Los atributos funcionales de un reporte incluyen el encabezado o título del reporte, el número de la página, la fecha del reporte, las etiquetas en las columnas, el agrupamiento de los datos relacionados y el uso de cortes de control, los cuales son separaciones entre los datos cuando aparecen cálculos de totales, por ejemplo, el sueldo promedio de empleados por área.

### 2.4 Etapa de desarrollo

Una vez terminada la etapa de diseño tendremos todos los elementos para poder desarrollar el sistema de información. El diseño de la base de datos nos ayudará para crear físicamente las tablas que forman el sistema y la relación que existe entre ellas; el diseño de la interfaz nos dará una guía para saber que pantallas componen nuestro sistema y empleando alguna herramienta de desarrollo procederemos a construirlas. Además tendremos que programar las líneas de código necesarias para hacer que los eventos que se encuentran en el sistema respondan de manera correcta con las acciones que realice el usuario.

Cabe destacar que en ésta etapa, serán necesarias la realización de las pruebas unitarias y de integración inevitables para asegurar el perfecto funcionamiento de los programas desarrollados, y de los procedimientos de usuarios, así como del plan de integración del sistema. Si fuera necesario llevar a cabo una migración de datos, es en esta etapa en donde se lleva a cabo la construcción de los componentes de migración, procedimientos de migración y carga de datos en la base de datos.

En esta etapa deberán ser desarrollados los planes y estrategias de acercamiento entre el sistema de información y los usuarios finales. Se deberá tomar nota de los resultados obtenidos en las pruebas iniciales del sistema por partes y también de manera unitaria.

El producto que se va a entregar al cliente consta de los siguientes elementos:

- Código fuente
- Procedimientos de operación y administración del sistema
- Procedimientos de seguridad y control de acceso
- Manuales de usuarios



- Condiciones de la capacitación hacia el usuario final
- Código fuente de los componentes de migración y carga de datos
- Documentación de los procedimientos de migración y carga inicial de datos
- Resultados de la evaluación de la migración y carga inicial de datos

### 2.5 IMPLANTACIÓN DEL SISTEMA

La implantación del sistema, consiste en la realización de las pruebas finales del sistema y las de aceptación por parte de los usuarios, de los procedimientos de producción, y verificación de la eficiencia de la integración para que, una vez aceptado el Sistema, se implante y pase a funcionar en un entorno real de producción. En este proceso se desarrolla un plan de mantenimiento del sistema, de tal manera que la persona encargada de darle mantenimiento al sistema esté familiarizada con este antes de que sea utilizado por los usuarios finales. Las fases que comprenden a la etapa de implantación son las siguientes:

- Plan de implantación del sistema en su totalidad
- Plan de mantenimiento previo a la liberación hacia los usuarios finales
- Evaluación del sistema de operación por parte de los usuarios finales
- Puesta en marcha del sistema de información.

### 2.6 MANEJADORES DE BASES DE DATOS

Un manejador de base de datos relacional (RDBMS, *Relational Database Management System*) o *Back-End*, es un conjunto de programas que se encargan de la administración de una o más bases de datos. El objetivo primordial de un RDBMS es crear un ambiente en que sea posible guardar y recuperar información de la base de datos en forma conveniente y eficiente.

Los manejadores de base de datos son utilizados para manejar grandes cantidades de información. El manejo de los datos incluye tanto la definición de las estructuras para el almacenamiento como los mecanismos para el manejo de la información. Además, el manejador de base de datos debe cuidar la seguridad de la información, tanto contra las caídas del sistema como contra los intentos de acceso no autorizado.

Se encarga de las siguientes actividades:

- Interacción con el manejador de archivos
- Implantación de la integridad de los datos
- Almacena, obtiene y modifica los datos
- Puesta en marcha de la seguridad
- Respaldo y recuperación
- Control de concurrencia

#### 2.6.1 Características de un manejador de base de datos relacional

- Representación de los datos por medio de tablas relacionadas
- Flexibilidad:
- Modificación de los datos
- Los cambios a la estructura de la base de datos resultan muy sencillos
- Contiene un diccionario de datos

¿Qué manejador de base de datos será conveniente utilizar? En ocasiones el cliente dirá cual, en otras tendremos que decidir cuál es el más conveniente.



## Sistema de Información de Aeropuertos



Los parámetros que influyen generalmente en la elección del manejador de base de datos son: la plataforma, el equipo que se tiene, la cantidad de información que se almacenará y el costo, en caso de adquirirse este último.

### 2.6.2 Manejadores de bases de datos comerciales

#### 2.6.2.1 Oracle

Es un manejador de base de datos relacional utilizado para misiones críticas de negocios. Actualmente por las características que posee lo hace ser el más utilizado por las empresas. Soporta las siguientes plataformas: UNIX, VMS, Windows NT, Windows 95, OS/2, NetWare, y actualmente con su nueva versión, Linux.

##### 2.6.2.1.1 Características

- Seguridad, utiliza sofisticados mecanismos de seguridad para controlar el acceso a los datos a través del manejo de privilegios.
- Realización de copias de seguridad y recuperación, la estrategia de copias de seguridad y recuperación de Oracle permite minimizar la pérdida de datos y el tiempo de caída del sistema cuando se produce un problema.
- Conectividad de carácter abierto, permite acceder hacia y desde paquetes de software de otros fabricantes mediante ODBC's y SQL Net, éste último, es su protocolo de comunicación.
- Herramientas de desarrollo, su principal herramienta de desarrollo es Developer/2000, el cual incluye un conjunto de herramientas para la creación de aplicaciones: Oracle Forms, Oracle Reports, Oracle Graphics y Procedure Builder; cuenta también con Designer/2000, el cual incluye herramientas para la reingeniería de procesos de negocios, modeladores y generadores.
- Incluye en su nueva versión la tecnología OLAP, Data Warehouse y el manejo de base de datos objeto - relacional.

#### 2.6.2.2 Sybase

Sybase es un manejador de bases de datos relacional utilizado para aplicaciones empresariales. Por su facilidad y la integración entre sus productos, lo ha hecho disputar junto con Oracle el mercado internacional. Puede ejecutarse bajo las siguientes plataformas: UNIX, Windows NT, Windows 95, etc.

##### 2.6.2.2.1 Características

Soporta conectividad con otros manejadores mediante ODBC's. Provee una alta seguridad, mediante contraseñas de entrada y el uso de privilegios. Proporciona mecanismos de respaldo y recuperación, aún cuando la base de datos esté levantada. Cuenta con productos de desarrollo para la creación de aplicaciones cliente/servidor (Power Builder) y aplicaciones en el Web mediante Jaguar de PowerSoft.

#### 2.6.2.3 Informix

Informix es otro manejador de base de datos relacional, pero no tiene tanto mercado en el ámbito mundial como Oracle y Sybase, sin embargo, es uno de los manejadores de base de datos más utilizados en organismos gubernamentales en México. Soporta las siguientes plataformas: UNIX, Windows 95, Windows NT, OS/2 y Macintosh.



### 2.6.2.3.1 Características

Multiprocesamiento simétrico diseñado para aprovechar al máximo la capacidad de arquitecturas multiprocesador y uniprocador.

Procesamiento transnacional excelente

Gran disponibilidad

Capacidad multimedia

Alta seguridad

Mecanismos de respaldo de la información en cintas y mecanismos de seguridad de la información como discos en espejo (*mirror*)

Herramientas para el monitoreo de las actividades en la de la base de datos y de los recursos del servidor.

Ofrece una gran gama de productos para el desarrollo de sistemas cliente/servidor, tales como: Informix-NewEra, producto para el desarrollo de interfaces de usuario, Informix-NewEra ViewPoint Pro, suite de herramientas gráficas de desarrollo que incluyen productos para la creación de formas, reportes, menús, etc.

Conectividad mediante Informix-Net, el cual permite la comunicación entre sus aplicaciones cliente y el servidor de base de datos, y con otros manejadores de bases de datos mediante ODBC's.

### 2.6.3.4 Microsoft SQL Server

Microsoft SQL Server es un manejador de bases de datos relacional multiusuario diseñado para soportar una gran cantidad de transacciones sobre computadoras personales, se ejecuta sólo bajo la plataforma Windows NT. El proyecto SQL Server, fue en sus inicios un desarrollo que en conjunto realizaron Microsoft y Sybase Corporation, originalmente diseñado para trabajar bajo plataforma OS/2 en el año de 1987, sin embargo, los sistemas operativos Windows empezaron a cobrar fuerza a principios de los años 90, por lo que Microsoft decidió diseñar una versión del producto para plataforma Windows para trabajo en grupo (posteriormente Windows NT). Por esta y otras razones, hubieron ciertas diferencias entre Microsoft y Sybase, por lo que Microsoft lanzó al mercado el producto Microsoft SQL Server 4.21, y Sybase su versión de SQL Server, conocida como Sybase Adaptive Server Enterprise. Todas las versiones de SQL Server de Microsoft, contenían gran cantidad de código Sybase, sin embargo, a partir de la versión 7.0, se reestructuró completamente el producto, dando como resultado, una muy buena opción dentro del mercado de los manejadores de bases de datos relacionales de medio nivel. SQL Server 2000 (8.0), representa la primera evolución de la versión 7.0, la cual obtuvo bastante aceptación en el mercado, por combinar el potencial de un motor de base de datos sumamente robusto, con un entorno de trabajo bastante amigable al usuario, a un precio muy competitivo.

#### 2.6.3.4.1 Características

- Administración de la base de datos en forma gráfica, a través de la herramienta Enterprise Manager, la cual facilita mucho las tareas administrativas.
- Ofrece un alto rendimiento sobre el sistema operativo Windows NT.
- Soporta la arquitectura cliente/servidor, es decir, divide una aplicación en dos partes: el front-end (cliente), el cual provee una interactividad alta y una interfaz fácil de usar, ya que es gráfico, y un back-end (servidor), el cual proporciona la administración de datos, información compartida, administración sofisticada, y seguridad.
- Soporta aplicaciones distribuidas.
- Alta seguridad, que se ve reforzada con la seguridad que ofrece el sistema operativo Windows NT.
- Replicación heterogénea, permite replicar la información a otros manejadores de bases de datos, tales como, Oracle y Sybase para mantener la consistencia en los datos.



## Sistema de Información de Aeropuertos



- Soporta aplicaciones Internet e Intranet, el Microsoft SQL Server Web Assistant permite a SQL Server generar automáticamente páginas HTML o rellenar plantillas HTML con datos del servidor.
- Soporta las aplicaciones Data-Warehouse
- Permite la recuperación y tolerancia de fallos.
- Alta integración con el servidor Web de Microsoft Internet Information Server (IIS)

### 2.6.3.5 MySQL.

Es un manejador de bases de datos, muy utilizado en plataforma Linux, del tipo OpenSource. No tiene costo de licencias, por lo que es uno de los manejadores de bases de datos más utilizados en software libre

#### 2.6.3.5.1 Características

- Su principal objetivo de diseño fue la velocidad de desempeño.
- Consume muy pocos recursos tanto de CPU como de memoria.
- Mayor rendimiento y velocidad tanto al conectarse al servidor, y al momento de realizar consultas sobre los datos
- Cuenta con buenas herramientas administrativas (backup, restore, recuperación de errores, etc.).
- Muy buena integración con PHP.
- No hay límites en el tamaño de los registros.
- Buen control de permisos de usuarios
- No soporta transacciones, roll-backs ni subconsultas
- No maneja integridad referencial en sus tablas

### 2.6.3.6 PostgreSQL

Al igual que MySQL es un manejador de bases de datos relacionales para Linux gratuito.

#### 2.6.3.6.1 Características:

- PostgreSQL pretende ser un RDBMS de mayor nivel que MySQL, aunque no está a la altura de Oracle o Sybase
- Por su arquitectura de diseño, escala muy bien al aumentar el número de procesadores y la cantidad de RAM.
- Soporta transacciones
- Tiene mejor soporte para *triggers* y procedimientos almacenados en el servidor.
- Soporta un subconjunto del SQL estándar mayor que el que soporta MySQL.
- Consume una cantidad considerable de recursos.
- Cuenta con menos funciones en PHP que MySQL
- Las consultas son más lentas que en MySQL
- Maneja integridad referencial en sus tablas

## 2.7 HERRAMIENTAS DE DESARROLLO

Existen en el mercado un gran número de herramientas para las distintas fases del proceso. A la hora de analizar diferentes herramientas, es necesario tener en cuenta, entre otros, parámetros como los siguientes:

- Entorno físico sobre el que funciona
- Sistema operativo sobre el que opera
- Gestores de bases de datos o fuentes a los que es aplicable
- Grado de especialización
- Profundidad del análisis que realiza



- Información adicional que aporta
- Necesidades de formación que exige a sus usuarios
- Costo económico
- En la fase de estudio habrá de solicitarse a los proveedores las oportunas demostraciones
- Técnicas que permitan evaluar la herramienta adecuadamente.
- Por último, siempre existe la posibilidad de fabricar una propia que contemple las particularidades del entorno sobre el que va a utilizarse.

Las herramientas de desarrollo o *Front-Ends*, son programas que sirven principalmente para el desarrollo de interfaces gráficas. Se pueden clasificar en cuatro categorías:

- *Add-ons* a productos ya existentes, los cuales son módulos que permiten que una aplicación de PC consulte al servidor de la base de datos, ejemplos de *add-ons* son los existentes para Dbase, Paradox, Access, Clarion, y hojas de cálculo como Lotus o Excel.
- Herramientas de desarrollo de aplicaciones, son usadas principalmente por los programadores y están diseñadas para facilitar el proceso de creación de aplicaciones *front-end*
- Reportadores, permiten hacer consultas no planeadas a la base de datos, facilitan la creación de consultas y reportes al *back-end* a los no programadores.
- Herramientas de análisis e integración de datos, que están diseñadas para que los encargados de tomar decisiones examinen los datos a partir de diferentes fuentes para así, construir cuadros de decisiones complejas.

### 2.7.1 Herramientas de desarrollo que existen en el mercado

#### 2.7.1.1 Visual Basic

Es una herramienta de desarrollo que corre sobre plataformas Windows y es una de las más utilizadas en el mercado.

##### 2.7.1.1.1 Características

- Permite desarrollar sistemas amigables y fáciles de utilizar
- Soporta la creación de aplicaciones cliente/servidor mediante el uso de ODBC's (*Open DataBase Connectivity*).
- Utiliza la programación orientada a eventos, es decir, cuando suceden ciertos eventos (mover el *mouse* o dar *click* en algún control) se realizan las acciones que se encuentran programados en éstos.
- Permite crear aplicaciones ejecutables que pueden ser distribuidos a todos los usuarios
- Permite crear aplicaciones para el Web basadas en VB Script con tecnología ASP.

#### 2.7.1.2 Microsoft Access

Además de ser un manejador de base de datos posee las herramientas necesarias para el desarrollo de aplicaciones de manera sencilla y rápida, aunque con algunas limitaciones. Trabaja sobre plataforma Windows

##### 2.7.1.2.1 Características

- Permite crear aplicaciones similares a las de VB en mucho menos tiempo



## Sistema de Información de Aeropuertos



- El generador de reportes que posee permite crear informes de rápida y sencilla
- Utiliza un lenguaje de programación llamado Access Basic muy similar al que maneja Visual Basic
- No puede generar aplicaciones ejecutables. Para hacerlo se tendrá que adquirir un producto adicional que se llama, Microsoft ADT (Access Developer Toolkit).
- Permite generar aplicaciones muy limitadas para el Web.

### 2.7.2.3 Power Builder

Es la herramienta de desarrollo nativa de Sybase utilizada para la creación de aplicaciones cliente/servidor. Permite desarrollar aplicaciones sumamente robustas en diversas plataformas

#### 2.7.2.3.1 Características

- Soporta el desarrollo de aplicaciones cliente/servidor.
- Permite crear aplicaciones ejecutables y generar los discos de instalación.
- Utiliza la programación orientada a eventos y hace uso de las características de la programación orientada a objetos: encapsulación, polimorfismo y herencia.
- Cuenta con una poderosa herramienta denominada DataWindow que le permite generar pantallas y reportes de cualquier base de datos de manera inmediata.

### 2.7.2.4 Delphi

Es una herramienta de desarrollado creado por la empresa Borland para las plataformas Windows (Windows 95, Windows 98 y Windows NT).

#### 2.7.2.4.1 Características

- Permite el desarrollo de aplicaciones cliente/servidor e Internet/Web mediante asistentes (*wizards*) que crean dichas aplicaciones de manera rápida y sencilla. Además dispone de una biblioteca de 25 componentes nativos para Internet para crear aplicaciones como lectores de noticias y servidores de correo.
- Soporte completo y alto desempeño a manejadores de bases de datos Oracle.
- Soporta múltiples manejadores de base de datos de alto desempeño en 32 bits, como SQL Link, Sybase, Informix, DB2, SQL e InterBase.
- Soporta los estándares creados por Microsoft como ActiveX.
- Optimización para el desarrollo cliente/servidor mediante Builder SQL, que construye directamente consultas complejas, además de un monitor y un explorador SQL.
- Incluye componentes para la creación de aplicaciones de información gerencial. Analiza y visualiza los datos corporativos, para la toma de mejores decisiones.

## 2.8 COMUNICACIONES Y REDES

### 2.8.1 Proceso de comunicación y concepto de red

Un proceso de comunicación permite transportar información entre dos puntos. En dicho proceso se tienen los siguientes elementos:

- Emisor: Dispositivo origen que transmite los datos
- Mensaje: Datos que son transmitidos
- Medio : Canal por el cual los datos transitan desde el origen hasta su destino
- Receptor: Dispositivo destino de los datos



## Sistema de Información de Aeropuertos



En términos muy generales, una red está conformada por un conjunto de puntos (nodos), enlazados a través de canales, por ejemplo una red telefónica. Una red de datos, está conformada por computadoras, equipos de comunicaciones y otros dispositivos que pueden compartir información entre sí a través de un medio en particular. En una red, la información debe ser entregada de manera confiable, segura, consistente y rápida. Los equipos que forman parte de la red deben ser capaces de identificarse entre sí. De esta manera, la información esté disponible en cualquier momento para cualquier usuario que esté autorizado para su consulta y manipulación.

### 2.8.2 Medios de transmisión

Los medios de transmisión, pueden ser Aéreos o Sólidos. Los Medios Aéreos están basados en señales radio-eléctricas, por lo que utilizan la atmósfera como medio de transmisión. Los medios Sólidos utilizan cables para la transmisión de datos, principalmente el cobre en cable coaxial y par trenzado así como la fibra óptica.

#### 2.8.2.1 Medios aéreos

##### 2.8.2.1.1 Ondas de radio

Las ondas de radio tienen como principales características: la facilidad para generarlas, la capacidad de viajar distancias considerablemente grandes así como la capacidad de atravesar paredes y puertas. La eficiencia de transmisión depende de factores como la frecuencia a la que son emitidas las ondas, ya que a frecuencias bajas, transmiten datos con menores interferencias, pero también con menor alcance. Una desventaja de emplear ondas de radio en exteriores, por ejemplo para una comunicación entre 2 edificios, es que factores meteorológicos como la lluvia y las tormentas eléctricas afectan considerablemente su desempeño. También afectan su desempeño otros factores como interferencias producidas por equipos eléctricos como motores.

##### 2.8.2.1.2 Enlaces con microondas

Las microondas trabajan con frecuencias superiores a los 100 MHz y viajan prácticamente en línea recta; se utilizan antenas parabólicas para su recepción. Cabe destacar que las antenas transmisoras y receptoras deben alinearse con precisión, y pueden conectarse múltiples emisores con múltiples receptores sin que éstos interfieran entre sí. Entre más separadas estén las antenas, mayor deberá ser la altura a la que deberán estar instaladas, debido a que la curvatura de la tierra puede interferir con la transmisión. Por consiguiente entre más altas estén colocadas las antenas, menos repetidores será necesario utilizar. La comunicación por microondas se utiliza ampliamente para la transmisión telefónica de larga distancia, telefonía celular, distribución de señales de televisión y enlaces de Internet a larga distancia entre otras.

##### 2.8.2.1.2 Medios Infrarrojos

Los medios infrarrojos, son útiles cuando las distancias entre emisor y receptor son muy pequeñas. Estos medios de transmisión son ampliamente usados en la comunicación de corto rango, por ejemplo, controles remotos de periféricos como ratones y teclados inalámbricos, comunicación entre computadoras portátiles (laptops con el puerto infrarrojo), etc. Su mayor inconveniente radica en la imposibilidad de atravesar objetos sólidos, como muebles, paredes y puertas.

#### 2.8.2.2 Medios Sólidos

##### 2.8.2.2.1 Cable Coaxial

El cable coaxial está conformado por un cable conductor interno cilíndrico separado de otro cable conductor externo por un aislante sólido.



Esto se recubre por otra capa aislante que es la funda del cable. Este medio físico, es más caro que el par trenzado, pero se puede utilizar a más larga distancia, con velocidades de transmisión superiores, menos interferencias, sin embargo presenta algunas desventajas, como la atenuación y la alta propensión al ruido eléctrico.

### 2.8.2.2.2 Par trenzado

El par trenzado está conformado por dos hilos de cobre aislados, trenzados entre sí y envueltos por una cubierta aislante. Los hilos están trenzados para reducir las interferencias electromagnéticas. Se pueden utilizar tanto para transmisión analógica como digital, y su ancho de banda depende de la sección de cobre utilizado y de la distancia que tenga que recorrer. Es el cableado más económico y más utilizado en las redes de computadoras.

El par trenzado está clasificado en las siguientes categorías:

- Categoría 1 Hilo telefónico trenzado de calidad de voz no adecuado para las transmisiones de datos. Velocidad de transmisión inferior a 1 Mbps.
- Categoría 2 Cable de par trenzado. Su velocidad de transmisión es de hasta de 4 Mbps.
- Categoría 3 Velocidad de transmisión de 10 Mbps. Con este tipo de cables se implementa en las redes Ethernet 10BaseT.
- Categoría 4 La velocidad de transmisión llega a 16 Mbps.
- Categoría 5 Puede transmitir datos hasta 100 Mbps.
- Categoría 5E, 6: Tienen la capacidad de transmitir datos a velocidades mayores a 100 mbps. El UTP categoría 6 tiene un rango de operación de 250 MHz a diferencia de los categoría 5 que operan a 100 MHz.

Existen dos tipos de pares trenzados: STP (*Shielded Twisted-Pair*) y UTP (*Unshielded Twisted-Pair*); el UTP es mejor aunque su costo es más elevado en comparación con el STP.

### 2.8.2.2.3 Fibra óptica

Se trata de un medio muy flexible que conduce información por medios ópticos: Los rayos de luz inciden con una gama de ángulos diferentes posibles en el núcleo del cable, dichos rayos irán brincando a lo largo de la fibra óptica hasta llegar a su destino. Su forma es cilíndrica con tres secciones radiales: núcleo, revestimiento y cubierta. El núcleo está conformado por varios hilos muy finos de cristal o plástico. Cada hilo está rodeado por su propio revestimiento que es un cristal o plástico con diferentes propiedades ópticas distintas a las del núcleo. La cubierta está constituida de material plástico que aísla y protege al núcleo contra presiones externas, humedad, etc. Ofrece muchas ventajas en comparación con los cables de cobre: Permite un mayor ancho de banda, su tamaño y peso son reducidos, presenta menor atenuación, no sufre de interferencias electromagnéticas, la distancia entre los repetidores es mayor, su rango de frecuencias es mucho más amplio.

### 2.8.3 Tipos de transmisión

La transmisión de los datos puede ser en serie o en paralelo. En una transmisión en serie los bits de datos se transmiten de uno a uno sobre un canal único. En una transmisión en paralelo: los bits se transmiten en grupo sobre varios canales al mismo tiempo. Es más rápida que la transmisión en serie pero en la medida que la distancia entre equipos se incrementa, suele presentar serios problemas debido al ruido generado dentro del medio.



Se tienen los siguientes tipos de transmisión:

- Transmisión simplex: La transmisión de los datos se produce en un mismo sentido, es decir, la parte emisora y la parte receptora siempre serán las mismas.
- Transmisión half-duplex: La transmisión de los datos se produce en ambos sentidos pero alternativamente, es decir, sólo se transmite en un sólo sentido a la vez. Mientras se estén recibiendo datos en un nodo, éste no puede transmitir hasta que se concluya la recepción de los mismos.
- Transmisión full-duplex: la transmisión de los datos se produce en ambos sentidos al mismo tiempo.

La transmisión de datos puede ser asíncrona y síncrona. En una transmisión asíncrona cada *byte* de datos incluye señales de inicio y fin al principio y al final de dicho *byte*. El objetivo de estas señales es el de avisar al receptor que se está enviando un dato, y darle suficiente tiempo al receptor de realizar funciones de sincronismo antes de que llegue el siguiente *byte*. En la transmisión síncrona se utilizan canales separados de reloj que administran la recepción y transmisión de los datos.

### 2.8.4 Clasificación de las redes

Las redes se clasifican por tecnología de transmisión y por escala o tamaño.

#### 2.8.4.1 Clasificación por tecnología de transmisión

Existen dos tipos de redes, las llamadas redes de difusión (broadcasting) que constan de un sólo canal o medio de comunicación que es compartido por todos los dispositivos de la red. El otro tipo está conformado por las redes de punto-a-punto (point to point), que consisten en múltiples conexiones entre pares individuales de computadoras.

#### 2.8.4.2 Clasificación por escala

De acuerdo a la escala de las redes, éstas se clasifican en:

- LAN(*Local Area Network*)
- MAN(*Metropolitan Area Network*)
- WAN(*Wide Area Network*)

##### 2.8.4.2.1 Redes LAN

Las redes LAN, como su nombre lo indica, están limitadas a un área geográfica relativamente pequeña, como podría ser un edificio individual, una planta o un campus en una universidad. El objetivo de este tipo de red es intercambiar información entre grupos de trabajo y compartir recursos tales como impresoras, discos duros y otros recursos de la red.

Se caracterizan por tres factores:

- Extensión (de unos cuantos metros hasta algunos kilómetros)
- Tecnología de transmisión(cable de par trenzado UTP o coaxial, fibra óptica, infrarrojo, radio, microondas en frecuencias no comerciales)
- Topología (anillo, estrella, bus).

Las velocidades en las LAN van desde los 10 mbps hasta los 1000 mbps. Los estándares más comunes son el IEEE 802.3 llamado Ethernet y el IEEE 802.5 llamado Token Ring. Ethernet opera entre 10 y 1000 mbps.



### 2.8.4.2.1.1 Token Ring

Este tipo de red es conocida como Anillo; en este estándar, todos los nodos pueden recibir todos los paquetes que circulan por la red, obtienen una copia y examinan el destinatario del paquete. Si el destinatario es el nodo mismo, lo procesa y si no lo descarta para recibir el siguiente paquete. Para enviar un paquete examina el medio para determinar si está libre; de ser así procede a enviar el dato. Si ocurre que dos nodos enviaron un paquete al mismo tiempo, se provoca una colisión y cada nodo vuelve a retransmitir su paquete después de esperar un tiempo aleatorio. Token ring opera entre 4 y 16 mbps y utiliza un token, que permite al nodo que lo posee, enviar paquetes a la red mientras los otros examinan el medio. Una vez que un nodo termina de enviar paquetes, pasa el token a otro nodo para que este transmita. El anillo es fácilmente expandido para conectar más nodos, aunque en este proceso interrumpe la operación de la red mientras se instala el nuevo nodo. Así también, el movimiento físico de un nodo requiere de dos pasos separados: desconectar para remover el nodo y otra vez reinstalar el nodo en su nuevo lugar.

### 2.8.4.2.1.2 Estrella

Todos los elementos de la red se encuentran conectados directamente mediante un enlace punto a punto al nodo central de la red, quien se encarga de gestionar las transmisiones de información por toda la estrella. Evidentemente, todas las tramas de información que circulen por la red deben pasar por el nodo principal, con lo cual un fallo dicho nodo provoca la caída de todo el sistema. Por otra parte, un fallo en un determinado cable sólo afecta al nodo asociado a él. Si bien esta topología obliga a disponer de un cable propio para cada terminal adicional de la red, es una buena elección siempre que se tenga varias unidades dependientes de un procesador, esta es la situación de típica de un *mainframe*.

En esta topología, los elementos que constituyen la red se disponen linealmente, es decir, en serie y conectados por medio de un cable: el bus. Las tramas de información emitidas por un nodo (terminal o servidor) se propagan por todo el bus en ambas direcciones, alcanzado a todos los demás nodos. Cada nodo de la red se debe encargar de reconocer la información que recorre el bus, para así determinar cual es la que le corresponde, la destinada a él. Es el tipo de instalación más sencillo y un fallo en un nodo no provoca la caída del sistema de la red. Por otra parte, una ruptura del bus es difícil de localizar (dependiendo de la longitud del cable y el número de terminales conectados a él) y provoca la inutilidad de todo el sistema, sin embargo existen dispositivos que permiten detectar errores en éste tipo de redes. El bus es la parte básica para la construcción de redes Ethernet.

### 2.8.4.2.2 Redes MAN

Este tipo de red corresponde es una versión más amplia de una red LAN, ya que puede abarcar un grupo de oficinas distantes o una ciudad, con una longitud máxima aproximada de 160 km y una velocidad aproximada de 155 mbps. Este tipo de red dispone de un estándar conocido como DQDB (*Distributed Queue Dual Bus*) IEEE 802.6. Esta red utiliza cables *half duplex*, por medio de los cuales se transmiten y reciben los datos.

### 2.8.4.2.3 Redes WAN

Estas redes se expanden en una gran zona geográfica, por ejemplo, un país o incluso un continente. Tiene una mayor tasa de errores que las redes LAN y MAN. El costo de implementación es elevado y los enlaces son de baja velocidad. Se suelen implementar con enlaces punto a punto. A la infraestructura que une los nodos de usuarios se le llama subred y abarca diversos dispositivos llamados ruteadores, así como líneas de comunicación que enlazan las diversas redes.



En la mayoría de las WAN se utilizan una gran variedad de medios de transmisión para cubrir grandes distancias. La transmisión puede efectuarse por microondas, por cable de cobre, fibra óptica o alguna combinación de los anteriores.

### 2.8.5 El modelo OSI

El modelo OSI fue definido por la Organización Internacional para la Estandarización, conocida como ISO (*Internacional Organization for Standardization*). Este modelo OSI describe la manera en la que los datos viajan desde una aplicación en un punto de la red, hacia otro punto de la red. Este modelo contempla 7 diferentes capas: Capa Física, Capa de enlace, Capa de Red, Capa de Transporte, Capa de Sesión, Capa de Presentación y Capa de Aplicación.

#### 2.8.5.1 Capa física

La capa física está directamente relacionada con el envío de bits en el medio físico de transmisión, asegurando que la transmisión esté libre de errores. También especifica si el medio permite la comunicación *simplex*, *half duplex* o *full duplex*.

#### 2.8.5.2 Capa de enlace

Esta capa contiene la estructura lógica de agrupación de bits, llamada trama (*frame*), con los datos provenientes de la capa anterior en la computadora origen. En la computadora destino, se encarga de agrupar los bits provenientes del nivel físico en tramas de datos que serán entregadas al nivel de red. Esta capa es la responsable de garantizar la transferencia de tramas libres de errores de una computadora otra a través de la capa física.

#### 2.8.5.3 Capa de red

Se encarga de controlar la operación de la subred. Su tarea principal es definir el origen y el destino de los paquetes. Otra función importante en este nivel es la resolución de cuellos de botella. En estos casos se pueden tener varias rutas para dar salida a los paquetes, ya que en base a algunos parámetros de eficiencia y disponibilidad se pueden elegir rutas dinámicas de salida.

#### 2.8.5.4 Capa de transporte

Permite tomar los datos de la capa de sesión y verificar que dichos datos lleguen a su destino. En ocasiones los datos que vienen de la capa de sesión exceden el tamaño máximo de transmisión (*Maximum Transmission Unit MTU*) de la interfaz de red, por lo cual es necesario fraccionarlos y enviarlos en unidades más pequeñas, lo que origina la fragmentación y ensamblado de paquetes cuyo control se realiza en esta capa.

#### 2.8.5.5 Capa de sesión

Esta capa establece, administra y analiza las sesiones de comunicación entre las entidades de la capa de presentación. Las sesiones de comunicación constan de solicitudes y respuestas de servicio que se presentan entre aplicaciones ubicadas en diferentes dispositivos de red. Otro servicio de este nivel es la sincronización y el establecimiento de puntos de chequeo de manera que si al momento de estar transmitiendo los datos la conexión se rompe, más tarde se pueda reiniciar a partir del punto de chequeo, lo cual ahorrará tiempo y permitirá la terminación de la transferencia de manera más rápida, ya que no se tiene que iniciar desde el principio la transmisión.



### 2.8.5.6 Capa de presentación

La capa de presentación provee servicios que permiten transmitir datos con algún formato propio para las aplicaciones o para el nodo en el que se está trabajando. Como existen computadoras que interpretan sus bytes de una manera diferente que otras, es en esta capa en donde es posible convertir los datos a un formato independiente de los nodos que intervienen en la transmisión. Otra tarea muy importante en esta capa es el manejo de cuestiones de seguridad, tal como la encriptación de datos.

### 2.8.5.7 Capa de aplicación

En esta capa se encuentran aplicaciones de red que permiten explotar los recursos de otros nodos. Dicha explotación se hace a través de emulación de terminales que trabajan en un nodo remoto, interpretando una gran variedad de secuencias de caracteres de control que permiten desplegar en la terminal local los resultados.

### 2.8.6 Protocolos

Un protocolo de red es un conjunto estandarizado de reglas que permiten la comunicación entre computadoras. Dentro del protocolo se especifica como las computadoras se identifican unas a otras sobre una red, la manera en la que los datos se deben procesar para ser transmitidos, la manera en la que los datos deben ser procesados una vez que llegaron a su destino, los procedimientos necesarios para la detección de errores en los datos, así como las medidas y acciones a tomar cuando se presenten dichos errores. Los datos son enviados a través de paquetes. Un paquete es una fracción del mensaje (datos enviados) de tamaño predefinido, en donde cada fracción o paquete contiene información de procedencia y de destino, así como información requerida para el reensamblado del mensaje. Los protocolos también definen los procedimientos para el manejo de transmisiones o "paquetes" dañados o perdidos totalmente.

Los principales protocolos de red son:

- IPX (para Novell NetWare)
- TCP/IP (para UNIX, Windows NT, Windows 95 y otras plataformas)
- DECnet (para redes de computadoras DEC de Digital Equipment Corp.)
- AppleTalk (para computadoras Macintosh)
- NetBIOS/NetBEUI (para redes LAN Manager y Windows NT)

Aunque cada protocolo de red es diferente, todos ellos son capaces de compartir el mismo cableado físico. Este método común de acceso a la red física permite a múltiples protocolos coexistir en el medio de red. Este concepto es conocido como independencia del protocolo (protocol independence) lo cual significa que los dispositivos son compatibles en las capas o niveles físico (physical layer) y de vínculo de datos, (data link layer) permitiéndole al usuario trabajar con protocolos diferentes sobre el mismo medio.

#### 2.8.6.1 Grupos de protocolos

Hay 4 grandes grupos de protocolos:

- Protocolos de gestión del sistema: todas las tareas relacionadas con el sistema de interconexión de los equipos en la red, por ejemplo, el protocolo TCP/IP
- Protocolos de gestión de la aplicación: procesos que controlan la asignación de recursos del equipo, bloqueos de la computadora, accesos no autorizados, etc.
- Protocolos de sistema: tareas del sistema operativo, tales como los accesos a discos y archivos, manejo de memoria, ejecución de tareas, etc.



- Protocolos específicos de aplicaciones: dependen del tipo de aplicación con la que se vaya a trabajar.

### 2.8.6.2 TCP/IP (Transmission Control Protocol/Internet Protocol)

El departamento de defensa de los Estados Unidos definió un conjunto de reglas que establecieron la manera de conectar computadoras entre sí para permitir el intercambio de información. Fue así como se concretó el conjunto de protocolos TCP/IP. Para los años 80 una gran cantidad de instituciones estaban interesadas en conectarse a esta red que se expandió por los Estados Unidos.

El TCP/IP es un conjunto de protocolos, que proporcionan una serie de reglas para la comunicación entre computadoras. Es una forma de resolver la cuestión de comunicación entre nodos que pertenecen a redes con distintos niveles lógicos y de enlace, por ejemplo, entre un nodo de una red Ethernet y otro nodo de una red de tipo Token Ring. Son los protocolos de interconexión de redes más utilizados en la actualidad y son la base de Internet.

#### 2.8.6.2.1 TCP Transmission Control Protocol

El protocolo TCP está formado por dos protocolos, el TCP y el UDP (User Datagram Protocol). El TCP permite establecer un circuito virtual entre dos nodos (protocolo orientado a conexión), mientras que el UDP permite el envío de un segmento entre nodos sin comprobar que el envío ha sido satisfactorio (protocolo no orientado a conexión).

Las funciones del protocolo TCP son:

- Segmentación de los mensajes: TCP decide cómo fraccionar el mensaje que le llega de las capas superiores y cómo formar el segmento.
- Etiquetado de procedencia: TCP etiqueta cada segmento indicando el lugar que ocupa en un mensaje completo.
- Control de flujo: TCP permite la eliminación de duplicados y la petición de reenvío de los segmentos que falten.
- Indicación de los datos urgentes: es posible indicar si los datos que porta el segmento son urgentes o no.
- Confirmación de la llegada de datos: permite confirmar la recepción de los segmentos en el destino.
- Direccionamiento de los servicios de red: permite orientar mensajes a un servicio de nivel superior u otro mediante el empleo de puertos. De esta manera, es posible distinguir servicios distintos recibiendo y enviando mensajes dentro de un mismo nodo.

#### 2.8.6.1.2 Internet Protocol IP

IP actúa en el nivel 3 del modelo OSI y permite el intercambio de mensajes entre redes que no compartan los mismos niveles físicos y de enlace de datos (niveles 1 y 2 del modelo OSI). Al conjunto de datos transmitidos por IP se le denomina Datagrama IP. Cada Datagrama IP posee información del destino, de forma que puede llegar automáticamente al nodo de destino. El mecanismo de conseguirlo es dotar a cada nodo de una dirección IP única.

En el esquema de IP existen dos tipos de dispositivos (nodos):

- Host: nodos pertenecientes a una subred
- Router: nodos que enlazan dos subredes distintas



### 2.8.6.1.3 Dirección IP

Consiste en una cadena de 32 bits que normalmente se expresa en forma decimal, en cuatro grupos de tres dígitos separados por puntos. Esos dígitos irán de 0 a 255.

### 2.8.6.1.4 Máscara de subred

La máscara de subred es un conjunto de 32 bits que permiten separar de una dirección IP la parte del identificador de la subred de la parte del nodo.

### 2.8.6.1.6 Servidores Proxy.

Un único nodo (el servidor proxy) tiene una dirección IP "real" y los demás utilizan direcciones falsas.

### 2.8.6.1.7 DNS

Para comunicarse con un nodo mediante IP es necesario conocer su dirección IP, que al ser un número no es fácil de recordar.

Por ello surgió un proyecto denominado DNS (*Domain Name System* o Sistema de Nombres de Dominio), que establece una relación entre una dirección IP y un nombre de *host*, haciéndolo más fácil de recordar. La relación entre IP y nombre de host está almacenada en unos nodos denominados servidores DNS. De esta forma, si se quiere utilizar el sistema de nombres de dominio es necesario conocer el nombre del *host* y la dirección IP de algún servidor de DNS que nos pueda proporcionar la relación entre direcciones IP y nombres de dominio. Otra ventaja de la DNS es que es independiente de la subred, es decir, si cambiamos un nodo de una subred a otra, en general deberemos cambiar su dirección IP, mientras que no será necesario cambiar su nombre de dominio.

#### 2.8.6.1.7.1 Servidores DNS

En los primeros tiempos de Internet, los servidores DNS mantenían en un archivo denominado "hosts.txt" las equivalencias de todos los nodos de Internet y era actualizado por el NIC (*Network Information Center*), pero debido al gran crecimiento de la red de Internet, el mantenimiento de una lista centralizada se hizo imposible y se introdujo el sistema DNS jerárquico, usado hoy en día. Actualmente cada servidor DNS gestiona y actualiza los nombres de *host* de un dominio o subconjunto de nodos de Internet que son administrados por el organismo denominado NIC. De esta forma, cuando se conecta un nuevo nodo a Internet, su nombre de host es dado de alta en el servidor DNS del dominio al que corresponda.

### 2.8.7 Ethernet

Ethernet es la más popular tecnología de soporte físico de LAN en uso en la actualidad. Otros tipos de soporte físico de LAN incluyen Token Ring, Fast Ethernet, Interfaz de Datos Distribuida por Fibra (óptica) (*Fiber Distributed Data Interface* o "FDDI"), Modo de Transferencia Asíncrono (*Asynchronous Transfer Mode* o "ATM") y *LocalTalk*. Ethernet es popular porque logra un buen balance entre velocidad, costo y facilidad de instalación. Estos puntos fuertes, combinados con una amplia aceptación en el mercado informático y la habilidad para soportar virtualmente todos los protocolos populares de red, hacen de Ethernet una tecnología de red ideal para la mayoría de los usuarios de computadoras hoy en día. El estándar Ethernet está definido por el Instituto de Ingenieros Eléctricos y Electrónicos (*Institute of Electrical and Electronic Engineers IEEE*) como el



estándar IEEE 802.3. Este estándar define las reglas para configurar una Ethernet y especifica a su vez como interactúan entre sí los elementos de una red Ethernet.

### 2.8.8 Fast Ethernet

Para redes Ethernet que requieren altas velocidades de transmisión, fue establecido el estándar Fast Ethernet (IEEE 802.3u). Este estándar eleva el límite de velocidad de transmisión de 10 Megabits por segundo (Mbps) a 100 Mbps con sólo mínimos cambios en los cableados existentes. Hay tres tipos de Fast Ethernet:

- 100BASE-TX: para el uso con cableados de par trenzado sin malla (Unshielded Twisted Pair o "UTP") nivel 5.
- 100BASE-FX: para el uso con cables de fibra óptica.
- 100BASE-T4: el cual utiliza un par extra de hilos para utilizar cableado existente tipo UTP nivel 3.

El estándar 100BASE-TX se ha vuelto el más popular debido a su gran compatibilidad con el estándar Ethernet 10BASE-T.

### 2.8.9 Gigabit Ethernet

Ethernet de 1.0 Gigabit es una tecnología futura que promete una migración más allá de Fast Ethernet de modo tal que las generaciones futuras de redes soportarán aún más altas velocidades de transferencia de datos. Las redes Fast Ethernet se extendieron con una gran rapidez, como consecuencia de esto los precios bajaron y su uso se popularizó hasta el punto de que se utiliza Fast Ethernet incluso en la conexión del usuario final. Debido a la popularización masiva de las redes Fast Ethernet, el comité IEEE 802.3 inició en 1998 el desarrollo de un nuevo tipo de red que incrementara la velocidad de nuevo en un factor diez, creando Gigabit Ethernet, que el 29 de junio de 1998 produjo la aprobación del suplemento 802.3z. De forma análoga a lo hecho con Fast Ethernet, se pretendía poder utilizar los mismos medios físicos que en Fiber Channel: emisores laser con fibra óptica multimodo y monomodo, cable UTP categoría 5 y 6. Se puede comentar también que siguiendo con la tradición ya establecida de aumentar cada vez la velocidad en un factor diez, el IEEE aprobó en enero del 2000 la creación de un grupo de estudio de alta velocidad para la eventual estandarización de una Ethernet de 10 Gigabits. Las decisiones sobre como se implementará el nivel físico de esta red se encuentran todavía en fases muy preliminares.

### 2.8.10 Principales dispositivos que conforman una red

#### 2.8.10.1 Tarjetas de red

Tarjeta electrónica que conecta a las computadoras a la red. Normalmente se inserta en una de las ranuras de expansión de la tarjeta madre de la computadora (*motherboard*), aunque las tarjetas madre más recientes ya las traen integradas

La tarjeta de red tiene las siguientes funciones:

- Formar los paquetes de datos
- Dar acceso al cable, con la conversión eléctrica y ajuste de velocidad
- Desempeñar el papel de transmisor y receptor en la computadora
- Analizar las tramas para detectar errores
- Identificar físicamente a la terminal o nodo



### 2.8.10.2 Estaciones de trabajo

Una estación de trabajo es cada una de las computadoras que están conectadas a la red a través de las cuales se puede acceder a los recursos compartidos en dicha red como discos, impresoras,

unidades de disco, módems, etc. Pueden carecer de la mayoría de los periféricos pero siempre tendrán una tarjeta de red, un monitor, un teclado y un CPU.

### 2.8.10.3 Servidores

Son las computadoras que proporcionan servicios a las estaciones de trabajo de la red tales como almacenamiento en discos, acceso a las impresoras, unidades para respaldo de archivos, acceso a otras redes o computadores centrales, etc.

### 2.8.10.4 Repetidores

Son dispositivos que generan la señal de un segmento de cable y pasan estas señales a otro segmento de cable sin variar el contenido de la señal. Son utilizados para incrementar la longitud entre conexiones en una LAN.

### 2.8.10.5 Puentes (*Bridges*)

Es un equipo que contiene dos puertos de comunicación y que crea tablas en memoria que contienen todas las direcciones de las tarjetas en la red, de ambos extremos, de tal manera que restringen el tráfico de datos de un segmento a otro, no permitiendo el paso de tramas que tengan como destino una dirección del mismo segmento al que pertenece la estación de origen.

Es conveniente el uso de los mismos cuando se requiere llevar a cabo la interconexión de dos redes LAN locales o remotas.

### 2.8.10.6 Ruteadores (*Routers*)

Son dispositivos que permiten unir varias redes (más de dos, a diferencia de los puentes), tomando como referencia la dirección de red de cada segmento. Al igual que los puentes, los ruteadores restringen el tráfico local de la red permitiendo el flujo de datos a través de ellos solamente cuando los datos son direccionados con esa intención.

### 2.8.10.7 *Brouters*

Son dispositivos con funciones combinadas de *bridge* y *router*. Cuando se configura se le indica la modalidad en la cual va a funcionar, como *bridge* o como *router*.

### 2.8.10.8 MAU (*Multistation Access Unit*)

Es un concentrador utilizado en redes tipo anillo, en donde se pueden insertar hasta 8 estaciones. El MAU detecta señales procedentes de las estaciones de trabajo, así como errores, evitando pérdidas de datos y del Token.

### 2.8.10.9 Concentradores (*Hubs*)

Concentradores de cableado para las redes Ethernet integrados por microprocesadores, memoria y protocolos, características que lo convierten en un nodo inteligente en la red capaz de controlar y diagnosticar el tráfico de información en la red, incluso por monitoreo remoto.



### 2.8.10.10 Switching Hub o Switch Ethernet

Divide la LAN en varios segmentos limitando el tráfico a uno o más segmentos en vez de permitir la difusión de los paquetes por todos los puertos.

### 2.9 INTERNET

Internet es una red mundial de computadoras, conformada por miles de redes conectadas entre sí, permitiendo que estas compartan recursos. Esto implica que desde cualquier parte del mundo es posible el intercambio de información con algún otro lugar, sin importar la distancia que exista. El origen de Internet está en la red conocida como ARPANet (Advanced Projects Agency Net).

Esta red fue implementada por el Ministerio de Defensa de Estados Unidos a finales de los años 60 con la idea de tener una sola red a lo largo del país que pudiera compartir información militar estratégica. En los años 60 se estableció una red interestatal, de modo que toda la defensa del país dependiera de la misma red y compartiera los recursos de ésta. ARPANet debía estar funcionando en el caso de que un desastre natural o una guerra, ocurrieran, de modo que aunque una parte de esta red resultara dañada, se pudiera seguir transmitiendo información.

Otra característica de esta red era la escalabilidad, ya que la incorporación de nuevos elementos no representaba ningún problema mayor.

Esta red empleaba protocolos que eran interpretados por cualquier computadora que estuviera conectada a la red, independientemente de su plataforma. La información que se enviaba a través de ARPANet era por paquetes, por medio del protocolo TCP/IP. A mediados de los años 80, esta red fue conformada también por computadoras conectadas en universidades de los Estados Unidos, lo que permitiría posteriormente integrar cualquier computadora a esta red, dentro y fuera de los Estados Unidos; en 1990 desapareció ARPANet para dar paso junto a otras redes TCP/IP a Internet.

Internet enlaza a muchas redes, incluyendo como más importantes la que proporciona acceso a los grupos de noticias (Usenet), que data de 1979 y la World Wide Web (WWW), diseñada por Tim Berners Lee a principios de los 90. Actualmente hay varios miles de redes de todos los tamaños conectadas a Internet, pudiendo compartir datos en base al protocolo TCP/IP.

#### 2.9.1 Direcciones URL

Todos los recursos en Internet tienen asignados una *dirección URL (Uniform Resource Locators, localizador uniforme de recursos)*. La información que contiene un URL hace posible navegar de un lugar de la red a otro haciendo clic sobre el link sin necesidad de saber el nombre físico del documento ni su ubicación. Un URL consiste en cuatro partes separadas que cuando son combinadas, definen la ubicación del recurso en Internet. Estas partes son: Protocol, Domain, Path y Filename.

**Domain:** (nombre del dominio o servidor) Indica la dirección del servidor en donde se encuentran los recursos. Dentro del nombre del dominio, existe un sufijo de tres letras indicando el tipo de página a la que se está haciendo referencia.

com = comercial  
edu = educacional  
mil = militar  
gov = gubernamental



gob=gubernamental  
net = network  
org = organizaciones no comerciales

Fuera de Estados Unidos es necesario agregar un sufijo de 2 letras que indica el país de origen, ejemplo:

mx = México  
jp = Japón  
ca = Canadá  
fr=francia  
uk = Reino Unido

Path: Indica la ruta dentro del servidor de la ubicación de un archivo.

Filename: Indica el nombre del archivo al que se hace referencia.

Por ejemplo, la dirección URL

<http://www.aicm.com.mx/Principal/Servicios/InfoTuristica/InfoTuristica.htm>

Hace referencia a un documento de hipertexto con información turística del Aeropuerto Internacional de la Ciudad de México, alojada dentro del subdirectorío InfoTurística del Subdirectorío Servicios del directorío Principal en el servidor del Aeropuerto.

### 2.9.2 Servicios de Internet

- **http:** Protocolo de Transferencia de Hipertexto (*HiperText Tranfer Protocol*), es el protocolo de intercambio de información en Internet. El hipertexto permite desplegar información en un documento, incluyendo texto, gráficos, ilustraciones, sonido, video y cualquier otro medio requerido. Los documentos están vinculados a otros documentos por una serie de hipervínculos (*hyperlinks*) llamados simplemente *links* ó ligas. Un *link* aparece como una frase o imagen dentro de una página que se destaca del resto del texto y de las imágenes. Un *link* podría ser una palabra subrayada, en otro color, otra fuente, o un icono. El usuario podría *navegar* por los documentos de Internet seleccionando estos *links*.
- **Gopher:** El nombre Gopher proviene de un tipo de ardilla, mascota de la Universidad de Minnesota, en donde se creó esta herramienta. El gopher es una herramienta que permite navegar en la red a través de menús en ambiente de comandos. Esta herramienta también Incluso permite buscar archivos en un servidor Archie FTP. Actualmente no es muy utilizado, debido a que las interfaces graficas de la red lo han desplazado.
- **ftp:** Protocolo de Transferencia de Archivos (*File Transfer Protocol*) Esta herramienta permite tener acceso a archivos de una computadora remota y traerlos a nuestra computadora, y viceversa, subir archivos a un servidor remoto desde nuestra computadora. Los comandos son muy similares a los del sistema operativo: ls, cd, get, put, etc.
- **News:** Grupos de Noticias: Es un servicio que aporta USENet a la red. Son grupos de discusión internacionales. Con métodos similares al envío de un correo electrónico, permite situar un mensaje en un grupo de discusión determinado, y que pueda ser leído por cualquier persona que acceda a ese grupo.



## Sistema de Información de Aeropuertos



- telnet: Servicio de conexión remota: permite que el usuario consulte información en una computadora localizada en otra ubicación sin poder bajar los archivos. Típicamente se utiliza para consultas de documentos en bibliotecas.
- e-mail: El e-mail o correo electrónico permite el intercambio de mensajes e incluso de archivos, de manera análoga al correo postal. Cada dirección de correo es única en el mundo, tal y como ocurre en el correo postal. Una dirección de correo incluye dentro de su cadena de texto el nombre del destinatario seguido de una arroba y seguido de la dirección del servidor de correo `username@host.domain`

Actualmente se puede tener acceso a todos los servicios de Internet a través de un navegador o *browser*, siendo los más populares el Netscape Navigator y el Internet Explorer en el ámbito de las computadoras personales.

### 2.10 PLATAFORMA CLIENTE/SERVIDOR

La tecnología Cliente/Servidor puede definirse como un conjunto de elementos de software como de hardware, entre los cuales se destacan tres componentes: El cliente, el servidor y la red. El servidor es el encargado de procesar los datos requeridos por el cliente, quien es el encargado de recibir el resultado del proceso; estos dos elementos son unidos por medio de una red.

#### 2.10.1 El Cliente

Es la computadora o el conjunto de computadoras encargadas de interactuar directamente con el usuario final. Mediante el cliente, el usuario realiza el acceso a la información sin importar el lugar en donde se encuentre.

El cliente maneja la presentación de los datos, realiza la captura y la validación de los mismos, genera consultas ejecuta operaciones y recibe información procedente del servidor o de otro cliente. Por lo tanto el cliente debe contar con una interfaz grafica de usuario amigable para el usuario final.

#### 2.10.2 El Servidor

El servidor es el encargado de satisfacer los requerimientos del cliente. Procesa las consultas del cliente, envía, recibe y almacena información, provee seguridad y control de acceso. Existen varias clases de servidores: de datos, de correo electrónico, de imágenes, de impresión, de Internet, entre otros. Los servidores deben contar con elementos que gestionen los datos, esto se lleva a cabo mediante un DBMS. Entre los servidores se puede contar con máquinas como AlphaServer, AS400, RS/6000, servidores PC's entre otros. Y entre los sistemas operativos de los servidores: Windows NT/2000 Server, OpenVMS, AIX, SunOS/Solaris, Unix, Linux, etc.

#### 2.10.3 La Red

La red es el elemento encargado de realizar la transmisión de los datos entre el cliente y el servidor, y viceversa. También controla la transmisión de datos entre los diferentes servidores que conformen el ambiente. La red debe proveer mucha seguridad, de tal forma como si estuviera trabajando de una manera local. Necesita de un propio sistema operativo para así ofrecer varios niveles de seguridad, tolerancia a fallas, alto rendimiento, herramientas de administración y sobre todo transparencia al usuario final.



### 2.10.4 Ventajas de la plataforma cliente servidor

- **Control Centralizado:** el servidor tiene el control sobre todos los clientes de la red. Un cliente podrá hacer las veces de servidor en el momento que se requiera.
- **sistemas Abiertos:** soportan múltiples ambientes, múltiples plataformas, múltiples manejadores de bases de datos. Permitiendo en la mayoría de los casos la adquisición de hardware y software sin problemas de compatibilidad.
- **Flexibilidad y Escalabilidad:** permite reemplazar, ampliar o agregar componentes sin necesidad de realizar grandes cambios a la aplicación, facultando a las empresas de una posición ventajosa en el mercado frente a sus competidores.
- **Incremento de la Productividad:** con las plataformas amigables, los usuarios podrán emplear menos tiempo en la realización de las tareas que antes eran muy laboriosas. La disponibilidad de nuevas tecnologías de programación (P.O.O, SQL, Herramientas Case, etc.) permite la implantación de Aplicaciones de Desarrollo Rápido (RAD), contribuyendo esto a una pronta satisfacción de las nuevas necesidades de la empresa. También facilita la toma de decisiones, ya que la información es oportuna y se encuentra disponible.
- **Optimización de Tráfico:** la Red se descongestiona por que la manipulación de los datos ocurre en el Cliente y en el Servidor, dependiendo de cuál sea la forma más efectiva para cada tarea. La base de datos se encuentra distribuida entre el Cliente y el Servidor. La red dedica mayor tiempo a transportar los resultados y no a la realización de consultas.

### 2.11 SEGURIDAD

Establecer un esquema de seguridad 100% libre de errores y de pérdidas es una tarea prácticamente imposible. El concepto Seguridad es un concepto asociado a la certeza, falta de riesgo o contingencia. Siempre existirán elementos de riesgo en el manejo de sistemas de información, sin embargo es aunque no es posible implementar sistemas absolutamente seguros, se puede hablar de niveles de seguridad. La seguridad informática está encaminada a obtener los más altos niveles de seguridad posibles.

Todos los componentes de un sistema de información (hardware, software, datos) están expuestos a un ataque, sin embargo, en los datos y la información se centra las políticas de seguridad establecidas en cualquier negocio. Se deben establecer políticas generales y controles que prevengan o detecten cualquier intento de acceso no autorizado a un sistema de cómputo.

Para evitar los riesgos en la seguridad de la información, se deberá considerar lo siguiente:

- Proteger los datos de cualquier uso indebido
- Definir y asignar niveles de control de acceso
- Monitorear los sistemas para detectar cualquier uso impropio

#### 2.11.1 Lineamientos de seguridad

Las normas y lineamientos que se deberán seguir para una buena seguridad en los sistemas de información, y así evitar la mala manipulación de los datos por las personas no autorizadas en el manejo de éstos, se listan a continuación.

##### 2.11.1.1 Privacidad de la Información

La información puede ser recopilada de varias fuentes pero ésta deberá usarse únicamente para los fines que la dependencia estipule y no dejar que usuarios ajenos tengan acceso a estas.



### 2.11.1.2 Integridad de la Información

Los valores en un banco de datos se pueden volver inválidos de varias maneras. Probablemente la causa más común es una entrada de datos o una actualización en la cual algunos valores son especificados incorrectamente, tanto de manera intencional como involuntaria. Por ello cuando se lleve a cabo la actualización en un dato, ésta actualización deberá llevarse a cabo en todos los datos relacionados con él.

### 2.11.1.3 Confiabilidad del personal

El mayor daño que puede sufrir un centro de cómputo es el que se hace dentro, ya que ni los sistemas más sofisticados pueden contra la deslealtad, la deshonestidad o la negligencia de los empleados de la organización. Los errores en los datos son otra fuente de riesgos, ya que los sistemas de Información no pueden ser eficientes si la información que manejan no es correcta, por esta razón se debe llevar a cabo:

- Una capacitación especificando los tipos de datos que se manejan en los sistemas.
- De donde y como obtener los datos para capturarlos en el sistema
- Como introducirlos al sistema y que hacer en caso de encontrar un datos no consistentes.

### 2.11.2 Clasificación de la información y sistemas de control de acceso

Se debe clasificar la información según su importancia para la organización, para saber que tanta seguridad se necesita para cada tipo de información, cuanto tiempo necesita ser retenida, a quien se le dará acceso, si se requieren duplicados o no, etc. Es necesario tener bien definido a que información puede tener acceso cada uno de los empleados.

Hay típicamente tres niveles de controles de acceso que deben existir en cualquier sistema información:

- Identificación de usuario
- *Password* (contraseña) o número de autenticación de usuario. Se usa para autenticar que la persona es quien dijo al identificarse.
- Mecanismo de autorización. Es aquel mecanismo que permitirá al usuario consultar la información que le corresponda únicamente; es el componente más importante en el sistema de seguridad de la información.

Los mecanismos de autorización o paquetes de control de acceso usualmente contienen las siguientes características:

- Identificación de usuario y contraseña.
- Salir del sistema si el usuario intenta una contraseña inválida de un número predeterminado de veces.
- Controles de autorización que permiten reglas de acceso a los archivos de datos.
- Reportes de seguridad que resaltan los intentos de acceso no autorizado.

### 2.11.3 Medidas de seguridad

- Tiempo de espera de la terminal. Si después de un cierto tiempo predeterminado no se registra ninguna actividad en la terminal o en la estación de trabajo, la sesión debe cerrarse automáticamente.



- Restricciones de tiempo de acceso. Pueden definirse periodos en los que los usuarios no tienen necesidad de entrar a su cuenta (fines de semana, por ejemplo) para negárseles el acceso automáticamente. Cuando el usuario requiere acceso a su cuenta fuera del horario normal de trabajo, deberá requerirlo por escrito.
- Detección y expulsión de intrusos del sistema. Debe fijarse un número máximo de intentos de ingresar al sistema (3). Si se excede este número, automáticamente se bloqueará la cuenta durante un tiempo predeterminado.
- Conexiones concurrentes. No debe permitirse a un mismo usuario tener sesiones simultáneas en diversas terminales.
- Asignaciones de propiedad. Desde el punto de vista de seguridad de la información, es muy importante definir bien los privilegios (crear, borrar, modificar, abrir, leer, escribir, etc.) que tiene cada usuario sobre la información. Aquí entran nuevamente las consideraciones sobre la propiedad de los archivos con el objeto de definir quien puede dar o negar los diversos tipos de acceso a la información y ser responsable de la misma.
- Definir y seguir procedimientos adecuados de respaldo de la información de la dependencia. Representa una actividad de suma importancia en la seguridad de la información y especialmente para la efectividad de los planes de contingencia. Deben mantenerse respaldos que contengan los archivos suficientes para recuperar la información dañada o destruida y que garanticen la posibilidad de continuar con el servicio y operación de la dependencia, así como los programas y documentación suficiente para facilitar este proceso en sus instalaciones o fuera de ellas.

#### 2.11.4 Virus

Son una fuente importante de pérdidas de datos. Los archivos ejecutables son los mecanismos más utilizados para la propagación de virus. Existe una categoría de virus, conocida como troyanos que también se propagan a través de ejecutables. Para disminuir el riesgo de pérdida de datos por virus, se deben seguir los siguientes lineamientos:

- No usar programas grabados en disquete (particularmente juegos o utilidades de procedencia desconocida)
- No usar programas bajados de sitios poco confiables de Internet
- No abrir archivos adjuntos de correo electrónico cuyo contenido o remitente se desconozcan o no sean de confianza.

Existe una gran variedad de virus cuyos efectos van desde mensajes en pantalla, hasta la destrucción de datos en el disco duro.

Un virus puede pasar mucho tiempo instalado en la computadora sin causar daño alguno, sin embargo, después de este periodo de inactividad puede afectar a un número considerable de programas y datos en la computadora. Los troyanos son una especie de virus que permiten a personas ajenas al sistema, ingresar a la computadora como un usuario propio del mismo. Los antivirus pueden detectar a la mayoría de los virus y troyanos que existen, sin embargo existen una cantidad reducida de estos que no son detectados por ningún antivirus.

#### 2.11.5 Cortafuegos (*Firewalls*)

En su aspecto más básico, la protección contra troyanos se basa en el uso de antivirus tales como el AVP, McAfee, Symantec, etc. quienes tienen la capacidad de detectar los troyanos más conocidos. Sin embargo existe la posibilidad de ataques más sofisticados por lo cual se hace necesario el uso de software conocido como *firewall*, que permite detectar los intentos de introducirse a un sistema sin la debida autorización.



El funcionamiento del software denominado *firewall* fundamentalmente consiste en filtrar los paquetes que entran y salen por la computadora a través de Internet. El *firewall* permite bloquear los puertos de conexión de la computadora con Internet de tal manera que este envía un mensaje a la pantalla cuando existe un paquete inseguro.

### 2.12 Desarrollo de aplicaciones en Internet

Las aplicaciones en Internet están basadas principalmente en páginas Web que incluyen código en hipertexto (html). Los sistemas de hipertexto son muy fáciles de utilizar y también es muy fácil encontrar lo que se busca, debido a que navegando sobre los enlaces se va accediendo a la información que interesa. Una aplicación Web no está restringida a texto únicamente, ya que puede manipular imágenes, videos, sonido, etc. De hecho, un nuevo concepto ha desplazado al hipertexto; este concepto es hipermedia que incluye todo tipo de contenidos multimedia.

#### 2.12.1 Lenguaje HTML (Hyper Text Markup Language)

Una página Web es visualizada en un navegador (browser). Dicha página típicamente es un archivo con extensión htm o html. Sin embargo esta página html generalmente está compuesta por diferentes archivos (imágenes, animaciones, sonidos, videos, entre otros). El código de las páginas está escrito en un lenguaje llamado HTML, que indica básicamente donde colocar cada texto, cada imagen o cada video y la forma que tendrán estos al ser colocados en la página. El lenguaje HTML consta de etiquetas; cada etiqueta se representa entre los caracteres `<>` y se cierra con los caracteres `</>`. Por ejemplo, la etiqueta título de específica de la siguiente manera:

```
<title> Proyecto de Tesis </title>
```

El lenguaje HTML está conformado por una serie de etiquetas que se utilizan para definir la forma o estilo de un documento. Un documento HTML está delimitado por la etiqueta `<html>` y `</html>`. El documento tiene un encabezado, delimitado por `<head>` y `</head>`, y un cuerpo delimitado por las etiquetas `<body>` y `</body>`.

#### 2.12.2 Páginas estáticas y páginas dinámicas

Se tienen dos tipos de páginas de Internet, las estáticas y las dinámicas. Las páginas estáticas se presentan sin movimiento y sin funcionalidades más allá de los enlaces. El usuario no puede interactuar con ellas.

En las páginas dinámicas, existe una interacción directa entre la página y el usuario. Las páginas estáticas se construyen con lenguaje html únicamente, mientras que las dinámicas se construyen con html y algún otro lenguaje orientado al Web.

#### 2.12.3 Clasificación de las páginas dinámicas

El desarrollo de aplicaciones en Internet es una tecnología cliente servidor. En este caso el cliente está dado por el navegador que abre una página desde una terminal, y el servidor es el equipo en donde está hospedada dicha página. Las páginas dinámicas se clasifican en función de dónde se lleva a cabo el procesamiento de la página, ya que la información puede ser procesada tanto del lado del cliente como del lado del servidor.



### 2.12.3.1 Páginas dinámicas de cliente

Son las páginas dinámicas que se procesan en el navegador directamente. Usos típicos de las páginas de cliente son efectos especiales para Web, control de ventanas, validación de formularios, cálculos aritméticos, etc. El código necesario para crear los efectos y funcionalidades se incluye dentro del mismo archivo HTML y es llamado *script*. Cuando una página HTML contiene *scripts* de cliente, el navegador se encarga de interpretarlos y ejecutarlos para realizar los efectos y funcionalidades. Las páginas dinámicas de cliente se escriben en dos lenguajes de programación principalmente: Javascript y Visual Basic Script (VBScript). Las páginas del cliente son dependientes del sistema donde se están ejecutando y esa es su principal desventaja, ya que cada navegador tiene sus propias características, incluso cada versión, y lo que puede funcionar en un navegador puede no funcionar en otro.

### 2.12.3.2 Páginas dinámicas de servidor

Las páginas dinámicas del servidor son reconocidas, interpretadas y ejecutadas por el propio servidor. Las páginas del servidor son más útiles y potentes, con ellas se puede hacer todo tipo de aplicaciones Web. Desde agendas a foros, sistemas de documentación, estadísticas, juegos, accesos a bases de datos, compras en Web, etc. En este caso se tiene un banco de información centralizada, situada en una base de datos en el servidor. Las páginas dinámicas del servidor se escriben en el mismo archivo html, mezclado con el código html las etiquetas del lenguaje al que se hace referencia. Por ejemplo, en el caso de ASP las etiquetas son `<% %>` y en el de PHP `<? ?>`. De esta manera, el servidor el maneja toda la información de las bases de datos y cualquier otro recurso, como imágenes o servidores de correo y luego envía al cliente una página en formato html que cualquier navegador puede interpretar con los resultados de todas las operaciones. Para escribir páginas dinámicas de servidor existen varias tecnologías: *Common Gateway Interface* (CGI), *Perl*, *Active Server Pages* (ASP), *Hipertext Preprocesor* (PHP), y *Java Server Pages* (JSP) entre otras. Una gran ventaja de estas tecnologías radica en que el cliente no puede ver los *scripts*, ya que se ejecutan y transforman en html antes de enviarlos. Además son independientes del navegador del usuario, ya que el código que reciben es html fácilmente interpretable.

### 2.12.3.3 Tecnologías de programación para Internet dinámico

#### 2.12.3.3.1 Javascript

Javascript es un lenguaje de programación utilizado para crear pequeños programas que realicen acciones dentro del ámbito de una página Web, es decir, del lado del cliente. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Entre las acciones típicas que se pueden realizar en Javascript se tienen dos caminos:

Por un lado los efectos especiales sobre páginas Web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, Javascript permite ejecutar instrucciones como respuesta a las acciones del usuario (eventos), con lo que se pueden crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.



### 2.12.3.3.2 Visual Basic Script (VBScript)

Es un lenguaje de programación de scripts del lado del cliente, pero sólo compatible con Internet Explorer de Microsoft. Está basado en Visual Basic. El modo de funcionamiento de Visual Basic script para construir efectos especiales en páginas Web es muy similar al utilizado en Javascript y los recursos a los que se puede acceder también son los mismos: el navegador.

### 2.12.3.3.3 DHTML

DHTML significa HTML dinámico, está basado en las capacidades de navegadores modernos de responder a eventos generados por el usuario. DHTML permite tener mayor control sobre una página, aunque nunca con el potencial que ofrece JavaScript o VBScript.

### 2.12.3.3.4 Cascading Style Sheets (CSS)

CSS, es una tecnología que permite incluir efectos dinámicos en páginas tales como márgenes, tipos de letra, fondos, colores, etc. Las hojas de estilo en cascada se escriben dentro del código html de la página Web, con la etiqueta style de html. Por ejemplo: `<style="color:blue;font-weight:bold">` Escribirá un texto en color azul y negrita. Dentro de la etiqueta style se deben indicar los atributos de estilos CSS separados por punto y coma (;). Una de las características más importantes de la programación con hojas de estilo consiste en definir los estilos de todo un sitio Web creando un sólo archivo en donde se colocan las declaraciones de estilos de la página para que y enlazando todas las páginas del sitio a ese archivo.

### 2.12.3.3.4 Applets de Java

Son pequeños programas escritos en lenguaje Java, que se transfieren a las páginas Web por medio de la etiqueta `<applet>` y los parámetros de dicho *applet*. La principal ventaja de utilizar *applets* consiste en que son mucho menos dependientes del navegador que los *scripts* en Javascript, incluso independientes del sistema operativo de la computadora en donde se ejecutan, sin embargo, son más lentos de procesar y que tienen espacio muy delimitado en la página donde se ejecutan, es decir, no se mezclan con todos los componentes de la página ni tienen acceso a ellos, es por ello que con los *applets* no se pueden abrir ventanas secundarias, controlar *frames*, formularios, capas, etc.

### 2.12.3.3.5 CGI (Common Gateway Interface)

Fue la primera tecnología para programar páginas dinámicas del lado del servidor, típicamente en lenguaje Perl. Esta tecnología se ha sido desplazada por otras tecnologías como PHP y ASP, principalmente por la dificultad con la que se desarrollan los programas y la pesada carga que supone para el servidor que los ejecuta. Los CGI se escriben habitualmente en el lenguaje Perl, sin embargo, otros lenguajes como C, C++ pueden ser también empleados para construirlos.

El funcionamiento básico de un programa CGI es similar al de las páginas dinámicas del servidor: Se realiza una petición http, a la que pueden acompañar datos llegados o bien por un formulario o bien a través un URL. El servidor ejecuta los programas CGI a los que se accede y trabaja con los recursos necesarios para llevar a cabo las acciones, como por ejemplo bases de datos.

El programa CGI va escribiendo en la salida estándar el resultado de la ejecución del CGI, que incluye etiquetas HTML. Esta tecnología presenta algunas desventajas: Los resultados se escriben directamente en el CGI, así que el código del programa se mezcla con el del HTML haciendo difícil su comprensión y mantenimiento.



### 2.12.3.3.6 Perl (*Practical Extracting and Reporting Language*)

Es un lenguaje de programación muy utilizado para construir aplicaciones CGI Web. Permite extraer información de archivos de texto y generar informes a partir del contenido de los archivos. Es un lenguaje gratuito diseñado específicamente para trabajar en ambiente Unix/Linux, aunque existen versiones para Windows. Perl es un lenguaje de programación de tipo interpretado, esto quiere decir que el código de los scripts en Perl no se compila sino que cada vez que se quiere ejecutar se lee el código y se pone en marcha interpretando lo que hay escrito línea por línea.

### 2.12.3.3.7 ASP (*Active Server Pages*)

Esta tecnología fue desarrollada por Microsoft para la creación de páginas dinámicas del servidor Web de Microsoft, conocido como Internet Information Server IIS que viene integrado con Windows 2000 Server. ASP se escribe en la misma página Web, utilizando el lenguaje Visual Basic Script o Jscript (Javascript de Microsoft). Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor Web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código html resultante de la ejecución de la página. Como la página resultante contiene únicamente código html, es compatible con todos los navegadores.

El tipo de servidores que emplean este lenguaje son, evidentemente, todos aquellos que funcionan con sistema Windows NT/2000 Server. Las páginas ASP permiten realizar muchos tipos de aplicaciones distintas: acceso a bases de datos, al sistema de archivos del servidor y en general a todos los recursos que tenga el propio servidor, incorporar componentes ActiveX, etc. La nueva versión de ASP es conocida como ASP.NET, que comprende algunas mejoras en cuanto a posibilidades del lenguaje para el manejo de eventos del lado del cliente, y un mejor desempeño en la ejecución de las páginas interactivas.

### 2.12.3.3.8 PHP (*Preprocesor Hipertext*)

Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran biblioteca de funciones y mucha documentación. PHP se escribe dentro del código HTML, lo que lo hace realmente fácil de utilizar, al igual que ocurre con el ASP de Microsoft, pero con algunas ventajas como su gratuidad, independencia de plataforma, rapidez y seguridad.

Es independiente de plataforma, puesto que existe un módulo de PHP para casi cualquier servidor Web incluido el de Microsoft. PHP, en el caso de estar montado sobre un servidor Linux u Unix, es más rápido que ASP dado que se ejecuta en un único espacio de memoria y esto evita las comunicaciones entre componentes COM que se realizan entre todas las tecnologías implicadas en una página ASP. PHP es más seguro que ASP debido a que se encuentra instalado sobre servidores Unix/Linux, los cuales son más veloces y seguros que el sistema operativo Windows NT/2000 Server que ejecuta las páginas ASP. Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones Web gracias a la extensa biblioteca de funciones con la que está dotado. La biblioteca de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, etc.

Algunas de las características más importantes de PHP son la compatibilidad con las bases de datos más comunes, como MySQL, Oracle, Informix, y ODBC. Incluye funciones para el envío de correo electrónico, subir archivos al servidor, crear dinámicamente en el servidor imágenes, etc. Típicamente el servidor Web que ejecuta los comandos de PHP es el Apache, que viene con la distribución RedHat del sistema operativo Linux, o que puede obtenerse de la página de la Fundación Apache



## Sistema de Información de Aeropuertos



### 3. REQUERIMIENTOS DEL CLIENTE

En Aeropuertos y Servicios Auxiliares, ASA se requiere implantar un Sistema de Información de Aeropuertos en ambiente Cliente Servidor, de tal manera que despliegue datos concernientes a los Aeropuertos Nacionales e Internacionales de cualquier parte del mundo. Este Sistema debe contar con un estricto control sobre los usuarios, ya que se tendrán diferentes perfiles de los mismos. Habrá información de carácter confidencial para lo cual se requerirá la validación con un nombre de usuario y una contraseña. Para alimentar al sistema, y mantener actualizada la información, se requerirá también de un permiso validado con un nombre de usuario y contraseña.

#### 3.1 ENTREVISTAS CON LOS USUARIOS.

3.1.1 Lic. Rosario Díaz de Guzmán, Secretaria Técnica General de Aeropuertos y Servicios Auxiliares.

Con la Licenciada Díaz de Guzmán se realizaron 5 entrevistas para determinar los requerimientos de información del sistema que se va a desarrollar. Destacaron en dichas entrevistas, en primer lugar, el conformar un sistema de información con los aeropuertos de Latinoamérica, así como los aeropuertos a nivel Nacional.

En la primeras dos entrevistas, se obtuvo la siguiente información:

Aeropuertos y Servicios Auxiliares (ASA), es una institución de carácter público, encargado de operar y administrar algunos de los Aeropuertos del país, así como proveer de combustible a todos los Aeropuertos de México. ASA está dividido principalmente en 4 áreas: finanzas, administración, jurídica y comunicación social. El objetivo principal de ASA es promover el desarrollo de la industria aeroportuaria a nivel nacional, y brindar la asesoría y apoyos a nivel internacional en los aeropuertos que así lo requieran. En ese sentido, se tienen tres vertientes principales: consultoría, operación aeroportuaria y suministro de combustibles.

El servicio de consultoría, a nivel nacional como internacional, consiste en realizar estudios técnicos de planeación, evaluación y diseño de proyectos, empleando tecnología de punta, planes de mantenimiento a equipos e instalaciones, estudios de impacto ambiental, estudios de factibilidad, entre otros. El área de operación aeroportuaria puede realizar la construcción de instalaciones siguiendo los lineamientos establecidos para cada proyecto, aprovechando al máximo, los recursos disponibles (humanos, materiales, naturales). El área de distribución de combustibles, lleva a cabo las tareas de recepción, almacenamiento, control de calidad y suministro de combustibles a las aeronaves comerciales, nacionales y extranjeras, aviación general, organismos gubernamentales y estatales, en la totalidad de las 65 terminales aéreas existentes en la república Mexicana, pudiéndose extender hacia otras partes del mundo, como Latinoamérica por ejemplo.

En ASA se requiere un sistema de información que suministre datos referidos a la operación de los aeropuertos nacionales que están bajo la responsabilidad de ASA y los que no están, y de los aeropuertos internacionales a los cuales ASA les puede prestar algún servicio, tales como los de Centro América y deberá operar bajo ambiente cliente servidor para la captura y consulta de datos.

Los datos más significativos de un Aeropuerto son: Siglas cada aeropuerto tiene sus siglas de manera independiente, dadas por la IATA (Internacional Air Transportation Association). Se tiene sin embargo una clasificación por la Organización de Aviación Civil Internacional (OACI) de 4 siglas que empieza a difundirse también.



## Sistema de Información de Aeropuertos

---



Cada Aeropuerto tiene un nombre por ejemplo: Benito Juárez. El Sistema debe distinguir entre los Aeropuertos Nacionales y los internacionales, aunque los datos que almacenará y desplegará son los mismos en ambos casos. Es importante conocer el país al que pertenece el Aeropuerto, así como su clasificación (Internacional, Nacional, Regional), y el tipo (Metropolitano, Turístico, Fronterizo). La fecha de inauguración, es importante, así como el tipo de administración que se tiene en el aeropuerto, ya que se tienen administradores gubernamentales y privados. Con respecto a los edificios terminales se requiere saber el número de edificios que se tienen, los servicios que prestan, la superficie total del edificio, el número de cajones de estacionamiento disponibles, la dimensión de los mismos y el número de estacionamientos que tiene. Cabe destacar que se requiere conocer la capacidad anual del aeropuerto, tanto para pasajeros como para operaciones.

El tipo de sistemas de navegación aérea, el tipo de sistemas de suministro de combustible, las instalaciones de prevención de accidentes y desastres, el historial de remodelaciones de los aeropuertos, el número de estacionamientos, el tamaño en metros cuadrados de los mismos, y la capacidad del número de vehículos, así como los servicios que ofrece el aeropuerto (taxis, casas de cambio, alquiler de vehículos, etc.). Es importante que el sistema despliegue las imágenes que se tengan disponibles de los aeropuertos, tales como torre de control, edificio principal, pistas, etc.

También deberá desplegar información de los aviones que pueden aterrizar en las pistas, tales como tamaño, peso, alcance y algunas fotografías de los mismos. De las pistas se requiere conocer su designación, sus dimensiones y el tipo de piso que tiene, ya que puede ser de asfalto o de concreto. Se tienen también los datos de las calles de rodajes y de las plataformas, en este caso interesa conocer sus dimensiones, y el tipo en el caso de las plataformas y la designación en el caso de las calles de rodajes, por ejemplo en la ciudad de México se tiene una plataforma Alfa de 54 x 340 metros.

Es importante también conocer los sistemas de suministro de combustible del aeropuerto así como su capacidad, y los sistemas de extinción de incendios y rescate con sus respectivas capacidades (número de vehículos, capacidades de los vehículos, tipo de agente extintor, etc.) Otro dato importante para el sistema de información son los servicios que presta un aeropuerto, ya que por ejemplo, se tiene el servicio taxis en la zona del edificio terminal, bancos y casas de cambio dentro del edificio terminal, hoteles en las zonas aledañas al edificio terminal, etc.

La Licenciada Díaz de Guzmán sugirió una entrevista con el Lic. Juan Luis Díaz, encargado de la unidad de Planeación y nuevos proyectos a con el objeto de evaluar los alcances del sistema de información que se está desarrollando.

### 3.1.2 Lic. Juan Luis Díaz, Jefe de la Unidad de Nuevos Proyectos de Aeropuertos y Servicios Auxiliares.

Al Lic. Juan Luis Díaz se le plantearon los alcances y características del sistema de información. En este sentido, se hicieron algunas observaciones mínimas a los requerimientos de información que la Licenciada Díaz de Guzmán había planteado. Por ejemplo, clasificar a los Aeropuertos por países, algo que ya estaba contemplado anteriormente. De los países es importante conocer su nombre, su población, su extensión territorial y el PIB, ya que en función de éste, se puede estimar el crecimiento económico del país y por lo tanto su demanda aeroportuaria.



## Sistema de Información de Aeropuertos



Es importante también conocer la estructura que cada país tiene con respecto a sus Aeropuertos, por ejemplo, en el caso de México, la responsabilidad de la Operación Aeroportuaria recae en diferentes organismos, en primer lugar se encuentra la Secretaría de Comunicaciones y Transportes, dependencia gubernamental encargada de controlar cualquier tipo de transporte a nivel federal. Sigue el SENEAM. (Servicios a la Navegación en el Espacio Aéreo Mexicano), organismo dependiente de la SCT, encargado de controlar el tráfico Aéreo a nivel Nacional y la DGAC (Dirección General de Aeronáutica Civil). A continuación se tienen los grupos Aeroportuarios, los cuales son los encargados de operar y administrar los Aeropuertos del país, aunque en el fondo los aeropuertos son propiedad de la SCT y están concesionados a los grupos aeroportuarios. Cabe destacar que se tienen grupos Aeroportuarios tanto gubernamentales como ASA, o privados como Aeropuertos del Sureste ASUR, aunque estos últimos operan bajo la supervisión de ASA.

En el caso de Costa Rica por ejemplo, la estructura que guarda con respecto a la operación de sus Aeropuertos es más o menos similar a la de México, la responsabilidad del espacio aéreo recae en el Ministerio de Comunicaciones, lo que equivaldría a la SCT en México, se tiene también una Dirección General de Aeronáutica Civil, que es el organismo encargado de manejar todos los aeropuertos de dicho país, los aeropuertos son operados por el gobierno, y únicamente se tiene un aeropuerto concesionado que es el Juan Santamaría a un organismo privado llamado Vectel. Otro caso interesante se presenta en Argentina, en donde el control Aéreo está dado por la Fuerza Aérea, es decir, depende de una institución de tipo Militar.

Es importante conocer la estructura que cada País tiene con respecto a la operación de sus Aeropuertos, así como conocer los nombres de los organismos encargados de la operación Aeroportuaria. Cada país tiene una estructura diferente, y aunque se pueden tener ciertas similitudes con respecto a otro país, las diferencias entre uno y otro suelen ser considerables.

El Lic. Juan Luis Díaz sugirió entrevistas con las siguientes personas: Ing. Federico Dovalí y el Ing. Jorge Mandri, con el fin de mostrarles los alcances de éste sistema y poder complementar los requerimientos de información que se tengan.

3.1.3 Ing. Jorge Mandri, Jefe de la Unidad de Nuevos Proyectos de Aeropuertos y Servicios Auxiliares.

El Ing. Jorge Mandri conoció los alcances y características del sistema, y se solicitó información con respecto a algunos catálogos del mismo. De manera similar que con el Maestro Juan Luis Díaz, las correcciones fueron pocas, por ejemplo, del Aeropuerto me comentó que faltaba considerar el horario de operación, ya que hay Aeropuertos que operan las 24 horas, como el Aeropuerto de la Ciudad de México, y hay Aeropuertos que tienen un horario, por ejemplo el Aeropuerto de Huatulco Oaxaca que opera de 9 a 6 p.m. La mayoría de los Aeropuertos tienen ya su página Web, por lo que es deseable que el Sistema tenga un vínculo a la página Web del Aeropuerto.

A juicio del Ing. Mandri no es necesario utilizar la clasificación OACI de 4 letras para el Aeropuerto, ya que la más estandarizada es la de 3 siglas. Con respecto a los aviones, las observaciones fueron referidas únicamente a la nomenclatura para describir a dichos Aviones, por ejemplo, la distancia entre alas es conocida como envergadura, la velocidad de operación es conocida como velocidad de crucero, que a su juicio no es necesario incluir.

En el caso de los pesos y capacidades de carga, son muy variables incluso para dos aviones del mismo modelo, ya que se equipan de acuerdo a los requerimientos específicos de una Aerolínea, así se pueden tener por ejemplo dos aviones Boeing 727, uno con 118 pasajeros de una sola clase, y otro con 96 pasajeros divididos en dos clases.



## Sistema de Información de Aeropuertos



Incluso la configuración de los motores puede cambiar, ya que se puede dar el caso de tener un 727 con motores RollsRoyce y otro con motores GE, por lo que se deben tomar los pesos máximos admisibles para describir las características del avión, por ejemplo, peso máximo de despegue, capacidad máxima de combustible, rango máximo de operación con el peso máximo, etc. Las ayudas como se había mencionado anteriormente pueden ser visuales o de radio, y un aeropuerto puede tener diferentes ayudas visuales o de radio. De las calles de rodaje es importante conocer la designación, sus dimensiones, y si están o no señalizadas.

3.1.4 Ing. Federico Dovalí, Jefe de la Unidad de Operación Aeroportuaria de Aeropuertos y Servicios Auxiliares.

Con el Ing. Dovalí se detallaron los requerimientos de información que se habían planteado anteriormente. En su opinión se cubren de manera satisfactoria los requerimientos de información proporcionados por la Licenciada Díaz de Guzmán. Comentó que era importante conservar la clasificación de 4 letras dada por la OACI, ya que se está empezando a estandarizar. En el caso de Estados Unidos, la clasificación de 3 letras y la 4 es prácticamente la misma, por ejemplo, el Aeropuerto John F. Kennedy tiene las siglas de IATA como JFK, y por la OACI como KJFK, es decir sólo se le agrega la K a la clasificación de 3 siglas, sin embargo en el caso de los demás países no sucede así, por ejemplo, el Aeropuerto de la ciudad de México tiene la clasificación IATA como MEX, mientras que la OACI es MMMX. Con respecto a los aviones, es importante desplegar sus velocidades en unidades Match y no en kilómetros por hora, ya que es la unidad estandarizada para medir dichas velocidades.

Explicó el proceso a seguir para designar el nombre de una pista, ya que están basadas en las coordenadas magnéticas de la tierra, así como las designaciones de las calles de rodajes que están dadas por el alfabeto fonético, a fin de evitar confusiones, por ejemplo, la calle a es alfa, b es bravo, etc. Comentó también acerca de una clasificación que da la OACI a los aeropuertos en base a su Sistema de Rescate, aunque no es muy utilizada y no amerita incluirla dentro del Sistema de Información.

3.1.5 Ing. Sergio Rivera Romero, Asesor de la Unidad de Informática

El Ing. Rivera planteó la necesidad de desarrollar el sistema bajo software libre, ya que no se iban a renovar las licencias de Sybase bajo las cuales se iba a desarrollar el proyecto, por lo que el sistema de información debería de ejecutarse sobre plataforma Linux, preferentemente en un ambiente 100 % Web por lo que habrá que elegir la mejor distribución Linux que haya disponible, el mejor lenguaje para el desarrollo de la aplicación y el mejor servidor que para estos fines se tenga.

### 3.2 ANÁLISIS DE LA ENTREVISTAS

En ASA y concretamente de parte de la Licenciada Díaz de Guzmán existe entusiasmo por el desarrollo de esta aplicación. Como se mencionó anteriormente, se debe tener un perfil de usuarios para el uso del sistema de información. Concretamente se tendrán 2 perfiles de usuarios: Usuarios de ASA que pueden consultar toda la información, y Administradores del sistema que pueden consultar toda la información, agregar, modificar y en caso necesario eliminar información, así como dar de alta nuevos usuarios, eliminar y modificar la información de los mismos.

Se puede tomar como ejemplo para este sistema de información el Aeropuerto de la Ciudad de México, ya que dentro de los Aeropuertos de América Latina es el más grande, y puede servir como ejemplo para algunos de los catálogos que se tendrán en el sistema, por ejemplo:



## Sistema de Información de Aeropuertos



Dentro de los servicios que el aeropuerto ofrece se tienen las siguientes categorías:

- Médicos
- Asistencia
- Emergencia
- Facilidades para discapacitados
- Alimentos y bebidas
- Artículos de viaje
- Comunicaciones
- Estacionamientos
- Farmacias
- Hoteles
- Librerías
- Puestos de Periódico
- Puestos de Lotería
- Tabaquerías
- Dulces y regalos
- Transportación terrestre
- Servicios Financieros
- Varios

Estas categorías pueden incluir a la vez otras subcategorías, ejemplo: los servicios de facilidades para discapacitados pueden ser

- Puertas e interiores de sanitarios
- Cabinas telefónicas a menor altura
- Cintas antiderrapantes
- Elevadores convencionales y elevadores para ascenso y descenso de escaleras
- Carritos eléctricos
- Transporte en sillas de ruedas
- Rampas de acceso a salas de última espera
- Rampas en vialidades
- Señalamientos
- Cajones especiales en estacionamientos
- Personal capacitado para la atención especializada a pasajeros con discapacidad.

Puede haber catálogos con características adicionales, ejemplo:

Transportación terrestre

Autobuses

Primera Plus Autobuses la Piedad

Destino: Aeropuerto Internacional de la Ciudad de México - Querétaro

Querétaro - Aeropuerto Internacional de la Ciudad de México

Ubicación: Sala "D" Puerta 8 del Aeropuerto Internacional

Teléfono: 5786-9358, 5786-9341 y 42

Horario de Servicio: 6:00 a 24:00 hrs.

Horarios de Salida: La salida de autobuses es cada hora. 7:00 a 22:00 hrs.



## Sistema de Información de Aeropuertos



### Metro

Línea 7 estación Terminal Aérea

Ubicación: Blvd. Aeropuerto Esq. Av. Capitán Carlos León González.

Horario: Lunes a viernes 5:00 A.M. A 1:00 A.M. Sábados 6:00 A.M. A 12:00 P.M.

Domingos 7:00 A.M. A 10:00 P.M.

### Renta de autos

Área Nacional Álamo Bests Enterprise, S.A. de C.V.

Ubicación: Sala "A1", Local 19

Teléfono: 2550-0055, 5255-3939

Horario: 24:00 hrs.

### Taxis

Taxistas Agremiados para el Servicio de Transportación Terrestre Sitio 300 A. C

Ubicación: Sala "A1", Local 19

Teléfono: 2550-0055, 5255-3939

Horario: 24:00 hrs.

En ese sentido, es necesario definir si se requiere conocer a detalle los datos de los catálogos, como en este ejemplo, o simplemente saber si se tienen o no los servicios. Por lo que en una entrevista adicional con la Lic. Díaz de Guzmán quedará definido éste punto.

Hay catálogos más especializados, como los tipos de aviones, características de las pistas, características de sistemas de combustible, plataformas, sistemas de extinción de incendios, etc. para los cuales algunos de los datos de ejemplo serán proporcionados por la Lic. Díaz de Guzmán y otros deberán ser investigados en libros como el AZUL y visitar algunas páginas Web como la OACI.

Se realizó una segunda entrevista con la licenciada Díaz de Guzmán para aclarar algunas dudas e inquietudes:

1. ¿Se tienen en el país aeropuertos Multimodales (Aeropuertos que incluyen terminal de tren, de barco y de carga terrestre), sobre todo en el norte de la república, el sistema los debe contemplar? No es necesario, de hecho sólo existe uno en todo el país en el estado de Sonora en Guaymas.
2. ¿Cual es la diferencia entre número de pasajeros y número de operaciones?  
Número de pasajeros se refiere a la cantidad de personas que pueden hacer uso del aeropuerto. Una operación puede ser el despegue o el aterrizaje de un avión.
3. ¿Un aeropuerto puede tener un operador y un administrador diferentes?  
Sí, sobre todo en Latinoamérica. En México no ocurre
4. ¿ASA es operador y administrador a la vez?  
Sí
5. ¿Cual es la diferencia entre ambos?  
El operador se encarga de la parte correspondiente al funcionamiento del aeropuerto, el administrador tiene que ver con la parte de gastos, nomina, ingresos, egresos, etc.
6. ¿Que es una calle de rodajes?  
Son las calles por las cuales circulan los aviones de las plataformas a las pistas.
7. ¿Aparte de ASA que otras instituciones similares existen en México?  
Grupo Sureste, Grupo Pacífico, Grupo Centro Norte, Grupo Ciudad de México y Grupo Puebla.
8. ¿Todas las siglas son exclusivamente de 3 letras?  
Sí, aunque hay una nueva clasificación que contempla 4 siglas, aunque la clasificación de 3 siglas no debe descartarse todavía.
9. ¿En el caso de los estacionamientos interesa la superficie en metros cuadrados y número vehículos como un total o por partes? Únicamente interesan totales.



## Sistema de Información de Aeropuertos



10. ¿En el caso de las pistas, análogamente con los estacionamientos, se requieren datos totales o por partes? Se requieren datos para cada pista por separado, ya que por ejemplo una pista puede medir más que otra.
11. ¿Cuáles son los tipos de aviones que aterrizan en las pistas?  
Básicamente se clasifican de acuerdo a los estándares de la marca Boeing
12. ¿En donde se puede encontrar un catálogo completo de aviones?  
En el Web, aunque para efectos del sistema, tomando como referencia a la marca Boeing se puede estandarizar el catálogo en función de pesos y tamaños a otras marcas de aviones. De hecho sólo es importante conocer el avión máximo operable, debido a que una pista que soporte un avión Boeing 737 que es un avión más o menos grande, soporta un avión Boeing 717 que es mucho más pequeño y ligero.
13. ¿Cuáles son los tipos de ayudas visuales?  
Ayudas de aproximación, Conos de Viento, Faros de Aeródromos, Luces de Aproximación, Pistolas de Señales. También se tienen radio ayudas: Radio Faros, Radares, ILS
14. ¿Sólo se tienen 3 tipos de plataformas? ¿Desembarco, Pernocta y Aviación General? Sí, son las más importantes
15. ¿Las dimensiones de las plataformas interesan en totales o por partes, así como el número de plataformas? Se consideran actualmente por totales
16. ¿Los sistemas de suministro de combustible interesan por partes o por totales?  
Por totales
17. ¿En donde se puede conseguir este catálogo?  
Revista ENLASA publicada por ASA
18. ¿Las instalaciones de prevención de accidentes interesan por partes o por totales?  
Interesa conocer tipos, número, y capacidades.
19. ¿Es correcto contemplar los servicios por detalle?, por ejemplo

### Transportación terrestre

#### Autobuses

Primera Plus Autobuses la Piedad

Destino: Aeropuerto Internacional de la Ciudad de México - Querétaro

Querétaro - Aeropuerto Internacional de la Ciudad de México

Ubicación: Sala "D" Puerta 8 del Aeropuerto Internacional

Teléfono: 5786-9358, 5786-9341 y 42

Horario de Servicio: 6:00 a 24:00 hrs.

Horarios de Salida: La salida de autobuses es cada hora. 7:00 a 22:00 hrs.

#### Metro

Línea 7 estación Terminal Aérea

Ubicación: Blvd. Aeropuerto Esq. Av. Capitán Carlos León González.

Horario: Lunes a viernes 5:00 A.M. A 1:00 A.M. Sábados 6:00 A.M. A 12:00 P.M.

Domingos 7:00 A.M. A 10:00 P.M.

#### Renta de autos

Área Nacional Álamo Bests Enterprise, S.A. de C.V.

Ubicación: Sala "A1", Local 19

Teléfono: 2550-0055, 5255-3939

Horario: 24:00 hrs.

#### Taxis

Taxistas Agremiados para el Servicio de Transportación Terrestre Sitio 300 A. C

Ubicación: Sala "A1", Local 19

Teléfono: 2550-0055, 5255-3939

Horario: 24:00 hrs.

No, sólo interesa conocer el tipo de servicios con los que cuenta el aeropuerto, por ejemplo si tiene terminal de autobuses, renta de autos y taxis, pero no es importante tener el detalle de los mismos.



## Sistema de Información de Aeropuertos



### 3.3 RECOPIACIÓN DE LOS REQUERIMIENTOS DEL SISTEMA

Se requiere un Sistema de Información sobre Aeropuertos que se ejecute sobre plataforma Web con acceso a bases de datos de tal manera que despliegue la siguiente información de los Aeropuertos: Nombre, siglas (con tres y cuatro letras), ubicación geográfica (País, Estado/Provincia, Ciudad), Clasificación (Nacional, Internacional, Local, Regional), Tipo (Metropolitano, Turístico, Fronterizo, Multimodal) la fecha de inauguración, nombre del operador y del administrador, así como saber si son gubernamentales o privados, la capacidad del aeropuerto tanto en número de operaciones anuales como en número de pasajeros anuales.

En este sentido sería correcto tener las capacidades promedio por día y presentar el dato calculado para un año. De las pistas se requiere conocer el número, la designación de las mismas, sus dimensiones, tanto de largo como de ancho, el tipo de pavimento (asfálticas o de concreto), el tipo de aviones que pueden aterrizar, en donde se considera como estándar la marca Boeing con el avión máximo operable con sus principales características.

Las unidades pueden ser almacenadas por ejemplo para velocidades en kilómetros por hora y visualizarlas en Mach, las capacidades en litros / galones, los pesos en kilogramos / libras. Se deben considerar el tipo de ayudas visuales que se tienen para los aterrizajes, el número de plataformas, tanto de desembarco como de aviación general y de pernocta, sus dimensiones en metros cuadrados, el número de calles de rodajes y las dimensiones de las mismas, los tipos de navegación aérea que se tienen en las torres de control, los tipos de sistemas de suministro de combustibles con sus respectivas capacidades, los equipos de prevención de accidentes y desastres, los datos generales del edificio terminal: número, dimensiones en metros cuadrados, el historial de remodelaciones y ampliaciones en los últimos 10 años siempre y cuando se tenga disponible esa información, el tamaño de los estacionamientos, tanto en metros cuadrados como en el número de vehículos que pueden albergar, los diferentes servicios que se ofrecen dentro del edificio terminal y zonas aledañas, así como fotografías de las zonas operacionales y del edificio terminal. Importa también conocer los sistemas de suministro de combustible con sus respectivas capacidades y tipos de combustibles que proveen.

Generalmente estos son Turbosina para los aviones tipo Jet y Gas avión para aviones de turbina. El Sistema deberá contar con un módulo de administración de usuarios en donde se concedan cuentas de acceso al sistema de información. Cabe destacar que únicamente se contemplarán 2 niveles de usuario: el usuario de ASA que podrá consultar toda la información del Sistema y el usuario administrador que podrá consultar toda la información del sistema, otorgar cuentas, cancelar cuentas, modificar claves, y consultar, modificar, actualizar y en caso necesario eliminar la información contenida en los catálogos del sistema con sus respectivas consideraciones en el sentido de permitir o no la eliminación de registros en cascada según sea el caso a fin de mantener la integridad de los datos.



### 4. DESARROLLO DE LA BASE DE DATOS

#### 4.1 BASES DE DATOS RELACIONALES

##### 4.1.1 Introducción a las Bases de Datos Relacionales

Una base de datos relacional, posee la característica de almacenar grandes volúmenes de datos en dentro de una estructura previamente establecida. Esa estructura, consiste en un conjunto de tablas, las cuales están relacionadas entre sí, con ciertas restricciones que permiten mantener la integridad referencial de los datos almacenados. Un sistema de base de datos, está conformado de diferentes elementos; en primer lugar, se tienen los datos, que no son más que *bytes* almacenados en la base de datos. La información, es referida a la interpretación que se les da a los datos, es decir, que al conjuntar diferentes datos, todos en conjunto representan información. La parte de hardware está referida al equipo o equipos en donde se van a almacenar físicamente los datos, mientras que el software, consiste en los programas que van a permitir trabajar con los datos. También los diferentes tipos de usuarios forman parte del sistema de base de datos.

En primer lugar, se tiene al analista que es aquella persona que por medio de una entrevista con el cliente o usuario, va a recabar las necesidades de información que se tengan en la empresa. Una vez recabadas las necesidades, el siguiente paso es realizar el análisis relacional de las mismas, es decir, generar un modelo entidad relación, abstraído de la realidad, que cubra al 100% los requerimientos de la empresa.

En segundo lugar, se encuentra el programador o desarrollador, su tarea es construir una aplicación, que manipule los datos contenidos en la base de datos, por medio de algún lenguaje de programación como Visual Basic o Power Builder por ejemplo.

Las tareas que realiza el administrador son diversas: Una vez generado el modelo entidad/relación, el siguiente paso es construir una base de datos en algún sistema manejador de bases de datos relacionales como es el caso de SQL Server, Oracle, Sybase, etc. Es importante diseñar una estructura de almacenamiento adecuada dentro del disco duro al momento de crear una base de datos nueva. Existen manejadores de bases de datos como SQL Server en los que las tareas administrativas son sumamente sencillas, ya que desde el momento de crear la base de datos, el espacio en disco duro es asignado de manera automática, y en la medida en la que crezca la base de datos, el espacio reservado para la misma irá creciendo de manera automática. El control de cuentas y permisos para usuarios y grupos de trabajo, es otra tarea administrativa que SQL Server realiza por medio de una interfaz muy amigable al usuario. Los respaldos y restauraciones de las bases de datos, los cuales son tareas administrativas también, se ejecutan con gran facilidad en los manejadores de bases de datos actuales.

El perfil de usuario que resta mencionar, es el usuario final, es decir, aquel que manipula los datos, y recupera la información generada por el sistema.

##### 4.1.2 Ventajas de las Bases de Datos Relacionales

Las ventajas de utilizar sistemas manejadores de bases de datos relacionales para almacenar información son muchas; con un buen diseño de base de datos relacional, se reduce la redundancia (repetición) de datos. Tener datos repetidos, tiene varios inconvenientes; el primero de ellos, es que al tener diferentes copias del mismo dato en la base de datos, las actualizaciones de esos datos son difíciles de realizar, y se corre el riesgo de tener diferentes copias del mismo dato, que no coincidan.



## Sistema de Información de Aeropuertos

---

La segunda desventaja, es referida al espacio de almacenamiento, es decir, la base de datos crecerá de manera innecesaria por tener varias veces almacenado el mismo dato. El propósito de tener restricciones (*constraints*) dentro de una base de datos relacional, es mantener la integridad referencial de los datos. El término integridad, se refiere a que los datos van a ser consistentes, confiables, es decir, van a tener un significado. Dentro de los sistemas de bases de datos basados en ambiente Cliente-Servidor, es importante tener el control de los usuarios sobre los datos, es decir, cuando diversos usuarios están conectados a la misma fuente de información, es importante tener el control de los mismos; así pues, hablar de concurrencia, es hablar de más de un usuario conectado al mismo tiempo.

Hablar de seguridad en una base de datos relacional es algo imperativo. Existen diferentes niveles de seguridad, los cuales pueden ser administrados desde el manejador de bases de datos relacionales en cuestión.

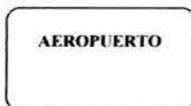
Una ventaja verdaderamente significativa de las bases de datos, es la independencia que estas tienen con los programas que las manipulan, en el sentido que si el programa llega a fallar, los datos permanecen intactos, y pueden ser accedidos por medio de lenguaje SQL, o por medio de otra aplicación. Los datos representan grandes volúmenes de dinero para una organización, en el sentido que existen cientos o miles de horas/hombre de trabajo, para recabar la información.

### 4.1.3 Modelo Entidad Relación

El Modelo de datos Entidad-Relación, es estructurado a raíz de problemas reales en una empresa, que se necesiten resolver. Esta conformado por entidades, las cuales, deben representar información de interés para el caso de estudio o problema que se tenga que resolver. Es importante, que antes de incluir una entidad dentro de un diseño de base de datos relacional, se esté seguro que ésta represente realmente información de interés para la empresa. Las entidades no actúan como objetos aislados, es decir, deben existir relaciones entre éstas, para que puedan intercambiar información. Las entidades deben ser escritas en singular, con letra mayúscula, sin acentos y sin espacios.

Ejemplos de entidades: AEROPUERTO, AVION, PERSONAL, LOCALIDAD, REGION, etc.

Una entidad se representa por un cuadro de esquinas redondeadas, con un nombre en singular y con letras mayúsculas



Un atributo, es una característica propia de una entidad; representa información de interés para la misma, por ejemplo, en el caso de la entidad AEROPUERTO, las siglas, el nombre, domicilio, la ubicación geográfica, la fecha de inauguración, etc. son características de dicha entidad, y representan información de interés para las mismas.

Todo atributo, debe ser descompuesto hasta su mínima expresión, por ejemplo, un atributo nombre en el caso de una persona, se descompone en apellido paterno, apellido materno y nombre; un atributo dirección, se descompone en calle número, colonia, código postal, delegación, ciudad, estado, etc.

Los atributos deben ser escritos en singular, con letra minúscula, sin acentos y sin espacios.



## Sistema de Información de Aeropuertos

---

Si existe algún atributo conformado por dos palabras, éstas pueden ser separadas por un guión bajo, por ejemplo calle\_numero. Cabe destacar que el carácter guión "-" no debe ser utilizado como separador entre dos o más palabras.

El término dominio del atributo, se refiere al tipo de valores permitidos para dicho atributo. Los tipos de datos, están clasificados de manera muy general, en numéricos, lógicos, de texto y de fechas, sin embargo los manejadores de bases de datos relacionales utilizan tipos de datos derivados de los anteriormente mencionados.

Se tienen diferentes tipos de atributos:

- **Únicos (#):** permiten identificar a un elemento de la entidad (registro) como único. Ejemplo: curp, no de cuenta en la UNAM, rfc, número de seguro social, número de serie, número de inventario, etc.
- **Obligatorios (\*):** deben contener valores no nulos de manera obligatoria. Ejemplo: Nombre (todas las personas deben tener un nombre), fecha de nacimiento, nacionalidad, estado civil, etc.
- **Opcionales (o):** admiten valores nulos. Ejemplo teléfono, e-mail, debido a que no todas las personas deben tener teléfono o e-mail de manera obligatoria.

Antes de empezar a diseñar una base de datos es importante, comprender al 100% los requerimientos de información de la empresa, apegarse a los estándares de entidades y atributos antes mencionados. Una vez entendido el problema, el paso siguiente es desarrollar un modelo entidad-relación que cubra todos los requerimientos. Después de desarrollar el modelo, el siguiente paso es construir una base de datos relacional a partir del modelo entidad-relación.

En la etapa de desarrollo del modelo entidad relación, se trabaja con entidades, atributos, identificadores únicos y relaciones. En la construcción de una Base de Datos Relacional, las entidades se convierten en tablas, los atributos en columnas (campos), los identificadores únicos en llaves primarias, y las relaciones en llaves foráneas.

Los nombres de las entidades, deben ser escritos en singular, mientras que los de las tablas en plural. Para identificar entidades, es importante examinar los sustantivos contenidos en las entrevistas con los clientes; antes de agregar una entidad en el modelo de datos, es necesario verificar que ésta represente información de interés para el caso de estudio que se quiere abordar; en caso de no representar información de interés, no es conveniente agregarla como entidad; entre más compacto sea el diseño de base de datos, mejor será su desempeño. Cabe destacar que todas las entidades deben ser identificables como únicas, si una entidad no se puede identificar como única, entonces no es una entidad.

Las llaves, son atributos con características particulares, que permiten identificar a cada elemento de una entidad como único; existen diferentes tipos de llaves:

- **Llave Candidato:** Una llave candidato, es un atributo que permiten identificar a la Entidad como única, por ejemplo, el número de serie de un equipo, el número de inventario, las placas de un vehículo, el número de empleado, el número de curp, el número de cuenta en la Universidad, etc.
- **Llave Primaria:** La llave Primaria de una entidad, es la llave candidato, seleccionada como identificador único de la entidad.
- **Llave Foránea:** Una llave foránea, es una llave primaria dentro de una entidad A, que permite establecer una relación entre dicha entidad A y una entidad B. Por ejemplo, en el caso de una entidad EMPLEADO, y una entidad DEPARTAMENTO, la llave foránea dentro de la entidad



## Sistema de Información de Aeropuertos



EMPLEADO, sería la clave del departamento, es decir, el identificador de la entidad DEPARTAMENTO, contenido en la entidad EMPLEADO.

Una entidad, puede tener más de una llave candidato, por ejemplo, una entidad EMPLEADO, tendrá como atributos únicos el rfc, el número de nómina, el número de seguro social, etc. De estos tres atributos únicos, sólo uno será llave primaria. Es recomendable, utilizar el identificador más sencillo como llave primaria, por ejemplo, el número de nómina en el caso de la entidad EMPLEADO

### 4.1.4 Relaciones entre entidades

Una relación es una regla de correspondencia entre dos entidades, con el objeto de intercambiar información. Todas las relaciones entre entidades, se llevan a cabo por medio de la llave foránea.

Por ejemplo, la relación entre AEROPUERTO y TIPO es:

Cada AEROPUERTO debe ser asignado a uno y solo un TIPO

Cada TIPO puede estar clasificando a uno o más AEROPUERTOS

Componentes de una relación

Cada dirección en una relación, tiene:

Un nombre: pertenecer a/ asignado a

Una opción: Debe ser/estar / puede ser/estar

Un grado: Uno y solamente uno, uno o más

Cabe destacar, que las relaciones se deben leer en primer lugar, de izquierda a derecha, o de arriba hacia abajo, y en segundo lugar, de derecha a izquierda, y de abajo hacia arriba.

### 4.1.5 Diagrama Entidad-Relación

Un diagrama entidad relación, representa la disposición de las entidades y sus relaciones, en otras palabras, conforma de manera grafica la estructura de la base de datos, mostrando todos sus elementos (entidades, atributos de las entidades, y relaciones entre entidades).

### 4.1.6 Tipos de relaciones

Existen diferentes tipos de relaciones entre entidades:

Relación uno a uno

Ocurre cuando a un elemento de la entidad B, le corresponde uno y solo un elemento de la entidad A

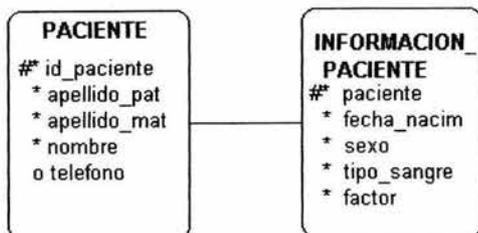


En realidad, este tipo de relación es poco común, y tiende a eliminarse por Normalización, de tal manera que los atributos formen parte de una sola Entidad. De esta manera, la entidad B desaparece, y sus atributos, forman parte de la entidad A.

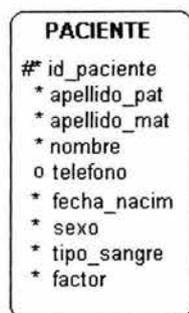


Ejemplo:

### DIAGRAMA ENTIDAD-RELACIÓN



- 1) De izquierda a derecha:  
Cada PACIENTE debe ser propietario de una y solo una INFORMACIÓN\_PACIENTE
- 2) De derecha a izquierda  
Cada INFORMACION\_PACIENTE debe pertenecer a uno y solo un PACIENTE



En realidad, Este tipo de relaciones, son muy poco comunes, y tienden a eliminarse. En este caso, lo más conveniente es eliminar la entidad INFORMACIÓN\_PACIENTE, de tal manera que sus atributos se encuentren ubicados en la entidad PACIENTE.

### Relación uno a muchos

Ocurre cuando a un elemento de la entidad B, le corresponden uno o más elementos de la entidad A

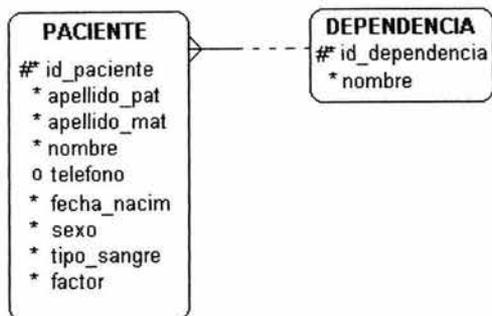


Este tipo de relación, es la más común en los diseños de bases de datos; tiende a ser obligatoria en un sentido, y opcional en el otro, es decir que cada elemento de la entidad A, debe (obligatoriamente) ser asignado a uno y solo un elemento de la entidad B, y cada elemento de la entidad B, puede (opcionalmente) estar asignado a uno o más elementos de la entidad A



Ejemplo:

### DIAGRAMA ENTIDAD-RELACION



- 1) De izquierda a derecha:  
Cada PACIENTE, debe estar perteneciendo a una y solo una DEPENDENCIA
- 2) De derecha a izquierda:  
Cada DEPENDENCIA puede ser responsable de uno o más PACIENTES

Relación muchos a muchos



Este tipo de relación se da cuando a uno o más elementos de la entidad B, le corresponden uno o más elementos de la entidad A. Este tipo de relaciones, no son soportadas por los manejadores de bases de datos relacionales, por lo que es necesario agregar una entidad de intersección entre las entidades con relación muchos a muchos, de tal manera que se genere una relación uno a muchos entre la entidad de intersección y cada una de las entidades involucradas en la relación muchos a muchos.

Resolución de relaciones muchos a muchos

Como se mencionó anteriormente, es necesaria una entidad de intersección entre las entidades con relación muchos a muchos.

Las entidades que tienen atributos suficientes para conformar una llave primaria, son conocidas como entidades fuertes o dominantes. Las entidades débiles o subordinadas, son aquellas que no tienen atributos suficientes para formar una llave primaria.

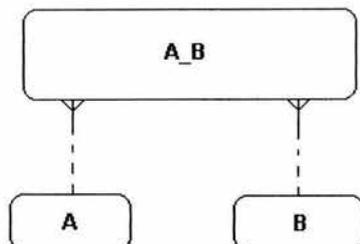
Concretamente, en el caso de las relaciones muchos a muchos, la entidad de intersección, es conocida como entidad débil o subordinada, porque toma los identificadores únicos de las entidades con relación muchos a muchos, en conjunto como llave primaria.



## Sistema de Información de Aeropuertos



De hecho, las llaves primarias de las entidades débiles, son conocidas como llaves primarias compuestas, debido a que están conformadas por más de un atributo.

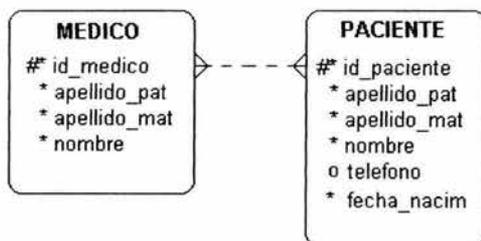


En este caso, la entidad **A\_B**, toma como identificador único, los identificadores únicos de las entidades **A** y **B** en conjunto, para formar su llave primaria.

Ejemplo: ¿Como sería la relación entre médico y paciente?, suponiendo que un paciente no tenga un médico asignado; es decir, que al momento de solicitar la consulta, a un paciente, se le asigne el médico que se encuentre disponible.

En este caso, se trata de una relación muchos a muchos:

### DIAGRAMA ENTIDAD-RELACION



1) De izquierda a derecha:

Cada **MEDICO** puede estar atendiendo a uno o más **PACIENTES**

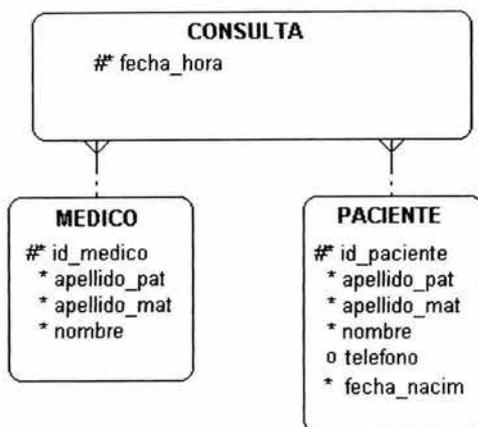
1) De derecha a izquierda:

Cada **PACIENTE**, puede ser atendido por uno o más **MEDICOS**

Es necesario, introducir una entidad de intersección, entre **MEDICO** y **PACIENTE**:



### DIAGRAMA ENTIDAD-RELACIÓN



En este caso, es necesario agregar el atributo adicional fecha\_hora, como parte de la llave primaria de la entidad CONSULTA, con el objeto de no duplicar la llave primaria, en el caso de que a un paciente lo atendiera el mismo médico, en más de una ocasión.

Relaciones recursivas uno a muchos:



Una relación recursiva, se da entre una misma entidad, es decir, existe una relación entre la misma entidad, ejemplo:

Se requiere obtener información de los médicos que laboran dentro de un hospital. Se toma su clave, nombre, y jefe inmediato. Cabe destacar que todos los médicos, tienen un solo jefe inmediato, y un jefe de médicos, tiene a su cargo a más de un médico, sin embargo, el director general del hospital, que también es médico, no tiene un jefe inmediato.

- 1) De izquierda a derecha:  
Cada MEDICO debe ser subordinado de uno y solo un MEDICO
- 2) De derecha a izquierda:  
Cada MEDICO, puede estar a cargo de uno o más MEDICOS

#### 4.1.7 Normalización

El proceso de normalización, permite evaluar a cada uno de los atributos, de cada una de las entidades, con el objeto de tener un modelo de datos eficiente, y sin redundancia.



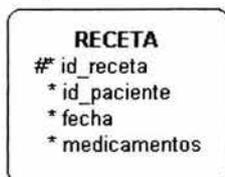
## Sistema de Información de Aeropuertos



Para normalizar una base de datos, existen reglas, conocidas como formas normales:

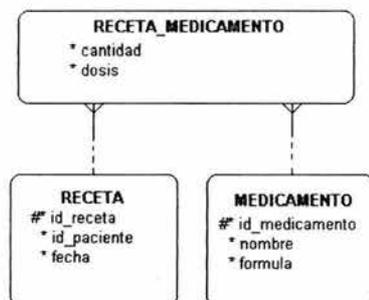
La Primera Forma Normal establece que todo atributo, debe tener un solo valor, es decir, un atributo no puede contener diferentes valores.

Ejemplo:



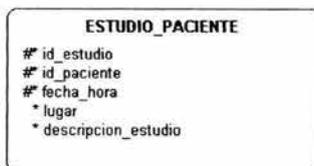
En este caso, el atributo `id_receta`, toma un solo valor, el número de la receta, es único; no existen dos recetas con el mismo identificador. El atributo `id_paciente`, también toma un solo valor, debido a que una receta, no puede ser para dos o más pacientes. El atributo `fecha`, toma un solo valor, debido a que una receta no tiene más de una fecha de emisión. El atributo `medicamentos`, no cumple la primera forma normal, debido a que puede tomar múltiples valores, es decir, una receta, puede tener varios medicamentos.

En este caso, es necesario reubicar el atributo `medicamentos`:



La segunda forma normal, establece que todo atributo, debe cumplir la primer forma normal, y que todo atributo que no sea identificador único de la entidad, debe depender completamente del identificador único de la entidad, y no solo de una parte de éste.

Ejemplo:





## Sistema de Información de Aeropuertos



En este caso, el identificador único de la entidad, está dado por los atributos `id_paciente`, `id_estudio` y `fecha_hora` en conjunto, por lo que los atributos no identificadores, en este caso, lugar y descripción del estudio, deben depender completamente del identificador único de la entidad.

En éste caso, el atributo `lugar`, depende del identificador único del paciente, sin embargo, el atributo `descripción_estudio`, no depende del identificador completo de la entidad, depende únicamente del atributo `id_estudio`, que es parte del identificador único de la entidad. En este caso, el atributo `descripción_estudio`, debe ser reubicado a la entidad ESTUDIO.

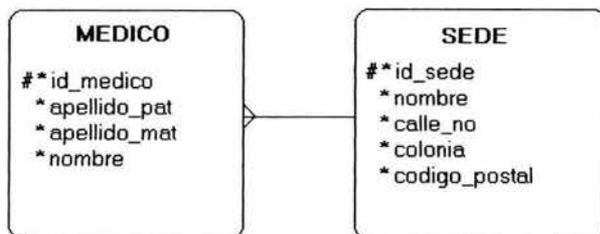
La tercera forma normal, establece que todo atributo, debe cumplir la segunda forma normal, y que todas las columnas que no sean identificador único de la entidad, no deben depender de ningún otro atributo que no sea identificador único de la entidad.

Ejemplo:



En este caso, todos los atributos, a excepción del atributo `nombre_sede`, dependen del identificador único de la entidad, que es el atributo `id_medico`. El atributo `nombre_sede`, depende del atributo `sede_adscpcion`, el cual, no es identificador único de la entidad, por lo que la tercera forma normal no se cumple en este caso.

El atributo `nombre_sede`, es reubicado a la entidad SEDE, ya que depende del identificador único de la entidad. En este caso, se reubicó, y se le cambió el nombre.





### 4.2 Lenguaje SQL

#### 4.2.1 Antecedentes

Antes de que existieran los sistemas Manejadores de Bases de Datos Relacionales, la información se manejaba por medio de sistemas de manejadores de archivos. A principios de la década de los 70, el Dr. Edgar F. Codd, quien trabajaba para IBM, propuso el modelo de Bases de Datos Relacional, basado en el Álgebra relacional, y en la teoría de conjuntos. IBM, denominó el proyecto de Codd Proyecto-R(R significaba Relational), y creó una división de investigación para dicho proyecto.

En este proyecto, se desarrolló un producto denominado Oracle, el cual utilizaba el lenguaje SQL (Structured Query Language), como lenguaje de manipulación de Datos.

En un principio, el proyecto obtuvo gran aceptación en el área de investigación en muchas universidades de Estados Unidos, sin embargo, comercialmente, no obtuvo los resultados esperados. En 1977, IBM se deshace del proyecto R, y vende todos los derechos a la empresa RSI (*Relational Software Inc*).

A finales de la década de los 70, RSI lanza el primer sistema manejador de bases de datos relacionales comercial: Oracle 2, el cual utilizó el lenguaje SQL para el manejo de información; para el año de 1983, RSI cambia de nombre a Oracle. Empiezan a surgir numerosos productos, que decían ser sistemas Manejadores de Bases de Datos Relacionales, sin realmente serlo; por esa razón, a mediados de los años 80, el ANSI (*American National Standard Institute*), toma las Reglas de Codd como un estándar que todos los sistemas Manejadores de Bases de Datos Relacionales deben cumplir. De la misma manera, el lenguaje SQL es estandarizado como el lenguaje universal de acceso a datos en todos los sistemas Manejadores de Bases de Datos Relacionales, en todas las plataformas.

#### 4.2.2 Estructura del Lenguaje SQL

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

##### 4.2.2.1 Comandos

Existen dos tipos de comandos SQL:

- los que permiten crear y definir nuevas bases de datos, campos e índices.

Comando	Descripción
CREATE	Utilizado para crear nuevas tablas, campos e índices
DROP	Empleado para eliminar tablas e índices
ALTER	Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.



## Sistema de Información de Aeropuertos



Los que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.

Comando	Descripción
SELECT	Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado
INSERT	Utilizado para cargar lotes de datos en la base de datos en una única operación.
UPDATE	Utilizado para modificar los valores de los campos y registros especificados
DELETE	Utilizado para eliminar registros de una tabla de una base de datos

### 4.2.2.2 Cláusulas

Las cláusulas son condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular.

Cláusula	Descripción
FROM	Utilizada para especificar las tablas de las cuales se van a seleccionar los registros
WHERE	Utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar
GROUP BY	Utilizada para separar los registros seleccionados en grupos específicos
HAVING	Utilizada para expresar la condición que debe satisfacer cada grupo
ORDER BY	Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico

### 4.2.2.3 Operadores Lógicos

Operador	Uso
AND	Es el "y" lógico. Evalúa dos condiciones y devuelve un valor de verdad sólo si ambas son ciertas.
OR	Es el "o" lógico. Evalúa dos condiciones y devuelve un valor de verdad si alguna de las dos es cierta.
NOT	Negación lógica. Devuelve el valor contrario de la expresión.



### 4.2.2.4 Operadores de Comparación

Operador	Uso
<	Menor que
>	Mayor que
<>	Distinto de
<=	Menor ó Igual que
=	Mayor ó Igual que
=	Igual que
BETWEEN	Utilizado para especificar un intervalo de valores.
LIKE	Utilizado en la comparación de un modelo
In	Utilizado para especificar registros de una base de datos

### 4.2.2.5 Funciones de Grupales

Las funciones de grupales se usan dentro de una cláusula SELECT en grupos de registros para devolver un único valor que se aplica a un grupo de registros.

Función	Descripción
AVG	Utilizada para calcular el promedio de los valores de un campo determinado
COUNT	Utilizada para devolver el número de registros de la selección
SUM	Utilizada para devolver la suma de todos los valores de un campo determinado
MAX	Utilizada para devolver el valor más alto de un campo especificado
MIN	Utilizada para devolver el valor más bajo de un campo especificado

## 4.3 JUSTIFICACION DEL DBMS

### 4.3.1 Recursos Disponibles en ASA

La elección del DBMS para el desarrollo, requirió investigar acerca de los diferentes productos disponibles en el mercado. En un principio se me planteó la posibilidad de desarrollar sobre Access, lo cual se consideró poco apropiado, por todas las limitantes que este manejador de bases de datos tiene. Por otra parte, estaban disponibles 2 de los manejadores de Sybase, el Adaptive Server Enterprise, el cual se descartó para el desarrollo de la aplicación, debido a que es un manejador de bases de datos relacionales demasiado grande para los propósitos de la aplicación que se va a desarrollar. Se tenía también el manejador Adaptive Server Anywhere de Sybase que se consideró mucho más apropiado ya que es un manejador de bases de datos relacionales de tamaño medio, apto para cliente servidor, e incluso para desarrollar sobre Web, ya sea utilizando el servidor Web de Sybase o el Internet Information Server de Windows. Sin embargo, hubo cambios en el departamento de sistemas de ASA, y los nuevos mandos entraron con nuevas ideas, sobre todo de ahorrar costos de licencias, ya que los costos de Sybase son elevados y no se cuenta con



## Sistema de Información de Aeropuertos



el presupuesto para la renovación de las mismas, por lo que el sistema deberá ser desarrollado en software libre sobre plataforma Linux.

En este caso, existen numerosos manejadores de bases de datos gratuitos para plataforma Linux. Dos de los más difundidos MySQL y PostgreSQL, por lo que la elección del manejador de base de datos más apropiado para el desarrollo se redujo a estos dos productos. Cabe destacar que existen productos como Adaptive Server Anywhere de Sybase, y algunas versiones de Oracle así como de Interbase disponibles en Linux, que tienen un costo de Licencia que aunque es menor que su contraparte de Windows no es software 100% gratuito.

### 4.3.2 Características de MySQL

Es un manejador de bases de datos de muy buen desempeño que consume pocos recursos de cpu y de memoria RAM, lo que lo hace ser un manejador de bases de datos muy estandarizado dentro de la plataforma Linux, debido a la velocidad de acceso que es superior en comparación con otros manejadores tales como PostgreSQL, el desempeño de MySQL puede decaer por un mal diseño de base de datos, o por tener un equipo pobre en cuanto a procesador y memoria RAM.

Es por eso que el modelo entidad relación sobre el cual se construya la base de datos deberá estar correctamente normalizado, con sus respectivas llaves primarias. MySQL contiene un registro de consultas lentas (*slow query log*) que permite monitorear el desempeño de las consultas a la base de datos. Un factor fundamental de MySQL es el uso de la memoria debido a que la aplicación es capaz de ejecutar varios subprocesos a la vez. Esto significa que cada vez que se realiza una conexión, MySQL crea un subproceso. Cada subproceso consume memoria. El almacenamiento en caché de los resultados también consume memoria.

Una característica muy importante de MySQL es su alta integración con el lenguaje PHP, ya que este contiene diferentes funciones preestablecidas que permiten simplificar enormemente el código PHP al realizar conexiones a la base de datos, sin embargo, a pesar de todas sus ventajas MySQL tiene algunos inconvenientes: No soporta transacciones, ni subconsultas, y en la versión 3.22 que es la más estable hasta el momento no están definidas las llaves foráneas a manera de *constraint*, por lo que es tarea del desarrollador realizar estos ajustes de tal manera que la integridad referencial de la base de datos nunca se pierda.

### 4.3.3 Características de PostgreSQL

PostgreSQL es un Sistema Manejador de Bases de Datos Relacionales, cuya primera versión de tipo Open Source salió en 1996. El proyecto PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel mundial, por lo que se liberan versiones nuevas continuamente. Es un Sistema de Bases de Datos Relacionales de gran tamaño y nivel, sus creadores comparan sus características con las de otros manejadores de bases de datos relacionales tales como Sybase u Oracle aunque en mi particular punto de vista es bastante la distancia entre dichos manejadores y PostgreSQL. Es un manejador de bases de datos relacionales de mayor nivel que MySQL en el sentido de que puede manipular una mayor cantidad de datos si perder estabilidad, esta planeado para sistemas de gran volumen de información. Este sistema manejador de bases de datos es 100% relacional, es decir, tiene muy bien definidos los *constraints* del tipo llave primaria y llave foránea, permite la ejecución de procedimientos almacenados y desencadenadores. Su principal desventaja en comparación con MySQL, es la velocidad de acceso a las tablas y la limitante de almacenar hasta 8kb por registro. Se han realizado diferentes estudios de desempeño de este manejador en comparación con MySQL, y las consultas en PostgreSQL tienden a ser 2 o 3 veces más lentas que en el caso de MySQL. Otra desventaja de PostgreSQL es la inestabilidad que presenta si se tiene un servidor con recursos limitados, en este sentido, una caída del sistema implica tener que volver a configurar todo desde el principio, ya que carece de herramientas para la recuperación de la configuración del servidor.



## Sistema de Información de Aeropuertos

---



Otra desventaja de PostgreSQL es su poca integración con PHP ya que no se tienen funciones como en el caso de MySQL que simplifique el trabajo del programador. Es común que las tablas con muchos campos (hablando de 30 o más) lleguen a tener problemas en los índices con el uso, en cuyo caso es necesario eliminar el índice y volverlo a generar o regenerarlo, operación que realiza las dos tareas al mismo tiempo.

### 4.3.4 Justificación de la elección de MySQL

El sistema que se va a desarrollar será un sistema de mediano tamaño, en donde se tendrán mucho menos de 10000 registros por tabla. Es muy importante que el desempeño de la aplicación sea bueno debido a que se ejecutará sobre Internet, en donde se tienen limitaciones por conexiones lentas. Otra característica del sistema es que será instalado en un servidor Linux Pentium III con 128 Mb en RAM, por lo que no se le puede exigir mucho en cuanto a carga de software. MySQL es mucho más utilizado que PostgreSQL por razones de desempeño en sistemas de mediano alcance, ya que no se requieren muchos recursos a nivel de hardware y es un manejador de bases de datos muy estable, soporta cargas de trabajo con tablas que tengan unos 50000 registros y unas 50 conexiones concurrentes a la base de datos.

La parte correspondiente a la integridad referencial de sistema, en cuanto a llaves foráneas será programada con código PHP, aunque se tiene ya una versión base de datos relacional en MySQL que admite *constraints* del tipo *foreign key*, por lo que se consideró que esta plataforma de desarrollo es la más adecuada para el tipo de sistema que se va a desarrollar.

## 4.4 ANALISIS

### 4.4.1 Entidades

Desglosando y analizando los requerimientos de información del capítulo 3, se identificaron las siguientes entidades:

**AEROPUERTO:** Es una terminal aérea tanto para el transporte de carga como para el transporte de pasajeros.

**CONTINENTE:** Es el catálogo que almacena los nombres de los continentes del mundo.

**REGION:** Es el catálogo que contiene las regiones en las cuales puede estar ubicado un país dentro de un continente.

**PAIS:** Esta entidad es en realidad el catálogo de países que almacena información relevante para la descripción de los mismos.

**ESTADO:** Es el catálogo de los estados o provincias que puede tener un país.

**LOCALIDAD:** Este catálogo contiene las ciudades o poblaciones en donde los AEROPUERTOS están ubicados.

**CLASIFICACION:** Es un catálogo que agrupa a los aeropuertos de acuerdo a su alcance.

**TIPO:** Es un catálogo que agrupa a los aeropuertos de acuerdo al tipo de población en donde se encuentran ubicados.



## Sistema de Información de Aeropuertos

---



**OBRA:** Es una entidad que contiene todas las obras tanto de remodelación como de ampliación realizadas a un AEROPUERTO.

**TIPO\_OBRA:** Es el catálogo de los tipos de obras que el sistema debe considerar para el AEROPUERTO.

**AVION:** Es la entidad que contiene las características principales de los aviones que aterrizan y despegan en los AEROPUERTOS.

**IMAGEN:** Es una entidad que contiene las fotografías que se tienen disponibles para un aeropuerto.

**PISTA:** Es la entidad que contiene las características de las pistas que tienen los AEROPUERTOS.

**TIPO\_PISO:** Es el catálogo que contiene los tipos de piso con el que puede estar conformada una PISTA.

**AYUDA\_ATERRIAJE:** Esta entidad contiene los tipos de ayudas para los aterrizajes de los AVIONES.

**TIPO\_AYUDA:** Es el catálogo que clasifica a las ayudas para los aterrizajes.

**PISTA:** Es el lugar destinado para el despegue y el aterrizaje de los aviones.

**AYUDA\_PISTA:** Es una entidad de intersección entre la entidad PISTA y la entidad AYUDA\_ATERRIAJE.

**RODAJE:** Entidad que describe las calles por las cuales los aviones transitan entre las pistas y las plataformas.

**PLATAFORMA:** Son los espacios asignados para la realización de diferentes actividades en el avión, tal como el ascenso y descenso de pasaje, carga, combustible, pernocta de aeronaves, etc.

**PLATAFORMA\_AEROPUERTO:** Es una entidad de intersección entre entidad PLATAFORMA y la entidad AEROPUERTO, ya que un AEROPUERTO puede tener varios tipos de plataformas y un tipo de plataforma la pueden tener diferentes aeropuertos.

**SUMINISTRO:** Es un catálogo que almacena las características principales de los diferentes sistemas de suministro de combustible para las aeronaves.

**SUMINISTRO\_AEROPUERTO:** Es una entidad de intersección entre la entidad AEROPUERTO y la entidad SUMINISTRO.

**CREI:** El CREI es el Cuerpo de Rescate y Extinción de Incendios del Aeropuerto, es decir, el cuerpo de bomberos que se tiene en el aeropuerto. Esta entidad es en realidad un catálogo de los tipos de sistemas de extinción que puede tener un aeropuerto.

**AEROPUERTO\_CREI:** Es la entidad de intersección entre AEROPUERTO y CREI

**SERVICIO:** Es un catálogo de los servicios que ofrece un aeropuerto a los pasajeros que hacen uso del mismo.



## Sistema de Información de Aeropuertos



**TIPO\_SERVICIO:** Es un catálogo auxiliar a la entidad SERVICIO que permite clasificar a los servicios de acuerdo al tipo, por ejemplo, el tipo de servicio de transportación terrestre puede abarcar taxis, metro, autobuses urbanos, autobuses foráneos, autobuses de alquiler, renta de autos, etc.

**UBICACIÓN\_SERVICIO:** Es un catálogo auxiliar a la entidad SERVICIO\_AEROPUERTO que permite clasificar la ubicación a los servicios, por ejemplo: interior del edificio terminal, exterior del edificio terminal, zonas aledañas, etc.

**SERVICIO\_AEROPUERTO:** Es una entidad de intersección entre la entidad AEROPUERTO y la entidad SERVICIO.

**RESPONSABLE:** Describe a los organismos tanto gubernamentales como privados encargados de operar y/o administrar los aeropuertos.

**TIPO\_RESPONSABLE:** Catálogo que permite indicar si el responsable es gubernamental o privado.

**ROL\_RESPONSABLE:** Catálogo que indica si un responsable es operador o administrador del aeropuerto, o ambos como en el caso de ASA.

**RESPONSABLE\_AEROPUERTO:** Entidad de intersección entre la entidad AEROPUERTO y la entidad RESPONSABLE. El responsable depende de cada país.

### 4.4.2 Relaciones

Las relaciones que existen entre cada una de estas entidades se describen a continuación:

#### PAIS y CONTINENTE

- Cada PAIS debe **pertenecer a** uno y sólo un CONTINENTE
- Cada CONTINENTE puede **contener a** uno o más PAISES

#### PAIS y REGION

- Cada PAIS debe **pertenecer a** una y sólo una REGION
- Cada REGION puede **contener a** uno o más PAISES

#### PAIS y ESTADO

- Cada ESTADO debe **pertenecer a** uno y sólo un PAIS
- Cada PAIS puede **contener a** uno o más ESTADOS

#### LOCALIDAD y ESTADO

- Cada LOCALIDAD debe **pertenecer a** uno y sólo un ESTADO
- Cada ESTADO puede **contener a** una o más LOCALIDADES

#### AEROPUERTO y LOCALIDAD

- Cada AEROPUERTO debe **pertenecer a** una y sólo una LOCALIDAD
- Cada LOCALIDAD puede **contener a** uno o más AEROPUERTOS

#### AEROPUERTO y TIPO

- Cada AEROPUERTO debe **corresponder a** uno y sólo un TIPO
- Cada TIPO puede **clasificar a** uno o más AEROPUERTOS

#### AEROPUERTO y CLASIFICACION

- Cada AEROPUERTO debe **corresponder a** una y sólo una CLASIFICACION
- Cada CLASIFICACION puede **contener a** uno o más AEROPUERTOS



## Sistema de Información de Aeropuertos



### AEROPUERTO y OBRA

Cada OBRA debe **corresponder a** uno y sólo un AEROPUERTO  
Cada AEROPUERTO puede **agrupar a** una o más OBRAS

### OBRA y TIPO\_OBRA

Cada OBRA debe **corresponder a** uno y sólo un TIPO\_OBRA  
Cada TIPO\_OBRA puede **clasificar a** una o más OBRAS

### IMAGEN y AEROPUERTO

Cada IMAGEN debe **corresponder a** uno y sólo un AEROPUERTO  
Cada AEROPUERTO puede **contener a** una o más IMAGENES

### RODAJE y AEROPUERTO

Cada RODAJE debe **corresponder a** uno y sólo un AEROPUERTO  
Cada AEROPUERTO puede **contener a** uno o más RODAJES

### PISTA y AEROPUERTO

Cada PISTA debe **pertenecer a** uno y sólo un AEROPUERTO

Cada AEROPUERTO puede **contener a** una o más PISTAS

### PISTA y TIPO\_PISO

Cada PISTA debe **corresponder a** uno y sólo un TIPO\_PISO

Cada TIPO\_PISO puede **clasificar a** una o más PISTAS

### PISTA y AVION

Cada PISTA debe **corresponder a** uno y sólo un AVION

Cada AVION puede **corresponder a** una o más PISTAS

Nota: Aparentemente esta es una relación muchos a muchos, debido a que en una pista aterrizan varios aviones, y un avión puede aterrizar en muchas pistas, sin embargo, para fines del sistema, sólo se requiere conocer el avión máximo que puede operar en una pista, ya que por ejemplo, una pista que soporte un Boeing 737, que es el avión más grande operado por aerolíneas en México, puede soportar un avión más pequeño, por ejemplo un Boeing 717

### PISTA y AYUDA\_ATERRIZAJE

Cada PISTA puede **contener a** una o más AYUDAS\_ATERRIZAJE

Cada AYUDA\_ATERRIZAJE puede **pertenecer a** una o más PISTAS

Nota: Existe una relación muchos a muchos que es necesario romper utilizando la entidad de intersección AYUDA\_PISTA

### AYUDA\_ATERRIZAJE y TIPO\_AYUDA

Cada AYUDA\_ATERRIZAJE debe **corresponder a** un y sólo un TIPO\_AYUDA

Cada TIPO\_AYUDA puede **clasificar a** una o más AYUDAS\_ATERRIZAJE

### PLATAFORMA y AEROPUERTO

Cada PLATAFORMA puede **pertenecer a** uno o más AEROPUERTOS

Cada AEROPUERTO puede **contener a** una o más PLATAFORMAS

Nota: Existe una relación muchos a muchos que es necesario romper utilizando la entidad de intersección PLATAFORMA\_AEROPUERTO

### SUMINISTRO y AEROPUERTO

Cada SUMINISTRO puede **pertenecer a** uno o más AEROPUERTOS



## Sistema de Información de Aeropuertos



Cada AEROPUERTO puede **contener a** uno o más SUMINISTROS

Nota: Existe una relación muchos a muchos que es necesario romper utilizando la entidad de intersección SUMINISTRO\_AEROPUERTO

SERVICIO y TIPO\_SERVICIO

Cada SERVICIO debe **corresponder a** uno y sólo un TIPO\_SERVICIO

Cada TIPO\_SERVICIO puede **clasificar a** uno o más SERVICIOS

CREI y AEROPUERTO

Cada CREI puede **pertenecer a** uno o más AEROPUERTOS

Cada AEROPUERTO puede **contener** uno o más CREI

Nota: Existe una relación muchos a muchos que es necesario romper utilizando la entidad de intersección CREI\_AEROPUERTO

SERVICIO y AEROPUERTO

Cada SERVICIO puede **pertenecer a** uno o más AEROPUERTOS

Cada AEROPUERTO puede **contener** uno o más SERVICIOS

Nota: Existe una relación muchos a muchos que es necesario romper utilizando la entidad de intersección SERVICIO\_AEROPUERTO

RESPONSABLE y AEROPUERTO

Cada RESPONSABLE puede **administrar u operar a** uno o más AEROPUERTOS

Cada AEROPUERTO puede **ser administrado u operado** por uno o más RESPONSABLES

Nota: Existe una relación muchos a muchos que es necesario romper utilizando la entidad de intersección RESPONSABLE\_AEROPUERTO. Aparentemente se debería tratar esta relación como una relación de uno a muchos, ya que en países como México, un aeropuerto es operado y administrado por un sólo organismo, sin embargo existen países en donde un aeropuerto puede ser operado por un organismo y administrado por otro, y a su vez esos organismos pueden operar y/o administrar diferentes aeropuertos. Cabe destacar que también la entidad TIPO\_ROL\_RESPONSABLE interviene en la relación muchos a muchos, es decir existe una relación muchos a muchos entre AEROPUERTO, RESPONSABLE y TIPO\_ROL\_RESPONSABLE en el sentido que un organismo responsable puede operar un aeropuerto, y administrar otro, o incluso operar y administrar algún otro aeropuerto.

RESPONSABLE y TIPO\_RESPONSABLE

Cada RESPONSABLE debe **corresponder a** uno y sólo un TIPO\_RESPONSABLE

Cada TIPO\_RESPONSABLE puede **clasificar a** uno o más RESPONSABLES

### 4.4.3 Atributos

Los atributos para cada una de las entidades, así como, las llaves primarias, foráneas e índices y observaciones adicionales se muestran a continuación:

Simbología:

\* Obligatorio

o Opcional

# Único



## Sistema de Información de Aeropuertos



### CONTINENTE

CAMPO	TIPO	OBSERVACIONES
#* Id_continente	NUMÉRICO	Identificador de la entidad, autoincrementable
* Nombre	TEXTO(20)	Nombre del continente

### REGION

CAMPO	TIPO	OBSERVACIONES
#* Id_region	NUMÉRICO	Identificador de la entidad, autoincrementable
* Nombre	TEXTO(20)	Nombre de la región

### PAIS

CAMPO	TIPO	OBSERVACIONES
#* Id_pais	NUMÉRICO	Identificador de la entidad, autoincrementable
# *Nombre	TEXTO(40)	Nombre del país, índice único
* Superficie	NUMÉRICO	Superficie del país en Km. cuadrados
* Población	NUMÉRICO	Número aproximado de habitantes
O PIB	NUMÉRICO	Producto interno bruto del país en el último año
O admón._aerea	TEXTO(255)	Describe como cada país hace uso de su espacio aéreo y los organismos encargados de regir el mismo

### ESTADO

CAMPO	TIPO	OBSERVACIONES
Id_estado	NUMÉRICO	Identificador de la entidad, autoincrementable
Nombre	TEXTO(20)	Nombre de l estado

### LOCALIDAD

CAMPO	TIPO	OBSERVACIONES
Id_localidad	NUMÉRICO	Identificador de la entidad, autoincrementable
Nombre	TEXTO(20)	Nombre de la localidad
Superficie	NUMÉRICO	Superficie en metros cuadrados de la localidad
Población	NUMÉRICO	Número de habitantes que tiene la localidad
Comentarios	TEXTO(200)	Campo opcional para datos adicionales



## Sistema de Información de Aeropuertos



### AEROPUERTO

CAMPO	TIPO	OBSERVACIONES
Siglas_3	TEXTO(3)	Identificador de la entidad, clasificación IATA
Siglas_4	TEXTO(4)	Clasificación OACI, índice único
Nombre	TEXTO(60)	Nombre del aeropuerto
Fecha_inauguracion	FECHA	Fecha en la que fue inaugurado
Capacidad_diaria_pasaje	NÚMERO	Número máximo de pasajeros por día
Capacidad_oper_hora	NÚMERO	Número máximo de operaciones por hora
No_edificio_terminal	NÚMERO	Número de edificios terminales de pasajeros
Superficie_edificio	NÚMERO	Superficie total en metros cuadrados de todos los edificios terminales
No_estacionamientos	NÚMERO	Número de estacionamientos
No_lugar_estacionamiento	NÚMERO	Número De cajones de estacionamiento
Superficie_estacionamiento	NÚMERO	Superficie en metros cuadrados totales de los estacionamientos
Web	TEXTO(50)	Dirección electrónica del aeropuerto
Distancia_ciudad	NUMÉRICO	Es la distancia en kilómetros del aeropuerto a la ciudad
Tiempo_ciudad	NUMÉRICO	Es el tiempo aproximado de recorrido del aeropuerto a la ciudad
Latitud	NUMÉRICO	Coordenada geográfica de latitud
Longitud	NUMÉRICO	Coordenada geográfica de longitud
Altitud	NUMÉRICO	Altura sobre el nivel del mar
Comentarios	TEXTO(200)	Campo opcional para datos adicionales

### OBRA

CAMPO	TIPO	OBSERVACIONES
Id_obra	NUMÉRICO	Identificador de la entidad, autoincrementable
Descripcion	TEXTO(100)	Datos relevantes de la obra
Fecha_inicio	FECHA	Fecha en la que se inicio la obra
Fecha_fin	FECHA	Fecha de conclusión de la obra

### TIPO\_OBRA

CAMPO	TIPO	OBSERVACIONES
Id_tipo_obra	NUMÉRICO	Identificador de la entidad, autoincrementable
Descripcion	TEXTO(50)	Describe los tipos de obra considerados en el sistema

### TIPO

CAMPO	TIPO	OBSERVACIONES
Id_tipo	NUMÉRICO	Identificador de la entidad, autoincrementable
Descripcion	TEXTO(30)	Describe los tipos de aeropuertos



## Sistema de Información de Aeropuertos



### CLASIFICACION

CAMPO	TIPO	OBSERVACIONES
Id_clasificacion	NUMÉRICO	Identificador de la entidad, autoincrementable
Descripcion	TEXTO(30)	Describe la clasificación de aeropuertos

### IMAGEN

CAMPO	TIPO	OBSERVACIONES
Id_imagen	NUMÉRICO	Identificador de la entidad, autoincrementable
Descripcion	TEXTO(100)	Describe la fotografía mostrada
Vinculo	TEXTO(100)	Vinculo Entre la imagen mostrada y el archivo en donde esta fisicamente ubicada

### AVION

CAMPO	TIPO	OBSERVACIONES
Id_avion	NUMÉRICO	Identificador de la entidad, autoincrementable
Modelo	TEXTO(100)	Nombre del modelo del avión
Num_max_pasaje	NUMÉRICO	Es el número máximo de pasajeros estándar en una clase para el modelo del avión
Capacidad_max_combustible	NUMÉRICO	Es la capacidad máxima de combustible que puede suministrarse a este avión
Peso_max_despegue	NUMÉRICO	Es el peso máximo que puede tener el avión para despegar considerando que esta al nivel del mar
Rango_max_operación	NUMÉRICO	Es el alcance máximo que puede tener el avión sin que se le suministre combustible
Envergadura	NUMÉRICO	Es la distancia que existe entre las dos alas del avión
Largo_total	NUMÉRICO	Es la longitud que tiene el avión
Ancho_total	NUMÉRICO	Es el ancho interior que tiene el avión
Alto_total	NUMÉRICO	Es la distancia entre el alerón trasero y el piso
Imagen	TEXTO(100)	Vinculo Entre la imagen mostrada y el archivo en donde esta fisicamente ubicada

### PISTA

CAMPO	TIPO	OBSERVACIONES
Id_pista	NUMÉRICO	Identificador de la entidad, autoincrementable
Designación	TEXTO(15)	Es la designación de la pista en base a sus coordenadas geográficas
Largo	NUMÉRICO	Longitud de la pista en metros
Ancho	NUMÉRICO	Ancho de la pista en metros

### TIPO\_PISO

CAMPO	TIPO	OBSERVACIONES
Id_tipo_piso	NUMÉRICO	Identificador de la entidad, autoincrementable
Descripcion	TEXTO(50)	Es la descripción de los tipos de piso que puede tener una pista



## Sistema de Información de Aeropuertos



### TIPO\_AYUDA

CAMPO	TIPO	OBSERVACIONES
Id_tipo_ayuda	NUMÉRICO	Identificador de la entidad, autoincrementable
Descripcion	TEXTO(50)	Es la descripción de los tipos de ayuda que puede tener una pista

### AYUDA\_ATERRIZAJE

CAMPO	TIPO	OBSERVACIONES
Id_ayuda_aterrizaje	NUMÉRICO	Identificador de la entidad, autoincrementable
Descripcion	TEXTO(50)	Es la descripción de las ayudas que puede tener una pista

### AYUDA\_PISTA

CAMPO	TIPO	OBSERVACIONES
Id_pista	NUMÉRICO	Identificador compuesto de la entidad
Id_ayuda	NUMÉRICO	Identificador compuesto de la entidad
Características	TEXTO(200)	Datos específicos para la ayuda en la pista

### RODAJE

CAMPO	TIPO	OBSERVACIONES
Id_rodaje	NUMÉRICO	Identificador de la entidad autoincrementable
Designacion	TEXTO(20)	Designación para la calle de rodaje
Largo	NUMÉRICO	Longitud de la calle de rodaje
Ancho	NUMÉRICO	Ancho de la calle de rodaje
Luces_borde	LÓGICO	Especifica si el rodaje tiene o no luces en el borde
Señalamiento	LÓGICO	Especifica si el rodaje tiene o no señalamientos

### PLATAFORMA

CAMPO	TIPO	OBSERVACIONES
Id_plataforma	NUMÉRICO	Identificador de la entidad autoincrementable
Descripcion	TEXTO(20)	Descripción

### PLATAFORMA\_AEROPUERTO

CAMPO	TIPO	OBSERVACIONES
Id_plataforma	NUMÉRICO	Identificador compuesto de la entidad
Siglas_3	NUMÉRICO	Identificador compuesto de la entidad
Superficie	NUMÉRICO	Especifica la superficie total de las plataformas de un sólo tipo
Cantidad	NUMÉRICO	Especifica la cantidad de plataformas que se tienen en un aeropuerto por tipo
Notas	TEXTO(200)	Datos específicos de la plataforma en el Aeropuerto

### CREI

CAMPO	TIPO	OBSERVACIONES
Id_crei	NUMÉRICO	Identificador de la entidad autoincrementable
Descripcion	TEXTO(20)	Descripción
Imagen	TEXTO(100)	Ruta de la imagen del CREI



## Sistema de Información de Aeropuertos



### CREI\_AEROPUERTO

CAMPO	TIPO	OBSERVACIONES
Id_crei	NUMÉRICO	Identificador compuesto de la entidad
Siglas_3	TEXTO(3)	Identificador compuesto de la entidad
Características	TEXTO(200)	Características del CREI por aeropuerto

### SUMINISTRO

CAMPO	TIPO	OBSERVACIONES
Id_suministro	NUMÉRICO	Identificador de la entidad autoincrementable
Descripción	TEXTO(20)	Descripción
Imagen	TEXTO(100)	Ruta de la imagen del suministro

### SUMINISTRO\_AEROPUERTO

CAMPO	TIPO	OBSERVACIONES
Id_suministro	NUMÉRICO	Identificador compuesto de la entidad
Siglas_3	TEXTO(3)	Identificador compuesto de la entidad
Características	TEXTO(200)	Características del Suministro por aeropuerto

### SERVICIO

CAMPO	TIPO	OBSERVACIONES
Id_servicio	NUMÉRICO	Identificador de la entidad autoincrementable
Descripción	TEXTO(20)	Descripción

### TIPO\_SERVICIO

CAMPO	TIPO	OBSERVACIONES
Id_tipo_servicio	NUMÉRICO	Identificador de la entidad autoincrementable
Descripción	TEXTO(20)	Descripción del tipo de servicio

### UBICACIÓN\_SERVICIO

CAMPO	TIPO	OBSERVACIONES
Id_ubicación_servicio	NUMÉRICO	Identificador de la entidad autoincrementable
Descripción	TEXTO(20)	Descripción de la ubicación del servicio

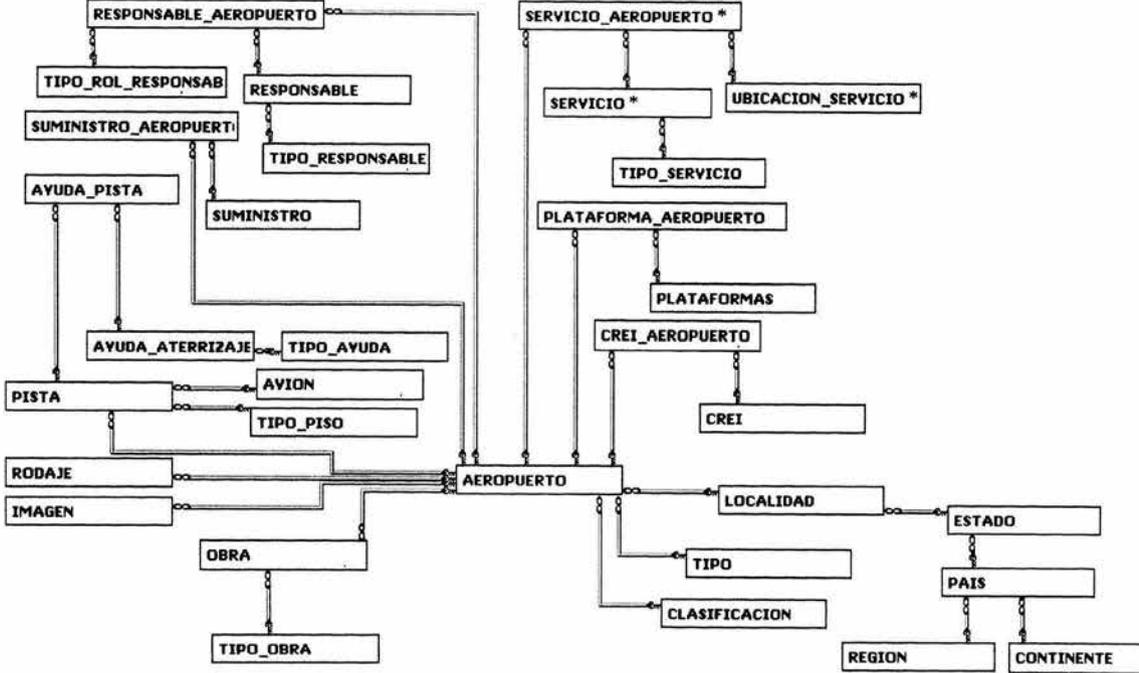
### SERVICIO\_AEROPUERTO

CAMPO	TIPO	OBSERVACIONES
Id_servicio	NUMÉRICO	Identificador compuesto de la entidad
Id_ubicacion	NUMERICO	Identificador compuesto de la entidad
Siglas_3	TEXTO(3)	Identificador compuesto de la entidad
Características	TEXTO(200)	Características del Servicio en el aeropuerto



4.4.4 Diseño

De acuerdo al análisis realizado en el punto anterior, el diseño de la base de datos es el siguiente:





### 5. ANALISIS, DISEÑO Y DESARROLLO DE LA APLICACIÓN.

#### 5.1 PROGRAMACIÓN ESTRUCTURADA

Como se mencionó anteriormente, la programación estructurada surgió en los años 70's, con el fin de tener las herramientas necesarias para el desarrollo de programas de calidad. Se compone de estructuras de control, técnicas de refinación sucesiva y el concepto de modularidad.

El teorema de la programación estructurada, conocido como teorema de Böhm y Jacopini expresa que un programa propio puede ser escrito utilizando únicamente tres tipos de estructuras de control de secuencia: Secuenciales, de decisión (selectivas) y de repetición (bucles). Las estructuras secuenciales se caracterizan porque una acción se ejecuta una tras otra. Las estructuras de decisión permiten al programa realizar diversas acciones en función de que una o más condiciones se cumplan o no. Los bucles son estructuras de datos que permiten la repetición de un bloque de instrucciones mientras una condición se cumpla. Las técnicas de refinación sucesiva de la solución permiten dividir un problema en partes y resolver cada parte del problema para posteriormente integrar la solución.

El primer paso es entender el problema a resolver, analizarlo y dividirlo en subpartes, identificando en cada una de estas los datos de entrada y salida, implementar los algoritmos necesarios para la codificación de cada una de las partes en las cuales fue dividido el problema, codificar los algoritmos en el lenguaje seleccionado, y verificar que no tengan errores asegurándose de obtener los resultados esperados. En la fase de diseño se utiliza el método Arriba - Abajo (*top-down*) para la elaboración del algoritmo que de solución al problema. El método *top-down* comienza con la formulación más abstracta del problema, trabajando la refinación hacia las partes más concretas y más detalladas de los subproblemas que componen el todo. También existe el enfoque inverso, de lo más concreto a lo más abstracto para resolver el todo, mediante la solución de sus partes.

Un módulo es un conjunto de sentencias encargadas de resolver una tarea o rutina que se repite frecuentemente en la solución de un problema. Tiene cierto grado de independencia del programa principal, resolviendo una parte del problema. Un módulo en un lenguaje de programación puede ser una función si devuelve un valor o una función si no devuelve valores. La idea de la técnica de la refinación progresiva es dividir en problema en subproblemas más sencillos de tal manera que la solución de problema esté dada por el conjunto de soluciones de los subproblemas.

La programación estructurada, pese al tiempo que tiene de haberse implementado, sigue siendo utilizada en los lenguajes actuales, tanto los orientados a objetos como c++ ó java, lenguajes visuales como visual Basic o Delhi, en donde los eventos están conformados de estructuras, incluso en las tecnologías de desarrollo en Web, como perl, java, javascript, asp. php, jsp, etc.

#### 5.2 JUSTIFICACIÓN DE LA PLATAFORMA DE DESARROLLO

La elección del lenguaje de programación mas adecuado para el desarrollo de esta aplicación implica un análisis de la vasta gama de tecnologías disponibles para el desarrollo de aplicaciones Web gratuitas son muy bastas actualmente. La herramienta de desarrollo debe ser una herramienta que trabaje a nivel de servidor, es decir, deberá ser interpretada en un servidor Web y no en un navegador, sin embargo, no se descarta el uso de lenguajes que se ejecutan sobre el lado del navegador para manejo de eventos y de validaciones hacia el usuario, y aunque dichas validaciones y manejo de eventos pueden hacerse desde el lado del servidor, es más óptimo el desempeño de una aplicación validada antes de ser interpretada por un servidor Web Linux.



## Sistema de Información de Aeropuertos



### 5.2.1 Plataforma Linux



Linux es un sistema operativo con las características de Unix desarrollado por Linus Torvalds como proyecto de tesis, y actualmente desarrollado por miles de programadores en todo el mundo, los cuales pertenecen a la GNU (*General Public Licence*). Este sistema operativo, es del tipo *Open Source*, lo que significa que es el código fuente es abierto a cualquier usuario. La idea original del sistema era tener un UNIX disponible para computadoras personales. Fue desarrollado originalmente en una máquina 386 Intel, y actualmente se ejecuta en cualquier procesador Intel, AMD, Cyrix, incluso en el PowerPC de Macintosh y procesadores Motorola.

Es un sistema operativo multitarea, multiusuario, multiplataforma y multiprocesador. En los sistemas Intel se ejecuta bajo el concepto de modo protegido en donde resguarda la memoria para que un programa no pueda hacer que el sistema se detenga. Este sistema permite compartir la memoria entre varios programas, y la memoria libre para el caché.

Las versiones se numeran como x.y.z; aquellas con y par son versiones estables; aquellas con y impar son versiones de desarrollo. Este modelo de desarrollo abierto permite tener disponible siempre la última versión, y las nuevas versiones se liberan en pocas semanas o meses. En ocasiones las funciones nuevas contienen errores, pero con cientos de personas en Internet probando y leyendo el código fuente suelen corregirse en cuestión de horas.

Esto permite tener un sistema operativo de muy alta calidad, menos errores que los sistemas operativos comerciales como Windows, gratuito o a un costo significativamente bajo y con aplicaciones de diversa índole como las aplicaciones Windows, software de oficina, aplicaciones en ingeniería, diseño, lenguajes de programación, bases de datos, etc.

Debido a la eficacia de Linux en el manejo de recursos de Hardware, éste requiere poco hardware, aunque el modo gráfico consume una mayor cantidad de recursos que el modo de comandos. Linux es una muy excelente alternativa frente a los sistemas operativos de la familia Windows, ya que aparte de ser un sistema que cuesta mucho menos dinero, es superior respecto al manejo de múltiples tareas y administración de memoria, conectividad en redes y seguridad.

### 5.2.2 Distribuciones Linux

Existen diferentes versiones del sistema operativo Linux conocidas como distribuciones. Todas las distribuciones tienen el mismo kernel que se desarrolla de manera conjunta. Las diferencias entre las distribuciones son muy significativas, ya que sus características de instalación, ambiente, utilidades y herramientas de configuración son variadas.

#### 5.2.2.1 Distribución Red Hat



Es distribuida por la compañía Red Hat Software en Connecticut, EE.UU. Es la distribución más difundida y sólida de Linux, viene acompañada de un ambiente gráfico por medio de los entornos KDE y Gnome que simplifica enormemente las tareas de instalación, mantenimiento, actualización de las últimas versiones y utilización del sistema operativo. Está dirigida principalmente a empresas. Tiene muy buen manejo de las actividades relacionadas con las redes de computadores, mejoras relacionadas con la administración de servidores Web y el paquete de programas StarOffice 5.1, con las mismas funciones que el Office de Microsoft y con la posibilidad de leer y escribir archivos en dicho formato. Aunque existen otras distribuciones como Slackware,



## Sistema de Información de Aeropuertos



Mandrake, Caldera, Gentoo, etc, se determinó que es la distribución que más se adapta a los requerimientos del Sistema de Información de Aeropuertos, debido a su estabilidad, confiabilidad y

sobre todo escalabilidad con respecto a otras distribuciones Linux. Una característica muy importante de esta distribución es que integra al Servidor Apache con el lenguaje PHP y el manejador de base de datos MySQL.

### 5.2.3 El Servidores Web

Apache



Un servidor Web permite obtener el acceso a la información de un sitio. Al escribir la dirección correspondiente en el navegador, éste solicita el texto e imágenes al servidor que desplegará posteriormente. Estas peticiones son enviadas por medio de Internet a un Servidor por medio de computadoras remotas ubicadas en cualquier parte del mundo. El servidor Web correspondiente a la dirección solicitada analiza la petición devolviendo los archivos necesarios para que el navegador de la computadora remota pueda hacer uso de éstos, y los pueda desplegar. En el caso de datos dinámicos que provengan de alguna base de datos, el servidor se encarga de ejecutar las aplicaciones, obteniendo a partir de datos variables una salida en formato html que puedan interpretar todos los navegadores. Apache se basó originalmente en el servidor http NCSA; Apache es un sistema que puede rivalizar y probablemente sobrepasar a casi cualquier otro servidor basado en UNIX HTTP en cuanto a funcionalidad, eficacia y rapidez. Apache es actualmente, el servidor WWW más popular en Internet, según el Netcraft Survey. El nombre de Apache viene de "A PATChy server", (Un servidor lleno de remiendos). Estaba basado en alguna codificación existente y en una serie de archivos "parche". Apache es uno de los servidores Web más utilizados a nivel mundial, dada su alta estabilidad, eficacia y seguridad. Es un servidor gratuito con características muy superiores de desempeño y seguridad a otros servidores, como el Internet Information Server. Este servidor continuamente se está actualizando, por lo que se adapta a los últimos protocolos http. Apache no es el único servidor Web disponible en el mercado, de hecho, fue creado a partir del servidor Web de NCSA.

NCSA

NCSA fue uno de los primeros servidores Web disponibles en el mercado, en un servidor relativamente seguro, rápido, y compatible con navegadores nuevos y pasados. Permite un fácil manejo de directorios, el manejo de scripts basados en CGI, establecer seguridad a nivel de directorios, con diferentes esquemas de seguridad y manejar hospedajes virtuales (*Virtual Host*) por medio de sus interfaces.

Netscape Communications Server

Un Servidor Web interesante es el Netscape Communications Server de la compañía Netscape, el cual es un Servidor Web fácil de usar, que permite proteger las comunicaciones privadas en línea por medio de certificados utilizando el protocolo abierto SSL (*Secure Sockets Layer*) para seguridad en Internet. El *Netscape Certificate Server* es un software para servidor del Web de plataforma universal que se basa en normas abiertas de Internet, que incluyen HTML, HTTP, SSL 3.0 y PKCS (*Public Key Cryptography Standards*). Está disponible para las principales plataformas UNIX y para Windows NT a un costo aproximado de \$995 dólares.



### Internet Information Server

Otro servidor sumamente utilizado es el *Internet Information Server* de Microsoft, disponible únicamente para plataforma Windows. Es un servidor muy fácil de implementar y de operar que por defecto maneja páginas asp, aunque puede ser capaz de manipular páginas php. Su principal desventaja en comparación con los servidores Web antes mencionados es la seguridad, ya que se tienen significativos huecos en cuanto al control y autenticación de usuarios. De hecho los servidores Web mas fácilmente atacados son los servidores Windows IIS.

### 5.2.4 Tecnologías para el desarrollo de aplicaciones Web

#### 5.2.4.1 CGI(*Common Gateway Interface*)

CGI es una tecnología que permite establecer la comunicación entre un servidor Web y un programa (script) de tal modo que pueda haber una interacción entre un navegador y un servidor Web. Un CGI es un programa que se ejecuta en tiempo real en un servidor Web en respuesta a la solicitud de un navegador. Cuando esto sucede el servidor ejecuta un proceso que recibirá los datos que envía el usuario, pone a disposición del mismo algunos datos en forma de variables de ambiente y captura la salida del programa para enviarlo como respuesta al navegador. El propósito de los CGI's es proveer "inteligencia" e interactividad a un sitio Web.

#### 5.2.4.2 Perl

Es un lenguaje interpretado que tiene varias utilidades, pero está principalmente orientado a la búsqueda, extracción y formateado de archivos de tipo texto. Es una combinación de las características de los lenguajes utilizados en el sistema operativo UNIX como C, incluso tiene características de Pascal. Perl está disponible en varios entornos como Unix, o Windows, es uno de los lenguajes más utilizados en la programación de scripts de CGI para intercambio de información entre aplicaciones de Internet. En un principio los programas de búsqueda fueron realizados utilizando este tipo de tecnología, aunque actualmente existen tecnologías que permiten el desarrollo de las aplicaciones Web de manera más sencilla y rápida.

#### 5.2.4.3 ASP (*Active Server Pages*)

Es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor *Internet Information Server*. El código ASP se escribe en la misma página Web intercalado con el código html utilizando el lenguaje Visual Basic Script o Javascript de Microsoft. Se ejecuta del lado del servidor justo antes de que se envíe la página al cliente (navegador) a través de Internet. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución del PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores. El tipo de servidores que emplean este lenguaje son, evidentemente, todos aquellos que funcionan con sistema Windows NT/2000 Server. Incluso en sistemas Linux es posible utilizar ASP si se instala un componente denominado Chilisoft.

#### 5.2.4.4 JSP (*Java server Pages*)

JSP es una tecnología basada en Java que simplifica el proceso de desarrollo de sitios Web dinámicos. Los archivos JSP son de texto y sustituyen a las páginas HTML tradicionales. Contienen contienen etiquetas HTML intercaladas con código JSP, el cual se ejecuta del lado del servidor. JSP se implementa utilizando la tecnología Servlet. Cuando un servidor Web recibe una petición de una página .jsp, la redirecciona a un proceso especial dedicado a manejar la ejecución



## Sistema de Información de Aeropuertos



de servlets (servlet container) llamado JSP container. Las páginas jsp tienen soporte de componentes reutilizables (*JavaBeans*).

Los *JavaBeans* utilizados en páginas .jsp pueden ser utilizados en servlets, applets o aplicaciones Java. Para poder utilizar esta tecnología es necesario un servidor Web que soporte a páginas .html y código que implemente un contenedor JSP donde ejecutar las etiquetas JSP. Existen servidores Web que incorporan dicha capacidad dentro de su código (Netscape Enterprise y Application Server 4.0) así como servidores escritos íntegramente en Java (Java Web Server de Sun y Jigsaw de W3 Consortium) que dan soporte directamente a esta tecnología.

### 5.2.4.5 PHP (HiperText Preprocesor)

PHP fue originalmente desarrollado en el año de 1994 por Rasmus Lerdorf como un CGI escrito en C que permitía la interpretación de un número limitado de comandos. El sistema fue denominado Personal Home Page Tools y adquirió relativo éxito gracias a que otras personas pidieron a Rasmus que les permitiese utilizar sus programas en sus propias páginas. Dada la aceptación del primer PHP y de manera adicional, su creador diseñó un sistema para procesar formularios al que le atribuyó el nombre de FI (Form Interpreter) y el conjunto de estas dos herramientas, sería la primera versión compacta del lenguaje: PHP/FI.

La siguiente gran contribución al lenguaje se realizó a mediados de 1997 cuando se volvió a programar el analizador sintáctico, se incluyeron nuevas funcionalidades como el soporte a nuevos protocolos de Internet y el soporte a la gran mayoría de las bases de datos comerciales. Todas estas mejoras sentaron las bases de PHP versión 3. Actualmente PHP se encuentra en su versión 4 y es un lenguaje que funciona del lado del servidor. Primero se compila y luego se ejecuta, lo que permite incrementar la rapidez de ejecución. A partir del año 2000, el número de servidores que utilizan PHP se ha disparado, logrando situarse cerca de los 5 millones de sitios y 800,000 direcciones IP, lo que le ha convertido a PHP en una tecnología muy popular.

### 5.2.5 Justificación de la plataforma de desarrollo RedHat-Apache-Php

La distribución RedHat de Linux es la más difundida y sólida que existe en el mercado. En su versión 4.0 viene acompañada del Servidor Web Apache, y de la tecnología PHP. Es una distribución confiable, con opciones de escalabilidad y facilidad de uso, incluso desde la instalación en comparación con otras distribuciones Linux. El entorno gráfico es muy similar al de Windows y tiene la capacidad para administrar los servicios de red de manera muy confiable y segura. El servidor Apache es el servidor más difundido dentro de los servidores libres. En este sentido, el sistema posee la restricción de utilizar tecnología gratuita, ya que el uso de un servidor como el IIS de Windows estaba descartado por el costo de las licencias y la pobre seguridad que ofrece. Un servidor interesante es el ofrecido por Netscape, sin embargo su costo se aproxima a los 1000 dólares, y aunque es un servidor significativamente más rápido que el Apache, el costo no amerita su elección, debido a que Apache a pesar de ser un servidor gratuito, es un servidor sumamente confiable, con características sobresalientes en cuanto a seguridad, incluso muy por encima del servidor IIS de Microsoft.

Se estima que casi el 50% de los servidores Web son Apache, las aplicaciones de Apache funcionan en sitios que tienen millones de usos al día, y éstas se ejecutan sin complicaciones. Apache está presente en más de 700,000 servidores en Internet y se ha probado completamente por servidores y usuarios. El Grupo Apache mantiene normas rigurosas antes de lanzar versiones nuevas de su servidor. PHP es el lenguaje de programación que más se acopla al servidor Apache. Apache puede ejecutar *scripts* ASP sin embargo el desempeño de los *scripts* PHP es muy superior al de un *script* ASP, incluso sobre el servidor Microsoft.



## Sistema de Información de Aeropuertos



La terna RedHat-Apache-PHP ha sido desarrollada para trabajar conjuntamente de manera sencilla, eficiente, confiable y segura. Existe mucha documentación disponible para el desarrollo de aplicaciones de esta naturaleza.

PHP es un lenguaje muy similar a C, y sus páginas pueden tener acceso a bases de datos, conexiones en red, envío de correo electrónico, servicios de ftp y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la página PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

El manejo de aplicaciones de bases de datos es mucho más simple que en un lenguaje basado en CGI. PHP y ASP son lenguajes un tanto similares, aunque PHP fue diseñado para ejecutarse en sistemas UNIX. ASP realiza numerosas tareas por medio de componentes que deben ser comprados. En el caso de PHP, se integran nuevas funciones de manera gratuita en cada versión que sale. Una posible alternativa pudo ser la plataforma jsp, sin embargo, la integración entre las plataformas RedHat-Apache-PHP es automática, están diseñados para trabajar de manera conjunta, y sin ningún costo de licencias, como es el caso de un servidor para jsp como puede ser el de Netscape. Jsp puede ser ejecutado incluso en el servidor Apache, sin embargo el mayor rendimiento de dicho servidor está dado con PHP.

### 5.3 ANÁLISIS

A partir del análisis de los requerimientos de información mostrado en capítulos anteriores y del modelo de base de datos propuesto, el siguiente paso consiste en desarrollar un sistema que manipule a dicha base de datos de la manera más eficiente y segura acorde con las necesidades de información que se tienen. Inicialmente podría pensarse en un modelado en cascada para el desarrollo de esta aplicación, incluso un modelo evolutivo sería válido para el desarrollo del mismo.

Se sugiere el ciclo en cascada, debido a que el sistema debe ser completado en cada etapa antes de continuar con la etapa siguiente. Habrá partes en el desarrollo del sistema en donde se tenga que utilizar el modelo evolutivo, sobre todo para comprobar la entera satisfacción del cliente, aunque los requerimientos de información han sido cuidadosamente recabados en varias entrevistas.

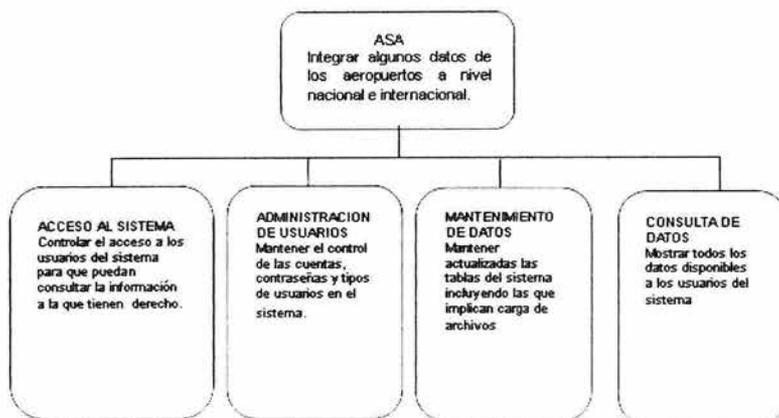
Cabe destacar que para el diseño y construcción de la base de datos se empleó la herramienta Case denominada *Application Modeler for Web* de Sybase, que permite a partir de un diagrama entidad relación la generación de los comandos DDL de SQL correspondientes a la creación de la Base de Datos. Como se mencionó anteriormente, el análisis comprende el reconocimiento del problema a resolver, la definición de dicho problema, la concentración de los objetivos, la evaluación del problema y la solución del mismo.

El siguiente paso consiste en evaluar el flujo de los datos, elaborar todas las funciones que el software debe cubrir y el comportamiento que el software deberá tener en función de los eventos que el usuario realice dentro del mismo, se deben establecer las características de la interfaz y descubrir restricciones adicionales. Las herramientas utilizadas en esta fase para visualizar la entrada, proceso y salida de los datos a través del sistema de información serán los diagramas jerárquico funcional, de flujo de datos y de contexto.



### 5.3.1 Diagrama Jerárquico Funcional

El diagrama jerárquico funcional tiene dos fines principalmente, el primero es el propio entendimiento del sistema y la segundo es definir la estructura de las actividades del sistema lo más simple posible de manera que puedan ser revisadas de manera sencilla.



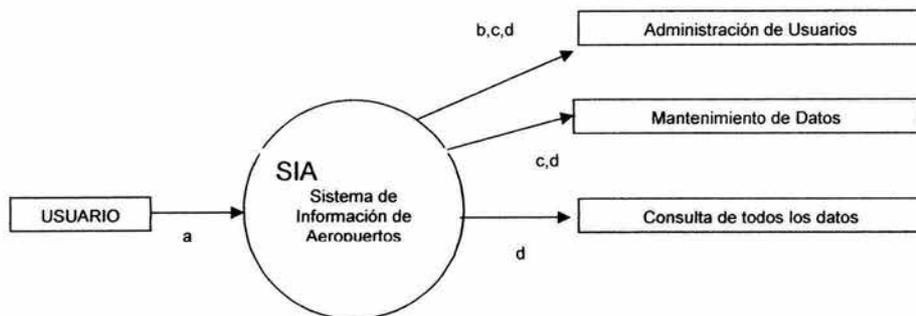


## Sistema de Información de Aeropuertos



### 5.3.2 Diagrama de Contexto

El diagrama de contexto se refiere al diagrama de flujo de datos de nivel 0. Este muestra de manera general el flujo de información del sistema.



- a. Validación de clave y contraseña
- b. modificación de cuentas de usuarios y datos
- c. modificación de datos
- d. lectura de datos

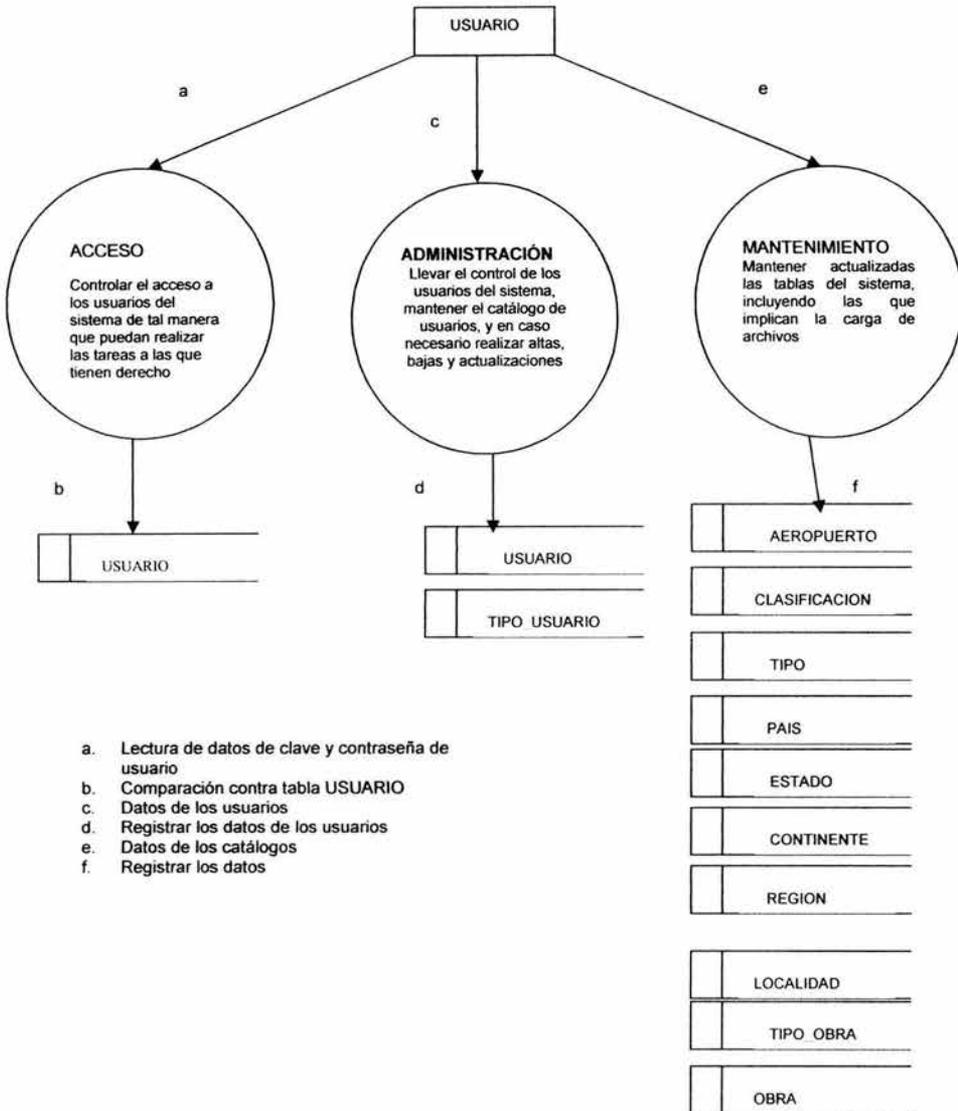


## Sistema de Información de Aeropuertos



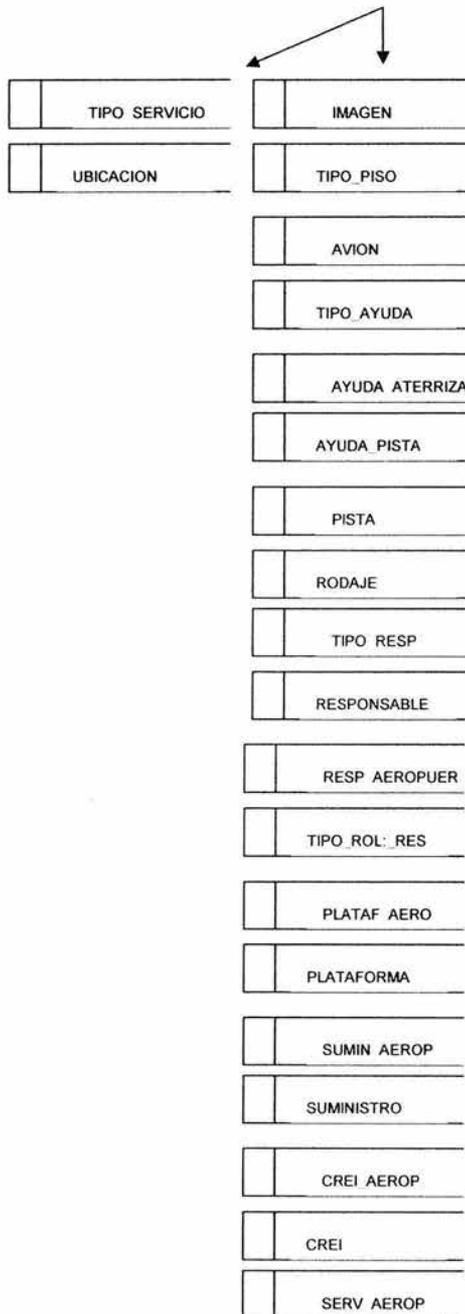
### 5.3.4 Diagrama de Flujo de Datos Del Sistema

Al entrar al sistema éste mostrará el menú principal, el cual contiene las opciones para poder acceder a los módulos del sistema o consultar información limitada de sólo lectura.



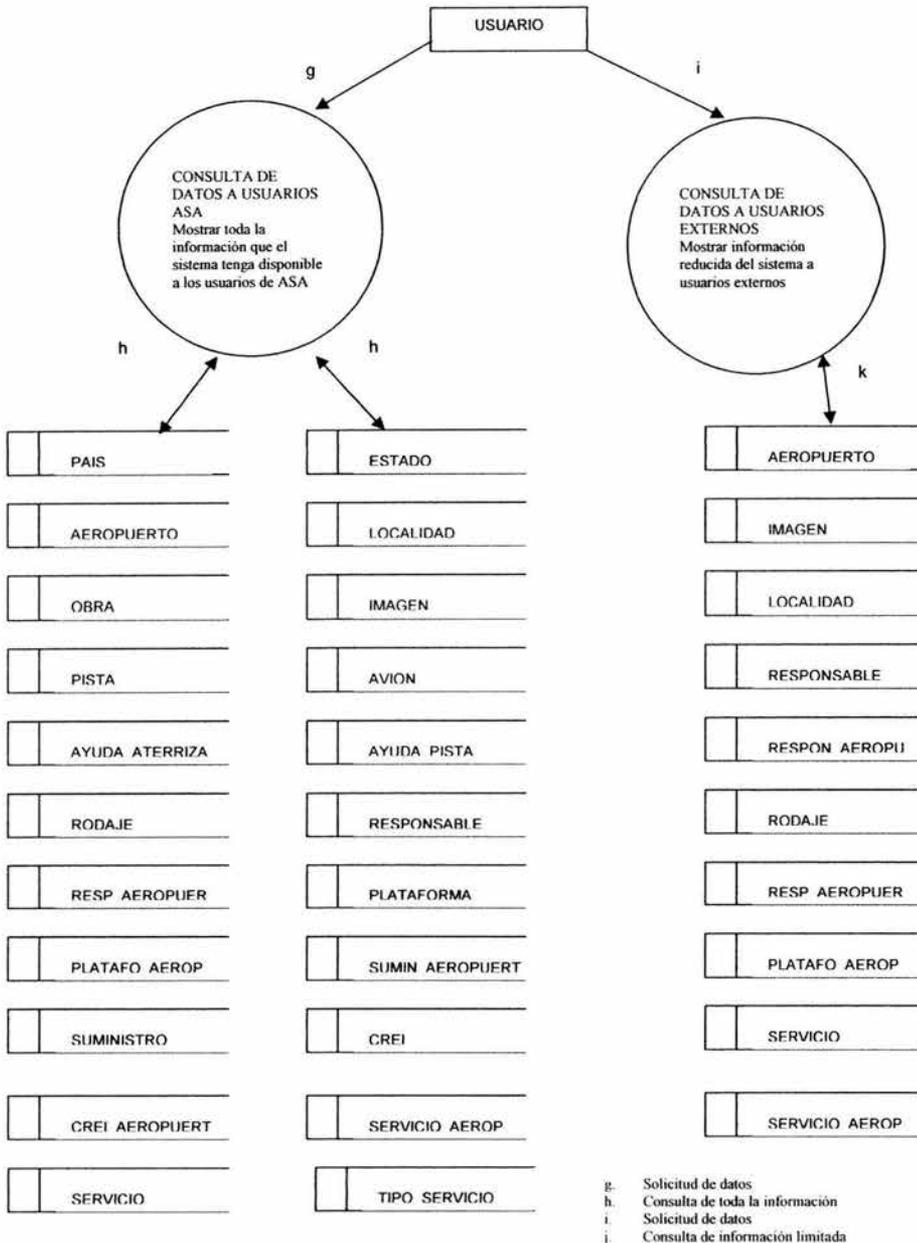


## Sistema de Información de Aeropuertos





# Sistema de Información de Aeropuertos





## Sistema de Información de Aeropuertos



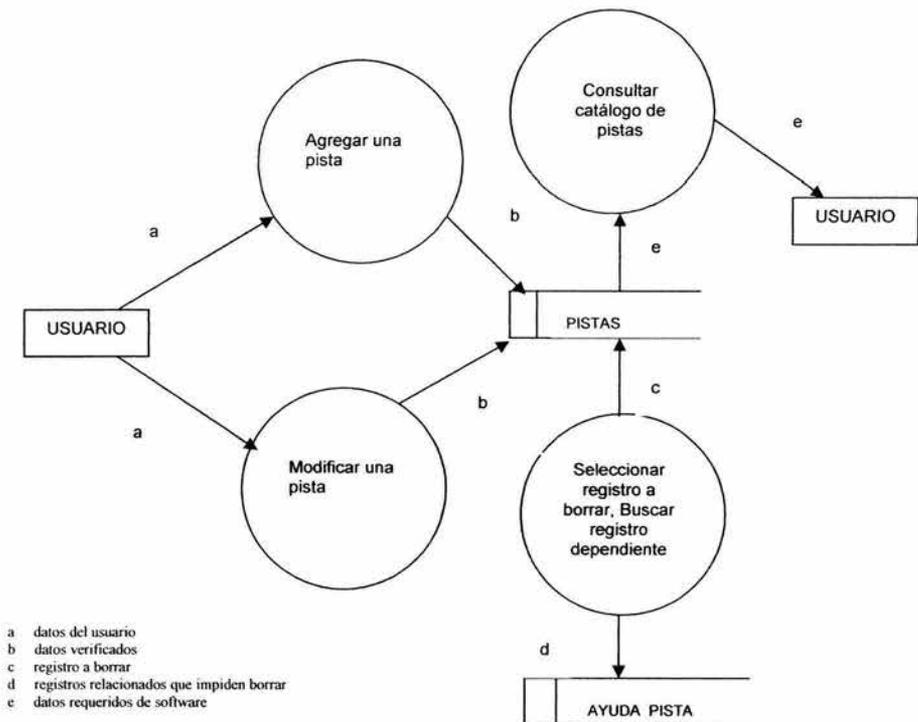
### 5.3.5 Mantenimiento a catálogos

El mantenimiento a catálogos permite asegurar que la integridad de la base de datos no se pierda al eliminar registros relacionados de una tabla hacia otra. En este caso, no se permitirán eliminaciones en cascada, y las actualizaciones no podrán ser sobre las llaves primarias, salvo en los casos en donde exista una relación muchos a muchos. Por ejemplo

#### PISTA

La siguiente figura muestra los procesos para realizar el mantenimiento del catálogo de PISTAS.

El usuario introduce los datos para agregar o modificar la información. Para borrar una pista, el usuario selecciona el registro que va ser eliminado, antes de eliminarla el sistema deberá buscar que no existan registros en tablas dependientes, en este caso concreto, sería la tabla AYUDA\_PISTA. Si no existen registros dependientes, el sistema procederá a eliminar el registro mostrando antes un mensaje para confirmar el borrado. En caso contrario, mostrará los registros en la tabla AYUDA\_PISTA que deberán ser eliminados antes de proceder a eliminar una pista. En el caso de las actualizaciones, estas serán en cascada hacia los registros relacionados, verificando antes que no produzcan una llave duplicada en el caso de una tabla intermedia.





### 5.4 DISEÑO Y DESARROLLO

La etapa de diseño sigue a la etapa de análisis y consiste en diseñar la interfaz entre el usuario y la base de datos. El diseño de la interfaz del usuario consiste en las formas, pantallas y reportes. En el caso de un sistema desarrollado en ambiente Web, las formas deberán ser todavía más sencillas e intuitivas, de tal manera que las formas sean fáciles de llenar, con un llenado preciso de datos de manera atractiva al usuario. Es recomendable que las formas sigan un flujo de izquierda a derecha, y/o de arriba hacia abajo.

El diseño debe tener la característica de usabilidad o facilidad de uso, la cual se define de acuerdo a la ISO (*Internacional Standard Organization*) como la capacidad de un software de ser comprendido, aprendido, usado y atractivo para el usuario, en condiciones específicas de uso. La usabilidad se basa en los siguientes principios básicos: facilidad de aprendizaje: referida a la facilidad con la que nuevos usuarios pueden tener una interacción efectiva. flexibilidad: referida a la variedad de posibilidades con las que el usuario y el sistema pueden interactuar, considerando la similitud de tareas actuales con tareas anteriores, optimizando el manejo de información entre el usuario y el sistema.

Algunos de los beneficios de la usabilidad son: reducción de los costos de aprendizaje, asistencia y ayuda al usuario, optimización de los costos de diseño y mantenimiento del sistema, mejora en la imagen del sistema, incremento en la productividad de los usuarios que hacen uso del sistema, incremento de la rapidez en la realización de tareas y reducción de las pérdidas de tiempo.

#### 5.4.1 Diseño de un Sitio Web

Un Sitio Web es un sistema de información con ciertas características particulares, que implican algunas consideraciones, por ejemplo, debe ser sumamente intuitivo, de tal manera que el entrenamiento sea mínimo y que el usuario pueda hacer uso del sitio desde el primer momento, sin perder desde luego la productividad que tenga este sistema de información. La protección del trabajo de los usuarios es una cuestión prioritaria, se debe asegurar que los estos nunca pierdan su trabajo como consecuencia de un error. Con respecto al diseño gráfico, los textos deben ser con colores que contrasten con el fondo y con fuentes suficientemente grandes fácilmente legibles.

Actualmente las tarjetas de video de las computadoras soportan perfectamente la resolución de 800 x 600 a 16 millones de colores, aunque la resolución de 1024 por 768 está estandarizándose de manera significativa. Los usuarios no deben sufrir sobrecarga de información, cuando un usuario visita un sitio Web y no sabe por donde comenzar ni a donde ir, existe sobrecarga de información.

Generalmente el usuario suele centrar su atención en el centro del sitio Web, por lo que si se incluye una cabecera, esta debe ser vistosa, o con elementos útiles para el usuario, como un buscador, un menú atractivo o una liga a un sitio de interés. Para los botones de un menú de cabecera es recomendable no utilizar iconos, ya que con un buen texto están de más, y ocupan mucho espacio. Para tener una navegación más fluida por el sitio, los menús deben tener unos buenos contenidos con una estructura bien definida, evitando menús con muchas opciones, considerando 5 o 7 como las opciones máximas permisibles.

Se recomienda el uso de hojas de estilo para simplificar el diseño en el documento. El documento deberá tener una buena distribución de texto e imágenes sin llegar a saturarse con las mismas, ya que la carga del sitio puede alentarse considerablemente por una imagen de manera innecesaria. Se recomienda el formato jpg para las imágenes del sitio, ya que posee una calidad de 16 millones de colores con muy poco espacio en comparación con el formato bmp.



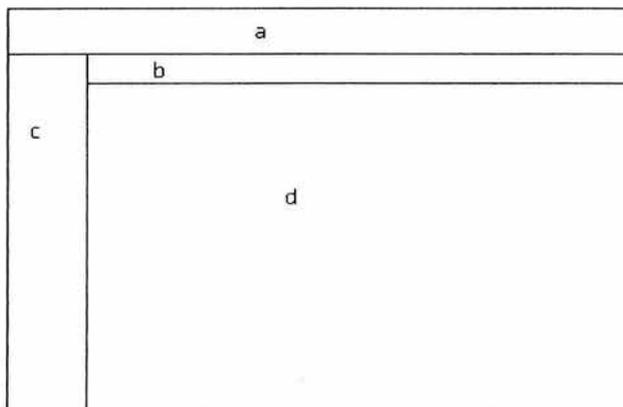
## Sistema de Información de Aeropuertos



El formato gif admite únicamente 256 colores, aunque tienen la ventaja de poder animarse con algunas herramientas para tales efectos. El sitio no deberá estar saturado de animaciones, ya que consumen mucho espacio y el tiempo de carga de la página se incrementa considerablemente. Se pueden utilizar animaciones en flash para darle vida a un sitio, aunque se debe tener una alternativa de consultar todo sitio sin animaciones flash. En el caso de tener versiones imprimibles de las páginas es recomendable tener tablas e imágenes de tamaño fijo inferior a 750 píxeles de ancho. La información estadística y financiera de un sitio se puede presentar en formato pdf de manera sencilla, entendible y actualizada.

El tener la información definida en módulos permite modificar el formato de la ficha en la medida que sea necesario. Es importante identificar que categoría de las fichas es más relevante para ser usada como elemento de la navegación principal, las menos relevantes y el orden de aparición más conveniente, etc.

### 5.4.2 Diseño del Sitio Web en cuestión de estudio



Este diseño contempla varios *frames* para los módulos

El sistema contempla dos secciones de navegación para el usuario: La sección a contendrá el encabezado del sitio.

La sección b contendrá los menús correspondientes a la información de lectura para todas las entidades que estén relacionadas con el Aeropuerto.

La sección c contempla las secciones de consulta para los aeropuertos nacionales, internacionales y una sección destinada a autenticar a los usuarios del sistema. La sección central derecha d está destinada a mostrar la información del sistema.

En la sección a el encabezado tendrá los siguientes elementos:

05/12/2001 14:35	Sistema de Información de Aeropuertos	
<b>UNAM</b>		
Facultad de Ingeniería		

El encabezado será común para todos los módulos.

El menú de navegación b hacia el usuario final, se compone de de los siguientes elementos:

Pistas	Ayudas	Plataformas	Rodajes	Suministros
CREI	Servicios	Responsables	Obras	Imágenes



## Sistema de Información de Aeropuertos



Datos Generales, que permite desplegar los datos de la tabla AEROPUERTO, CLASIFICACION, TIPO, OBRA, TIPO\_OBRA, LOCALIDAD, ESTADO, PAIS, CONTINENTE, REGIÓN E IMAGEN.

Pistas: que despliega los datos de las tablas AVION, TIPO\_AYUDA, AYUDA\_ATERRIZAJE, AYUDA\_PISTA, PISTA

Plataformas: que despliega los datos de las tablas PLATAFORMA, PLATAFORMA\_AEROPUERTO

Rodajes: que despliega los datos de las tablas RODAJES

Suministros: Despliega los datos de las tablas SUMINISTRO y SUMINISTRO\_AEROPUERTO

CREI: Despliega los datos de las tablas CREI y CREI\_AEROPUERTO

Servicios: Despliega los datos de las tablas SERVICIO, TIPO\_SERVICIO, UBICACIÓN\_SERVICIO Y SERVICIO

Responsables: Despliega los datos de las tablas RESPONSABLE\_AEROPUERTO, RESPONSABLE, TIPO\_RESPONSABLE, TIPO\_ROL\_RESPONSABLE

La sección c comprende la consulta de datos para los aeropuertos nacionales, los internacionales y la entrada al sistema de información por medio de la clave y de la contraseña de acceso.

<b>Aeropuertos</b>
<a href="#">Nacionales</a>
<a href="#">Internacionales</a>

<b>Usuarios</b>
Clave <input type="text" value="arturo548"/>
Contraseña <input type="text" value="*****"/>
<b>Enviar</b> <input type="button" value=""/>

La sección d es en sí el área de trabajo del sistema y tiene como objeto mostrar la información que despliegue el sistema de información.

### CONSULTA DE PAÍSES

México	
Superficie	1455522 metros cuadrados
Población	1000000 habitantes
PIB	1.4
Administración Aérea	A cargo de SCT y de SENEAM
Continente	América
Región	Norte



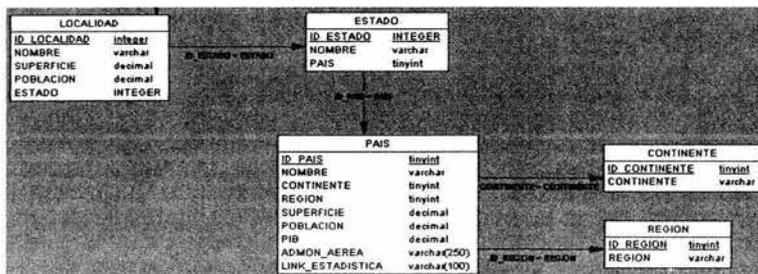
## Sistema de Información de Aeropuertos

Los usuarios van a interactuar con el sistema de Información de la siguiente manera:

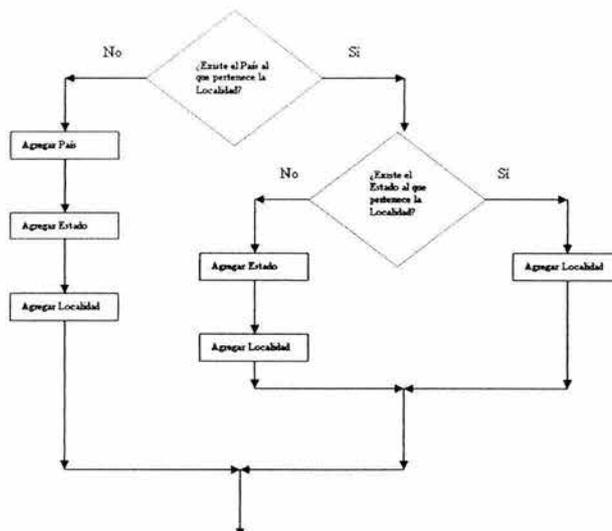
Se tienen 3 niveles de usuario: El usuario administrador, que se encarga de administrar la aplicación, el capturista que se encarga de darle mantenimiento a los catálogos y el usuario de asa que puede consultar toda la información del sistema

Cuando el usuario introduce su clave y su contraseña de acceso, el sistema inmediatamente detecta el tipo de usuario que se tiene y en función de eso, envía al usuario a las páginas correspondientes de acuerdo al tipo de usuario en cuestión.

Se tienen las siguientes opciones: Agregar, Modificar, Eliminar, Consultar para todos los catálogos de tal manera que el administrador los pueda tener actualizados con mucha sencillez. Para aquellos catálogos que provienen de tablas relacionadas, se tendrán asistentes para la captura de datos, por ejemplo, en el caso del catálogo de localidades, se tiene la siguiente estructura de base de datos:



El diagrama de flujo correspondiente al proceso de agregar una localidad es el siguiente:





## Sistema de Información de Aeropuertos



### 5.4.3 Desarrollo

Los pasos del Asistente en el formulario del Navegador se resumen a continuación. Por ejemplo: para agregar la ciudad de Huatulco, que pertenece al estado de Oaxaca en el país México:

El primer paso es seleccionar el país de la lista de países, si no está en la lista, será necesario agregarlo.

Dirección http://www.fora.com.mx/TesisASA/agrega\_ciudad.php

**ASISTENTE PARA AGREGAR CIUDADES**

1.-Seleccione el País al que pertenece la Ciudad y presione el botón Seleccionar.En caso de no estar en dicha lista presione el botón Agregar País.

**Lista de Países**

<input type="text" value="Canadá"/>	<input type="button" value="Seleccionar"/>
<input type="text" value="Canadá"/>	
<input type="text" value="Estados Unidos"/>	
<input type="text" value="Guatemala"/>	
<input type="text" value="México"/>	<input type="button" value="Agregar País"/>
<input type="text" value="Nicaragua"/>	

Una vez seleccionado el país México, será necesario seleccionar el estado al que pertenece la ciudad que se quiere agregar.

En la siguiente pantalla será necesario seleccionar el estado del país al que pertenece la ciudad. Cabe destacar que el sistema desplegará únicamente los estados dados de alta el país en cuestión.

Dirección http://www.fora.com.mx/TesisASA/agrega\_pais\_c1v.php

**ASISTENTE PARA AGREGAR CIUDADES**

2.-Seleccione el Estado/Provincia al que pertenece la Ciudad y presione el botón Seleccionar.En caso de no estar en dicha lista presione el botón Agregar Estado/Provincia.

**Lista de Estados/Provincias de México**

<input type="text" value="Oaxaca"/>	<input type="button" value="Seleccionar"/>
-------------------------------------	--



## Sistema de Información de Aeropuertos



Al presionar el botón Seleccionar, en la siguiente pantalla aparecerán los campos para capturar los datos correspondientes a la localidad en donde se ubicará el Aeropuerto

Dirección

### ASISTENTE PARA AGREGAR CIUDADES

3.-Escriba los datos de la Ciudad que vaya a agregar al Sistema.

País: México, Estado: Oaxaca	
Nombre	<input type="text" value="Bahías de Huatulco"/>
Superficie	<input type="text" value="250000"/> metros cuadrados
Población	<input type="text" value="300000"/> habitantes
Información Adicional	<input type="text" value="Destino de playa conformado por 9 bahías y 36 playas, con 34600 acres de exótica vegetación, terreno formado por la accidentada tierra de las montañas de la Sierra Madre del Sur."/>
<input type="button" value="Enviar"/> <input type="button" value="Restablecer"/>	

Al enviar los datos, el Sistema desplegará el siguiente mensaje:

La localidad Huatulco en Oaxaca México fué agregada con éxito

[Consultar](#) | [Modificar](#) | [Eliminar](#)

Como se mencionó anteriormente, la plataforma de desarrollo es PHP, utilizando módulos de funciones, por ejemplo, todas las páginas php requieren un script de conexión a la base de datos, éste fue incluido en un archivo denominado conecta.php, el cual tiene la siguiente estructura:

```

<?php
function Conectarse()
{
    if(!($link=mysql_connect("localhost","basedatos","contraseña")))
    {
        exit();
    }
    if(!mysql_select_db("usuario", $link))
    {
        exit();
    }
    return $link;
}
?>

```

El script se conecta a un servidor de Base de Datos, que no es el de ASA debido a que por políticas de la Institución no se autorizó la configuración de un servidor para hacer el desarrollo, únicamente para instalarlo una vez terminado, por lo que el mismo se realizó en un servidor comercial Linux de prueba con la debida autorización del propietario ([www.geoanalitica.com.mx/arth/TesisASA](http://www.geoanalitica.com.mx/arth/TesisASA))





## Sistema de Información de Aeropuertos



Para mostrar datos descriptivos en lugar de numéricos hablando de llaves foráneas, se utilizaron consultas basadas en cláusulas *where* referidas al campo que contiene la información relacionada a desplegar, por ejemplo, para visualizar el nombre del país y del estado en el formulario de captura de ciudades, se utilizó el siguiente código:

```
<?php
include("conecta.php");
$link=conectarse();
$sql0="select * from ESTADOS where nombre like '$nombre'";
$result0=mysql_query($sql0,$link);

while($row = mysql_fetch_array($result0)) {
    $sql1="select * from PAISES where id_pais=$row[pais]";
    $id_pais=$row[pais];
    $result1=mysql_query($sql1,$link);
    while($row1 = mysql_fetch_array($result1)) {
        $nombre_pais=$row1[nombre];

        $id_estado=$row[id_estado];
        $nombre_estado=$row[nombre];
        $pais=$row[pais];
    }
}
```

La consulta a la base de datos está en función del nombre del Estado, sin embargo el dato del país es numérico por lo que hubo que hacer una consulta anidada a la base de datos que desplegara el nombre del país.

El proceso de captura de datos para todas las tablas de la base de datos tiene la misma estructura que en el caso de las Ciudades, con asistentes que tienen los mismos pasos a seguir.

El proceso de actualización de las tablas es por medio de un asistente, en donde se selecciona de una lista desplegable el registro a modificar, por ejemplo, para modificar un país es necesario seleccionarlo de la lista y presionar el botón continuar:

MODIFICACIÓN DE PAÍSES	
Seleccione el País a Modificar	
Argentina	<input type="button" value="Modificar"/>
Argentina	
Canadá	
Estados Unidos	
Guatemala	
México	
Nicaragua	



## Sistema de Información de Aeropuertos



El siguiente paso es editar los campos que se quieran modificar, y presionar el botón modificar

PAÍS: México	
<b>Datos Generales</b>	
Superficie:	<input type="text" value="14552255"/> superficie
Población:	<input type="text" value="20000000"/> habitantes
PIB:	<input type="text" value="1.4"/>
Continente:	<input type="text" value="América"/> km
Región:	<input type="text" value="Norte"/> km/hr
Administración Aérea :	<input type="text" value="A cargo de SCT"/>
<input type="button" value="Modificar"/>	

Una vez presionado el botón modificar, el sistema indicará que los datos fueron modificados.

**Los datos del País México fueron Actualizados**

[Consultar](#) | [Agregar](#) | [Eliminar](#)

La modificación se realiza en primer lugar por medio de una lista desplegable que toma los datos del registro a modificar de la tabla correspondiente:

```

<select size="1" name="nombre">
<?
    while($row = mysql_fetch_array($result0)) {
        printf("<option value='%s'%s<option>", $row[nombre], $row[nombre]);
    }
?>
</select>

```

Una vez seleccionado el registro a modificar, es necesario crear un formulario que traiga los datos de la tabla correspondiente de tal manera que el usuario pueda modificar los datos que necesite:

```

<input type="text" name="nombre" value="$row0[nombre]" size="20">

```

En este caso el cuadro de texto llamado nombre tomará como valor el contenido del campo nombre para el registro que se seleccionó anteriormente. Este registro sale de una consulta filtrada por medio de una cláusula where proveniente de la lista desplegable del formulario anterior:

```

$sql0="select * from PAISES where nombre like '$nombre'";
$result=mysql_query($sql0,$link);

```



## Sistema de Información de Aeropuertos



Al presionar el botón modificar, se ejecutará un comando *update* hacia la base de datos con los valores de los campos que fueron cambiados y los que no cambiaron.

```
$sql0="update PAISES set nombre='$nombre2', superficie = '$superficie',  
poblacion='$poblacion', pib = '$pib', continente='$continente', region='$region',  
admon_aerea='$admon_aerea' where nombre like '$nombre';
```

```
mysql_query($sql0,$link);
```

Para eliminar los registros será necesario verificar que no haya datos relacionados en otra tabla que dependan de los registros actuales que se van a eliminar:

Si no existen registros relacionados, pueden eliminarse, en caso contrario, el sistema desplegará un mensaje indicando que el registro no puede ser eliminado. La eliminación del registro será seleccionando desde una lista desplegable el registro a eliminar, por ejemplo para eliminar el país Guatemala, será necesario seleccionarlo de una lista desplegable:

ELIMINACIÓN DE PAÍSES	
Seleccione el País a Eliminar	
Argentina ▼	Eliminar
Argentina	
Canadá	
Estados Unidos	
Guatemala	
México	
Nicaragua	

Una vez seleccionado, será necesario presionar el botón eliminar en donde el sistema enviará un mensaje de confirmación indicando que el registro fue eliminado.

ELIMINACIÓN DE PAÍSES
El País Guatemala fué eliminado

[Consultar](#) | [Modificar](#) | [Agregar](#)



## Sistema de Información de Aeropuertos



En este caso Guatemala pudo ser eliminado porque no había registro alguno que dependieran de dicho país en la tabla ESTADOS, en caso contrario el sistema hubiera mandado el siguiente mensaje:

### ELIMINACIÓN DE PAÍSES

El País México no pudo ser eliminado porque los siguientes Estados dependen de dicho País

Aguascalientes

Baja California

Baja California Sur

Campeche

Coahuila

Los datos de ésta tabla se obtienen de la siguiente consulta:

```
<? $sql0="select id_estado,nombre from ESTADOS where pais= '$id_pais';  
$result0=mysql_query($sql0,$link); ?>
```

Si esta consulta no hubiera dado resultados, el código para eliminar el país sería el siguiente:

```
<? $sql0="delete from PAISES where id_pais='$id_pais';  
mysql_query($sql0,$link);  
?>
```

Cabe destacar que la variable \$id\_pais se obtiene del cuadro de lista desplegable que seleccionó el usuario para eliminar el país deseado:

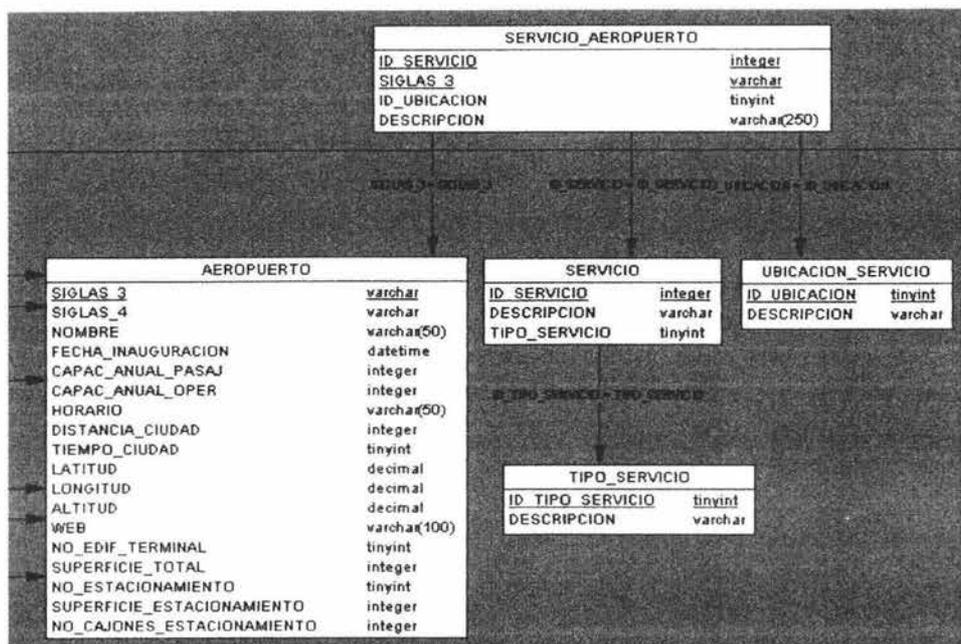
México	▼
Argentina	
Canadá	
Estados Unidos	
México	
Nicaragua	



## Sistema de Información de Aeropuertos



Con respecto a las entidades con relación muchos a muchos, el proceso de captura, actualización, eliminación y selección de datos es muy similar a los procesos antes mencionados, por ejemplo, existe una relación muchos a muchos entre AEROPUERTO y SERVICIOS:



Para hacer más simple la operación del sistema, será necesario implementar un asistente en donde se seleccione en primer lugar el aeropuerto en cuestión, y en segundo lugar el servicio que se le desee asignar al aeropuerto en cuestión, filtrado por tipo de servicio, por ejemplo para agregar el servicio de autobuses foráneos al Aeropuerto de la Ciudad de México será necesario seleccionar dicho aeropuerto. La captura del Aeropuerto tiene la misma lógica que la captura de la Ciudad descrita anteriormente, será necesario ubicar al Aeropuerto por país, estado y localidad por medio de un asistente, el cual contempla capturar previamente el país, estado o localidad en caso necesario, si estos ya existen se procede a capturar el Aeropuerto en cuestión. Una vez llenados los datos generales del Aeropuerto (contenidos en la tabla AEROPUERTO), el siguiente paso es capturar los datos de las relaciones M:M entre la tabla AEROPUERTO y las tablas con este tipo de relación:

### 5.5 CONCLUSIONES

La plataforma de desarrollo PHP es una plataforma muy robusta para el desarrollo de aplicaciones Web, PHP es un lenguaje híbrido de C ideal para el desarrollo de aplicaciones Web, incluso, se puede montar el servidor Apache y desarrollar una aplicación en ambiente de escritorio, con la gran ventaja de no pagar costos de licencias. PHP es un lenguaje muy fácil de asimilar, muy superior en cuanto a funcionalidad sobre plataformas como ASP. La mayoría de los servidores Web son del tipo Apache, y funcionan con este lenguaje.



## Sistema de Información de Aeropuertos



---

### CAPITULO 6 SEGURIDAD EN LOS SISTEMAS DE INFORMACIÓN

---

La seguridad es un concepto sumamente amplio que comprende varias etapas para garantizar la confiabilidad dentro de un sistema de información. Hablar de seguridad es imperativo dentro de un Sistema de Información, y comprende el estudio de diferentes niveles, entre los que destacan la seguridad a nivel del Sistema Operativo, a nivel de la red, a nivel de la Base de Datos, a nivel de la aplicación, así como a nivel de las políticas internas que se tengan en la empresa.

#### 6.1 SEGURIDAD DEL SISTEMA OPERATIVO

Linux es un Sistema Operativo que está cobrando mucha fuerza dentro del campo de sistemas operativos, debido a sus características, tales como potencia, flexibilidad, apertura de código, costo reducido y en algunos casos nulo, facilidad para descargar actualizaciones y herramientas, estabilidad, y el buen manejo de seguridad que tiene. Linux incorpora las características de seguridad fundamentales de los sistemas operativos Unix, por lo que le está ganando la batalla en muchos aspectos a los sistemas Windows. En muchos casos, incluso, será necesario sacrificar la funcionalidad del sistema, decayendo la facilidad de uso, a cambio de una plataforma más segura. Es conveniente que el administrador verifique constantemente los registros de transacciones (logs) del sistema, y que el sistema tenga siempre las últimas actualizaciones disponibles y confiables.

Debido a que Linux es un sistema multiusuario el sistema debe tener medidas de seguridad para proteger a los usuarios de otros usuarios e incluso protegerse así mismo, garantizando todos los servicios que puede prestar en una red. Tener sistemas seguros en la actualidad es sumamente imperativo, ya que Internet una red sumamente insegura. La información que fluye a través de Internet pasa por muchos puntos durante su camino, por lo que existe la posibilidad de que esta sea interceptada por otros usuarios, al grado de poder consultar e incluso alterar la información que se está enviando. Incluso, dentro de la empresa, existe la posibilidad de tener usuarios que de manera intencional intenten realizar tareas que no les corresponden dentro del sistema de información. Otra amenaza sumamente importante es representada por la enorme cantidad de programas existentes en la red que permiten vulnerar la seguridad de los sistemas de información, accedendo a los mismos de manera indebida. Es importante por lo tanto estar actualizado y tener disponibles publicaciones acerca de las últimas debilidades del sistema operativo detectadas.

Cabe destacar que ningún sistema operativo es completamente seguro al 100%, sin embargo, se puede incrementar la dificultad para que la integridad del sistema se vea comprometida por usuarios ajenos al sistema. Existen diversos documentos relacionados con la seguridad de un sistema de información, así como algunas medidas que deben ser tomadas en el caso de un ataque, tal como el documento RFC2196, disponible en la dirección <http://www.fags.org/rfcs/rfc2196.html>. Una característica muy importante de Linux es referida al código abierto (*Open Source*) de sus fuentes, cualquier persona puede obtenerlas, analizarlas y modificarlas. Esto tiene dos implicaciones, por un lado, este modelo permite tener sistemas de seguridad muy superiores a los de otras plataformas como Microsoft, así como la posibilidad de tener actualizaciones muy rápidas para los huecos de seguridad que se tengan, en comparación con los sistemas Microsoft, en donde las actualizaciones tardan más en aparecer.

La seguridad inicia desde la elección de la ubicación física que tendrá el servidor, ya que el acceso del personal deberá estar restringido para los usuarios no administradores. Por otra parte es recomendable no otorgar permisos de inicio de sesión en el servidor a todos los usuarios de manera local, ya que deberán acceder a éste de manera remota. Es importante tener el control de los usuarios que intentan ingresar al sistema operativo. Linux permite registrar todos los intentos de acceso, incluidos los fallidos al sistema operativo. Es importante configurar el LILO de tal manera que al inicio del sistema no se pueda manipular el arranque (por ejemplo con el parámetro `restricted` en el archivo `/etc/lilo.conf`).



## Sistema de Información de Aeropuertos



Tener en cuenta que ningún sistema es realmente seguro si alguien, con los conocimientos necesarios, puede usar su propio disco para arrancar. Es importante tener siempre una clave para el arranque del sistema, de tal manera que el equipo no reinicie sin esta clave, incluso en el caso de que el sistema se detenga de manera imprevista, será necesario suministrar dicha clave.

Es conveniente también hacer uso de programas que permitan bloquear la consola a fin de evitar la ejecución de programas que simulan la pantalla de inicio del sistema. Estos programas pueden tomar la clave y contraseña del usuario, para entrar posteriormente al sistema con sus datos. Una solución a este problema puede ser el programa `vlock`, el cual permite bloquear las consolas abiertas en la terminal.

Algo que es importante destacar en los sistemas Linux, es referido al usuario `root`, el objetivo de un atacante es conseguir acceso como `root` al sistema operativo, para tener el control total del sistema. En ese sentido, es algo típico, conseguir el acceso como un usuario diferente a `root`, para posteriormente ir incrementando los niveles de privilegios hasta lograr acceder como `root` si el sistema tiene deficiencias en su administración de la seguridad: programas con errores, configuraciones deficientes de los servicios o el descifrado de claves cifradas.

Linux es un sistema operativo multiusuario, en donde cada usuario está contenido en un registro del archivo `/etc/passwd` y cada grupo de usuarios en un registro del archivo `/etc/group`. Cada usuario puede pertenecer a uno o varios grupos y cada grupo puede tener acceso a uno o más usuarios o grupos. Será necesario definir los permisos de las tareas que un usuario puede y no puede realizar, por ejemplo, el `floppy` tiene generalmente los siguientes permisos

```
brw-rw-r-- 1 root floppy 2,0 oct 5 2001 /dev/fd0
```

Esto significa que el propietario es `root` con permisos de lectura y escritura, el grupo es `floppy` con permiso de lectura y escritura, y el resto de los usuarios tendrán el permiso de lectura. Si es necesario que algún usuario pueda escribir en el disco, sólo tendrá que ser incluido en el grupo `floppy`, para tener acceso a la unidad en modo de lectura y escritura. Será necesario que el administrador analice las necesidades específicas de cada uno de sus usuarios, y que se les asignen los recursos mínimos para que éstos puedan realizar sus tareas, sin que representen un peligro para la integridad del sistema operativo.

Las cuentas y contraseñas de los usuarios deben tener ciertas características importantes a fin de evitar que puedan vulnerar la seguridad del sistema, por ejemplo, evitar el uso de palabras conocidas, tanto en el `login` como en el `password`; es recomendable que contengan combinaciones de letras, números y caracteres especiales. Un `login` y un `password` deben ser fáciles de recordar para un usuario, y difíciles de adivinar, y sobre todo, hacer conciencia en los usuarios en no revelar los datos de sus cuentas a otros usuarios, o no dejar anotada su clave y contraseña de acceso en el escritorio o pegado al monitor por ejemplo.

Las contraseñas de los usuarios son almacenadas en el archivo `/etc/passwd`. Todos los usuarios del sistema tienen permisos de lectura en este archivo. Si por medio de un navegador con un usuario que tenga permisos de lectura sobre este archivo, los datos llegan a un hacker, es posible que este tenga acceso a las cuentas de los usuarios, incluyendo al usuario `root`. Existe el concepto de contraseñas de sombra (`shadow passwords`), en donde las contraseñas son tomadas del archivo `/etc/passwd` para llevarlas al archivo `/etc/shadow` a fin de que sólo el usuario `root` pueda tener acceso a este archivo.

El archivo `/etc/shadow` contiene la siguiente información:

`login` : `password` : `ultimo` : `puede` : `debe` : `aviso` : `expira` : `desactiva`



## Sistema de Información de Aeropuertos



login: El nombre del usuario.  
password: La clave cifrada  
ultimo: Días transcurridos del último cambio de login desde el día 1/1/70  
puede: Días transcurridos antes de que el login se pueda modificar.  
debe: Días transcurridos antes de que el login tenga que ser modificado.  
aviso: Días de aviso al usuario antes de que termine el login.  
expira: Días que se desactiva la cuenta tras terminar la el login.  
desactiva: Días de duración de la cuenta desactivada desde el 1/1/70.

Por ejemplo:

cacheton : hRzn6qoPKYHgg : 10554 : 0 : 99999 : 7 : -1 : -1

El paquete de Shadow Passwords se puede descargar desde la página de redhat o desde alguna otra distribución conocida.

Una recomendación muy importante, es la de crear una cuenta alterna a root para el usuario administrador, y evitar en todos los casos navegar en Internet como root, para evitar el acceso al archivo shadow password incluso. Es importante evitar que la contraseña de root viaje por la red, un canal seguro puede ser el Secure Shell por ejemplo.

Para disminuir el riesgo de pérdida de datos, es conveniente estructurar los directorios en diferentes particiones de manera que si se daña una partición todo el sistema se pierda. Una recomendación para sistemas que requieren almacenamiento para documentos o datos de cada usuario es tener al directorio /home en una sola partición, con el espacio en disco duro adecuado. Si se cuenta con un servidor de correo o impresión, el directorio var también puede tener su propia partición. Los directorios de arranque como /dev, /etc, /bin, /lib e incluso /boot pueden quedar en otra partición. El directorio /usr/local tiene todos los programas instalados y compilados por el usuario root, por lo que puede ser conveniente también tener a directorios en una partición por separado.

La estructura de permisos de archivos y directorios es sumamente importante, ya que de los permisos de directorios y archivos depende en una gran medida la vulnerabilidad que tenga el sistema operativo. En los sistemas Unix/Linux, los permisos de los archivos constan de 3 partes: usuario o propietario(u), grupo(g) y otros(o). El propietario y el grupo son únicos para cada archivo/directorio, en el entendido de que un grupo puede contener varios usuarios. La parte de Otros es referida a aquellos usuarios que no son del grupo, ni son propietarios. Los permisos se dividen en lectura denotados con una r, escritura con una w y ejecución, denotados con una x. Los datos del usuario y del grupo se almacenan con sus respectivos identificadores. Se tiene un bit adicional en el caso de los directorios, conocido como Sticky bit. Si está puesto a *true*, entonces un usuario sólo puede borrar archivos que le pertenezcan o para los cuales tiene permiso de escritura. Por ejemplo: drwxrwxrwt 19 root root 8192 Oct 05 14:35 tmp

Existen herramientas como el programa *tripwire* que permiten verificar la integridad del sistema de archivos, en donde se ejecutan varias revisiones hacia la integridad de dichos archivos contra valores de dichos archivos contenidos en una base de datos que se consideran como válidos, de esta manera se pueden identificar algunos archivos que puedan estar corruptos. Los archivos sin propietario son un indicio claro de que un posible intruso ha estado accediendo al sistema. Con el comando find es posible ubicar archivos que no cuenten con propietario alguno utilizando el modificador -nouser -group  
find / -nouser -o -nogroup -print



## Sistema de Información de Aeropuertos



Como el sistema operativo estará interactuando dentro de una red con diferentes usuarios, las claves y contraseñas estarán viajando a través de la red. En ese sentido es importante utilizar canales seguros de comunicación de diversa índole, por ejemplo el SSH y el STelnet, que son programas que permiten mantener conexiones cifradas con sistemas remotos, evitando que los datos viajen a través de la red sin cifrar, en comparación con programas como el telnet en donde los datos viajan sin cifrar, especialmente en el caso del usuario root. Si es necesario entrar a una terminal para realizar tareas de root es altamente recomendable ingresar como usuario, y posteriormente utilizar el comando *su* para acceder como root, de esta manera, el paso para descifrar la contraseña es más indirecto. Existen analizadores de puertos como Satan que permiten alertar al administrador acerca de los posibles huecos de seguridad que puede tener el sistema.

### 6.2 SEGURIDAD DE LA BASE DE DATOS

La seguridad de la base de datos es de suma importancia dentro de un sistema de información, debido a que la base de datos es la fuente de datos de cualquier sistema. En ese sentido, es recomendable seguir algunas políticas de seguridad e integridad de los datos, con el fin de que el sistema de información trabaje de manera adecuada, garantizando que la información que despliega es realmente la que se le está solicitando. En primer lugar es importante tener una política de seguridad a nivel de usuarios adecuada.

Es de suma importancia realizar respaldos de manera periódica, a fin de garantizar la operación del sistema de información en cualquier eventualidad que pueda ocurrir. Los respaldos de bases de datos pequeñas (inferiores a 650 MB) pueden ser realizados en unidades de CDROM, incluso utilizando discos regrabables, que se pueden ir rotando en función de las necesidades de la empresa, y de la cantidad de datos que ingresen diariamente al sistema. En el caso de este sistema, se recomienda un respaldo diario en la etapa de captura de datos, y un respaldo semanal durante su etapa de operación. Será importante tener el respaldo del diccionario de datos en una ubicación segura.

Es importante establecer políticas de seguridad contra causas físicas de pérdida de datos, causas lógicas y también humanas, las cuales pueden o no ser intencionales. Dentro de la seguridad a nivel de base de datos se tienen algunos conceptos importantes como la integridad de los datos aplicable cuando se tiene una dependencia de tablas, implicando que lo que se tenga en la tabla dependiente a nivel de llave foránea exista previamente en la tabla independiente, pudiéndose o no realizar actualizaciones o eliminaciones en cascada. Lo más importante es evitar las inconsistencias de los datos, de tal manera que el sistema proteja de manera automática las operaciones incorrectas. En sistemas multiusuario es importante tener control sobre la concurrencia de los usuarios con el fin de conservar siempre la integridad de los datos, ya que se pueden producir inconsistencias cuando dos usuarios intenten modificar el mismo registro de manera simultánea.

La consistencia de datos implica que los datos siempre deben tener el mismo significado aunque sean consultados de tablas diferentes. La confidencialidad de los datos es referida a la capacidad de establecer permisos de acceso a las tablas de tal manera que los usuarios no autorizados a consultar cierto tipo de información no lo puedan hacer. El término accesibilidad implica que todos los datos puedan ser consultados en caso de ser necesario.

El sistema manejador de bases de datos relacionales debe permitir establecer permisos de acceso a las diferentes tablas de la base de datos, de tal manera que los usuarios no puedan acceder de manera total a toda la base de datos. En este sentido una buena parte de los sistemas manejadores manipulan la autenticación de los usuarios de dos maneras: por medio del sistema manejador de bases de datos relacionales y por medio del sistema operativo.



## Sistema de Información de Aeropuertos

---



En la mayoría de los casos, es más seguro dejarle la tarea al sistema operativo que al manejador de bases de datos relacionales. Los permisos deberán ser otorgados por el administrador de la base de datos, en cuyo caso deberá poder crear cuentas, conceder privilegios, quitar privilegios, suspender e incluso cancelar cuentas de los usuarios en caso de ser necesario. Es recomendable que exista la posibilidad de asignar cuentas y permisos por grupo, debido a que esta tarea facilita la administración de los usuarios, en donde todos los usuarios de un mismo grupo tendrán los mismos privilegios de acceso. El manejador de bases de datos deberá contar con las capacidades de realizar copias de seguridad y restauraciones de manera sencilla y rápida.



## 7 AFINACIÓN DEL SISTEMA DE INFORMACIÓN

### 7.1 PRESENTACIÓN PRELIMINAR AL CLIENTE.

En este capítulo se describen algunas experiencias al momento de presentar el sistema en ASA.

El proyecto fue realizado utilizando un servidor Linux que actualmente está siendo rentado por la compañía Geoanalítica ([www.geoanalitica.com/arth/TesisASA](http://www.geoanalitica.com/arth/TesisASA)) con la respectiva autorización de sus directivos. En este sentido lo ideal hubiera sido desarrollar el proyecto directamente en un servidor de ASA, pero no hubo la disponibilidad en ese momento, por lo que se acordó desarrollar el proyecto por separado y posteriormente anexarles los archivos fuentes y el diseño de la base de datos a fin de que lo pudieran instalar posteriormente en el servidor de su conveniencia. En este sentido, quedan excluidas del alcance del proyecto las tareas de instalación de Linux, configuración de Apache, PHP y MySQL, así como las tareas de seguridad que involucran al Sistema Operativo y a la Red. La parte de seguridad que al sistema concierne es la autenticación del usuario hacia la base de datos por medio de sesiones en PHP, así como validar el tamaño de los archivos que sean subidos al servidor por medio de validaciones, también realizadas desde PHP.

El Sistema contempló la búsqueda de aeropuertos por nombre, utilizando un cuadro de texto y un operador *like* para los datos del nombre, sin embargo se tenía la necesidad de ubicar aeropuertos por diferentes criterios, como el nombre, el estado, el país, el tipo, e incluso la clasificación, por lo que sería necesario modificar la forma de búsqueda de aeropuertos.





## Sistema de Información de Aeropuertos

Cabe destacar que la captura de datos es responsabilidad de ASA, por lo que el compromiso concluye con la entrega del software así como de los manuales técnico y de usuario, y el proceso de capacitación a los usuarios finales, el cual consiste en una clase de 3 horas para que los usuarios comprendan todos los términos que incluye el software en cuestión. Como se trata de páginas en php, no será necesario compilar ningún archivo, únicamente instalarlo en un directorio protegido contra lectura y escritura para otros usuarios, y con acceso total para los usuarios del grupo, los cuales serán autenticados por medio de una sesión en php.

### 7.2 RECOPIACIÓN DE OBSERVACIONES

Se recomienda habilitar una opción de búsqueda para ubicar aeropuertos por nombre, país, clasificación y tipo, así como realizar la distinción entre aeropuertos nacionales y aeropuertos internacionales. También es recomendable consultar las especificaciones de los aviones desde las pistas, con datos expresados en unidades del sistema métrico decimal y del sistema inglés. Los datos del aeropuerto deben aparecer en capacidades diarias y en capacidades anuales. Se recomienda hacer uso de la interfaz MySQLAdmin que incluye la opción de respaldar y restaurar la base de datos con acceso únicamente para el administrador con el objeto de que este pueda respaldar y restaurar en caso necesario su base de datos. También se recomienda poder imprimir los datos devueltos por las consultas, por lo que habrá que programar un script que simplemente envíe la tabla que se esté mostrando a la impresora utilizando el navegador.

### 7.3 AFINACIÓN DEL SISTEMA DE INFORMACIÓN

Las correcciones que se le hicieron al sistema fueron pocas, sobresaliendo la pantalla de búsqueda de aeropuertos por diferentes criterios: nombre del país, nombre del estado, nombre del aeropuerto, clasificación y tipo de aeropuerto a fin de que la búsqueda de sus datos sea más eficiente. En este sentido, se programó un control *Select* de html (conocido como Combo Box o Cuadro Combinado en otros lenguajes) de tal manera que la búsqueda de los datos del aeropuerto se realice en función de los criterios solicitados en el Sistema de Información.

**Seleccione el criterio de búsqueda y presione el botón Siguiente**

Nombre del Aeropuerto	▼
Nombre del Aeropuerto	
Estado del Aeropuerto	
Ciudad del Aeropuerto	
Tipo de Aeropuerto	
Clasificación del Aeropuerto	

En este sentido, dependiendo del criterio elegido, el sistema automáticamente despliega en pantalla los posibles valores de los filtros de datos en función de dicho criterio, por ejemplo, si la búsqueda se realizó por estado, aparecerán en la siguiente pantalla todos los estados del país al que pertenece el aeropuerto en cuestión, por ejemplo para México, al seleccionar la opción Estado del Aeropuerto, aparecerá una lista con los estados para seleccionar el aeropuerto de acuerdo al criterio solicitado:



## Sistema de Información de Aeropuertos



Por ejemplo:

Seleccione el Estado al que pertenece el Aeropuerto y presione el botón Siguiente.

Lista de Estados de México	
Aguascalientes	Siguiente
Chiapas	
Chihuahua	
Coahuila	
Colima	
Distrito Federal	
Durango	
Estado de México	
Guanajuato	
<b>Guerrero</b>	
Hidalgo	
Jalisco	

De esta manera, al presionar el botón Siguiente aparecerá una lista con los aeropuertos del estado seleccionado.

Consulta de Aeropuertos del País México del Estado Guerrero				
Elegir	Nombre del Aeropuerto	Siglas OACI	Siglas ACILAC	Ciudad
<input type="radio"/>	Aeropuerto de Ixtapa-Zihuatanejo	ZIH	MMZH	Zihuatanejo
<input checked="" type="radio"/>	General Juan N. Álvarez	ACA	MMAA	Acapulco

En función del aeropuerto que elija el usuario, serán los datos que despliegue el sistema de Información

Platas	Ayudas	Plataformas	Rodajes	Suministros
CREI	Servicios	Responsables	Obras	Imágenes

Consulta de Datos Generales del Aeropuerto ACA	
Siglas ACILAC	ACA
Siglas OACI	MMAA
Nombre	General Juan N. Álvarez
Clasificación	Internacional
Tipo	Turístico
Ciudad	Acapulco
Estado	Guerrero
País	México
Fecha Inauguración	1962-10-02
Capacidad Anual de Pasajeros	10000000
Capacidad Anual de Operaciones	200000
Num. de Edificios Terminales	1

El sistema cuenta con una validación, indicando si se tienen o no capturados los datos correspondientes en las tablas de la base de datos, por ejemplo, en el caso de este aeropuerto no se tienen obras capturadas por lo que al hacer una consulta de las mismas, el resultado es un mensaje indicando que no se tienen obras capturadas.



## Sistema de Información de Aeropuertos



Si el usuario cuenta con los permisos necesarios, aparecerá un botón de captura de datos.

**No hay Obras capturadas en el Aeropuerto: General Juan N. Álvarez de Acapulco del País México**

Agregar Obra

En este caso concreto se tiene una estructura de decisión múltiple, la cual en base a las opciones elegidas por el usuario validará los datos.

Por ejemplo para desplegar los aeropuertos clasificados por estados en un país, el código PHP de la estructura de decisión es el siguiente:

```

if ($criterio=='estado_aeropuerto')
{
$sql0="select ESTADOS.*, PAISES.nombre as pais from PAISES inner join ESTADOS on
ESTADOS.pais=PAISES.id_pais where PAISES.nombre like '$pais' order by ESTADOS.nombre";
$result0=mysql_query($sql0,$link);

echo"
<font face='Tahoma' size='2'> <b>
2.-Seleccione el Estado al que pertenece el Aeropuerto y presione el botón Siguiente.
</font>
<table border='2' cellpadding='0' style='border-collapse: collapse' bordercolor='#0099FF' width='100%' height='1'>
<tr>
<td width='100%' bordercolor='#0099FF' bgcolor='#99CCFF'>
<p align='center'><font face='Tahoma'><b>Lista de Estados de $pais</b></font>
</td>
</tr>
<tr>
<td width='100%' bordercolor='#0099FF' bgcolor='#FFFFFF' height='1'>
<p align='center'>
<br>
<FORM method='post' action='nacionales2.php' enctype='multipart/form-data'>
<select size='1' name='estado'>;
while($row = mysql_fetch_array($result0)) {
printf("<option value=%s>%s</option>", $row[id_estado], $row[nombre]);}
echo"</select>
<input type='hidden' value=$pais name='pais'>
<input type='hidden' value=$criterio name='criterio'>
<input type='submit' value='Siguiente' name='Seleccionar'></form>
</td></tr> </table>;";

```

En este caso, el sistema despliega los datos contenidos en la tabla estado de acuerdo al país por medio de un control *select* para que posteriormente sean desplegados los aeropuertos que el usuario puede seleccionar de ese estado, empleando el siguiente código:

```

if ($criterio=='estado_aeropuerto'){
$result=mysql_query("SELECT AEROPUERTOS.siglas_3, AEROPUERTOS.siglas_4,AEROPUERTOS.nombre as
nombre_aero, LOCALIDADES.nombre as localidad, AEROPUERTOS.ubicacion,
LOCALIDADES.estado,ESTADOS.nombre as nombre_estado from AEROPUERTOS inner join LOCALIDADES on
AEROPUERTOS.ubicacion=LOCALIDADES.id_localidad inner join ESTADOS on
LOCALIDADES.estado=ESTADOS.id_estado
where LOCALIDADES.estado = '$estado'
order by AEROPUERTOS.nombre",$link); }

```



## Sistema de Información de Aeropuertos



La estructura variable *result* tomará los campos generados por la consulta anterior en SQL, de tal manera que al utilizar la función *mysql\_fetch\_array*, los datos se despliegan en una tabla a fin de que el usuario los pueda seleccionar dentro de un botón de opciones. En este caso, por *default* el primer elemento de la lista estará seleccionado, utilizando una estructura de decisión anidada dentro del ciclo *While*, por medio de la propiedad *checked*.

```
<?php
$contador=1;
while($row = mysql_fetch_array($result)) {
?>
<tr>
  <td width="8%" align="center" bordercolor="#0099FF" bgcolor="#FFFFFF">
<font face="Tahoma" size="2">
<?if ($contador=1){
  $contador=2;?>
<input type="radio" value="<?printf("%s",$row[siglas_3]);?>" name="siglas_3" checked>

<?> else {?>
<input type="radio" value="<?printf("%s",$row[siglas_3]);?>" name="siglas_3">
<?>?>

</font></td>
  <td width="37%" align="center" bordercolor="#0099FF" bgcolor="#FFFFFF">
<font face="Tahoma" size="2"> <?printf("%s",$row[nombre_aero]);?></font></td>
  <td width="10%" align="center" bordercolor="#0099FF" bgcolor="#FFFFFF">
<font face="Tahoma" size="2"> <?printf("%s",$row[siglas_3]);?></font></td>
  <td width="22%" align="center" bordercolor="#0099FF" bgcolor="#FFFFFF">
<font face="Tahoma" size="2"> <?printf("%s",$row[nombre_estado]);?></font></td>
  <td width="23%" align="center" bordercolor="#0099FF" bgcolor="#FFFFFF">
<font face="Tahoma" size="2"> <?printf("%s",$row[localidad]);?></font></td>
</tr>
```

El resultado del código se muestra a continuación:

**Seleccione el Aeropuerto deseado y presione el botón Continuar**

Elegir	Nombre del Aeropuerto	Siglas OACI	Estado	Ciudad
<input checked="" type="radio"/>	Aeropuerto de Ixtapa-Zihuatanejo	ZIH	Guerrero	Zihuatanejo
<input type="radio"/>	General Juan N. Álvarez	ACA	Guerrero	Acapulco

[Consultar](#)

En caso de no encontrar datos en la base de datos mandará el mensaje correspondiente al usuario, por ejemplo, al iniciar la captura de datos en el sistema actual, no se tenían todavía Aeropuertos de tipo fronterizo, por lo que el sistema enviaría el siguiente mensaje:

**No existen Aeropuertos con ese criterio en México**

[Pulse aquí para REGRESAR...](#)



## Sistema de Información de Aeropuertos



El código PHP es el siguiente:

```
<?$renglones=mysql_numrows($result);
if($renglones<0){
?>
<table border="2" cellpadding="0" cellspacing="0" style="border-collapse: collapse" bordercolor="#0099FF"
width="100%" bgcolor="#99CCFF">
<tr>
<td width="100%">
<p align="center"><b><font face="Tahoma"><?printf("No existen Aeropuertos con ese criterio en
%s", $pais); ?></font></b></td>
</tr>
</table>
</CENTER>
<p align="center"><font color="#666666" face="Arial" size="2">
<A HREF="javascript:history.go(-1);">Pulse aquí para REGRESAR...</a>
</p>
```

Con respecto a la impresión de los datos contenidos en las tablas, únicamente fue necesario agregar una función de javascript en la página php, de tal manera que los datos sean enviados directamente a la impresora:

```
<script language="javascript">
function imprimirPagina() {
if (window.print)
window.print();
else
alert("El navegador no admite impresión automática, utilice el menú de impresión o descargue una versión más reciente del Navegador");
}
</script>
```

De tal manera que cuando el usuario realice un clic sobre el vínculo de impresión, automáticamente abra el cuadro de diálogo de la impresora:





## Sistema de Información de Aeropuertos



El código dentro de la página php para mandar llamar a la función de impresión es simplemente un hipervínculo:

```
<a href=JavaScript:imprimirPagina();>Imprimir</a>
```

Otro tipo de validaciones realizadas en php-javascript son referentes a las descripciones repetidas de los catálogos. En este sentido, es importante destacar que no puede haber dos registros dentro de un mismo catálogo con la misma descripción:

### CONSULTA DE LOS SERVICIOS QUE OFRECE UN AEROPUERTO

Servicio	Características	Tipo
Autobuses foráneos	Servicio de primera clase a las principales ciudades más cercanas del aeropuerto	Transportación Terrestre
Cafeterías	Café, desayunos, sandwiches	Alimentos y Bebidas
Restaurantes	Servicio a la carta	Alimentos y Bebidas
Servicio de Taxis	Servicio dentro del área metropolitana de la ciudad en cuestión	Transportación Terrestre

Así, si se intenta dar de alta nuevamente el servicio de Taxis en el catálogo de servicios:

El sistema enviará el siguiente mensaje a la pantalla:

### **Error!**

El Servicio ya está dado de alta en el Sistema...

[Pulse aquí para REGRESAR...](#)

El código que realiza ésta validación es el siguiente:

```
$$sql0="select * from SERVICIOS where SERVICIOS.descripcion like '$descripcion';  
$result=mysql_query($$sql0,$link);  
$renglones=mysql_numrows($result);  
if($renglones>0){  
    echo"<h3>Error!</h3><p>El Servicio ya está dado de alta en el Sistema...<br><br>  
<h4><a href='\"javascript:history.go(-1);\">Pulse aquí para REGRESAR...</a></h4>";  
}  
  
else{  
$sql1="insert into SERVICIOS(descripcion,caracteristicas,tipo_servicio)  
values ('$descripcion','$caracteristicas','$tipo')";  
mysql_query($sql1,$link);
```



## Sistema de Información de Aeropuertos



En este caso se realiza un comando *SELECT* de *SQL* hacia la tabla en cuestión. Si el dato existe, entonces se le envía al usuario un mensaje indicándole que el dato ya está dado de alta, y por medio de una función en *JavaScript* permite regresar el navegador a la página anterior. En caso contrario, se da de alta el registro con un comando *INSERT* de *SQL*.

Otra validación realizada fue sobre los campos requeridos, en donde si el dato que el usuario está insertando es nulo, el sistema lo detecta y antes de guardarlo en la base de datos indica al usuario acerca del error cometido:

### **Error!**

Falto definir la descripción del Servicio ...

[Pulse aquí para REGRESAR...](#)

En este caso si el campo está vacío se le indica al usuario el error, y por medio del método *go* de *JavaScript* es posible regresar a la página anterior.

```
if ($descripcion=="") {  
  echo"<h3>Error!</h3><p>Falto definir la descripción del Servicio ...<br><br>  
<h4><A HREF='\"javascript:history.go(-1);\">Pulse aquí para REGRESAR...</a></h4>";  
}
```



### 8. OPERACIÓN DEL SISTEMA

#### 8.1 DOCUMENTACIÓN

La documentación de un sistema de información debe ir dirigida principalmente los siguientes grupos de usuarios: analistas, desarrolladores, administradores y usuarios finales, cada una con sus respectivas características. Las ventajas de un sistema de información correctamente documentado son varias: el tiempo de aprendizaje del sistema será mucho menor, es posible obtener la ayuda de otras personas en la parte de análisis, desarrollo, y administración, la interacción entre el sistema y los usuarios finales es más estrecha, y sobre todo, si alguna persona del equipo de desarrollo sale de la empresa, el sistema puede seguir trabajando, es decir, no hay dependencia hacia una persona o un grupo de personas. En este sentido, la documentación de un manual está dividida en dos grandes bloques, la documentación técnica, dirigida al analista, al desarrollador y al administrador, y la documentación de usuario, dirigida a los usuarios finales los cuales representan a fin de cuentas la mayoría de personas que van a hacer uso del sistema de información.

#### 8.2 MANUAL DE USUARIO

El manual de usuario debe estar pensado en los usuarios finales, explicando de manera clara, concisa y sin tecnicismos las principales funciones del sistema, así como la definición de los pasos necesarios para realizar las tareas correspondientes, apoyándose en imágenes o pantallas capturadas del propio sistema. Deben contar con una introducción que describa los objetivos del sistema, los requerimientos de hardware y de software, la descripción general del sistema, el esquema de operación: como ingresar y como salir de manera correcta del sistema, las pantallas de captura de datos, tanto principales como secundarias, las pantallas de envío de información, la pantalla acerca de, y preferentemente la ayuda en línea, es decir que el usuario pueda consultar su manual directamente desde la computadora, en formato html por ejemplo. La parte del sistema correspondiente a las consultas, búsquedas e impresión de los datos deben estar claramente detalladas, con sus respectivas pantallas de captura, así como un esquema de los principales errores que el usuario puede cometer durante la operación del sistema.

#### 8.3 ESQUEMA DE OPERACIÓN

El Sistema de Información de Aeropuertos tiene como finalidad brindar la búsqueda de datos importantes referidos a los aeropuertos, tanto nacionales como internacionales. Está diseñado para trabajar sobre Internet, con cualquier navegador comercial en versiones actuales, incluso con cualquier sistema operativo, sea Windows o Linux. Es un sistema sumamente amigable al usuario, debido a que el manejo se realizan a través de un navegador de Internet, como el Internet Explorer o el Netscape Navigator. Cabe destacar que su aprendizaje es sumamente rápido, debido a su diseño intuitivo basado en Web.

#### 8.4 PERSONAL TÉCNICO RESPONSABLE

El personal técnico encargado de la operación del sistema se divide en dos partes: El administrador de la red y del sistema operativo, es el encargado de verificar que el servidor Web en el que reside dicho sistema trabaje de manera adecuada, será el encargado de realizar los respaldos a la base de datos, así como la restauración de los mismos en caso necesario, así como dar de alta a nuevos usuarios, cancelar cuentas y cambiar contraseñas de usuarios. Otro usuario con perfiles importantes es el capturista, el cual tendrá los permisos de lectura y escritura sobre los catálogos del sistema, lo que implica que podrá eliminar registros, insertar registros y modificar datos existentes, de tal manera que la base de datos del sistema esté actualizada de manera correcta.



## Sistema de Información de Aeropuertos



El último perfil de usuario está dado por los usuarios finales, los cuales con su respectiva cuenta y contraseña podrán hacer uso del sistema de información a fin de poder consultar y en caso necesario imprimir los datos requeridos de los aeropuertos, los cuales son muy fácilmente localizables, debido al diseño del sistema y a los asistentes con los que cuenta.

### 8.5 REQUERIMIENTOS MINIMOS DE OPERACIÓN

Los requerimientos de operación se dividen en dos partes: los requerimientos a nivel de hardware, y los requerimientos a nivel de software.

Requerimientos de hardware:

PC Pentium o equivalente con 32 MB en RAM, monitor a color de 800 x 600 pixeles a 256 colores, conexión a Internet ya sea por Módem o por red

Requerimientos de software:

Cualquier sistema operativo, tanto Windows como *Linux*, correctamente configurado para trabajar en Internet, y tener instalado cualquier navegador que admita marcos (*frames*), *scripts* de *javascript* y sea compatible con *HTML 4.0*. Se recomienda *Microsoft Internet Explorer 5* o superior, *Netscape Communicator 6* o superior, *Konqueror 2.1* o superior.

### 8.6 AMBIENTE DE TRABAJO

El ambiente en el que trabaja el sistema es un ambiente 100% Web, por lo que se recomienda que los usuarios tengan conocimientos sobre el manejo de un navegador Web, así como de las bases para navegar por Internet. Cabe destacar que el Sistema de información de Aeropuertos es un sistema sumamente intuitivo, de tal manera que las tareas de consulta e impresión son sumamente fáciles de aprender.

#### Asistente para Seleccionar Aeropuertos Nacionales

1.-Seleccione el criterio de búsqueda y presione el botón Siguiente

Ciudad del Aeropuerto ▾

Siguiente

Ingreso al sistema: Para ingresar al sistema, será necesario tener el navegador abierto y una conexión a Internet ya establecida. Será necesario escribir la dirección URL del sistema directamente en la barra de navegación del mismo: [www.asa.gob.mx/general/rosario/sia](http://www.asa.gob.mx/general/rosario/sia)

Una vez escrita la dirección URL, será necesario ingresar la clave y la contraseña del usuario en la sección de usuarios. Si los datos son correctos, el usuario tendrá acceso al menú principal del Sistema.

Usuarios	
Clave	rosario
Contraseña	*****
<input type="button" value="Enviar"/>	



## Sistema de Información de Aeropuertos



En caso contrario, enviará cualquiera de los siguientes mensajes según sea el caso.

**Error!**

**Usuario Inexistente**

[Pulse aquí para REGRESAR...](#)

**Error!**

**La Contraseña está mal escrita**

[Pulse aquí para REGRESAR...](#)

### 8.6.1 Edición de datos

Para poder capturar datos deberá tener los permisos de captura. Será necesario contactar al administrador para tales permisos. La captura de datos se da por medio de formas, las cuales están clasificadas de acuerdo al tipo de información que contiene cada una. En este sentido, se tienen las formas primarias del sistema, las cuales empiezan por la palabra Tipo. Se les denomina primarias, porque deben ser llenadas antes de capturar información en las demás formas, por ejemplo, al seleccionar la forma de captura de servicios y presionar el botón Enviar

-
Ciudades
Estados/Provincias
Países
-
Tipo de Aeropuertos
Tipo de Ayudas
Tipo de Obras
Tipo de Pisos
Tipo de Plataformas
<b>Tipo de Servicios</b>
Tipo de Servicios

Aparecerán los servicios existentes en el aeropuerto que estén dados de alta en el sistema ordenados de manera alfabética:

CONSULTA DE TIPOS DE SERVICIOS EN AEROPUERTOS	
Descripción	
<a href="#">Alimentos y Bebidas</a>	
<a href="#">Artículos de Viaje</a>	
<a href="#">Comunicaciones</a>	
<a href="#">Farmacias</a>	
<a href="#">Guarda Equipajes</a>	
<a href="#">Hoteles</a>	
<a href="#">Librerías</a>	
<a href="#">Lotería</a>	
<a href="#">Servicios Financieros</a>	
<a href="#">Tabaquerías, Dulces y Regalos</a>	
<a href="#">Tiendas Libres de Impuestos</a>	
<a href="#">Varios</a>	
<a href="#">Transportación Terrestre</a>	

[Agregar Tipo de Servicio](#) | [Modificar Tipo de Servicio](#) | [Eliminar Tipo de Servicio](#)

Si se requiere la edición de los mismos será necesario seguir los vínculos a las opciones de Agregar, Modificar o Eliminar el Tipo de Servicio. Cabe destacar que sólo el capturista y el administrador tendrán acceso a esta parte del Sistema.



## Sistema de Información de Aeropuertos



### 8.6.1 Edición de datos

#### 8.6.1.1 Agregar datos

Al elegir la opción Agregar Tipo de Servicio, aparecerá la pantalla de captura para los tipos de Servicios que ofrece un aeropuerto:

### CAPTURA DEL CATÁLOGO DE TIPOS DE SERVICIOS DE UN AEROPUERTO

Escriba los datos del Tipo de Servicio que vaya a agregar al Sistema.

Tipo de Servicios	
Descripción	<input type="text"/>
<input type="button" value="Enviar"/> <input type="button" value="Restablecer"/>	

En el campo descripción será necesario escribir la descripción del tipo de servicio, por ejemplo, si se escribe en la descripción el valor Transportación Terrestre, el cual ya está dado de alta y se presiona el botón Enviar:

Tipo de Servicios	
Descripción	<input type="text" value="Transportación Terrestre"/>
<input type="button" value="Enviar"/> <input type="button" value="Restablecer"/>	

El resultado será el siguiente:

### **Error!**

El Tipo de Servicio ya está dado de alta en el Sistema...

[Pulse aquí para REGRESAR...](#)

Al presionar la liga para regresar y escribir algún tipo de servicio que no exista, por ejemplo, Servicios Médicos y presionar el botón Enviar



## Sistema de Información de Aeropuertos



Tipo de Servicios	
Descripción	Servicios Médicos
<input type="button" value="Enviar"/> <input type="button" value="Restablecer"/>	

El resultado será el siguiente:

**El Tipo de Servicio Servicios Médicos fue agregado con éxito**

[Consultar Tipo de Servicios](#)

El botón Restablecer está presente en todas las pantallas de captura del Sistema.

Tipo de Servicios	
Descripción	
<input type="button" value="Enviar"/> <input type="button" value="Restablecer"/>	

En este sentido, cuando el usuario presiona el botón Restablecer antes de Enviar los datos capturados, éstos son puestos en blanco.

### 8.6.1.2 Modificar datos

La tarea de modificar datos existentes es sumamente simple, por ejemplo, para modificar los datos del servicio de autobuses será necesario seleccionar la opción servicios del menú Edición de datos, y seguir el vínculo Modificar Servicio

CONSULTA DE LOS SERVICIOS QUE OFRECE UN AEROPUERTO		
Servicio	Características	Tipo
Autobuses foráneos	Servicio de primera clase a las principales ciudades más cercanas del aeropuerto	Transportación Terrestre
Cafeterías	Café, desayunos, sandwiches	Alimentos y Bebidas
Restaurantes	Servicio a la carta	Alimentos y Bebidas
Servicio de Taxis	Servicio dentro del área metropolitana de la ciudad en cuestión	Transportación Terrestre

Consulta de Datos: 
  
                                   Nacionales
   
                                   Internacionales

Edición de Datos:

[Agregar Servicio](#) | [Modificar Servicio](#) | [Eliminar Servicio](#)



## Sistema de Información de Aeropuertos



Una vez presionada la liga Modificar Servicio el siguiente paso será seleccionar el servicio a modificar y presionar el botón correspondiente:

MODIFICACIÓN DE SERVICIOS	
Seleccione el Servicio a Modificar	
Cafeterías	Modificar
Cafeterías	
Restaurantes	
Servicio de Taxis	
Autobuses foráneos	

A continuación aparecerán los datos referidos al servicio en cuestión

MODIFICACIÓN DE SERVICIOS	
Datos Generales:	
Descripción:	Autobuses foráneos
Características:	Servicio de primera clase a las principales ciudades más cercanas del aeropuerto
Tipo:	Transportación Terrestre
<a href="#">Modificar</a>	

De esta manera los datos pueden ser editados y para guardar los cambios será necesario presionar el botón Modificar, por ejemplo, en este caso, se reemplazará Autobuses foráneos por Autobuses Foráneos, así como las características del servicio en cuestión:

Datos Generales:	
Descripción:	Autobuses Foráneos
Características:	Servicio de clase ejecutiva a las principales ciudades cercanas al aeropuerto
Tipo:	Transportación Terrestre



Al presionar el botón modificar, el sistema enviará el mensaje correspondiente a la acción:

**Los datos del Servicio Autobuses Foráneos fueron Actualizados**

[Consultar Servicios](#)

De tal manera que cuando se presione el botón Consultar Servicios los datos aparecen actualizados.

### CONSULTA DE LOS SERVICIOS QUE OFRECE UN AEROPUERTO

Servicio	Características	Tipo
Autobuses Foráneos	Servicio de clase ejecutiva a las principales ciudades cercanas al aeropuerto	Transportación Terrestre
Cafeterías	Café, desayunos, sandwiches	Alimentos y Bebidas
Restaurantes	Servicio a la carta	Alimentos y Bebidas
Servicio de Taxis	Servicio dentro del área metropolitana de la ciudad en cuestión	Transportación Terrestre

#### 8.6.1.3 Eliminar registros

El sistema contempla la eliminación de registros en cascada únicamente para los aeropuertos. Esto quiere decir que al eliminar un aeropuerto del sistema, todos los datos que dependen de dicho aeropuerto serán eliminados.

Sin embargo, es el único caso en el que los datos son eliminados en cascada, ya que en los demás casos, si se intenta eliminar un registro y hay datos que dependen de dicho registro el sistema no lo permitirá. Por ejemplo, se tiene el servicio de Autobuses Foráneos, el cual es del tipo Transportación Terrestre, es decir, Autobuses Foráneos depende de Transportación Terrestre. Si se intenta eliminar el servicio de transportación terrestre, el sistema no lo permitirá.

Para intentar eliminar el Tipo de Servicio Transportación Terrestre, seleccione dentro del menú Edición de Datos la opción Tipo de Servicio y presione el botón Enviar.

#### Edición de Datos

Tipo de Servicios ▼

Enviar



## Sistema de Información de Aeropuertos



A continuación será necesario presionar la liga correspondiente a Eliminar Tipo de Servicios

[Agregar Tipo de Servicio](#) | [Modificar Tipo de Servicio](#) | [Eliminar Tipo de Servicio](#)

Posteriormente será necesario seleccionar el servicio en cuestión, que en este caso es Transportación Terrestre y presionar el botón Eliminar. A continuación el sistema desplegará el mensaje correspondiente indicando si se pudo o no eliminar el tipo de servicio, por ejemplo en este caso, el tipo de servicio no pudo ser eliminado:

**No se puede eliminar el Tipo de Servicio.**

Los siguientes Servicios dependen de dicho Tipo. Elimine primero los Servicios y repita la operación

Servicios
Servicio de Taxis
Autobuses Foráneos

[Pulse aquí para REGRESAR...](#)

Si por ejemplo se intentara eliminar el Tipo de Servicio Librerías que en este caso no tiene todavía ningún servicio que dependa de él, el resultado sería el siguiente:

**El Tipo de Servicio fue eliminado con éxito**

[Consultar Tipo de Servicios](#)

Al consultar el tipo de servicio, ya no debe estar en el Sistema el tipo Librería.

### 8.7 CATÁLOGOS DEL SISTEMA

Se tienen dos tipos de catálogos en el sistema, los catálogos primarios, y los catálogos secundarios. Los primarios, como se mencionó anteriormente, constituyen la base de captura de catálogos secundarios, por ejemplo, para poder dar de alta un servicio dentro de un aeropuerto, es necesario que previamente exista el tipo de servicio.

Datos Generales:	
Descripción:	Autobuses Foráneos
Características:	Servicio de clase ejecutiva a las principales ciudades cercanas al aeropuerto
Tipo:	Transportación Terrestre

- Alimentos y Bebidas
- Artículos de Viaje
- Comunicaciones
- Servicios Financieros
- Tabaquerías, Dulces y Regalos
- Servicios Médicos
- Varios



## Sistema de Información de Aeropuertos



Esto implica que para poder dar de alta un servicio, el tipo de servicio debe ser dado de alta previamente. Si en alguna pantalla de captura, alguna de las listas desplegadas está vacía, significa que no se dio de alta previamente el catálogo primario correspondiente, por ejemplo:

### 8.8 EDICIÓN DE DATOS DE AEROPUERTOS

La captura de datos de los Aeropuertos se realiza por medio de un Asistente el cual permite agregar los datos de manera sumamente sencilla, siguiendo los pasos. Una vez capturados los datos generales del Aeropuerto será necesario capturar la información específica de cada Aeropuerto. Para agregar un nuevo Aeropuerto seleccione la opción agregar Aeropuerto dentro de la sección de Aeropuertos en la opción Edición de datos

#### Agregar Aeropuerto | Modificar Aeropuertos | Eliminar Aeropuertos

El asistente solicitará el País, y el Estado al que pertenece el Aeropuerto. Una vez indicados estos datos, aparecerá la pantalla de captura de los datos generales, que incluyen la Ciudad, Clasificación y Tipo. Una vez que estos datos fueron dados de alta, la pantalla de datos generales de Aeropuertos incluye 10 opciones adicionales con información específica de cada Aeropuerto:

Pistas	Ayudas	Plataformas	Rodajes	Suministros
CREI	Servicios	Responsables	Obras	Imágenes

Consulta de Datos Generales del Aeropuerto ZIH	
Siglas ACILAC	ZIH
Siglas OACI	MMZH
Nombre	Aeropuerto de Ixtapa-Zihuatanejo
Clasificación	Internacional
Tipo	Turístico
Ciudad	Zihuatanejo
Estado	Guerrero
País	México

Cada uno de estos botones tiene la misma lógica de visualización y edición de datos que los catálogos vistos anteriormente, por ejemplo, al presionar el botón Servicios, la pantalla mostrada será la siguiente:

Datos Generales	
Consulta de los Servicios Existentes en el Aeropuerto: Aeropuerto de Ixtapa-Zihuatanejo de Zihuatanejo del País México	
Descripción	Autobuses Foráneos
Características	Servicio a las localidades cercanas
Ubicación	Exterior del Edificio Terminal
Tipo	Transportación Terrestre

Agregar Servicio	Modificar Servicio	Eliminar Servicio
------------------	--------------------	-------------------

En este caso al presionar el botón Datos Generales, el sistema regresará a la pantalla anterior. En el caso de los botones ubicados en la parte inferior, las opciones para agregar, modificar y eliminar el servicio son las mismas que en los catálogos comentados anteriormente, incluyendo la validación de datos duplicados y valores nulos.



## Sistema de Información de Aeropuertos



Por ejemplo, si se intenta dar de alta un servicio que ya se tiene en el aeropuerto, el mensaje que aparecerá es el siguiente:

### **Error!**

El Servicio ya está dado de alta en el Aeropuerto...

**Pulse aquí para REGRESAR...**

En caso de que se intente ingresar un servicio que no esté dado de alta en el Aeropuerto, el resultado será satisfactorio:

**El Servicio fue agregado con éxito en el Aeropuerto**

Consultar Servicios

Incluso se tiene el botón de consulta de los servicios que ofrece dicho Aeropuerto

#### Consulta de los Servicios Existentes en el Aeropuerto: Aeropuerto de Ixtapa-Zihuatanejo de Zihuatanejo del País México

Descripción	Autobuses Foráneos
Características	Servicio a las localidades cercanas
Ubicación	Exterior del Edificio Terminal
Tipo	Transportación Terrestre

Descripción	Servicio de Taxis
Características	Servicio a Ixtapa, Zihuatanejo y a municipios cercanos
Ubicación	Exterior del Edificio Terminal
Tipo	Transportación Terrestre



## Sistema de Información de Aeropuertos



### 8.9 DEPENDENCIA DE CATÁLOGOS

La siguiente tabla muestra la dependencia de catálogos, esto significa que para poder dar de alta algún catálogo, como el de Servicios de los Aeropuertos debe existir previamente el de Tipo de Servicios. Otro ejemplo lo constituye la Ciudad ya que para dar de alta una Ciudad, será necesario ingresar primeramente los datos del estado o provincia, y para dar de alta un Estado, el País al que pertenece dicho Estado debe existir previamente, por lo que el orden de captura de datos será País, Estado, Ciudad.

<b>Catálogo</b>	<b>Depende de los Catálogos</b>
Aeropuertos	Clasificación de Aeropuertos, Tipo de Aeropuertos, Países
Aviones	
Ayudas de Aterrizaje	Tipo de Ayudas
Clasificación de Aeropuertos	
Ciudades	Estados
Cuerpo de Rescate	
Estados/Provincias	Países
Países	
Responsables	
Servicios	Tipo de Servicios
Suministros	
Tipo de Aeropuertos	
Tipo de Ayudas	
Tipo de Obras	
Tipo de Pisos	
Tipo de Plataformas	
Tipo de Servicios	
Pistas en Aeropuertos	Aeropuertos, Aviones, Tipo de Pisos
Ayudas en Aeropuertos	Aeropuertos, Ayudas de Aterrizaje
Plataformas en Aeropuertos	Aeropuertos, Plataformas
Rodajes en Aeropuertos	Aeropuertos
Suministros en Aeropuertos	Aeropuertos, Suministros
Cuerpo de Rescate en Aeropuertos	Aeropuertos, Cuerpo de Rescate
Servicios en Aeropuertos	Aeropuertos, Servicios
Responsables en Aeropuertos	Aeropuertos, Responsables
Obras en Aeropuertos	Tipo de Obras
Imágenes en Aeropuertos	Aeropuertos

### 8.10 CONSULTA DE DATOS POR PARTE DE LOS USUARIOS FINALES

La consulta de datos por parte de los Usuarios Finales es un proceso sumamente sencillo, en primer lugar el usuario al abrir el sistema tendrá que ingresar los datos referidos a su clave y contraseña:



## Sistema de Información de Aeropuertos



Una vez ingresados los datos, el usuario deberá seleccionar entre la opción de consultar Aeropuertos Nacionales y Aeropuertos Internacionales:

### Consulta de Datos

<b>Aeropuertos</b>
Nacionales
Internacionales

La diferencia entre ambas selecciones radica en que en el caso de seleccionar Aeropuertos nacionales, el sistema asume directamente la búsqueda de Aeropuertos para México, mientras que en el caso de los internacionales, será necesario especificar el país al que se refiere. Incluso, desde la sección de Aeropuertos Internacionales se pueden consultar los Aeropuertos de México seleccionando el País de la lista:

1.-Seleccione el País al que pertenece el Aeropuerto y presione el botón Seleccionar.

<b>Lista de Países</b>																									
<table border="1"><tr><td>Panamá</td><td>▼</td></tr><tr><td>Colombia</td><td>▲</td></tr><tr><td>Costa Rica</td><td></td></tr><tr><td>Ecuador</td><td></td></tr><tr><td>Estados Unidos</td><td></td></tr><tr><td>Francia</td><td></td></tr><tr><td>Guatemala</td><td></td></tr><tr><td>México</td><td></td></tr><tr><td>Panamá</td><td></td></tr><tr><td>Paraguay</td><td></td></tr><tr><td>Uruguay</td><td></td></tr><tr><td>Venezuela</td><td>▼</td></tr></table>	Panamá	▼	Colombia	▲	Costa Rica		Ecuador		Estados Unidos		Francia		Guatemala		México		Panamá		Paraguay		Uruguay		Venezuela	▼	<input type="button" value="Siguiete"/>
Panamá	▼																								
Colombia	▲																								
Costa Rica																									
Ecuador																									
Estados Unidos																									
Francia																									
Guatemala																									
México																									
Panamá																									
Paraguay																									
Uruguay																									
Venezuela	▼																								

A partir de este punto, las opciones para búsqueda son las mismas tanto en Aeropuertos Nacionales como en Aeropuertos Internacionales. El siguiente paso es buscar los Aeropuertos de acuerdo al criterio seleccionado por el usuario. Cabe destacar que únicamente se mostrarán los Aeropuertos del País seleccionado.



## Sistema de Información de Aeropuertos



Para ello habrá que seleccionar el criterio de búsqueda, por ejemplo, el Estado del Aeropuerto.

### Asistente para Seleccionar Aeropuertos Internacionales del País México

2.-Seleccione el criterio de búsqueda y presione el botón Siguiente

Ciudad del Aeropuerto	▼
Nombre del Aeropuerto	
Estado del Aeropuerto	■
Ciudad del Aeropuerto	
Tipo de Aeropuerto	
Clasificación del Aeropuerto	

Al presionar el botón siguiente, aparecerán únicamente los Estados del País seleccionado:

2.-Seleccione el Estado al que pertenece el Aeropuerto y presione el botón Siguiente.

Lista de Estados de México	
Aguascalientes	▼
Aguascalientes	▲
Baja California Norte	
Baja California Sur	
Campeche	
Chiapas	
Chihuahua	
Coahuila	
Colima	
Distrito Federal	■
Durango	
Estado de México	▼

Una vez seleccionado el estado, el sistema permitirá escoger entre los aeropuertos que tenga el Estado, por ejemplo, si se selecciona el Distrito Federal, aparecerá el único Aeropuerto que se tiene en el Distrito Federal

3.-Seleccione el Aeropuerto deseado y presione el botón Continuar

Elegir	Nombre del Aeropuerto	Siglas OACI	Estado	Ciudad
☞	Aeropuerto Internacional de la Ciudad de México	MEX	Distrito Federal	Ciudad de México



## Sistema de Información de Aeropuertos



Si se hubiera seleccionado por ejemplo el Estado de Guerrero, el sistema hubiera mostrado los Aeropuertos que se tienen capturados en el Estado de Guerrero:

### 3.-Seleccione el Aeropuerto deseado y presione el botón Continuar

Elegir	Nombre del Aeropuerto	Siglas OACI	Estado	Ciudad
<input type="radio"/>	Aeropuerto de Ixtapa-Zihuatanejo	ZIH	Guerrero	Zihuatanejo
<input type="radio"/>	General Juan N. Álvarez	ACA	Guerrero	Acapulco

Consultar

De tal manera que el usuario pueda seleccionar el Aeropuerto que requiera consultar. Por ejemplo, en el caso del Aeropuerto de la Ciudad de México, al presionar el botón consultar los datos que aparecerán en pantalla serán los siguientes:

Pistas	Ayudas	Plataformas	Rodajes	Suministros
CREI	Servicios	Responsables	Obras	Imágenes

#### Consulta de Datos Generales del Aeropuerto MEX

Siglas ACILAC	MEX
Siglas OACI	MMEX
Nombre	Aeropuerto Internacional de la Ciudad de México
Clasificación	Internacional
Tipo	Metropolitano
Ciudad	Ciudad de México
Estado	Distrito Federal
Pais	México
Fecha Inauguración	1964-10-05
Capacidad Anual de Pasajeros	500000000
Capacidad Anual de Operaciones	15768000
Num. de Edificios Terminales	3
Superficie Total	2000000 metros cuadrados
Numero de Estacionamientos	3
Superficie de Estacionamientos	40000 metros cuadrados

En la parte posterior de la pantalla se tienen todas las opciones con los datos específicos del Aeropuerto: Pistas, Ayudas, Plataformas, Calles de Rodajes, Suministros de Combustible, Cuerpo de Rescate, Servicios, Responsables, Obras e Imágenes.



## Sistema de Información de Aeropuertos



Por ejemplo, al seleccionar la opción Servicios la ventana desplegará los servicios que se tengan capturados en el Aeropuerto.

Datos Generales

### Consulta de los Servicios Existentes en el Aeropuerto: Aeropuerto Internacional de la Ciudad de México de Ciudad de México del País México

Descripción	Autobuses Foráneos
Características	Servicio a las ciudades de Toluca, Pachuca, Cuernavaca y Querétaro
Ubicación	Exterior del Edificio Terminal
Tipo	Transportación Terrestre

Descripción	Servicio de Taxis
Características	Servicio en el Distrito Federal y Area Metropolitana
Ubicación	Exterior del Edificio Terminal
Tipo	Transportación Terrestre

Al presionar el botón Datos Generales el sistema regresará a la pantalla anterior.

### 8.11 RESPALDO DE LA BASE DE DATOS

Los respaldos del Sistema deberán ser realizados por el Administrador del Sistema. En este sentido, se recomienda utilizar la interfaz gratuita MySQLAdmin disponible en la página [www.mysql.com](http://www.mysql.com).

Esta interfaz deberá ser instalada en una carpeta a la que tenga acceso únicamente el administrador. Para respaldar las tablas de la Base de datos, el administrador deberá seleccionar todas las tablas de la base de datos, elegir la opción Estructura y Datos, activar las casillas de verificación Añadir Drop Table y Complete Inserts, y presionar el botón Siga. Se generará un archivo txt con el volcado de la Base de Datos.

- Ver volcado esquema de la base de datos

AEROPUERTOS  
AREAS  
AVIONES  
AYUDAS\_ATERRIJAJES  
AYUDAS\_PISTAS

- Únicamente estructura  Añadir 'drop table'
- Estructura y datos  Complete inserts
- Solo datos  enviar (  "gzipped" )

Siga

### 8.12 RESTAURACIÓN DE LA BASE DE DATOS

La restauración se puede hacer dentro de la misma herramienta, seleccionando la opción Localización del Archivo de Texto y eligiendo el archivo con el último respaldo que se tenga.



O Localización del archivo de texto :

C:\respaldo\_05102003.txt

### 8.13 SALIDA DEL SISTEMA

Para salir del Sistema, será necesario presionar la opción Salir. De esta manera la sesión del usuario quedará cerrada.



### CONCLUSIONES DEL PROYECTO

El Sistema de Información de Aeropuertos permitirá a los Usuarios de ASA tener el registro en línea de todos los aeropuertos que se tienen en México, así como la captura de datos para aeropuertos extranjeros de manera rápida y sencilla, ya que actualmente los datos de los aeropuertos están contenidos en bitácoras de papel, las cuales son muy grandes y a veces difíciles de consultar.

La tecnología de desarrollo Web está cobrando mucha popularidad dentro de la comunidad desarrolladora en estos últimos tiempos. La ventaja de este tipo de sistemas es que pueden ser consultados y explotados desde cualquier computadora en el mundo con conexión a Internet.

El sistema opera en un ambiente Cliente Servidor, en donde los clientes son los navegadores, y el servidor en éste caso es el Apache de Linux. La plataforma Linux es una plataforma de desarrollo sumamente robusta, y con numerosas herramientas para el desarrollo de aplicaciones, como PHP por ejemplo.

Este sistema me permitió obtener los conocimientos y experiencia necesaria para desarrollar sistemas basados en Web con Acceso a bases de datos.

En particular me gustó mucho desarrollar sobre PHP, debido a que su sintaxis es muy similar a la de lenguaje C, por lo que la etapa de aprendizaje de esta nueva plataforma fue inmediato.

Fue muy grato poder trabajar con la Lic. Díaz de Guzmán a la cual le agradezco la posibilidad de haberme involucrado en este proyecto.

Le agradezco a la Ing. Liliana Muñoz Zafra su tiempo, dedicación y asesoría en el desarrollo del proyecto.



---

### BIBLIOGRAFIA

- Atre Shakuntala, Técnicas de Bases de Datos Estructuración en Diseño, Trillas 1988, México
- Deen Sayyeed, Fundamentos de los Sistemas de Bases de Datos, Mc Graw Hill 1987, Barcelona España
- Elmasri Ramez, Sistemas de Bases de Datos, Addison Wesley 1997, Wilmington
- Hawryszkiewicz Igor Titus, Análisis y Diseño de Bases de Datos, Limusa 1994, México
- Korth Henry F, Fundamentos de Bases de Datos, McGrawHill 1987, México
- Lyon John K, Bases De Datos, Ateneo 1983, México
- Zikopoulos : DBA S Guide To Database On Linux, Publishers GRP SBS 2000, Rockland
- Flanagan David , : Javascript The Definitive Guide, OReilly 1997, Cambridge
- Johnson Marc, Javascript Manual of Style, Davis 1996, Emeryville
- Atkinson Leon, Using PHP to Build Dynamic Web, Prentice Hall 1999, Estados Unidos
- Hilton Craig, Building Database Applications on The Web Using PHP, Addison Wesley 2000, Estados Unidos
- Medinets David, PHP 3 Programming Browser Based Applications, McGraw Hill 2000, New York
- Clements Paul, Constructing Superior Software, Mcmillan Computer, 2000, Estados Unidos
- Douglas Bell, Software Engineering a Programming Approach, Addison Wesley, 2000 Harlow England
- McGoveran D, A Guide to Sybase And SQL Server a Users Guide to, Addison Wesley 1992 , New York

### SITIOS DE INTERNET

- <http://www.php.net/>
- <http://www.mysql.com/>
- <http://www.redhat.com/>
- <http://www.apache.org/>



### APENDICE A COMANDOS DDL PARA LA CREACIÓN DE LA BASE DE DATOS

```
/* ===== */
/* Table: CONTINENTE */
/* ===== */
create table CONTINENTE
(
  ID_CONTINENTE      tinyint      not null,
  CONTINENTE         varchar      null ,
  constraint PK_CONTINENTE primary key (ID_CONTINENTE)
)
go

/* ===== */
/* Table: REGION */
/* ===== */
create table REGION
(
  ID_REGION          tinyint      not null,
  REGION             varchar      null ,
  constraint PK_REGION primary key (ID_REGION)
)
go

/* ===== */
/* Table: CLASIFICACION */
/* ===== */
create table CLASIFICACION
(
  ID_CLASIFICACION   tinyint      not null,
  CLASIFICACION      varchar      null ,
  constraint PK_CLASIFICACION primary key (ID_CLASIFICACION)
)
go

/* ===== */
/* Table: TIPO */
/* ===== */
create table TIPO
(
  ID_TIPO            tinyint      not null,
  TIPO               varchar      null ,
  constraint PK_TIPO primary key (ID_TIPO)
)
go
```



## Sistema de Información de Aeropuertos



```
/* ===== */
/* Table: TIPO_RESPONSABLE */
/* ===== */
create table TIPO_RESPONSABLE
(
  ID_TIPO_RESPONSABLE tinyint not null,
  DESCRIPCION varchar null ,
  constraint PK_TIPO_RESPONSABLE primary key (ID_TIPO_RESPONSABLE)
)
go

/* ===== */
/* Table: TIPO_ROL_RESPONSABLE */
/* ===== */
create table TIPO_ROL_RESPONSABLE
(
  ID_TIPO_ROL_RESPONSABLE tinyint not null,
  DESCRIPCION varchar null ,
  constraint PK_TIPO_ROL_RESPONSABLE primary key
(ID_TIPO_ROL_RESPONSABLE)
)
go

/* ===== */
/* Table: TIPO_PISO */
/* ===== */
create table TIPO_PISO
(
  ID_TIPO_PISO tinyint not null,
  DESCRIPCION varchar null ,
  constraint PK_TIPO_PISO primary key (ID_TIPO_PISO)
)
go

/* ===== */
/* Table: AVION */
/* ===== */
create table AVION
(
  ID_AVION integer not null,
  MODELO varchar null ,
  NUM_MAX_PASAJE integer null ,
  PESO_MAX_DESPEGUE decimal null ,
  CAPACIDAD_MAX_COMBUSTIBLE decimal null ,
  RANGO_MAX_OPERACION integer null ,
  VELOCIDAD_CRUCERO integer null ,
  ENVERGADURA float null ,
  LARGO_TOTAL float null ,
  ALTO_TOTAL float null ,
  ANCHO_INTERIOR float null ,
  IMAGEN varchar(100) null ,
  constraint PK_AVION primary key (ID_AVION)
)
```



```
)  
go
```

```
/* ===== */  
/* Table: TIPO_AYUDA */  
/* ===== */  
create table TIPO_AYUDA  
(  
  ID_TIPO_AYUDA      tinyint      not null,  
  DESCRIPCION        varchar      null ,  
  constraint PK_TIPO_AYUDA primary key (ID_TIPO_AYUDA)  
)  
go
```

```
/* ===== */  
/* Table: PLATAFORMAS */  
/* ===== */  
create table PLATAFORMAS  
(  
  ID_PLATAFORMA      tinyint      not null,  
  DESCRIPCION        varchar      null ,  
  constraint PK_PLATAFORMAS primary key (ID_PLATAFORMA)  
)  
go
```

```
/* ===== */  
/* Table: SUMINISTRO */  
/* ===== */  
create table SUMINISTRO  
(  
  ID_SUMINISTRO      tinyint      not null,  
  DESCRIPCION        varchar      null ,  
  IMAGEN             varchar      null ,  
  constraint PK_SUMINISTRO primary key (ID_SUMINISTRO)  
)  
go
```

```
/* ===== */  
/* Table: TIPO_OBRA */  
/* ===== */  
create table TIPO_OBRA  
(  
  ID_TIPO_OBRA      tinyint      not null,  
  DESCRIPCION        varchar      null ,  
  constraint PK_TIPO_OBRA primary key (ID_TIPO_OBRA)  
)  
go
```



## Sistema de Información de Aeropuertos



```
/* ===== */
/* Table: TIPO_SERVICIO */
/* ===== */
create table TIPO_SERVICIO
(
  ID_TIPO_SERVICIO tinyint not null,
  DESCRIPCION varchar null,
  constraint PK_TIPO_SERVICIO primary key (ID_TIPO_SERVICIO)
)
go

/* ===== */
/* Table: UBICACION_SERVICIO */
/* ===== */
create table UBICACION_SERVICIO
(
  ID_UBICACION tinyint not null,
  DESCRIPCION varchar null,
  constraint PK_UBICACION_SERVICIO primary key (ID_UBICACION)
)
go

/* ===== */
/* Table: CREI */
/* ===== */
create table CREI
(
  ID_CREI integer not null,
  DESCRIPCION varchar(100) not null,
  IMAGEN varchar(100) null,
  constraint PK_CREI primary key (ID_CREI)
)
go

/* ===== */
/* Table: PAIS */
/* ===== */
create table PAIS
(
  ID_PAIS tinyint not null,
  NOMBRE varchar null,
  CONTINENTE tinyint null,
  REGION tinyint null,
  SUPERFICIE decimal null,
  POBLACION decimal null,
  PIB decimal null,
  ADMON_AEREA varchar(250) null,
  constraint PK_PAIS primary key (ID_PAIS)
)
```



## Sistema de Información de Aeropuertos



```
go
/* ===== */
/* Table: ESTADO */
/* ===== */
create table ESTADO
(
  ID_ESTADO          INTEGER          not null,
  NOMBRE             varchar          null ,
  PAIS               tinyint         null ,
  constraint PK_ESTADO primary key (ID_ESTADO)
)
go

/* ===== */
/* Table: LOCALIDAD */
/* ===== */
create table LOCALIDAD
(
  ID_LOCALIDAD       integer          not null,
  NOMBRE             varchar          null ,
  SUPERFICIE        decimal          null ,
  POBLACION         decimal          null ,
  ESTADO            INTEGER          null ,
  constraint PK_LOCALIDAD primary key (ID_LOCALIDAD)
)
go

/* ===== */
/* Table: AEROPUERTO */
/* ===== */
create table AEROPUERTO
(
  SIGLAS_3          varchar          not null,
  SIGLAS_4          varchar          not null,
  NOMBRE            varchar(50)      null ,
  LOCALIDAD         integer          not null,
  CLASIFICACION     tinyint         null ,
  TIPO              tinyint         null ,
  FECHA_INAUGURACION datetime       null ,
  CAPAC_ANUAL_PASAJ integer          null ,
  CAPAC_ANUAL_OPER integer          null ,
  DISTANCIA_CIUDDAD integer          null ,
  TIEMPO_CIUDDAD   tinyint         null ,
  LATITUD           decimal          null ,
  LONGITUD          decimal          null ,
  ALTITUD           decimal          null ,
  WEB               varchar(100)     null ,
  NO_EDIF_TERMINAL tinyint         null ,
  SUPERFICIE_TOTAL integer          null ,
  NO_ESTACIONAMIENTOS tinyint      null ,
  SUPERFICIE_ESTACIONAMIENTO integer null ,
)
```



## Sistema de Información de Aeropuertos



```
NO_CAJONES_ESTACIONAMIENTO integer null ,
NOTAS varchar null ,
constraint PK_AEROPUERTO primary key (SIGLAS_3)
)
go

/* ===== */
/* Table: PISTA */
/* ===== */
create table PISTA
(
  ID_PISTA integer not null,
  DESIGNACION varchar(30) null ,
  LARGO float null ,
  ANCHO float null ,
  AVION_MAXIMO tinyint null ,
  PISO tinyint null ,
  AEROPUERTO varchar null ,
  constraint PK_PISTA primary key (ID_PISTA)
)
go

/* ===== */
/* Table: RESPONSABLE */
/* ===== */
create table RESPONSABLE
(
  ID_RESPONSABLE int not null,
  NOMBRE varchar null ,
  TIPO_RESPONSABLE tinyint null ,
  constraint PK_RESPONSABLE primary key (ID_RESPONSABLE)
)
go

/* ===== */
/* Table: AYUDA_ATERRIZAJE */
/* ===== */
create table AYUDA_ATERRIZAJE
(
  ID_AYUDA_VISUAL tinyint not null,
  DESCRIPCION varchar null ,
  TIPO_AYUDA tinyint null ,
  IMAGEN varchar(100) null ,
  constraint PK_AYUDA_ATERRIZAJE primary key (ID_AYUDA_VISUAL)
)
go
```



## Sistema de Información de Aeropuertos



```
/* ===== */
/* Table: SERVICIO */
/* ===== */
create table SERVICIO
(
  ID_SERVICIO          integer          not null,
  DESCRIPCION          varchar         null ,
  TIPO_SERVICIO        tinyint         null ,
  constraint PK_SERVICIO primary key (ID_SERVICIO)
)
go

/* ===== */
/* Table: RESPONSABLE_AEROPUERTO */
/* ===== */
create table RESPONSABLE_AEROPUERTO
(
  SIGLAS_3             varchar         not null,
  ID_RESPONSABLE       int             not null,
  ID_TIPO_ROL_RESPONSABLE tinyint     not null,
  constraint PK_RESPONSABLE_AEROPUERTO primary key (SIGLAS_3,
  ID_RESPONSABLE, ID_TIPO_ROL_RESPONSABLE)
)
go

/* ===== */
/* Table: AYUDA_PISTA */
/* ===== */
create table AYUDA_PISTA
(
  ID_PISTA             integer          null ,
  ID_AYUDA_VISUAL      tinyint         null ,
  CANTIDAD             tinyint         null
)
go

/* ===== */
/* Table: PLATAFORMA_AEROPUERTO */
/* ===== */
create table PLATAFORMA_AEROPUERTO
(
  SIGLAS_3             varchar         not null,
  ID_PLATAFORMA        tinyint         not null,
  SUPERFICIE           integer         null ,
  constraint PK_PLATAFORMA_AEROPUERTO primary key (SIGLAS_3,
  ID_PLATAFORMA)
)
go
```



## Sistema de Información de Aeropuertos



```
/* ===== */
/* Table: RODAJE */
/* ===== */
create table RODAJE
(
  ID_RODAJE          integer      not null,
  DESIGNACION       varchar      null ,
  LARGO              integer      null ,
  ANCHO              integer      null ,
  LUCES_BORDE       bit          null ,

  SENALAMIENTO      bit          null ,
  AEROPUERTO        varchar      null ,
  constraint PK_RODAJE primary key (ID_RODAJE)
)
go
```

```
/* ===== */
/* Table: SUMINISTRO_AEROPUERTO */
/* ===== */
create table SUMINISTRO_AEROPUERTO
(
  SIGLAS_3          varchar      not null,
  ID_SUMINISTRO     tinyint      not null,
  CAPACIDAD         integer      null ,
  constraint PK_SUMINISTRO_AEROPUERTO primary key (SIGLAS_3, ID_SUMINISTRO)
)
go
```

```
/* ===== */
/* Table: OBRA */
/* ===== */
create table OBRA
(
  ID_OBRA           integer      not null,
  DESCRIPCION       varchar      null ,
  FECHA             date         null ,
  AEROPUERTO        varchar      null ,
  TIPO              tinyint      null ,
  constraint PK_OBRA primary key (ID_OBRA)
)
go
```



## Sistema de Información de Aeropuertos



```
/* ===== */
/* Table: SERVICIO_AEROPUERTO */
/* ===== */
create table SERVICIO_AEROPUERTO
(
  ID_SERVICIO          integer      not null ,
  SIGLAS_3            varchar      not null ,
  ID_UBICACION        integer      not null ,
  CARACTERISTICAS     varchar      not null ,
  constraint PK_SERVICIO_AEROPUERTO primary key (ID_SERVICIO, SIGLAS_3,
  ID_UBICACION))
)
go

/* ===== */
/* Table: IMAGEN */
/* ===== */
create table IMAGEN
(
  ID_IMAGEN           integer      not null,
  DESCRIPCION         varchar      null ,
  IMAGEN              varchar      null ,
  AEROPUERTO          varchar      null ,
  constraint PK_IMAGEN primary key (ID_IMAGEN)
)
go

/* ===== */
/* Table: CREI_AEROPUERTO */
/* ===== */
create table CREI_AEROPUERTO
(
  ID_CREI             integer      not null,
  SIGLAS_3            varchar(3)   not null,
  CANTIDAD            integer      null ,
  constraint PK_CREI_AEROPUERTO primary key (ID_CREI, SIGLAS_3)
)
go

alter table PAIS
  add constraint FK_PAIS_REF_21_CONTINEN foreign key (CONTINENTE)
  references CONTINENTE (ID_CONTINENTE)
go

alter table PAIS
  add constraint FK_PAIS_REF_25_REGION foreign key (REGION)
  references REGION (ID_REGION)
go
```



## Sistema de Información de Aeropuertos

---



```
alter table ESTADO
  add constraint FK_ESTADO_REF_1242_PAIS foreign key (PAIS)
    references PAIS (ID_PAIS)
go

alter table LOCALIDAD
  add constraint FK_LOCALIDA_REF_1246_ESTADO foreign key (ESTADO)
    references ESTADO (ID_ESTADO)
go

alter table AEROPUERTO
  add constraint FK_AEROPUER_REF_50_LOCALIDA foreign key (LOCALIDAD)
    references LOCALIDAD (ID_LOCALIDAD)
go

alter table AEROPUERTO
  add constraint FK_AEROPUER_REF_54_CLASIFIC foreign key (CLASIFICACION)
    references CLASIFICACION (ID_CLASIFICACION)
go

alter table AEROPUERTO
  add constraint FK_AEROPUER_REF_58_TIPO foreign key (TIPO)
    references TIPO (ID_TIPO)
go

alter table PISTA
  add constraint FK_PISTA_REF_157_AEROPUER foreign key (AEROPUERTO)
    references AEROPUERTO (SIGLAS_3)
go

alter table PISTA
  add constraint FK_PISTA_REF_169_TIPO_PIS foreign key (PISO)
    references TIPO_PISO (ID_TIPO_PISO)
go

alter table PISTA
  add constraint FK_PISTA_REF_185_AVION foreign key (AVION_MAXIMO)
    references AVION (ID_AVION)
go

alter table RESPONSABLE
  add constraint FK_RESPONSA_REF_75_TIPO_RES foreign key (ID_RESPONSABLE)
    references TIPO_RESPONSABLE (ID_TIPO_RESPONSABLE)
go

alter table AYUDA_ATERRIZAJE
  add constraint FK_AYUDA_AT_REF_199_TIPO_AYU foreign key (TIPO_AYUDA)
    references TIPO_AYUDA (ID_TIPO_AYUDA)
go
```



## Sistema de Información de Aeropuertos



```
alter table SERVICIO
  add constraint FK_SERVICIO_REF_629_TIPO_SER foreign key (TIPO_SERVICIO)
  references TIPO_SERVICIO (ID_TIPO_SERVICIO)
go

alter table RESPONSABLE_AEROPUERTO
  add constraint FK_RESPONSA_REF_80_AEROPUER foreign key (SIGLAS_3)
  references AEROPUERTO (SIGLAS_3)
go

alter table RESPONSABLE_AEROPUERTO
  add constraint FK_RESPONSA_REF_84_RESPONSA foreign key (ID_RESPONSABLE)
  references RESPONSABLE (ID_RESPONSABLE)
go

alter table RESPONSABLE_AEROPUERTO
  add constraint FK_RESPONSA_REF_88_TIPO_ROL foreign key
  (ID_TIPO_ROL_RESPONSABLE)
  references TIPO_ROL_RESPONSABLE (ID_TIPO_ROL_RESPONSABLE)
go

alter table AYUDA_PISTA
  add constraint FK_AYUDA_PI_REF_203_PISTA foreign key (ID_PISTA)
  references PISTA (ID_PISTA)
go

alter table AYUDA_PISTA
  add constraint FK_AYUDA_PI_REF_207_AYUDA_AT foreign key (ID_AYUDA_VISUAL)
  references AYUDA_ATERRIZAJE (ID_AYUDA_VISUAL)
go

alter table PLATAFORMA_AEROPUERTO
  add constraint FK_PLATAFOR_REF_301_AEROPUER foreign key (SIGLAS_3)
  references AEROPUERTO (SIGLAS_3)
go

alter table PLATAFORMA_AEROPUERTO
  add constraint FK_PLATAFOR_REF_305_PLATAFOR foreign key (ID_PLATAFORMA)
  references PLATAFORMAS (ID_PLATAFORMA)
go

alter table RODAJE
  add constraint FK_RODAJE_REF_322_AEROPUER foreign key (AEROPUERTO)
  references AEROPUERTO (SIGLAS_3)
go

alter table SUMINISTRO_AEROPUERTO
  add constraint FK_SUMINIST_REF_404_AEROPUER foreign key (SIGLAS_3)
  references AEROPUERTO (SIGLAS_3)
go
```



```
alter table SUMINISTRO_AEROPUERTO
  add constraint FK_SUMINIST_REF_408_SUMINIST foreign key (ID_SUMINISTRO)
  references SUMINISTRO (ID_SUMINISTRO)
go

alter table OBRA
  add constraint FK_OBRA_REF_433_TIPO_OBR foreign key (TIPO)
  references TIPO_OBRA (ID_TIPO_OBRA)
go

alter table OBRA
  add constraint FK_OBRA_REF_442_AEROPUER foreign key (AEROPUERTO)
  references AEROPUERTO (SIGLAS_3)
go

alter table SERVICIO_AEROPUERTO
  add constraint FK_SERVICIO_REF_634_SERVICIO foreign key (ID_SERVICIO)
  references SERVICIO (ID_SERVICIO)
go

alter table SERVICIO_AEROPUERTO
  add constraint FK_SERVICIO_REF_816_AEROPUER foreign key (SIGLAS_3)
  references AEROPUERTO (SIGLAS_3)
go

alter table SERVICIO_AEROPUERTO
  add constraint FK_SERVICIO_REF_625_UBICACIO foreign key (ID_UBICACION)
  references UBICACION_SERVICIO (ID_UBICACION)
go

alter table IMAGEN
  add constraint FK_IMAGEN_REF_826_AEROPUER foreign key (IMAGEN)
  references AEROPUERTO (SIGLAS_3)
go

alter table CREI_AEROPUERTO
  add constraint FK_CREI_AER_REF_1277_CREI foreign key (ID_CREI)
  references CREI (ID_CREI)
go

alter table CREI_AEROPUERTO
  add constraint FK_CREI_AER_REF_1286_AEROPUER foreign key (SIGLAS_3)
  references AEROPUERTO (SIGLAS_3)
```