

**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**

FACULTAD DE CIENCIAS

**SISTEMA PARA LAS INSTALACIONES FÍSICAS DE LAS
INSTITUCIONES INCORPORADAS A LA UNAM**

T E S I S

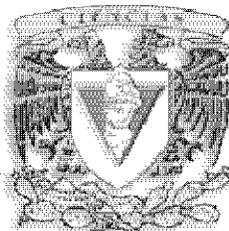
QUE PARA OBTENER EL TÍTULO DE:

LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A

Enrique Sánchez Trueba

**Tutora de Tesis: Mat. Olivia Suárez Martínez
Asesora de Tesis: M. en C. María Guadalupe Elena
Ibargüengoitia González**



**FACULTAD DE CIENCIAS
UNAM**

2006



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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



UNIVERSIDAD NACIONAL
AVENIDA DE
MEXICO

ACT. MAURICIO AGUILAR GONZÁLEZ
Jefe de la División de Estudios Profesionales de la
Facultad de Ciencias
Presente

Comunicamos a usted que hemos revisado el trabajo escrito:

“Sistema para las Instalaciones Físicas de las Instituciones Incorporadas a la UNAM”

realizado por **Enrique Sánchez Trueba**

con número de cuenta **09538040-5**, quien cubrió los créditos de la carrera de:

Lic. en Ciencias de la Computación

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis
Propietario
Asesora de Tesis
Propietario

Mat. Olivia Suárez Martínez

Propietario

M. en C. María Guadalupe Elena Ibarguengoitia González

Suplente

M. en C. Gustavo Arturo Márquez Flores

Suplente

M. en C. Javier García García

Dra. Hanna Oktaba

Consejo Departamental de Matemáticas



Dr. Francisco Hernández Quiroz

FACULTAD DE CIENCIAS
DEPARTAMENTO DE
MATEMÁTICAS

Agradecimiento

Agradezco ante todo a DIOS por darme la oportunidad de ver realizado este logro y por darme la oportunidad de venir a aprender aquí.

A ti Addity mi esposa. Gracias..., por todo tu apoyo y por inspirar todos los sueños que he logrado desde que te conocí y por todo lo que vendrá. Te Adoro Nena.

Gracias a María Antonieta Trueba Franco, mi Mamá, simplemente... por todo cuanto has formado en mi, por lo que soy y por quien eres. Tu carácter y determinación para llegar a los objetivos marca cada uno de mis aun pequeños pasos.

A Enrique Sánchez Pérez, mi Pa', respetablemente gracias. Por tu apoyo incondicional, por el ejemplo de perseverancia, de esfuerzo interminable y sobretodo de esa visión y madurez que sólo a los hombres grandes como tú caracteriza; sabe que hoy entiendo que las personas valen por lo que son y no por lo que dicen ser.

Karencilla...gracias por dejarme aprender de ti, por recordar y transmitirme la capacidad de soñar, imaginar, divertirme, decir burradas y en fin todo eso que los niños tienen y que a fuerza de inconciencia olvidamos los adultos.

Diana alias:(ma2), por nuestra niñez tan padre, y por todos los momentos que pasamos, eres para mi ejemplo de superación y entusiasmo.

A ti Alfredo, gracias amigo, creeme que decir esta palabra sólo y sólo en contadas ocasiones se logra pronunciar.

Gracias Abuelito Luis, por enseñarme que las cosas se hacen bien hechas y por darme en cada platica lingotes de oro en sabiduría.

Maga(+) y Conchita(+)...GRACIAS MIL!.

Gracias infinitamente a la UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO, cuna y abrigo del conocimiento. A la Facultad de Ciencias, por formar no sólo nuestras mentes sino también nuestro carácter, gracias por todos los recuerdos, los cuates(Radamantis, Mauro, Rene), las clases, los días de dos o más noches, etc.

Mi más sincero agradecimiento a todos mis maestros en especial a Javier García García(por lo realista de sus clases, por las espantadas en el salón y por los exámenes kilométricos), José Galavis(por sus anécdotas, consejos y por su excelencia), Mónica Leñero(por todo el apoyo durante la carrera), Elisa Viso, Lupita Ibarguengoitia(

por todo el entusiasmo mostrado en este trabajo), Hanna Oktaba(por hacer de la Ingeniería de software un arte), Pedro Miramontes y porsupuesto a mis ayudantes Karla Ramírez y Julio Barreiro.

Mi reconocimiento, aprecio y gratitud a la Matemática Olivia Suárez Martínez, por todo el apoyo brindado en esta Tesis, gracias por darme la oportunidad de trabajar con usted. Al Matemático Héctor Ayuso Naranjo, gracias por su guía, paciencia y por toda la confianza depositada en mi.

Contenido

1. Introducción	11
2. La DGIRE y el Sistema Incorporado a la UNAM	13
2.1. Dirección General de Incorporación y Revalidación de Estudios	13
2.1.1. Sistema Incorporado de la UNAM	14
3. Tecnología para sistemas Web	21
3.1. Proceso de Desarrollo de Sistemas(PDS)	21
3.1.1. Captura de Requerimientos	22
3.1.2. Diseño	23
3.1.3. Arquitectura	24
3.1.4. Diseño de las tablas de Base de Datos	25
3.1.5. Implementación	29
3.1.6. Diagramas de Actividades UML	29
3.2. Entorno General del Software	31
3.3. Herramientas de Software	32
3.3.1. SMBD SYBASE	36
3.4. Entorno General del Hardware	40
3.5. Telecomunicaciones	42
3.5.1. Condiciones actuales de telecomunicaciones en la DGIRE	44
3.6. Entorno de Seguridad del sistema	46
3.6.1. Servidores Web	48
4. Desarrollo del sistema “Planta Física”	49

4.1. ¿Qué es el sistema “Planta Física”?	49
4.2. Requerimientos	49
4.2.1. Objetivo del Sistema	49
4.2.2. Glosario de términos	50
4.3. Usuarios en Planta Física	51
4.3.1. Requerimientos Funcionales	52
4.3.2. Requerimientos No Funcionales	55
4.4. Interfaz Gráfica o Front-End del Sistema	57
4.5. Arquitectura del sistema	63
4.6. Almacenamiento de Información	65
4.6.1. Normalización de las tablas	69
4.6.2. Acceso a Sybase	71
4.6.3. Factores para una migración	77
4.7. Implementación	81
4.8. Validaciones	81
4.9. Funcionamiento de los módulos del Sistema	83
5. Conclusiones	93
Bibliografía	97

Lista de Figuras

1.	Votos Aprobatorios	2
2.1.	Organigrama D.G.I.R.E.	15
2.2.	Presencia del Sistema Incorporado en la República Mexicana	16
2.3.	Sistema Incorporado en el Distrito Federal y Edo. de México.	16
3.1.	Estructura de directorios en UNIX	34
3.2.	Estructura Multicapa de UNIX	35
3.3.	Rack del 1er. piso de la DGIRE	44
3.4.	Rack de la planta baja de la DGIRE	45
4.1.	Entrada a Planta Física	57
4.2.	Menú de Planta Física	58
4.3.	Aulas de uso Común	59
4.4.	Dirección Técnica.	61
4.5.	Servicios Escolares.	61
4.6.	Aulas de Actividades Estéticas.	62
4.7.	Laboratorios de Licenciatura.	62
4.8.	Laboratorios de Bachillerato.	63
4.9.	Diagrama de tres niveles de la aplicación	64
4.10.	Diagrama E-R: Tablas con herencia de Planta Física.	67
4.11.	Diagrama E-R: Tablas relacionadas para almacenamiento.	68
4.12.	Diagrama E-R: Tablas independientes.	68
4.13.	Entrada al Sistema.	83
4.14.	Funcionalidad Global.	85

4.15. Funcionamiento del módulo Bibliotecas	87
4.16. Funcionamiento del módulo de Áreas Deportivas.	88
4.17. Funcionamiento del módulo de Áreas de dispersión	90
4.18. Funcionamiento del módulo de Extintores	91
4.19. Funcionamiento del módulo de Notas Adicionales	92

Lista de Tablas

2.1. Presencia del SI en el D.F. y Edo. de Méx.	17
2.2. Accesos al Sitio Web de la DGIRE.	19
3.1. Servidor Azul y Unamsi0	40

Capítulo 1

Introducción

En los años ochenta se inicia el **INTERNET**(Interconnected Networks), y a sólo poco más de dos décadas la palabra Internet se ha hecho común en la mayor parte de los habitantes de nuestro planeta, divertido para jóvenes, importante para los negocios e indispensable para la educación y la investigación.

Por ahora, cualquier información o servicio resulta eficiente si cumple con el objetivo de llegar a los individuos que lo necesitan en cualquier lugar y en cualquier momento; esto puede ser posible gracias a las nuevas tecnologías de comunicación e información, es decir con el Internet.

En la Dirección General de Incorporación y Revalidación de Estudios (DGIRE), Dependencia de la Universidad Nacional Autónoma de México (UNAM), el cómputo se viene utilizando, desde hace más de treinta años, en sistemas para la administración escolar. Dichos sistemas han evolucionado con el cómputo; importante fue el uso de los lectores ópticos y la impresión de formularios, para intercambio de información, con el paso del tiempo y las recientes tecnologías disponibles y las telecomunicaciones, también los modelos de operación han cambiado.

La DGIRE se inicia en el Internet a principios de los años noventa con una página Web cuyo objetivo fue mostrar información de diversa naturaleza, principalmente relacionada con sus funciones. Paralelamente al uso del Internet se consolidaron los sistemas sobre Administración Escolar y con la llegada de los años 2000 se inician las aplicaciones en línea.

En el presente trabajo se describirá el subsistema denominado Instalaciones

Físicas, “Planta Física”, que forma parte del Sistema para la Administración Escolar del Sistema Incorporado de la UNAM, sistema UNAMSI.

En el segundo capítulo se define qué es la DGIRE, su entorno, funciones, organigrama y los servicios que ofrece al Sistema Incorporado de la UNAM, así como también se brindan algunas estadísticas de dichos servicios .

En el tercer capítulo definiremos el proceso de desarrollo de software utilizado para la creación del sistema “Planta Física”, así también las herramientas de software, hardware, telecomunicaciones y algunos aspectos de seguridad interna que respaldan a dicho sistema.

En el cuarto capítulo se abordará lo referente a la creación sistema de “Planta Física”, incluyendo su diseño, desarrollo e implementación, tablas de la base de datos, submódulos que lo integran, así como herramientas de uso específico del sistema.

Finalmente, en la conclusión, realizaremos un resumen de las bondades más importantes con las que cuenta el sistema, una propuesta para agregar algunas herramientas a éste, y la importancia del sistema a nivel desarrollo en la DGIRE.

Capítulo 2

La DGIRE y el Sistema Incorporado a la UNAM

2.1. Dirección General de Incorporación y Revalidación de Estudios

La Universidad Nacional Autónoma de México es una institución pública, un organismo descentralizado del estado dotada de plena capacidad jurídica y que tiene por fines impartir educación superior para formar profesionistas, investigadores, profesores universitarios y técnicos útiles a la sociedad, organizar y realizar investigación, principalmente acerca de las condiciones y problemas nacionales, y extender con la mayor amplitud posible los beneficios de la cultura.

La Dirección General de Incorporación y Revalidación de Estudios (DGIRE) es la dependencia universitaria a través de la cual la UNAM brinda los servicios de incorporación y revalidación de estudios, los cuales se definen a continuación.

- Se entiende por Incorporación de estudios, al reconocimiento académico que hace la UNAM, a través de la Comisión de Incorporación y Revalidación de Estudios (CIRE) del H. Consejo Universitario, de los planes y programas que se imparten en instituciones particulares, en razón de la identidad que existe con los que ella misma ofrece y que quedan bajo su supervisión académico-administrativa.

- Se entiende por plan de estudios, el conjunto de asignaturas y requisitos académicos que aseguran la preparación integral del alumnado.
- Se entiende por revalidación y equivalencia de estudios, la validez académica que la UNAM otorga a los estudios realizados en instituciones educativas que no formen parte de ella.

Las instituciones y su comunidad educativa deberán cumplir con los requisitos académico-administrativos señalados en el Manual de Disposiciones y Procedimientos Relativos a las Instituciones con Estudios Incorporados a la UNAM. En especial, la institución que la CIRE otorgue incorporación de estudios deberá contar con los recursos humanos y materiales necesarios para operar eficientemente los planes de estudio incorporados.

Misión de la DGIRE

Derivado de sus funciones, la DGIRE es la institución encargada de apoyar a la UNAM a mantener su carácter rector en materia académica, brindando a la sociedad los servicios de incorporación y revalidación.

Para cumplir con ello, la DGIRE se organiza a través de cuatro Subdirecciones, una Secretaría, Unidad Administrativa y la Unidad de Control de Gestión, según se muestra en el siguiente organigrama:

Es a través de la Subdirección de Incorporación que se realiza la verificación de los requisitos académico-administrativos para impartir planes y programas de estudio, en particular, el que se refiere a la planta física, es decir, al requisito de las instalaciones de cada una de las instituciones incorporadas, motivo que nos ocupa para esta tesis.

Para una mejor comprensión definiremos y daremos un panorama general de lo que es el sistema Incorporado de la UNAM.

2.1.1. Sistema Incorporado de la UNAM

Las autoridades, los profesores y los alumnos de las instituciones autorizadas para impartir planes de estudio de la Universidad Nacional, constituyen el Sistema Incorporado de la UNAM (SI).

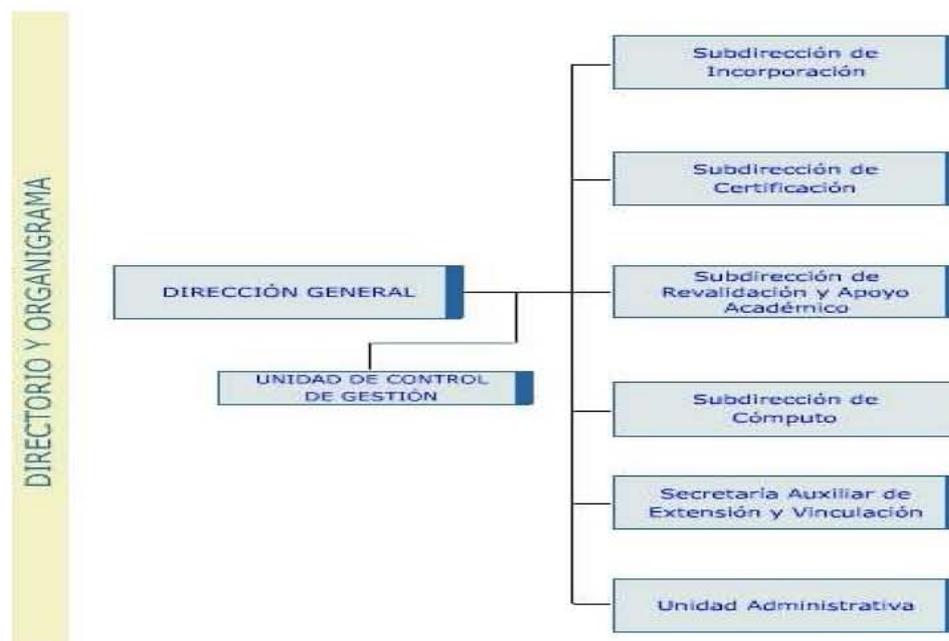


Figura 2.1: Organigrama D.G.I.R.E.

Evolución y Conformación del Sistema Incorporado

Lo que hoy conocemos como Sistema Incorporado vivió sus primeras etapas en los años treinta, con la incorporación de estudios de, entre otras instituciones educativas, la Universidad Motolinia, a nivel licenciatura, la Escuela de Enfermería de la Cruz Roja Mexicana en el nivel técnico y, en el nivel bachillerato, los Colegios Alemán y Francés, este último, actualmente, Centro Universitario México (CUM).

Hacia 1960, al Sistema Incorporado lo conformaban ya 61 instituciones y en la década de los 70 se incrementaron a 135. Fue tal la expansión del SI, que en el año de 1967 la UNAM decide crear una Dirección General específica para su atención.

El Sistema Incorporado de la UNAM responde a una demanda importante de la sociedad que busca opciones de educación para realizar estudios de nivel bachillerato y licenciatura en instituciones que garanticen la formación de los alumnos con excelencia académica y con reconocimiento oficial en sus estudios.

El Sistema Incorporado se compone actualmente por 317 instituciones que en 350 sedes imparten 512 planes de estudio a nivel bachillerato y licenciatura, sus alumnos suman aproximadamente los 95 mil, los cuales son atendidos por casi 11 mil profesores, distribuidos en diferentes puntos del país. En la República Mexicana



Figura 2.2: Presencia del Sistema Incorporado en la República Mexicana

se aprecia la distribución del SI en la Figura 2.2, y en el D.F. y Estado de México en la Figura 2.3 y la Tabla 2.1.

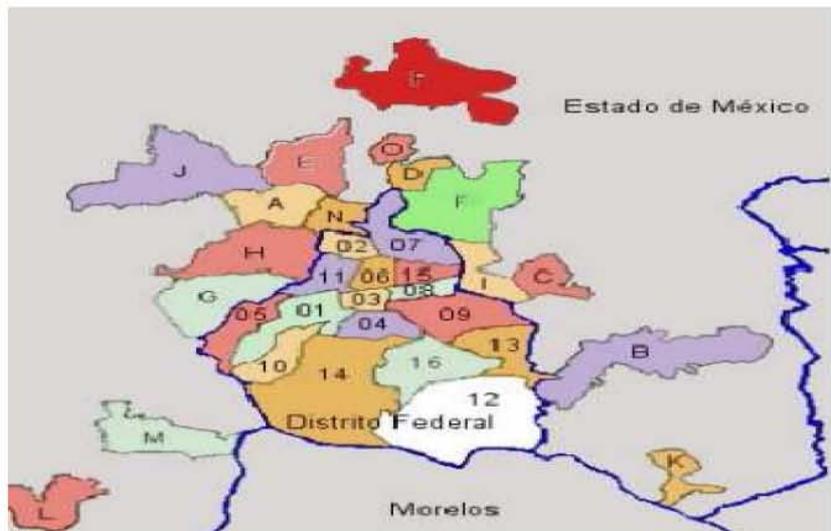


Figura 2.3: Sistema Incorporado en el Distrito Federal y Edo. de México.

La administración escolar del Sistema Incorporado ha experimentado diferentes circunstancias, principalmente las que han hecho posible plantearse la posibilidad de contar con un sistema que permita la captura, resguardo y consulta de la información referente a la planta arquitectónica de las instalaciones incorporadas. Desde

Distrito Federal		Estado de México	
01	Del. Alvaro Obregón	A	Atizapan de Zaragoza
02	Del. Azcapotzalco	B	Chalco
03	Del. Benito Juárez	C	Chimalhuacán
04	Del. Coyoacán	D	Coacalco
05	Del. Cuajimalpa	E	Cuautitlán Izcalli
06	Del. Cuauhtémoc	F	Ecatepec de Morelos
07	Del. Gustavo A. Madero	G	Huixquilucan
08	Del. Iztacalco	H	Naucalpan de Juárez
09	Del. Iztapalapa	I	Nezahualcoyotl
10	Del. Magdalena Contreras	J	Nicolás Romero
11	Del. Miguel Hidalgo	K	Ozumba
12	Del. Milpa Alta	L	Santiago Tianguistenco
13	Del. Tláhuac	M	Tenancingo
14	Del. Tlalpan	N	Tlanepantla de Baz
15	Del. Venustiano Carranza	O	Tultepec
16	Del. Xochimilco	P	Zumpango

Tabla 2.1: Presencia del SI en el D.F. y Edo. de Méx.

el punto de vista de evolutivo, la Administración Escolar del UNAMSI, se puede agrupar en las siguientes etapas:

- En la primera etapa se definió un Sistema Central operado desde las instalaciones de la DGIRE, donde se generaba papel con diversas opciones y características, con este modelo de sistema los datos eran capturados tanto en las instituciones incorporadas como en la DGIRE. Lo que caracterizó a esta etapa fueron los grandes volúmenes de papel y duplicidad de captura en la información.
- En la segunda etapa se definió un modelo de sistema distribuido con bases de datos replicadas. Esta distribución se definió para cada uno de los niveles de estudio de cada una de las instituciones; es decir, para cada institución incorporada y nivel de estudios se creó una réplica de la base de datos y se definió que la operación se realizara desde las instituciones. Esto permi-

tió eliminar la doble captura de la información.

- Paralelamente a este sistema se inició la tercera etapa, es decir, el uso de la Web con información que fue utilizada principalmente para consulta de las instituciones, como circulares, minutas, planes de estudio, eventos, avisos, etc. Además experimentó con el primer servicio de consulta hacia las bases de datos, es decir, se puso en línea la consulta de las historias académicas desde la Web, con datos conocidos por los alumnos.

Para el año 2001 surge el primer sitio Web con el concepto definido como Servicios en Línea, a través del cual las instituciones pueden consultar información con una clave de acceso previamente asignada por la DGIRE. El uso y la confianza de estos servicios ha permitido ampliar las aplicaciones, en especial, los que como este proyecto, reúnen información que permanece en una base de datos central y puede ser consultada y actualizada por los propios usuarios.

En los últimos años los accesos al sitio Web han aumentado de manera importante. Algunos datos estadísticos se muestran a continuación:

Año	Total de accesos
2001	14,000
2002	80,000
2003	242,372
2004	254,325
2005	412,631

Tabla 2.2: Accesos al Sitio Web de la DGIRE.

Con los nuevos sistemas se tienen accesos diarios muy por arriba del número de alumnos que están vigentes en el Sistema Incorporado¹, lo cual nos da una referencia de cuan importante ha sido la creación de estos.

Para soportar la carga informática que emana de brindar estos servicios, ha sido necesaria la adquisición de nuevos equipos, reestructuración de telecomunicaciones para éstos, rediseño de los esquemas de datos actuales, así como enfatización de la seguridad y en fin, toda aquella tarea que coadyuve a un servicio óptimo y eficaz.

¹Se tienen registrados actualmente 95,000 alumnos del SI, y 10,000 profesores activos

Capítulo 3

Tecnología para sistemas Web

Este capítulo está dedicado a especificar de manera general algunos aspectos fundamentales en cuanto a hardware, software y telecomunicaciones, necesarios para la implementación de los sistemas en WEB. De esta manera exponer la tecnología y el proceso de desarrollo con los que la DGIRE ha realizado la estructuración de los sistemas en esta dependencia Universitaria, “**Sistema de Administración Escolar en Línea**”.

Si bien es importante que en cualquier tipo de desarrollo de software se realice el análisis de recursos de hardware, herramientas de desarrollo y necesidades del usuario, en este trabajo se hace énfasis por tratarse de un desarrollo de software en Internet.

La razón de ello responde a aspectos que emanan del número potencial de usuarios, disponibilidad casi instantánea de la información y bien, de las consecuencias técnicas de la realización de las diferentes tareas a las que la WEB está destinado a realizar y que conjuntamente deben llegar a un equilibrio, logrando así un rendimiento óptimo de nuestros recursos en general.

3.1. Proceso de Desarrollo de Sistemas(PDS)

El proceso de desarrollo de software puede definirse como un conjunto de herramientas, métodos y prácticas que se emplean para producir software. Como cualquier otra organización, las dedicadas al desarrollo de software mantienen entre

sus principales fines la producción de software de acuerdo con la planificación inicial realizada, además de una constante mejora con el fin de lograr los tres objetivos últimos de cualquier proceso de producción: alta calidad y bajo costo, en el mínimo tiempo.

Un proceso de software tiene fases, las fases actividades que utilizan y generan productos. Las actividades son responsabilidades de roles.

3.1.1. Captura de Requerimientos

El objetivo de esta fase es iniciar la construcción propiamente dicha del producto y ser la base para la construcción. Por esta razón es importante capturar y especificar los requerimientos para tener una descripción clara y no ambigua de lo que será el producto, qué realmente se necesita y cuáles son las necesidades claves.

Un requerimiento o necesidad, es lo que el cliente o quien usará el software creen necesitar que haga el software para resolver un cierto problema. Muchas veces el cliente y el usuario son la misma persona. A estas personas se les llama en inglés stakeholders que se traduce por interesados o involucrados.

Una característica de los requerimientos es que cambian constantemente por muchas razones, cambian las necesidades del cliente, establecer los requerimientos es un proceso de negociación constante entre cliente y desarrolladores, ambos aprenden en el transcurso del desarrollo, etc.

Por otro lado, es una fase fundamental en la construcción de un software con calidad. Se entiende por calidad en un producto de software, que debe cumplir con las necesidades del usuario.

La captura de requerimientos termina cuando se obtiene la Especificación de los requerimientos, que es un documento que resume lo que cliente necesita y que servirá de base a los desarrolladores para analizar esos requerimientos, validarlos y administrarlos.

Hay dos tipos de necesidades o requerimientos: los funcionales y los no funcionales. Los funcionales son:

- Las entradas y salidas de datos, los cálculos.
-

Entre los requerimientos no funcionales están:

- Las necesidades de la interfaz externa: tipo de usuario, hardware, software, comunicaciones, facilidad de uso por sus usuarios.
- Los atributos del software: eficiencia, disponibilidad, seguridad, conversión, portabilidad, de mantenimiento.
- Restricciones del diseño: formatos, de archivos, lenguajes, estándares, compatibilidad.
- Otros: base de datos, instalación, etc.

El proceso para la captura de los requerimientos es:

1. Entender las necesidades del sistema.
2. Identificar a los interesados en el sistema y solicitarles sus requerimientos.
3. Definir los requerimientos funcionales y no funcionales
4. Hacer prototipo para que el cliente entienda si esos son sus requerimientos.
5. Hacer la Especificación de requerimientos.

3.1.2. Diseño

El objetivo de esta fase es definir la arquitectura. Definir y describir las partes de las cuales se va a componer el sistema y mostrar sus relaciones.

El diseño es la parte de la construcción de software en que se cambia el enfoque de las necesidades del cliente, a las necesidades del desarrollador y que servirá de base para la construcción del software propiamente dicha.

Un diseño completo debe describir las partes o componentes principales de que se compondrá, cómo interactúan, y cómo se ponen juntos para formar el producto terminado. En él, se desglosan las partes principales partiendo del conocimiento de los requerimientos y se define la arquitectura.

3.1.3. Arquitectura

La arquitectura de software es el diseño del más alto nivel. Consiste en definir cuáles serán las partes que formarán el software. Estas partes o componentes definen la estrategia para dividir el trabajo entre los miembros del equipo y después de la implementación para ensamblar lo que cada quien hizo.

Definir la arquitectura adecuada a cada producto de software depende de los atributos o requerimientos no funcionales especificados. Algunos de esos atributos pueden ser:

- Extensión. La posibilidad de agregar nuevas características.
- Cambio. Facilitar los cambios en los requerimientos.
- Sencillez. Que sea fácil de comprender y de implementar.
- Eficiencia. Que corra rápido en la compilación y/o ejecución. Que el código sea pequeño.

La arquitectura de 3 capas es una forma de dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realiza se efectúe con la mayor eficiencia, y permita simplificar las actualizaciones y mantenimiento del sistema.

La ventaja principal de esta arquitectura, es que el desarrollo se puede llevar a cabo en varios niveles y en caso de algún cambio sólo se ataca al nivel requerido sin tener que revisar entre el código mezclado.

Además permite distribuir el trabajo de creación de una aplicación por capas, de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, simplemente es necesario conocer el conjunto de herramientas que existe entre capas.

En el diseño de sistemas informáticos actual se suele usar las arquitecturas por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten).

Las tres capas que conforman esta arquitectura son:

1. **Capa de presentación:** Es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio.
2. **Capa de negocio o lógica:** Es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar, al gestor de base de datos, para almacenar o recuperar datos de él.
3. **Capa de almacenamiento o de datos:** Es donde residen los datos. Está formada por uno o más Sistemas Manejadores de Bases de Datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Todas estas capas pueden residir en una única computadora, si bien lo más usual es que haya una multitud de computadoras donde residan la capa de presentación (son los clientes de la arquitectura cliente/servidor). Las capas de negocio y de datos pueden residir en la misma computadora, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más computadoras.

En esta arquitectura, los terminos Capas o Niveles no significan lo mismo ni son similares. El término capa hace referencia a la forma como una solución es segmentada desde el punto de vista lógico: Presentación/ Lógica de Negocio/ Datos. El término nivel, corresponde a la forma como las capas lógicas, se encuentran distribuidas físicamente.

3.1.4. Diseño de las tablas de Base de Datos

Para el diseño de las tablas de la base de datos, es esencial la identificación de las funciones e interfaces; se requiere la especificación del flujo, estructura y asociatividad de la información y se debe contar con un documento que contenga la definición de las tablas.

Existen diversas herramientas para ilustrar de manera gráfica la estructura y relación de las tablas en las bases de datos; Los diagramas Entidad Relación ofrecen al desarrollador una amplia gama de elementos para dicho fin.

Tal como su nombre lo indica, estos diagramas tienen como elementos básicos las entidades, representadas por un rectángulo en cuyo interior se encuentra el nombre de la tabla, y líneas que unen dichas entidades, las cuales definen relaciones. Las entidades pueden tener múltiples atributos o bien campos, estos últimos representados por óvalos.

Existen diversos tipos de atributos. Dicha tipificación se expresa por medio de las características del óvalo dibujado. Dentro de cada óvalo se encuentra el nombre del atributo representado. Este nombre puede tener un subrayado de ciertas características, las cuales determinarán qué papel juega el atributo a nivel de relación entre tablas de la BD.

Normalización de las tablas

Pueden existir múltiples diseños de las tablas de una base de datos, sin embargo, es importante que sepamos, que el nuestro es un buen diseño. Existen muchas razones para pensar en ello, las cuales están relacionadas con el poder dar garantía que en cualquier momento los datos almacenados en las tablas, y la gestión con ellas, guarden relación con respecto a la semántica de uso que le daremos. Esto es que se conserve la integridad en nuestros datos.

Para ello se ha desarrollado teoría que nos brinda la capacidad de evaluar si nuestro diseño es correcto.

Uno de los retos en el diseño de la base de datos es el de obtener una estructura estable y lógica de tal manera que:

1. El sistema de base de datos no sufra de anomalías de almacenamiento.
2. El modelo lógico pueda modificarse fácilmente para admitir nuevos requerimientos.

Una base de datos implantada sobre un modelo bien diseñado tiene mayor esperanza de vida aun en un ambiente dinámico, que una base de datos con un diseño

pobre. Una base de datos bien diseñada tendrá un buen desempeño aunque aumente su tamaño, y será lo suficientemente flexible para incorporar nuevos requerimientos o características adicionales.

Existen diversos riesgos en el diseño de las bases de datos relacionales que afecten la funcionalidad de la misma, los riesgos generalmente son la redundancia de información y la inconsistencia de datos.

La normalización es el proceso de simplificar la relación entre los campos de un registro. Por medio de la normalización un conjunto de datos en un registro se reemplaza por varios registros que son más simples y predecibles (descomposición de tablas sin pérdida) y, por lo tanto, más manejables.

En términos más sencillos la normalización trata de simplificar el diseño de una base de datos, esto a través de la búsqueda de la mejor estructuración que pueda utilizarse con las entidades involucradas en ella.

La normalización se lleva a cabo por cuatro razones:

- Estructurar los datos de forma que se puedan representar las relaciones pertinentes entre ellos.
- Permitir la recuperación sencilla de los datos en respuesta a las solicitudes de consultas y reportes.
- Simplificar el mantenimiento de los datos actualizándolos, insertándolos y borrándolos.
- Reducir la necesidad de reestructurar o reorganizar los datos cuando surjan nuevas aplicaciones.

Para realizar la normalización de las tablas se utilizan métodos llamados **Formas Normales**.

Las formas normales son las técnicas para prevenir las anomalías en las tablas. Dependiendo de su estructura, una tabla puede estar en primera forma normal, segunda forma normal o en cualquier otra.

Según el orden de las formas, estas van requiriendo mayores restricciones en cuanto a la forma y organización de los datos de una tabla y su relación con otras.

Primera forma normal

Una relación R se encuentra en 1era.FN sí y sólo sí cada columna contiene valores atómicos¹.

Abreviada como 1FN, se considera que una relación se encuentra en la primera forma normal cuando cumple lo siguiente:

1. Las celdas de las tablas poseen valores simples y no se permiten grupos ni arreglos repetidos como valores, es decir, contienen un solo valor por cada celda.
2. Todos los ingresos en cualquier columna (atributo) deben ser del mismo tipo.
3. Cada columna debe tener un nombre único.
4. Dos filas o renglones de una misma tabla no deben ser idénticas, aunque el orden de las filas no es importante.

Segunda forma normal

Para definir formalmente la segunda forma normal requerimos saber qué es una dependencia funcional:

Definición: Sean X e Y subconjuntos de atributos de una relación. Diremos que Y tiene una dependencia funcional de X , o que X determina a Y , si cada valor de X tiene asociado siempre un único valor de Y .

Una relación se encuentra en segunda forma normal, cuando cumple con las reglas de la primera forma normal y todos sus atributos que no son claves (llaves) dependen por completo de la clave. De acuerdo con esta definición, cada tabla que tiene un atributo único como clave, esta en segunda forma normal.

Tercera forma normal

Para definir formalmente la 3FN necesitamos definir dependencia transitiva: En una afinidad (tabla bidimensional) que tiene por lo menos 3 atributos (A,B,C) en donde A determina a B, B determina a C, pero no determina a A.

Una relación R o tabla está en 3FN sí y solo sí está en 2FN y todos sus atributos no primos² dependen no transitivamente de la llave primaria.

¹Un valor ó dominio es atómico si los elementos del dominio son indivisibles

²Que no pertenecen a la llave

Consiste en eliminar la dependencia transitiva que queda en una segunda forma normal, en pocas palabras una relación está en tercera forma normal si está en segunda forma normal y no existen dependencias transitivas entre los atributos. Nos referimos a dependencias transitivas cuando existe más de una forma de llegar a referencias a un atributo de una relación.

3.1.5. Implementación

En esta fase se realiza la construcción del código del sistema, dando seguimiento a los requerimientos y diseño de las fases anteriores.

3.1.6. Diagramas de Actividades UML

Los Diagrama de Actividad son una especialización de los Diagramas de Estado UML³, organizados respecto de las acciones y usados para especificar:

- Un método
- Un caso de uso
- Un proceso de negocio (Workflow)

Un estado de actividad representa una actividad: un paso en el flujo de trabajo o la ejecución de una operación. Un diagrama de actividades describe grupos secuenciales y concurrentes de actividades. Los diagramas de actividades se muestran en diagramas de actividades. Las actividades se enlazan por transiciones automáticas. Cuando una actividad termina se desencadena el paso a la siguiente actividad. Un diagrama de actividades es provechoso para entender el comportamiento de alto nivel de la ejecución de un sistema, sin profundizar en los detalles internos de los mensajes. Los parámetros de entrada y salida de una acción se pueden mostrar usando las relaciones de flujo que conectan la acción y un estado de flujo de objeto.

³Muestra el conjunto de estados por los cuales pasa un objeto durante su vida en una aplicación, junto con los cambios que permiten pasar de un estado a otro. Los Diagramas de Estado representan autómatas de estados finitos.

Un diagrama de actividades contiene estados de actividad que representa la ejecución de una secuencia en un procedimiento, o el funcionamiento de una actividad en un flujo de trabajo. En vez de esperar un evento, como en un estado de espera normal, un estado de actividad espera la terminación de su cómputo. Cuando la actividad termina, entonces la ejecución procede al siguiente estado de actividad dentro del diagrama. Una transición de terminación es activada en un diagrama de actividades cuando se completa la actividad precedente. Los estados de actividad no tienen transiciones con eventos explícitos, pero pueden ser abortados por transiciones en estados que los incluyen.

Un diagrama de actividades puede contener también estados de acción, que son similares a los de actividad pero son atómicos y no permiten transiciones mientras están activos. Los estados de acción se deben utilizar para las operaciones cortas de mantenimiento. Un diagrama de actividades puede contener bifurcaciones, así como divisiones de control en hilos concurrentes. Los hilos concurrentes representan actividades que se pueden realizar concurrentemente por los diversos objetos o personas. La concurrencia se representa a partir de la agregación, en la cual cada objeto tiene su propio hilo. Las actividades concurrentes se pueden realizar simultáneamente o en cualquier orden. Un diagrama de actividades es como un organigrama tradicional, excepto que permite el control de concurrencia además del control secuencial.

Notación

Una actividad se representa como una caja con los extremos redondeados que contiene una descripción de actividad. Las transacciones simples de terminación se muestran como flechas. Las se representan como diamantes con múltiples flechas de salida etiquetadas. Una división o una unión de control se representa con múltiples flechas que entran o salen de la barra gruesa de sincronización.

A menudo es útil organizar las actividades en un modelo según su responsabilidad. Esta clase de asignación puede mostrarse organizando las actividades en regiones distintas separadas por líneas en el diagrama. Debido a su aspecto, esto es conocido como carriles.

3.2. Entorno General del Software

Trabajar con sistemas web predispone a una afluencia de usuarios importante, por lo que mantener su estabilidad y crecimiento requiere de un análisis bien fundamentado para así cambiar de sistemas tradicionales que funcionan en una red interna a sistemas abiertos al WEB.

Desde la manera de identificar usuarios hasta la manera de presentar la información hay inmersos cambios en la forma de como se manejan las situaciones, no es difícil exponer lo anterior con un ejemplo, la seguridad en cuanto a validaciones en el ingreso de un login o un password por parte del usuario, en el que se puede analizar que en un sistema convencional en red las validaciones se realizan al vuelo y en el mismo lenguaje en donde se programaron. Sin embargo, en un sistema en WEB hay dos maneras posibles de hacer esta misma tarea. La primera consiste en validar la información ingresada, enviándola por medio de una transacción al servidor WEB en la que el código, en ejecución, del programa deberá contener el motor de validación incluido (funciones de validación centralizadas) lo cual constituye tiempo reservado de ejecución tanto en el servidor mismo como en el servidor WEB. La segunda forma de realizar la validación es vía javascript, VBscript, o alguna otra opción de lenguaje que dependa del navegador, validación que en un primer plano resulta suficiente, aunque inseguro, pero que con acciones de más bajo nivel, como lo es el chequeo directo en una base de datos, ya no lo es; por tanto, es necesaria también la tarea de validación propuesta en primera opción. De esta manera al integrar estas dos opciones, reducimos en cierta medida, las posibles transacciones erróneas al servidor WEB y con ello logramos reducir la carga de uso de CPU⁴ del servidor.

Con lo anterior ejemplificamos aspectos que en ocasiones parecen sencillos de resolver en sistemas diseñados para una red interna, no lo son en el caso de sistemas en WEB.

Haciendo referencia al anterior capítulo, la DGIRE es una de las dependencias con más alto índice de usuarios en la UNAM. Por tal motivo, resulta no trivial la planeación de los sistemas, por lo que el compromiso de implementar sistemas en WEB significa: Garantizar que nuestros usuarios puedan, por medio de una computadora con servicio de Internet, acceder de manera transparente a cada uno de los servicios y recursos que se liberan, sin importar en que lugar se encuentren. Te-

⁴Central Process Unit o procesador

niendo en consideración lo anterior, es necesario realizar una elección adecuada de herramientas eficientes de software para dichos fines, tales como: la plataforma de trabajo, Sistema Operativo, servidor WEB, lenguajes de programación, herramientas en el WEB.

Cualquier herramienta a utilizar, y teniendo una visión de crecimiento potencial, debe ser una herramienta compatible con los elementos de software y hardware con los que habrá de relacionarse; más aún, debe tratarse de herramientas ampliamente documentadas, robustas y que conlleven el menor costo posible en cuanto al hardware requerido para su funcionamiento y aún en cuanto al capital invertido.

¿Capital invertido? Sí, debe tomarse en cuenta que utilizar un software que a la larga tenga inmersa una dependencia en cuanto a costos altos resulta inconveniente, si referimos el hecho de que el desarrollo se lleva a cabo en una institución pública, la UNAM, este aspecto retoma importancia.

Como puede verse la tarea es ardua. Aunado a todo lo anterior y tomando el papel de usuarios, algo importante a lograr es poder ofrecer claridad en los sistemas en cuanto a su uso. En sistemas en WEB con un alto índice de concurrencia, prácticamente todo está ligado al rendimiento del equipo, es primordial poder reducir en lo más posible la cantidad de peticiones erróneas, y disminuir el número de tareas realizadas por nuestro servidor en un momento dado, coadyuvando a un mejor desempeño de éste. Un sistema amigable, de uso claro y eficiente, es un sistema funcional y atractivo, por lo que en todo tipo de desarrollo de software, a nuestro particular punto de vista, se debe tener personal destinado a la evaluación de éste, en el cual se encuentren involucrados potenciales usuarios de los sistemas, próximos a liberarse.

3.3. Herramientas de Software

Dentro de la generalidad de la palabra software, existen categorías que se han realizado con base en la funcionalidad y su nivel de ejecución. Básicamente existen en cualquier computadora tres tipos de software:

- Aplicaciones: Aquel con que un usuario interactúa a través de una interfaz, dando pie a la generación de resultados de cualquier índole.
-

- Interfaz de Sistema Operativo o Shell⁵: Es el programa intermediario para la interacción de los usuarios con el sistema Operativo. Proporciona los medios e instrucciones para tener comunicación e información sobre nuestra computadora, de cualquier tarea, en cualquier momento.
- Sistema Operativo: Es el programa más importante de una computadora, por tal motivo es menester hablar de manera más amplia de este aspecto.

En la DGIRE se cuenta con servidores con capacidad de servicio y almacenamiento necesario para satisfacer las demandas de los usuarios no sólo internos, sino también del SI, vía el Internet.

Dichos servidores cuentan con el sistema operativo UNIX Solaris que a efectos de dar una perspectiva un poco más amplia al lector de la arquitectura funcional con la que cuenta, describiremos a continuación.

Su sistema de directorios está organizado en una forma arborecente, en la cual el directorio raíz es el comienzo de la expansión, tal como se muestra en la siguiente figura 3.1.

Ha sido interesante la manera en que UNIX maneja sus archivos, internamente son manejados como secuencias de bytes carentes de estructura y su almacenamiento se basa en bloques de tamaño fijo, indexados a través de apuntadores llamados i-nodos.

UNIX es un sistema multicapa, esto es, que se ha organizado en capas definidas de manera precisa para la ejecución de los programas. Existen de esta manera 4 capas como se describen en la figura 3.2:

⁵El shell es un término utilizado en los sistemas derivados de UNIX

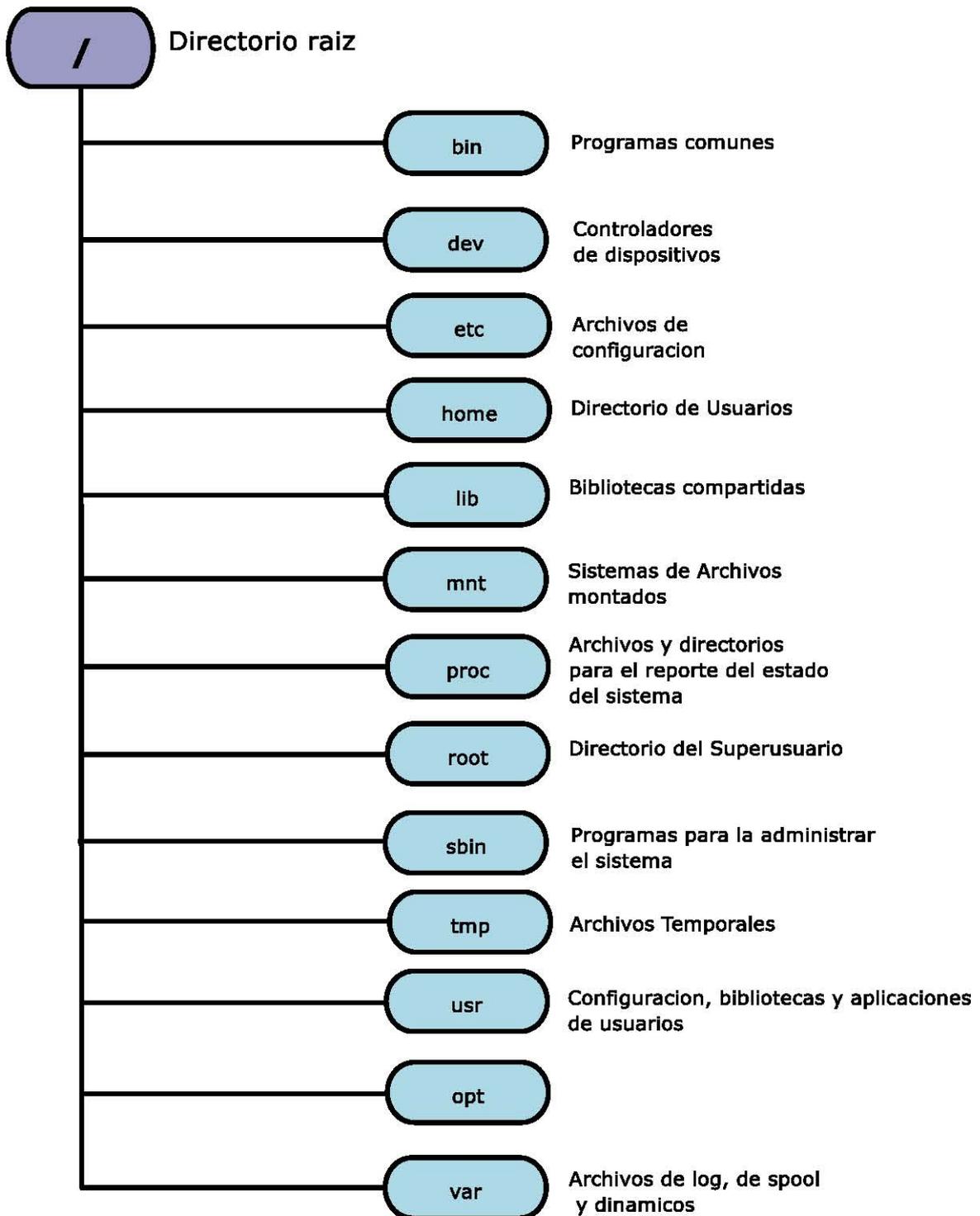


Figura 3.1: Estructura de directorios en UNIX

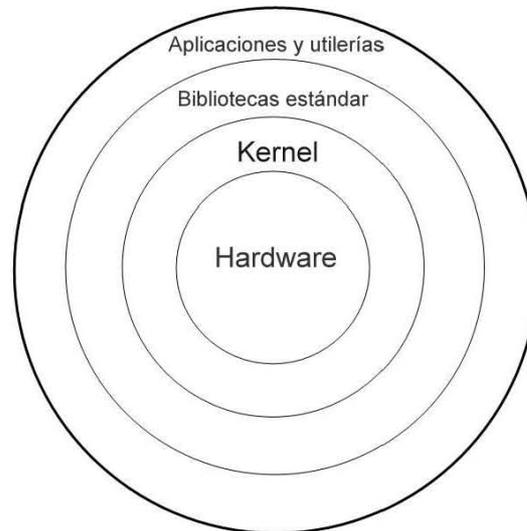


Figura 3.2: Estructura Multicapa de UNIX

- **Hardware:** En esta capa encontramos los recursos físicos con los que se realiza la interacción, tales pueden ser: CPU, memoria, impresoras, monitor, teclado, etc...
- **Kernel:** llamado también núcleo, es el encargado de gestionar el uso de la memoria, CPU, es decir, asignar recursos del sistema a los procesos que en sus momento requieran de un dispositivo, es también el encargado de mantener el sistema de archivos.
- **Bibliotecas estándar:** Escritas principalmente en el lenguaje C, están encargadas de crear y manipular recursos del sistema, para ello se realizan las *llamadas al sistema*.
- **Aplicaciones y utilerías:** Son todas aquellas aplicaciones destinadas a utilizarse directamente por usuarios para realizar tareas en específico.

El apoyo para trabajo en redes es una de las características más importantes de UNIX, ya que gracias a su sistema de archivos orientado a redes **NFS** *Network File System* diseñado por SUN Solaris, se tiene una especificación para importar o exportar archivos remotos de una máquina servidor a otra, a través de redes de área local. Es, así, responsabilidad del usuario privilegiado o administrador el decidir qué usuarios o recursos acceden a este tipo de servicios.

3.3.1. SMBD SYBASE

Dentro de las herramientas para el almacenamiento y gestión de la información, las Bases de Datos relacionales, sin duda han venido a revolucionar la tecnología desde hace ya más de una década, permitiendo a particulares, organizaciones e instituciones, llevar un control mucho más seguro y preciso de sus datos, almacenar grandes cantidades de información sin el peligro latente de la pérdida de ésta, y garantizar el acceso a dicha información desde sitios remotos con ayuda de la tecnología en telecomunicaciones.

En la actualidad existen múltiples y muy diversos Sistemas de Gestión de Bases de Datos, mejor conocidos como Sistemas Manejadores de Bases de Datos⁶, como ejemplo podemos mencionar entre los más conocidos a: *SYBASE*, *ORACLE*, *MySQL*, *SQLServer*, *Informix*, *Postgresql*. A pesar de las diferentes propiedades y funcionalidades que los caracterizan, en ellos persiste un concepto común, el cual sigue la tendencia a proporcionar un sistema que funcione como intermediario entre los sistemas finales de una aplicación y las bases de datos, así mismo proporcionar al usuario métodos con los cuales se pueda brindar un acceso rápido, una estructuración consistente de las bases, y un conjunto de herramientas que nos permitan el modelado de datos con respecto a las necesidades. Aunado a lo anterior, una de las características más importantes de un DBMS es la capacidad de proporcionar al desarrollador soporte, en cuanto a conectividad, a través de los lenguajes de programación⁷ que requieran un servicio de gestión con la Base de Datos almacenada.

Se ha logrado también la construcción de un lenguaje estándar, *SQL Structured Query Language* creado en 1979, con el cual podamos acceder a los recursos de Bases de Datos de forma funcional y práctica, de manera tal que el programador tiene ahora herramientas para realizar transacciones⁸ a la base de datos con fragmentos de código SQL, embebido en el código del lenguaje elegido para la programación de una cierta aplicación.

En particular, en esta sección hablaremos de las características esenciales del

⁶En lo posterior nos referiremos a ellos como DBMS (siglas en inglés de *Data Base Management System*)

⁷Actualmente es importante que el DBMS proporcione interfaces de conexión a lenguajes comunes, como pueden ser: *C/C++*, *Java*, *Perl*, *PHP*, *Delphi*, *Python*.

⁸Una transacción es un conjunto de instrucciones que se ven como una unidad lógica. A efectos de la consistencia de los datos, las transacciones deben cumplir cuatro reglas esencialmente (ACID): Atomicidad, Consistencia, Aislamiento(Isolation) y Durabilidad.

DBMS SYBASE.

Descripción de Sybase

El servidor Sybase consiste en: a) 2 procesos activos permanentemente, servidor de datos(Dataserver) y servidor de respaldo(Backupserver). b) Dispositivos(Devices) los cuales alojan las bases de datos; una base de datos (maestra) contiene datos de configuración y del sistema. c) Un archivo de configuración(Interfaces), el cual contiene los atributos del servidor.

Modelo de memoria

El modelo de memoria de Sybase consiste en:

- Sybase realiza el manejo de memoria con base en el modelo de paginación, que consiste en administrar la memoria en bloques de tamaño fijo(potencias de dos) llamados páginas.
- El área de programa donde el ejecutable, que accede al servidor de datos, es almacenado.
- El caché de datos, almacena las páginas que recientemente llegan desde el dispositivo de base de datos.
- El caché de procedimientos almacenados, el cual contiene sentencias sql optimizadas.

El servidor de datos Sybase trabaja como un proceso único dentro del SO, en este caso Unix; cuando múltiples usuarios están conectados a la base de datos, sólo un proceso es manejado por el SO. Cada conexión a base de datos sybase, requiere de 40 a 60 kb's de memoria. El parámetro de configuración "Memoria Total" determina la cantidad de memoria asignada al servidor. Esta memoria es utilizada inmediatamente después del arranque (start up) del servidor de DB, y no incrementa.

Procedimiento de transacción

Las transacciones son escritas al caché de datos, de donde llegan al registro de transacciones(log transaction) y al dispositivo de DB. Cuando ocurre un rollback, las páginas son desechadas de la caché de datos. El registro de transacciones es usado para restaurar los datos en situaciones de falla de cualquier índole. Una operación

checkpoint lleva todas las páginas actualizadas de memoria hasta sus respectivas tablas.

El almacenamiento de transacciones en el registro de transacciones (transaction logging) es requerido por todas las bases de datos; sólo los campos de imagen (blob) pueden quedar exentos. Durante un proceso de actualización, las páginas de datos que contienen renglones de una cierta tabla son bloqueadas. Esto dará pie al soporte de problemas cuando una transacción no este escrita eficientemente.

Procedimiento de respaldo

Una operacion dump database puede ser desarrollada cuando la base de datos esté o no en línea. Después los comandos dump transaction necesitan ser llevados a cabo durante el día, para asegurar recuperación aceptable de ventanas.

Procedimiento de recuperación

Un comando load database carga la base de datos designada con el llamado dump file. Después los comandos load transaction pueden entonces ser llevados a cabo para cargar múltiples transacciones dump files.

Configuración y seguridad de cuenta

El primer usuario en entrar a Sybase es “as” (administrador de sistema en español) o sa (sistem administrator en inglés). éste usuario tiene el papel de “sa_role” que es el súper usuario, en términos de Sybase.

Los registros de usuarios son ingresados a nivel servidor y entonces se les permite acceder a cada base de datos, cuando lo necesiten. En cada base de datos, el acceso a tablas puede ser concedido por requerimientos de aplicaciones. Un usuario también puede ser llamado como “dbo”, el cual automáticamente les concede todos los derechos dentro de una base de datos.

Creaciones de base de datos

Las bases de datos son inicializadas con el comando “create database”. No es inusual para un servidor de sybase contener diferentes bases de datos. Las tablas son creadas dentro de cada base de datos; los usuarios hacen referencia a las tablas usando la nomenclatura ownername.tablename. Nombrando a los usuarios en la base de datos, elimina la necesidad del prefijo. Típicamente, un usuario será nombrado

como “dbo” (database owner), lo cual también da el mismo resultado.

Una típica base de datos de sybase consistirá de 6 segmentos dispersos a través de varios dispositivos.

Tipos de datos

Los tipos de datos aceptados son integer, decimal, float, money, char, varchar, datetime, image, y text datatypes.

Los datos Image y text son declarados por punteros dentro de la estructura física de registro. El contenido del campo es guardado en páginas específicas. Por eso cada campo de texto o imagen requiere al menos 2 kilobytes de almacenamiento (En la mayoría de las plataformas).

Para el tipo cadena(string) y el tipo varchar pueden ser usados con longitudes de más de 255; el tipo TEXT puede ser usado para campos de datos más grandes.

Dentro de una declaración “Create table”, una columna puede ser identificada como una columna “identidad”. Lo cual ocasiona que sea incrementada automáticamente cuando las filas son insertadas.

Conceptos de almacenamiento

Las tablas son almacenadas en segmentos; un segmento es un área dentro de un dispositivo, con un nombre y un tamaño que es alojado por una base de datos. El registro de transacciones es guardado en su propio segmento, usualmente en un dispositivo separado.

Transact-sql

Es un lenguaje sólido de programación en el cual pueden ser escritos los procedimientos almacenados. Los procedimientos son almacenados en un formato compilado el cual permite una más rápida ejecución de código. Los cursores son aceptados procesando fila por fila. Son soportadas tablas temporales, las cuales pueden ser modificadas, las tablas de trabajo privado para ser creadas con procesos complejos. Cualquier número de resultados encontrados pueden ser regresados al llamado de las aplicaciones por medio de enunciados SELECT.

3.4. Entorno General del Hardware

Comenzaremos este panorama de tecnología con el aspecto hardware.

A medida que el tiempo transcurre las organizaciones han tenido la necesidad de ampliar sus modelos de funcionalidad, para ello han tenido que recurrir a nuevas y mejores alternativas de equipo para mayor productividad.

En estos tiempos es inevitable pensar que la computadora juega un papel preponderante en este proceso, siendo éste cada vez más ambicioso e interminable por ende, la tecnología avanza con el fin de minimizar el costo y tiempo de los servicios que proporcionan la empresas, instituciones y organizaciones que día a día necesitan mayores recursos y rapidéz.

La DGIRE tal como se mencionó anteriormente, está dentro de este proceso, para ello ha tenido que realizar grandes inversiones, en lo que a recursos de hardware se refiere, para poder así dar un servicio eficiente tanto a nuestros usuarios internos como al Sistema Incorporado.

Actualmente cuenta con 2 servidores, que brindan estos servicios, las características técnicas de ellos las podemos mencionar de la manera siguiente:

SERVIDOR:	“Azul”	“Unamsi0”
Marca	Sun Microsystems	Sun Microsystems
Modelo	Sunfire v880	Sun Fire v890
HDD Num.	12	12
HDD Capacidad	6 x 36 & 6 x 72 GB	438 GB
CPU Num.	4	4
CPU Vel.	900 MHz	1.2GHz
RAM	8GB	8GB

Tabla 3.1: Servidor Azul y Unamsi0

Para la ejecución de las aplicaciones en general las necesidades del usuario no son realmente exigentes. Se requiere de una computadora de propósito general con procesador mínimo pentium, al menos 256 MB en RAM y un módem o un medio de comunicación a Internet. Javascript y DHTML.

En cuanto a Software basta con el SO Windows, MacOS, o bien SO's que puedan soportar la ejecución de un navegador o Explorador en Internet que a su vez

tenga soporte para HTML, DHTML, JavaScript.

3.5. Telecomunicaciones

La palabra telecomunicaciones viene a ser un concepto contemporáneo, la mayoría de nosotros ha sido beneficiado de una u otra manera con su advenimiento e integración a los rubros más importantes en nuestra sociedad: educación, salud, negocios, gobierno, etc..

En sentido amplio, las telecomunicaciones comprenden los medios para transmitir, emitir o recibir signos, señales, o datos de cualquier naturaleza, entre más de dos puntos geográficos a cualquier distancia, a través de cables, radioelectricidad, medios ópticos u otros medios electromagnéticos. Al seno de la Unión Internacional de Telecomunicaciones(UIT) se tuvieron que realizar grandes esfuerzos en los años setenta y ochenta para avanzar hacia una definición aceptable. Su significado ha evolucionado rápidamente por la convergencia de diferentes tecnologías que han posibilitado la interconexión de dispositivos electrónicos y por la necesidad de comunicación entre personas.

El concepto de telecomunicaciones se ha enriquecido con el surgimiento de medios interactivos como la misma telefonía, computación, televisión y televisión por cable, que paulatinamente vienen disminuyendo las diferencias tecnológicas existentes entre ellos. A menudo, la expresión transmisión de datos ha sido confundida a la de radiodifusión, comunicación de voz, quizá porque esta última nos es más familiar, sin embargo la radiodifusión ha llegado a tener mucha similitud, siendo ésta parte de los sistemas de telecomunicaciones ya que, por ejemplo, las transmisiones de radio y televisión se realizan también vía telefónica, a través de los sistemas de de satélites.

Las telecomunicaciones en la actualidad se conforman básicamente por tres grandes medios de transmisión: cables, señales de radio y, como mencionamos en el anterior párrafo, satelitales. Las transmisiones por cables refieren la conducción de señales eléctricas a través de distintos tipos de líneas; un ejemplo común de ellos son las redes de cables metálicos (de cobre, coaxiales, hierro galvanizado, aluminio) y fibra óptica. Los cables metálicos se tienen en torres o postes formando líneas aéreas, o bien en conductores subterráneos y submarinos, donde se colocan también las fibras ópticas. Para las transmisiones por radio se utilizan señales eléctricas por aire o el espacio en bandas de frecuencia. Las comunicaciones satelitales presuponen el uso de satélites artificiales estacionados en la órbita terrestre para proveer de la

transmisión de señales a puntos geográficos predeterminados.

La digitalización es la base para el despliegue de las redes integradas, que proporcionan conectividad de extremo a extremo para apoyar una amplia variedad de servicios a los que los usuarios tienen acceso mediante un conjunto de dispositivos e interfaces. El Comité Consultivo Internacional de Telefonía y Telegrafía (CCITT) de la UIT inició, en 1966, los primeros estudios para normar el funcionamiento de redes digitales. Posteriormente se visualizó la potencialidad de dichos sistemas para gestionar una gama de servicios en forma simultánea. Los análisis se consolidaron hasta 1984 y se vió que la red integrada (RDI) evolucionaría tomando como base la red telefónica, por ser la más grande e interconectada en el mundo; se basaría en la tecnología digital y se integraría entre áreas de conmutación y transmisión.

Realmente la historia de las telecomunicaciones y su desarrollo a través del tiempo ha tenido uno de sus mayores logros en el surgimiento de INTERNET, consecuencia de investigaciones hechas en los sesentas por varias organizaciones⁹ que dieron pie a la creación de los antecesores¹⁰ de INTERNET y a los protocolos por medio de los cuales se organizaría el envío y recepción de datos.

Dichos protocolos¹¹ fueron al mismo tiempo integrando características de mayor robustez con el objetivo de hacer más eficientes la entrega y el envío de datos.

TCP/IP sigue siendo hasta nuestros días el protocolo estándar para transmisión de información, de esta manera hemos podido constatar que cada día más y más usuarios de las computadoras se están conectando con otros miles de usuarios y con otros servicios diversos, mantenidos todos ellos por redes locales, conexiones a Internet, sitios web, etc. En específico TCP es un conjunto de protocolos usados por casi cualquier sistema operativo (UNIX, macOS, Windows, Windows NT, entre otros). Es el lenguaje de internet y junto con IP definen una interfaz uniforme que puede utilizarse en distintos tipos de hardware, garantizando el intercambio de datos de una manera segura.

⁹ARPA(Agencia de Proyectos Avanzados), MIT(Instituto Tecnológico de Massachusetts), DARPA(Agencia de proyectos de investigación para la Defensa de Estados Unidos), IPTO(Oficina de Técnicas de Proceso de información), NSF(Fundación Nacional de Ciencia)

¹⁰ARPANET(1969): Creada por Bob Taylor, director del IPTO, como una red orientada a compartir recursos informáticos a trabajadores del IPTO,

¹¹NCP(1969)(Network Control Protocol, creado por DARPA-Rand Corporation), FTP(1971)-impulsado por el MIT

3.5.1. Condiciones actuales de telecomunicaciones en la DGIRE

Actualmente, la DGIRE se encuentra en un proceso de actualización de sus sistemas, de tal manera que permita el acceso a éstos desde sitios remotos. Todo ello con la ayuda de una infraestructura de telecomunicaciones que depende de la propia infraestructura de la UNAM. La red de telecomunicaciones abarca desde la llegada de la fibra óptica controlada por la Dirección General de Servicios de Cómputo Académico (DGSCA), hasta el acceso a los servidores. Para ello se ha utilizado una estructura de cableado cuya señal es distribuida por switch's.

Las siguientes figuras 3.3 y 3.4 se ejemplifican de manera general la estructura de conexión de estos switch's:

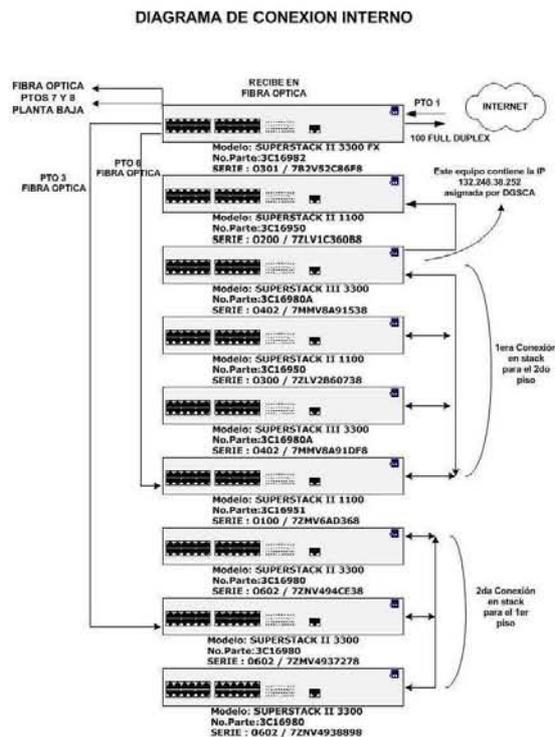


Figura 3.3: Rack del 1er. piso de la DGIRE

Es por medio de este sistema de telecomunicaciones que la DGIRE cuenta con una intranet que proporciona servicio a los trabajadores de la dependencia, aunado a ello, las escuelas del SI pueden ingresar a los sistemas.

La dependencia, con la adquisición de nuevos equipos y renovación de su tec-

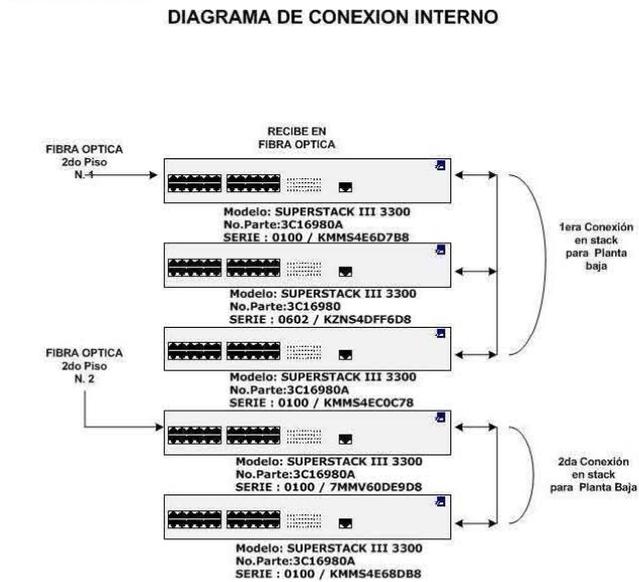


Figura 3.4: Rack de la planta baja de la DGIRE

nología en telecomunicaciones se encuentra preparada para futuros retos y proyectos basados en tecnologías orientadas a INTERNET, con lo cual podrá estar a la vanguardia.

3.6. Entorno de Seguridad del sistema

Seguridad..., quizá uno de los factores de más importancia en estos tiempos, y no tan solo referente al cómputo. La intención de cualquier tipo o sistema de seguridad necesariamente esta orientada a garantizar que un servicio sea confiable, estable y general.

En el cómputo, el Internet y con ello las aplicaciones web son temas que, en cuestión de seguridad, resultan nada triviales tanto a nivel usuario como a nivel de servicio (Servidores). Se trata de hacer que dichos sistemas y servicios hereden las características mencionadas en el anterior párrafo, pero para ello hay que tomar en cuenta algunos aspectos.

El primero de ellos es que estamos trabajando en sistemas donde el entorno es un medio público(Internet). Por tanto, cualquier persona en cualquier parte del mundo puede tener acceso a este. Por sí mismo, el Internet no es seguro, ya que dentro de su gran capacidad para la distribución de información surgen lo que conocemos como ojos de seguridad, vulnerabilidades tanto en las máquinas locales como servidores cada vez más difíciles de evitar. Como ejemplo de ello podemos hablar de la entrada de virus, espías, y más recursos informáticos malignos que con dicha tecnología y con el uso de los protocolos de comunicación han podido construirse.

Por otra parte, uno de los aspectos actuales que mantiene la atención puntual de los administradores en cuestión de seguridad son los llamados "hackers", también conocidos como white hats (sombremos blancos) o black hats (sombremos negros), según una clasificación de sus acciones (según sean solo destructivas o no, etc.), que son personas dedicadas a la investigación de vulnerabilidades o fallas de seguridad en cierto equipo, regularmente remoto.

Existe un número bastante amplio de herramientas de software que los administradores tienen a su alcance para la detección de intrusos y espías: bloqueo de puertos, encriptación de passwords, detectores de rutas, etc. Pero de todo ello lo más importante es llevar protocolos de seguridad sólidos que vayan orientados a crear filtros que minen la capacidad informática de nuestros enemigos.

El SO aquí retoma un papel preponderante, ya que éste es el que nos provee de las herramientas y posibilidades de crear un entorno de seguridad. Unix y sus descendientes: Linux, como hemos mencionado anteriormente, representan un SO

orientado a redes y por su diseño permite al administrador autenticar de manera eficaz a cada usuario y darle a este capacidades distintas. Por otra parte, realiza un manejo muy eficiente de los procesos que se ejecutan en el servidor, mantiene vitácoras* de entrada y salida, puntos de chequeo en un proceso, y configuraciones que permiten el bloqueo de máquinas remotas(intrusas), por tanto, será responsabilidad del administrador determinar las jerarquías de cada usuario con respecto a las actividades realizadas en un equipo.

Muchos de los administradores y organizaciones han decidido mantener los sistemas y aplicaciones en un servidor con SO descendiente de UNIX, ante la ya conocida vulnerabilidad de los servidores con un SO Windows atacados al por mayor.

En lo que a la DGIRE respecta, desarrolladores y administradores, la tarea es crear herramientas estándares con amplio soporte, con bases de diseño sólidas para que de esta manera puedan difundirse entre las personas relacionadas al cómputo y usuarios.

Para todo administrador de sistemas, tener máquinas conectadas a una red pública como es Internet, conlleva una serie de responsabilidades implícitas. Una de las más importantes es salvaguardar la integridad de dichas máquinas, así como proteger los datos de sus usuarios ante eventuales intrusos.

La DGIRE cuenta actualmente con 2 servidores en funcionamiento; uno de ellos trabaja en línea¹² y el otro ha sido cerrado¹³ al acceso público por razones de seguridad, con lo que sólo equipos registrados en la INTRANET pueden utilizarlo para fines específicos¹⁴.

Para el servidor que se encuentra en línea (con acceso público) se cuenta con herramientas comunes para impedir el acceso de intrusos al servidor, tales como una "Fire wall"¹⁵, un control de conexiones con TCP-Wrappers¹⁶, un detector de

¹²Por medio de este servidor se tiene acceso a las páginas y aplicaciones en WEB de la DGIRE

¹³Se niega el acceso a direcciones ip ajenas a un rango establecido por el administrador del servidor

¹⁴Usuarios comunes sólo acceden al servidor vía las aplicaciones y páginas liberadas, y usuarios avanzados(desarrolladores, administración y proceso de datos) interactúan con el servidor en niveles más bajos

¹⁵En español Pared de fuego: es un programa que sirve para filtrar lo que entra y sale de un sistema conectado a una red. De esta manera se pueden prevenir algunos tipos de comunicaciones a prohibidos por las políticas de red, fundamentadas en las necesidades del usuario

¹⁶Permite monitorear, controlar y proteger los servicios de red, limitando el acceso, y registrado todos las conexiones para hacer el trabajo de detectar y resolver problemas de forma más sencilla y ágil.

intrusos Portsentry¹⁷, entre otros.

Para el servidor que se encuentra fuera de línea se tienen las mismas herramientas, sin embargo el rango de posibles intrusos se reduce notablemente al no contar con un acceso público.

3.6.1. Servidores Web

Un servidor web es un programa que se ejecuta como servicio en el SO cuya función es implementar el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos.

Este servicio es también el encargado de direccionar y dejar el control de la ejecución de programas a instancias correspondientes dentro del SO. El Web Server retoma el control en el momento de término del proceso para realizar la siguiente orden.

Para los dos servidores se encuentra instalado el Apache Web Server, desarrollado por “The Apache Software Foundation”. Apache ha demostrado ser un software con amplio soporte para aplicaciones de muy diversos tipos, entre ellas los Cgi’s¹⁸ en C.

La seguridad del servidor WEB se basa en tener una configuración correcta y lo más adaptada al entorno de desarrollo; de manera tal que se brinde un marco de transparencia entre peticiones HTTP y la ejecución de los programas en el servidor.

Teniendo dicho entorno de seguridad en los servidores, no queda más que delegar la responsabilidad a los desarrolladores, cuyas aplicaciones deben estar orientadas a afrontar los riesgos con medidas eficaces y lo más eficientemente posibles.

¹⁷Es un programa que corre como servicio o demonio y que impide escaneo de puertos. Digamos que es un escucha de los puertos TCP/UDP.

¹⁸Common Gateway Interface.

Capítulo 4

Desarrollo del sistema “Planta Física”

4.1. ¿Qué es el sistema “Planta Física”?

Planta Física es un módulo dedicado al **Inventario de Instalaciones Físicas de las Instituciones del Sistema Incorporado**, que consiste en desarrollar un sistema que permita al SI el almacenamiento de características, específicas, de sus instalaciones, así como una herramienta que administre esta información y permita utilizarla en la DGIRE.

4.2. Requerimientos

4.2.1. Objetivo del Sistema

Contar con un sistema de cómputo que permita al SI el almacenamiento de la información sobre las características específicas de sus instalaciones físicas, para utilizar dicha información en los procesos de administración escolar.

Antes de abordar de lleno la parte de requerimientos propiamente, es menester tomar en cuenta algunas premisas que fueron de fundamental importancia para el desarrollo del sistema y que definen de manera general la estructura informática que

se ha utilizado para la gestión en el SI.

4.2.2. Glosario de términos

Enunciamos a continuación las siguientes definiciones y estructuras:

1. **Institución del SI:** Organización privada de carácter educativo, cuyos estudios académicos son reconocidos por la UNAM.
 2. **Sede:** Es el conjunto de instalaciones de una Institución (domicilio) donde se imparte uno o más planes de estudios. Para llevar el control de Sedes la DGIRE ha asignado un número consecutivo de sede en la Institución.
 3. **Plantel(s):** Son las diferentes claves de incorporación vigentes de la o las sedes que conformen la institución. Cada plantel cuenta con características académicas particulares.
 4. **Planta física:** Es cada una de las diferentes áreas en que está dividida una Sede. Cada una de ellas podrá ser ubicada por un “identificador”, según la nomenclatura asignada por la DGIRE.
 5. Los elementos de Planta Física que se han tomado en cuenta para la realización del sistema son las siguientes:
 - ▶ **Dirección Técnica.** Espacio de la dirección técnica del plantel
 - ▶ **Servicios Escolares.** Espacio de para los servicios escolares del plantel
 - ▶ **Aulas de Uso Común.** Impartición de clases de materias teóricas
 - ▶ **Aulas de dibujo.** Areas para la realización de la actividad de dibujo
 - ▶ **Aulas de actividades estéticas y artísticas.** Realización de las diferentes actividades estéticas
 - ▶ **Laboratorio Nivel Licenciatura.** Espacio de los laboratorios para materias correspondientes al nivel licenciatura
 - ▶ **Laboratorio de Cómputo.** Espacio donde se imparten materias relacionadas a computación
 - ▶ **Laboratorio Nivel Bachillerato.** Espacio de los laboratorios para materias correspondientes al nivel bachillerato
-

- ▶ **Auditorios y Salas de usos múltiples** Espacio utilizado para proyecciones y eventos diversos respectivamente.
 - ▶ **Bibliotecas.** Diferentes áreas de la biblioteca.
 - ▶ **Sanitarios.** Área de Sanitarios en el Plantel.
 - ▶ **Enfermerías.** Espacio destinado a enfermerías en el plantel.
 - ▶ **Extintores** En este módulo se encuentran la información sobre el número y ubicación de los extintores en las instalaciones.
 - ▶ **Otras áreas.** En este módulo se encuentran englobadas las áreas destinadas a actividades deportivas y áreas de esparcimiento y dispersión: entradas, salidas, escaleras, tomas de agua, pasillos, rampas para discapacitados y desniveles.
 - ▶ **Notas Adicionales** Espacio destinado para información adicional sobre instalaciones que el plantel quiera mencionar.
6. Cada planta física, por su naturaleza, contendrá elementos particulares a considerar.

4.3. Usuarios en Planta Física

Los usuarios del sistema “Planta Física” son los Directores Técnicos de las Instituciones del SI y los supervisores adscritos a la Subdirección de Incorporación de la DGIRE.

La importancia de la identificación de los usuarios en *Planta Física* radica en que la consulta o ingreso de datos es responsabilidad única de los Directores Técnicos de cada institución. Para ello se tiene un login, la clave de incorporación, y un password o contraseña asignada por la DGIRE. Para mayor seguridad, la contraseña puede ser modificada en el momento en que se decida por medio de un sistema habilitado en la página de la DGIRE¹.

El proceso de entrada de usuarios, consiste en proporcionar su login y password. Se han tomado las medidas necesarias en caso de que el usuario teclee mal ya sea el login, password ó los dos, presentándole una pantalla en la que se le indica un error de entrada y se le invita a regresar a retectar los datos solicitados en un principio.

¹www.dgire.unam.mx, en la parte correspondiente a Directores Técnicos.

De esta manera el usuario se hace responsable del almacenamiento y uso de la información de Planta Física.

Por otra parte en la entrada del sistema se tiene implementado un filtro con JavaScript, el cual consiste en checar las longitudes de cada uno de los campos, de esta manera no se realiza una petición inútil al servidor.

La seguridad correspondiente a usuarios que intentaran introducirse al sistema teniendo como objeto de intrusión la ruta de alguno de los Cgi's que componen el sistema, está implementada por medio de la petición de cadenas de nombre específico, así, el supuesto usuario tendría que saber no sólo la ruta, login y password sino también el nombre de la variable o variables requeridas por el cgi en acción. De no ser dados los datos correspondientes el sistemas bloqueará al supuesto usuario.

Para el uso interno de la DGIRE, se liberó una aplicación, que accede a la base de datos y tablas de almacenamiento en el servidor cerrado cuya funcionalidad está limitada para sólo consulta a efectos de que no haya manipulación de datos. El login de entrada será, de igual forma, la clave de incorporación y no será necesario un password.

4.3.1. Requerimientos Funcionales

Los requerimientos funcionales describen las partes modulares de funcionamiento entre el sistema y su ambiente, de forma independiente a su implementación. El ambiente incluye al usuario y cualquier otro sistema externo con el cual interactue el sistema Planta Física.

Se identificaron los siguientes requerimientos funcionales:

La Planta Física deberá permitir:

- Validar usuarios del sistema
 - Identificar la Institución
 - Seleccionar la Planta Física a capturar
 - Capturar los datos
 - Validar en la captura
-

- Validar internamente la información
- Control de fecha de ingresos y actualizaciones
- Lista dinámica de los elementos capturados
- Consulta de información de un elemento capturado
- Agrupación de la información por Sede
- Información al usuario.

⊙ Validar Usuarios del sistema

Planta Física deberá proporcionar una interfaz gráfica del Sistema con la cual se permita la entrada de datos del plantel o usuario, vía un login y password, para efectos de realizar la validación a nivel base de datos, de esta forma proporcionar o no el acceso.

⊙ Identificar la Institución

El sistema deberá identificar el nombre de la institución correspondiente al login y password, por medio de un comprobación en la base de datos, además de que en toda instancia del sistema muestre información que permita al usuario saber en que institución está trabajando.

⊙ Selección de la Planta Física a capturar

Una vez iniciada la sesión en Planta Física, el sistema presentará un menú de captura, cuyo objetivo es seleccionar un elemento de la planta física para iniciar su captura.

⊙ **Captura de datos**

El ingreso de datos de una sección del sistema en específico se hará por medio de campos cuya tipificación estará en función de la semántica inherente al nombre de dicho campo.

⊙ **Validación en la captura**

La captura de datos en una sección deberá ser validada por el sistema al vuelo, es decir, se realizará el chequeo de la información introducida en un cierto campo, con respecto a su tipificación. En caso de que el dato introducido sea erróneo, se envía un mensaje indicando la falla y se limpia el contenido del campo.

⊙ **Validación Interna**

Se debe realizar una segunda validación justo antes de que el sistema introduzca los datos capturados por el usuario, de manera que si algún dato resultó erróneo después de la primera validación, o simplemente se violó ésta, los datos no sean introducidos de manera permanente. Si existe un error se envía un mensaje y se marcan los campos que tuvieron el error.

⊙ **Control de fechas de Ingresos y Actualizaciones**

Se definió que el sistema lleve el registro de fechas de alta de un elemento y de su última modificación. Así, cada vez que un elemento existente sea modificado, se ingresará una nueva fecha de última modificación.

⊙ **Lista dinámica de los elementos Capturados**

El sistema deberá llevar una lista visible en la cual se incluyan los elementos que se hayan ingresado en una sección de Planta Física, tomando en cuenta que

dicho ingreso será registrado en la lista de todos los planteles que pertenezcan a una cierta sede².

⊙ Consulta de información de un elemento capturado

Debemos poder consultar la información de cualquier elemento definido anteriormente. Para ello será necesario poder seleccionar un elemento y mostrar la información de éste, almacenada en la base de datos.

⊙ Agrupación de la información por Sede

En cada sección de la planta física será necesario mostrar todos los planteles pertenecientes a una sede, a efectos de orientar al usuario acerca del conjunto de planteles que serán afectados por los cambios realizados.

⊙ Información al Usuario

Se debe brindar ayuda al usuario mientras captura la información de los campos del formulario de una sección.

4.3.2. Requerimientos No Funcionales

Los requerimientos no funcionales describen aspectos de nuestro sistema transparentes para el usuario, por tratarse de aspectos que no se relacionan en forma directa con el comportamiento funcional de éste, pero que son fundamento para el funcionamiento del sistema

⊙ Ambiente Distribuido

Planta Física deberá ejecutarse en un ambiente distribuido; esto es, la información de las bases de datos y la aplicación misma se mantendrán de forma centralizada en un servidor; pero los usuarios podrán acceder al sistema usando sus

²La razón de este “registro simultáneo” tiene fundamento en que se trata con distintos planteles en las mismas instalaciones físicas.

propios equipos. De esta manera muchos usuarios accederán de forma concurrente al sistema desde distintas localizaciones; la parte referente al usuario de la aplicación deberá poder ejecutarse en equipos con recursos menores a las de un servidor.

⊙ Seguridad

Dado que Planta Física será utilizado por usuarios “privilegiados”³, será necesario garantizar que el acceso a éste y por ende a los datos contenidos, este controlado.

⊙ Portabilidad

Planta Física, deberá ser un sistema que pueda utilizarse en cualquier máquina, sin necesidad de recursos de alto nivel o especializados. Por tanto, la portabilidad de dicho sistema estará dada por el ambiente de trabajo en el que opera.

³Usuarios quienes cuentan con una clave del SI.

4.4. Interfaz Gráfica o Front-End del Sistema

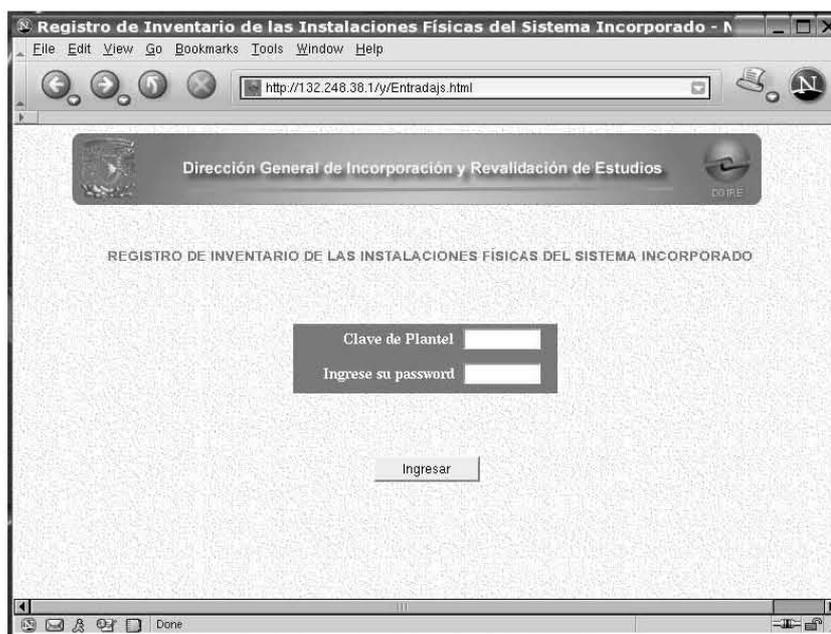


Figura 4.1: Entrada a Planta Física

Esta ventana es la puerta de entrada a “Planta Física”. Es necesario ingresar el login y password correspondiente y hacer click en el botón **Ingresar** para poder pasar al menú de opciones o menú principal.

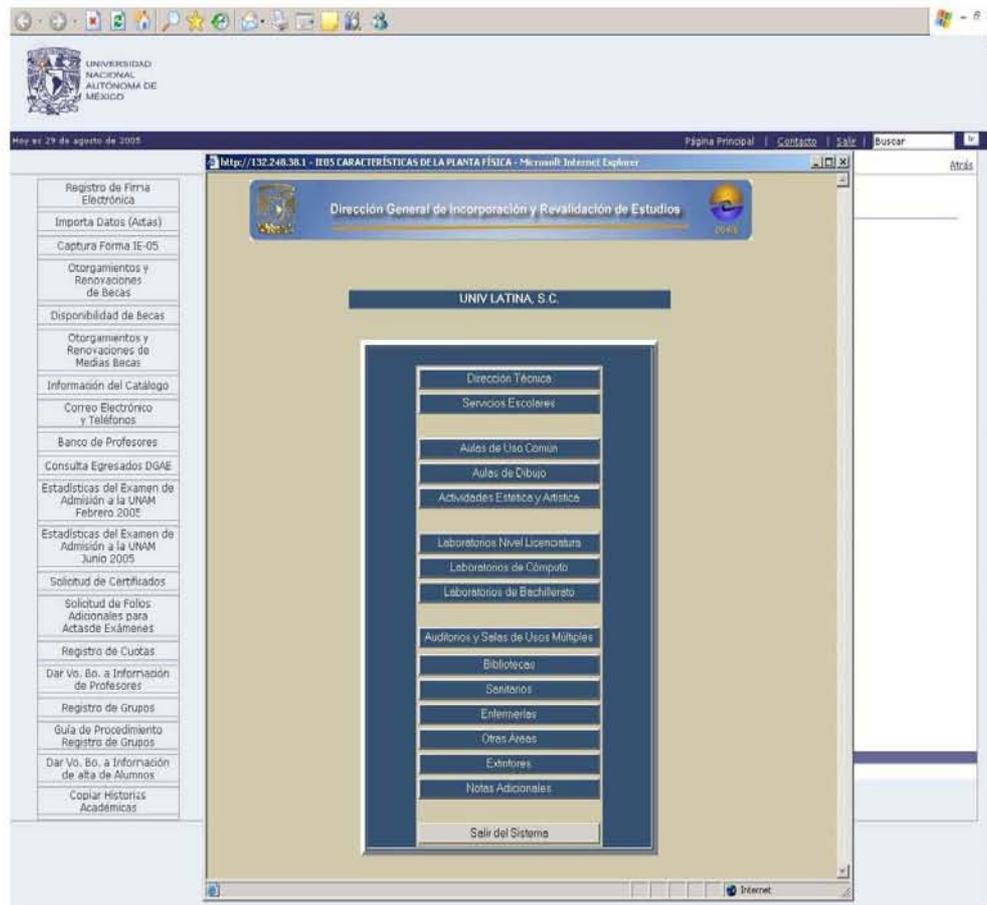


Figura 4.2: Menú de Planta Física

Este es el menú de opciones o menú principal. Por medio de éste podemos navegar a través de “Planta Física”, accionando el botón de la opción requerida para capturar, o bien el botón **Salir del Sistema** para regresar a la página de entrada. En la parte superior de esta ventana se muestra el nombre de la institución correspondiente al login y password ingresados.

Figura 4.3: Aulas de uso Común

Describiremos de manera general las características de los módulos por medio de **Aulas de Uso Común**.

Podemos tener acceso a cada uno de los módulos oprimiendo el botón correspondiente en el menú de opciones. Al ingresar a un módulo podremos observar ciertas características que identifican a éste.

En la parte superior izquierda se encuentran la clave del plantel o los planteles agrupados en una sede, cualquier cambio en este módulo, afectará a dichos planteles.

Un poco más abajo se encuentra el nombre del módulo de captura. Debajo del nombre del módulo se encuentra la lista de identificadores ingresados por medio de esta forma. Los detalles o contenido de dichos identificadores podrán ser vistos al señalar uno de ellos y posteriormente oprimir el botón **Ver Detalles**; en ese momento los datos serán mostrados en la forma.

Debajo del botón **Ver Detalles** se mostrará la cantidad de identificadores contenidos en la lista.

De igual manera, si es necesario eliminar un identificador junto con su información se señalará dicho identificador y se oprimirá el botón **Eliminar Registro**.

Algunos de los datos de ingreso en la forma, pueden dar una sugerencia de llenado; esto es, al pasar el ratón sobre el nombre del campo, emergerá dicha sugerencia.

El campo **Área(m²)** se calculará de manera inmediata al ingresar los datos a los campos de largo y ancho.

Dentro de las formas existen dos tipos de Fechas: **Fecha de Alta** y **Fecha de Modificación**.

En la primera aparecerá la fecha actual de la primera vez que se ingresen y sean guardados los campos de la forma con identificador específico; esta fecha no cambiará. La segunda fecha cambiará a la fecha actual, cada vez que sea modificado algún campo de la forma.

Para realizar el envío de la información se debe oprimir el botón **Enviar Información**, la información será enviada siempre y cuando la validación de cada campo sea positiva.

El botón **Limpiar Datos**, nos ayudará a borrar cualquier dato ingresado en la forma, incluso el campo de ingreso del identificador, dando pie a un nuevo ingreso.

Por último, en la parte inferior derecha se encuentra el botón de **Regresar al Menú**, el cual, como su nombre lo indica, traslada al usuario a la parte del Menú de Opciones.

En general, la forma de captura contienen estos elementos, esenciales para la gestión de la información almacenada por “Planta Física”.

A continuación se muestra la interfaz gráfica de algunos de los módulos con los que cuenta el sistema:

Figura 4.4: Dirección Técnica.

Figura 4.5: Servicios Escolares.

CLAVES:
108420843084

Universidad Nacional Autónoma de México
Dirección General de Incorporación y Revalidación de Estudios.

Aulas Actividades Estéticas

Características Generales	Cantidad de Mobiliario	Iluminación y Ventilación
ACTIVIDADES Largo (m): 6.73 Ancho (m): 5.89 Área (m ²): 39.63 Altura (m): 2.94 Nombre Actividad: Danza Clasi	Sillas: 0 Escritorios: 1 Mesas: 0 Pizarrones: 1	Tipo de Ventilación: <input type="checkbox"/> Circada <input checked="" type="checkbox"/> Sensilla <input checked="" type="checkbox"/> Mecánica Tipo de Iluminación Artificial: Total Watts Fluorescentes: 234 Total Watts Incandescentes:

Ventanas:
 Área de Iluminación Natural: 6.91 m²
 Área de Ventilación Natural: 5.37 m²
 Fecha de Alta: 17/08/2005
 Última Fecha de Modificación:

Enviar Información Regresar al Menú
Limpieza Datos

Figura 4.6: Aulas de Actividades Estéticas.

CLAVES:
108420843084

Universidad Nacional Autónoma de México
Dirección General de Incorporación y Revalidación de Estudios.

Laboratorios de Licenciatura

Características Generales	Cantidad de Mobiliario	Iluminación y Ventilación
LABORATORIO Largo (m): 7 Ancho (m): 6 Área (m ²): 42 Altura (m): 3 Tipo Laboratorio: LINUX	Mesas: 6 Sillas: 6 Pizarrones: 2	Tipo de Ventilación: <input checked="" type="checkbox"/> Circada <input type="checkbox"/> Sensilla <input checked="" type="checkbox"/> Mecánica Tipo de Iluminación Artificial: Total Watts Fluorescentes: 1500 Total Watts Incandescentes:

Ventanas:
 Área de Iluminación Natural: 3.5 m²
 Área de Ventilación Natural: 2.7 m²
 Fecha de Alta: 09/08/2005
 Última Fecha de Modificación:

Enviar Información Regresar al Menú
Limpieza Datos

Figura 4.7: Laboratorios de Licenciatura.

CLAVES:
1084,2004,3094

Laboratorios de Bachillerato

LABFISICA
LABQ-2
LABQ-B1
LABQ-B2

Ver Detalles
4 Registrados
Eliminar registro

Características Generales	Cantidad de Mobiliario	Iluminación y Ventilación	
Identificador: LABQ-B1	Mesas: 4	Tipo de Ventilación:	Tipo de Iluminación Artificial:
Largo (m): 7.82	Bancos: 25	Cruzada: <input type="checkbox"/>	Total Watts Fluorescentes: 312
Ancho (m): 5.86	Pizarrones: 1	Senilla: <input checked="" type="checkbox"/>	Total Watts Incandescentes: <input type="text"/>
Área (m2): 45.82	Estantes: 0	Mecánica: <input checked="" type="checkbox"/>	
Altura (m): 2.94	Botiquín: <input checked="" type="checkbox"/>		
Tipo de Laboratorio: QuimBiolo			

Ventanas

Área de Iluminación Natural: 9.36 m2
Fecha de Alta: 17/08/2005
Última Fecha de Modificación: 17/08/2005

Área de Ventilación Natural: 4.68 m2

Enviar Información
Regresar al Menú
Limpiar Datos

<<Laboratorios 2da Parte>>

INSTALACIONES

Medidas de las Mesas	Número de Instalaciones	Instalaciones Visibles	Controles Maestros	Seguridad
Altura (m): 1	Tallas: 10	Gas: <input checked="" type="checkbox"/>	Gas: <input checked="" type="checkbox"/>	Regadera de Presión sin Sardinat: <input checked="" type="checkbox"/>
Largo (m): 5	Tomas de Gas: 10	Eléctricas: <input checked="" type="checkbox"/>	Eléctricas: <input checked="" type="checkbox"/>	Contenedor de Residuos Tóxicos: <input checked="" type="checkbox"/>
Ancho (m): 1	Tomas Eléctricas: 10	Agua: <input type="checkbox"/>	Agua: <input checked="" type="checkbox"/>	Arco para resguardo de material y equipo: <input checked="" type="checkbox"/>
	Tomas de Agua: 10			

<<Volver a Lista>>

Figura 4.8: Laboratorios de Bachillerato.

4.5. Arquitectura del sistema

Planta Física es una aplicación de 3 capas en 2 niveles. Sin embargo, y para un contexto más amplio, hemos definido un nivel intermedio: el de telecomunicaciones, que es el medio de transporte de la información.

A nivel físico, Planta Física se divide como se muestra en la figura 4.10.

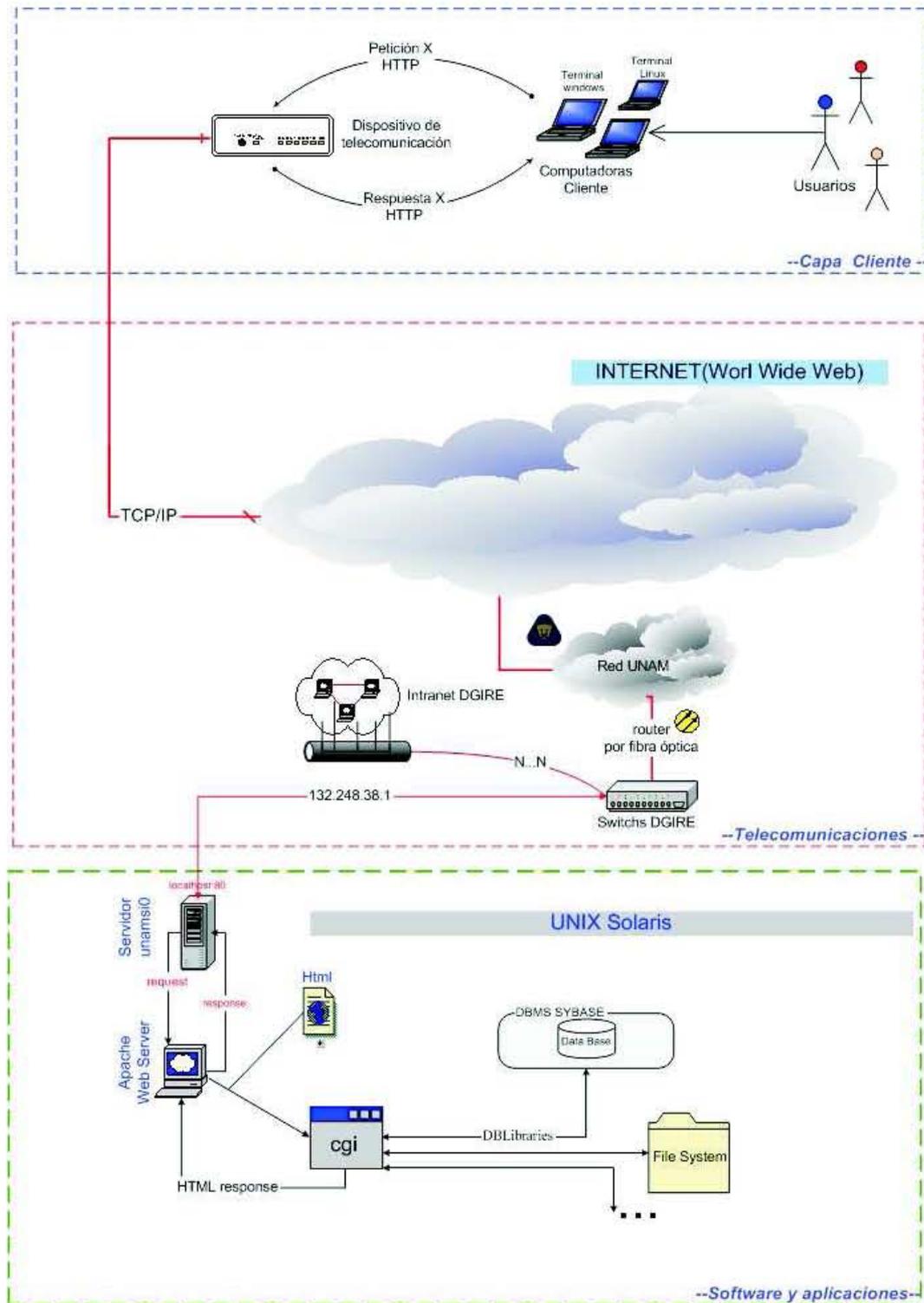


Figura 4.9: Diagrama de tres niveles de la aplicación

4.6. Almacenamiento de Información

La información ingresada por medio de Planta Física, se ha almacenado en tablas de una base de datos específica en Sybase, siguiendo para ello, el modelo bases de datos relacionales.

Para diseñar las tablas del sistema se hizo uso de la herencia, con el objeto de minimizar la repetición de campos. De esta manera se crea la tabla *pf_rasgos_grales* cuya función es almacenar, como su nombre lo indica, los datos generales que deberán contener las tablas que de ella heredan.

Para poder relacionarse, estas tablas cuentan un campo llave: el identificador de la planta Física correspondiente. Las tablas que heredan de *pf_rasgos_grales* son las siguientes (fig. 4.11).

- *pf_aesteticas*
- *pf_lbach*
- *pf_lablic*
- *pf_lcomputo*
- *pf_auditorios_salas*
- *pf_sanitarios*

Existen cinco módulos en Planta Física cuyo almacenamiento de información depende de la relación entre dos tablas: *pf_rasgos_grales* y *pf_mobiliario*(fig. 4.12).

Los módulos integrados en dichas tablas son:

- Aulas de uso común
 - Dirección Técnica
 - Servicios Escolares
 - Aulas de Dibujo
 - Enfermerías
-

Para el almacenamiento de los módulos de: Bibliotecas, Areas Deportivas, Extintores, Areas de Esparcimiento y Disperción se cuenta con tablas individuales sin relación con otras (fig. 4.13).

Se presenta a continuación el diagrama entidad-relación de las tablas que han dado estructura para el almacenamiento de los datos de Planta Física.

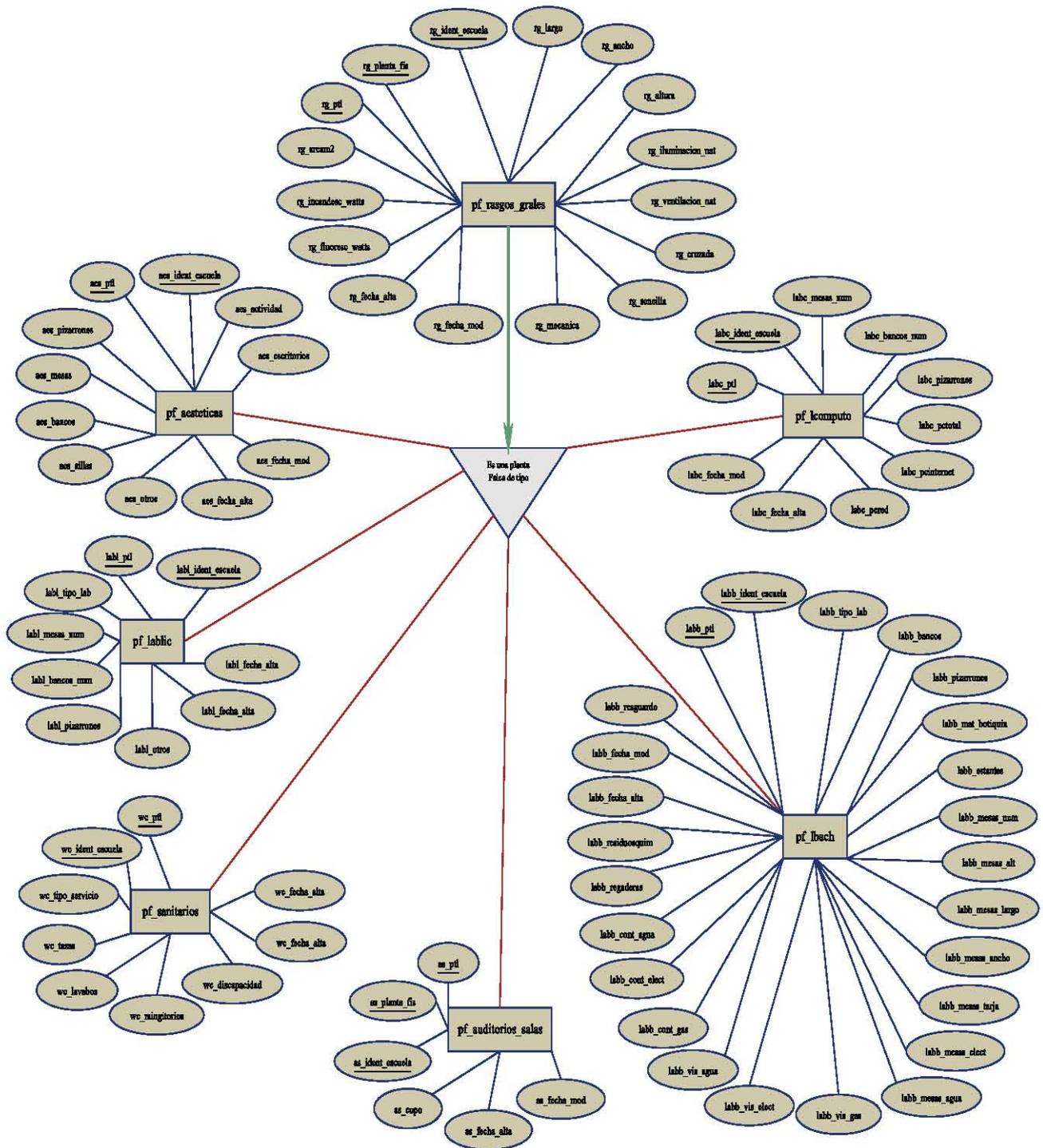


Figura 4.10: Diagrama E-R: Tablas con herencia de Planta Física.

4.6.1. Normalización de las tablas

Por otra parte, podemos afirmar que el diseño de las tablas de Planta Física es un buen diseño, ya que dichas tablas están en 3ra forma normal, como se expresa a continuación.

- Las tablas están en primera forma normal ya que cada uno de los elementos de estas son indivisibles, son de un sólo tipo respectivamente, y ninguna columna es idéntica a otra dentro de las relaciones.
- Las tablas se encuentran en segunda forma normal, ya que:

1. Están en primera forma normal.
2. Las tablas que complementan a *pf_rasgos_grales*, es decir: *pf_aesteticas*, *pf_lablic*, *pf_sanitarios*, *pf_auditorios_salas*, *pf_lbach* y *pf_lcomputo*, contienen atributos que dependen de manera única de los campos claves: *ptl*, *ident_escuela*. Los campos clave determinan de manera única cada registro contenido en dichas tablas.

Por otra parte en la tabla de *rasgos_grales* y *pf_mobiliario* los elementos clave de: *ptl*, *planta_fis* e *ident_escuela* determinan de manera única a los campos no primos de las relaciones respectivamente.

Cada uno de los campos no primos de las tablas, independientes a su vez, también son determinados por la llave formada por: *ptl* e *ident_escuela*. La tabla de bibliotecas, contiene un elemento más en la llave que es *bib_zona*, con lo cual se identifican de manera única a cada área de la biblioteca *X* del plantel *M*.

- Las tablas se encuentran en 3ra. FN. ya que:
 1. Se encuentran en segunda forma normal.
 2. La única forma de determinar cada atributo de cada tabla es por medio de sus llaves primarias, las cuales a su vez son independientes de cualquier otra relación, y no existe transitividad en los elementos no primos contenidos en cada una de ellas ya que ningún atributo que no sea la llave conjunta logra determinar a otro atributo. Por tanto no existen dos caminos distintos para identificar un campo. De esta manera podemos asegurar la consistencia e identificabilidad de cada registro en estas tablas.
-

Las restricciones de integridad con las que cuenta cada tabla de Planta Física son la definición de llaves primarias e índices únicos en los campos clave. Aunado a ello, la actualización y borrado de cada registro es en cascada, lo cual nos permite asegurar que la integridad referencial se conserva.

Por el diseño de las tablas no existe la posibilidad de que hayan registros duplicados o no identificables.

4.6.2. Acceso a Sybase

Para tener la capacidad de acceder con C a cualquier DBMS, en particular a Sybase, es necesario tener herramientas que nos provean de una interfaz con elementos necesarios para gestionar, desde el código fuente, con la Bases de Datos a utilizar; así también es importante que dichas herramientas nos proporcionen un medio rápido, seguro, y eficaz para el tratamiento de errores propios de la gestión con una base de datos.

Sybase cuenta con dos familias de productos para escribir tanto aplicaciones cliente como servidor, *Open Client* y *Open Server*.

Las *Open Client* proveen al usuario de interfaces necesarias para comunicarse vía SQL con Sybase. Dicha interfaz tiene fundamento en bibliotecas diseñadas para desarrollar aplicaciones cliente, *Client-Library*, *DB-Library* y *CS-Library*.

Para nuestros propósitos, hemos hecho uso de las *DB-Library/C*, que son una colección de rutinas y macros⁴ escritas en C, para la programación de aplicaciones cliente. Las *DB-Libraries* incluyen también una serie de funcionalidades para hacer uso de los algoritmos two-phase commit y para tener acceso a versiones antiguas de Sybase. Dichas bibliotecas se encuentran totalmente documentadas y son, a la fecha, consideradas, por su eficiencia, uno de los medios más comunes entre los programadores en C.

Las *DB-Library's* así, dan posibilidad de tener el lenguaje de SQL embebido en el código en C de nuestros CGI's, pora ello se hace uso de las bibliotecas *sybdb.h* y *sybfront.h* como sigue:

```
#include <sybfront.h>
#include <sybdb.h>
```

Las consultas a la base de datos se realizan pormedio de las funciones contenidas en las bibliotecas. Para abrir una sesión con Sybase es necesario especificar la ubicación absoluta del archivo interfaces, la cual contiene la dirección del servidor Sybase, localización e identificación de dispositivos, y el nombre de la base de datos correspondiente, acontinuación se presenta:

⁴Son palabras reservadas con algún significado semántico en el código, definidas ya sea por el usuario o lenguaje

```
dbsetifile("/home/sybase/interfaces");
```

A continuación se especifica el nombre de la sesión, el login, password del usuario autorizados para tener acceso a la base de datos y el nombre de la base de datos default a acceder:

```
RETCODE    result_code; //Almacena los mensajes de error o de exito de la
              //transacci'on

DBPROCESS *Alumnos; //Almacena los resultados de los Query's

LOGINREC   *login; // Estructura de datos que almacena, login, password y
              // nombre de sesi'on

login = dblogin(); // Se inicializa la estructura.

DBSETLUSER(login, "p_fisica"); // Se asigna el login
DBSETLPWD(login, "XXXXXXXX"); // Se asigna el password
DBSETLAPP(login, "SesionAulas");// Se asigna el nombre de la sesi'on
dbuse(Sedes, "unamsi3");
```

Posteriormente se verifica que la conexión se haya realizado exitosamente:

```
if (dbinit() == FAIL)
    exit(ERREXIT);
```

Así se inicia con la estructuración de la consulta que nos dará como resultado los datos u operaciones requeridas:

```
dbfcmd(Sedes, "select B.ins_ptl from unamsi3.dbo.instituciones A,
              unamsi3.dbo.instituciones B ");
dbfcmd(Sedes, "where A.ins_ptl = '%s' and A.ins_cve=B.ins_cve and
              A.ins_sede = B.ins_sede" ,ptlw);

dbsqlxexec(Sedes); //Ejecuci'on de la transacci'on
```

Una vez realizada la consulta se da pie a el almacenamiento de resultados en las cadenas asignadas. Es importante resaltar que la longitud de las cadenas para almacenamiento debe ser igual que la longitud de las cadenas declaradas en Sybase, por tanto si tenemos un campo *bsd1 char(5)* debemos de declarar un DBCHAR x[6], para el almacenamiento en C, la razón del incremento en uno corresponde a que debemos de asignar espacio al valor nulo de la cadena.

En ocasiones podremos tener, según la consulta, varias tuplas como resultado de una consulta, por tanto el almacenamiento debe hacerse con la ayuda de un while en C. No debemos confundir el while principal que es usado para extraer los datos de la consulta.

Sybase arroja el valor *SUCCEED* cuando existe por lo menos una tupla resultante y cuando se han acabado de traer los resultados despliega el valor *NO_MORE_ROWS*.

En el ejemplo a continuación se puede ver como se da el almacenamiento de las tuplas:

```
while ((result_code = dbresults(Sedes)) != NO_MORE_RESULTS) {
    if (result_code == SUCCEED && DBROWS(Sedes) == SUCCEED) {

        //se almacenan los datos en la variable ptlxsede
        dbbind(Sedes, 1, NTBSTRINGBIND, (DBINT)0, ptlxsede);

        //se agotan resultados
        while (dbnextrow(Sedes) != NO_MORE_ROWS);
    }
}
```

En este momento ya se puede hacer uso de los datos contenidos en la variable *ptlxsede*.

Es importante que recordemos que la sesión *Sedes* continúa abierta, por tanto debemos cerrarla:

```
dbexit();
```

Las DB-Libraries poseen recursos no solamente para ejecutar transacciones, sino también para evaluar cómo se ejecutan. Es decir, podemos crear una biblioteca de mensajes de error como sigue:

```
int err_handler(dbproc, severity, dberr, oserr, dberrstr, oserrstr)
DBPROCESS      *dbproc;
int severity;
int dberr;
int oserr;
char *dberrstr;
char *oserrstr;
{
    printf ("Content-type: text/html\n\n"); //Imprimimos el Header.
    printf ("<HTML>\n");
    printf ("<HEAD>\n");
    printf ("<TITLE>DGIRE UNAM</TITLE>\n");
    printf ("</HEAD>\n");
    printf ("<BODY bgcolor=\"#99DDFF\" text=\"#355270\">\n");
    printf ("<HR>\n");
    if (dberr == 20014)
        printf("Revise su Login y su password <BR>\n");
    else {
        printf(" dberr          : %d <BR>\n", dberr); //20014 = login y/o
                                                    //pass incorrectas
        printf(" dberrstr       : %s <BR>\n", dberrstr);
    }
    printf ("<HR>\n");
    printf ("</BODY></HTML>\n");
    if ((dbproc == NULL) || (DBDEAD(dbproc)))
        return(INT_EXIT);
    else {
        fprintf (stderr, "DB-Library error:\n\t%s\n", dberrstr);
        if (oserr != DBNOERR)
            fprintf (stderr, "Operating-system error:\n\t%s\n", oserrstr);
        return(INT_CANCEL);
    }
}
```

```

    }
}

```

La cual se inicializa en el main del programa:

```
dberrhandle(err_handler);
```

También podemos crear una función para mensajes de error especializados:

```

int msg_handler(dbproc, msgno, msgstate, severity, msgtext, srvname, procname, line)
    DBPROCESS    *dbproc;
    DBINT        msgno;
    int          msgstate;
    int          severity;
    char         *msgtext;
    char         *srvname;
    char         *procname;
    DBUSMALLINT line;
{
    if ((msgno == 5701) || (msgno == 5703))//no son errores
    {}
    else
    {
        printf ("Content-type: text/html\n\n"); //Imprimimos el Header.
        printf ("<HTML>\n");
        printf ("<HEAD>\n");
        printf ("<TITLE>MSG</TITLE>\n");
        printf ("</HEAD>\n");
        printf ("<BODY>\n");
        printf ("<HR>\n");
        // printf ("Msg %ld, Level %d, State %d\n", msgno, severity, msgstate);
        // if (strlen(srvname) > 0)
        //     printf ("Server '%s', " , srvname);
        // if (strlen(procname) > 0)

```

```
    //      printf ("Procedure '%s', ", procname);
    // if (line > 0)
    //      printf ("load-----Line %d", line);
printf ("\t%s", msgtext);
fprintf (stderr, "\t%s", msgtext);

    printf ("</BODY></HTML>\n");
}
return(0);
}
```

La cual se inicializa de la siguiente forma:

```
dbmsghandle(msg_handler);
```

4.6.3. Factores para una migración

Conocemos del poderío del DBMS Postgresql, por lo cual hacemos un razonamiento enfocado a ¿cuales serían los requerimientos para una posible migración de las tablas de Planta Física a éste DBMS?

Primeramente hay que crear las tablas, para lo cual fue necesario un análisis de los tipos de datos que existen en este sistema. Como resultado obtuvimos que existen variaciones en cuanto a la definición de tipos de datos con Sybase, por lo cual habría que definirlos de la siguiente manera:

Sybase type :- Postgresql Type

- tinyint :- smallint
- smalldatetime :- timestamp
- float :- float4
- text :- text(sin límite definido)

Existen más diferencias en cuanto a tipo de datos, sinembargo, éstas son las que causan conflicto con Planta Física.

En cuanto a índices y estructura son similares, aunque podríamos resaltar el hecho de que en Sybase hay que especificar que los atributos definidos serán nulos, contrario a lo que pasa en Postgresql, en el cual los atributos deben ser definidos como no nulos.

En cuanto al acceso, las bibliotecas de acceso a Postgresql cambiarán a la biblioteca: *”libpq-fe.h”* con la cual se podrá tener SQL embebido en C. Un ejemplo de ello es:

```
#include <stdio.h>
#include <stdlib.h>
#include "libpq-fe.h"
```

```
static void exit_nicely(PGconn *conn)
{
    PQfinish(conn);
    exit(1);
}

int main(int argc, char **argv)
{
    const char *conninfo;
    PGconn      *conn;
    PGresult     *res;
    int          nFields;
    int          i,
               j;

    if (argc > 1)
        conninfo = argv[1];
    else
        conninfo = "dbname = template1";
    /* Realiza la conexion a la base de datos */
    conn = PQconnectdb(conninfo);
    /* Checa si La conexi'on fue exitosa*/
    if (PQstatus(conn) != CONNECTION_OK)
    {
        fprintf(stderr, "Connection to database '%s' failed.\n", PQdb(conn));
        fprintf(stderr, "%s", PQerrorMessage(conn));
        exit_nicely(conn);
    }

    /* Comienza el bloque de conexi'on */
    res = PQexec(conn, "BEGIN");
```

```
    if (PQresultStatus(res) != PGRES_COMMAND_OK)
    {
        fprintf(stderr, "BEGIN command failed: %s", PQerrorMessage(conn));
        PQclear(res);
        exit_nicely(conn);
    }
    /*
    *Limpia el contenedor de consultas
    */
    PQclear(res);

    /*
    * Arroja resultados
    */
    res = PQexec(conn, "DECLARE myportal CURSOR FOR select * from pg_database");
    if (PQresultStatus(res) != PGRES_COMMAND_OK)
    {
        fprintf(stderr, "DECLARE CURSOR failed: %s", PQerrorMessage(conn));
        PQclear(res);
        exit_nicely(conn);
    }
    PQclear(res);
    res = PQexec(conn, "FETCH ALL in myportal");
    if (PQresultStatus(res) != PGRES_TUPLES_OK)
    {
        fprintf(stderr, "FETCH ALL failed: %s", PQerrorMessage(conn));
        PQclear(res);
        exit_nicely(conn);
    }
    /* first, print out the attribute names */
    nFields = PQnfields(res);
    for (i = 0; i < nFields; i++)
        printf("%-15s", PQfname(res, i));

    /* next, print out the rows */
```

```
for (i = 0; i < PQntuples(res); i++)
{
    for (j = 0; j < nFields; j++)
        printf("%-15s", PQgetvalue(res, i, j));

}
PQclear(res);
/*Cierra la interfaz de conexi'on*/
res = PQexec(conn, "CLOSE myportal");
PQclear(res);
/* termina la transacci'on */
res = PQexec(conn, "END");
PQclear(res);
/* Cierra la conexi'on con Postgresql*/
PQfinish(conn);
return 0;
}
```

De esta manera tenemos la base para lo que en algún momento podríamos necesitar para establecer una gestión con el DBMS Postgresql.

4.7. Implementación

“Planta Física” se ha desarrollado con cgi’s en el lenguaje C. La entrada al sistema se encuentra alojada en la siguiente dirección de Internet:

<http://www.dgire.unam.mx/y/Entrada.html>

4.8. Validaciones

Las validaciones en el sistema están hechas en dos niveles diferentes, el primero de ellos es por medio de **JavaScript**, que interactúa como un módulo del navegador, en donde se ejecuta la aplicación, cuya función es validar el tipo de dato ingresado en un campo. Esta validación se realiza campo a campo, por lo que cada error en la captura de datos será sancionado con un mensaje que indica cuál es la falla, aunado al borrado de los datos en este lugar.

JavaScript, también nos permite dar funcionalidad a nivel página HTML, de manera que el proceso de captura parezca más dinámico y sin la necesidad de hacer peticiones al servidor.

En otro nivel, una vez que los datos se han enviado, ie, que se ejecuta la adquisición de datos capturados por medio de un CGI, cuya finalidad es insertar dicha información en la base de datos, y realizar una segunda validación. La finalidad y a su vez justificación de esta doble validación resulta del hecho de que es posible deshabilitar el motor de JavaScript en algunos exploradores, con lo cual dejaríamos al descubierto la inserción de la información, que por cierto debe corresponder con la semántica de cada campo.

Las validaciones con JavaScript nos ayudan a reducir el número de envíos erróneos al servidor, de esta manera, eficientar la ejecución y uso de la aplicación.

Las validaciones en C nos permiten, por medio de una biblioteca de manejo de errores, creada expresamente para este fin y con posibilidades de reuso, identificar los campos con datos semánticamente incorrectos. Así mismo, subrayar los campos, a nivel página HTML, que hayan incurrido en dicho suceso. Por tanto el sistema indica al usuario cuál es el campo que capturó de manera incorrecta.

La seguridad, como mencionamos al final del capítulo anterior, depende en

mucho del programador. Así, mientras mejor esté validado nuestro sistema, será más independiente del manejo humano.

4.9. Funcionamiento de los módulos del Sistema

En general el manejo de cada uno de los módulos de Planta Física es muy similar, de los quince módulos que componen el sistema, cuatro de ellos están diseñados de diferente manera.

La descripción funcional de dichos módulos se han realizado a través diagramas de actividad UML (Unified Modeling Language) ⁵ para fines de documentación y mejor entendimiento del sistema.

Entrada al Sistema Planta Física

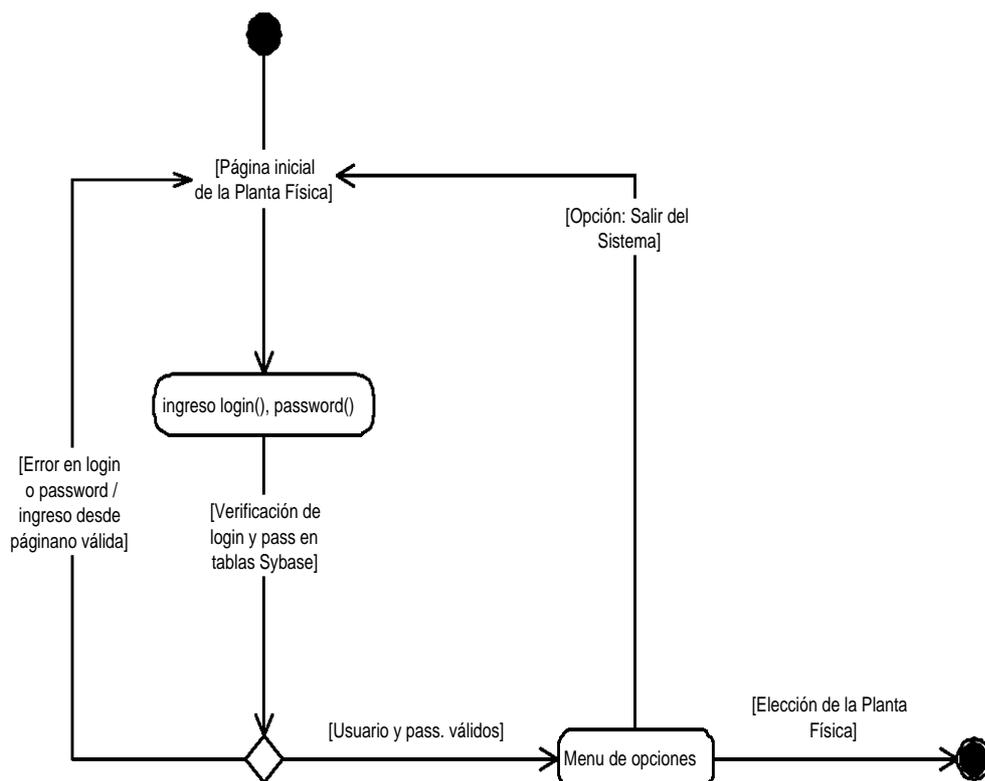


Figura 4.13: Entrada al Sistema.

⁵UML es un lenguaje de modelado visual de propósito general usado para especificar, visualizar, y documentar los componentes de sistemas computacionales

El diagrama del funcionamiento global de Planta Física contiene las opciones que corresponden a los módulos siguientes(fig 4.15).

1. Dirección Técnica
2. Servicios Escolares
3. Aulas de uso Común
4. Aulas de Dibujo
5. Actividades Estética y Artística
6. Laboratorios Nivel Licenciatura
7. Laboratorios de Cómputo
8. Laboratorios de Bachillerato
9. Auditorios y Salas de Usos Múltiples
10. Sanitarios
11. Enfermerías

La información de cada espacio declarado en uno de estos módulos se distingue por un identificador, asignado por el propio usuario, de hasta diez caracteres incluyendo números y/o letras. Así, los atributos correspondientes de cada área registrada de un módulo podrán ser visualizadas de manera única.

Funcionamiento global de Planta Física

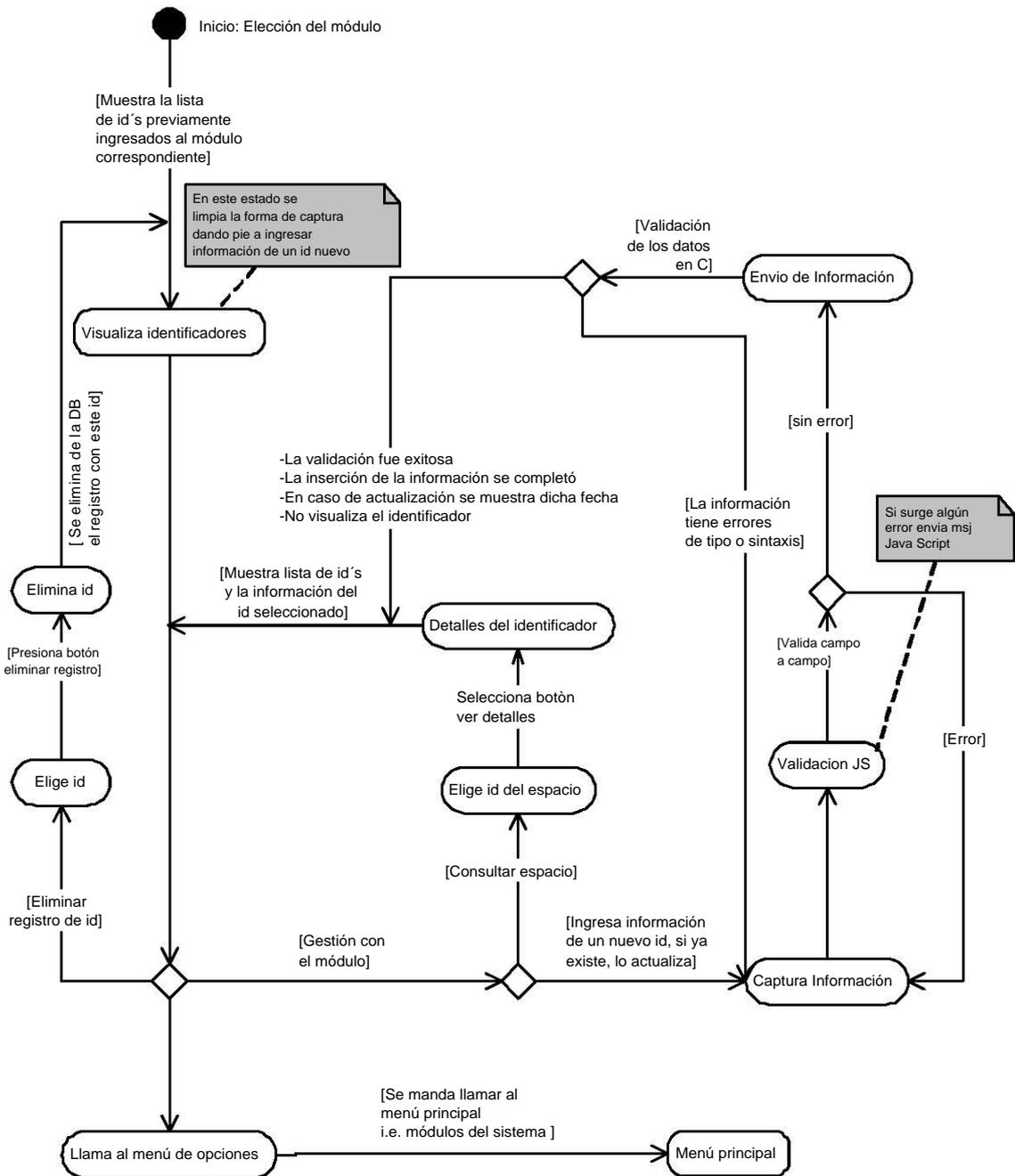


Figura 4.14: Funcionalidad Global.

Como puede verse en las anteriores áreas el comportamiento y resultados que emite el sistema depende necesariamente de un identificador. El área de bibliotecas se pensó para aprovechar dicho identificador de tal manera que este englobara todas las áreas de una biblioteca, la razón de ello podemos verla como una abstracción de cada biblioteca en un objeto que contiene a su vez áreas predeterminadas de manera general, así podremos acceder a cada una de las diferentes áreas de una biblioteca con un mismo identificador y poder tener varias bibliotecas informáticamente organizadas. El siguiente diagrama expresa el funcionamiento del módulo Bibliotecas.

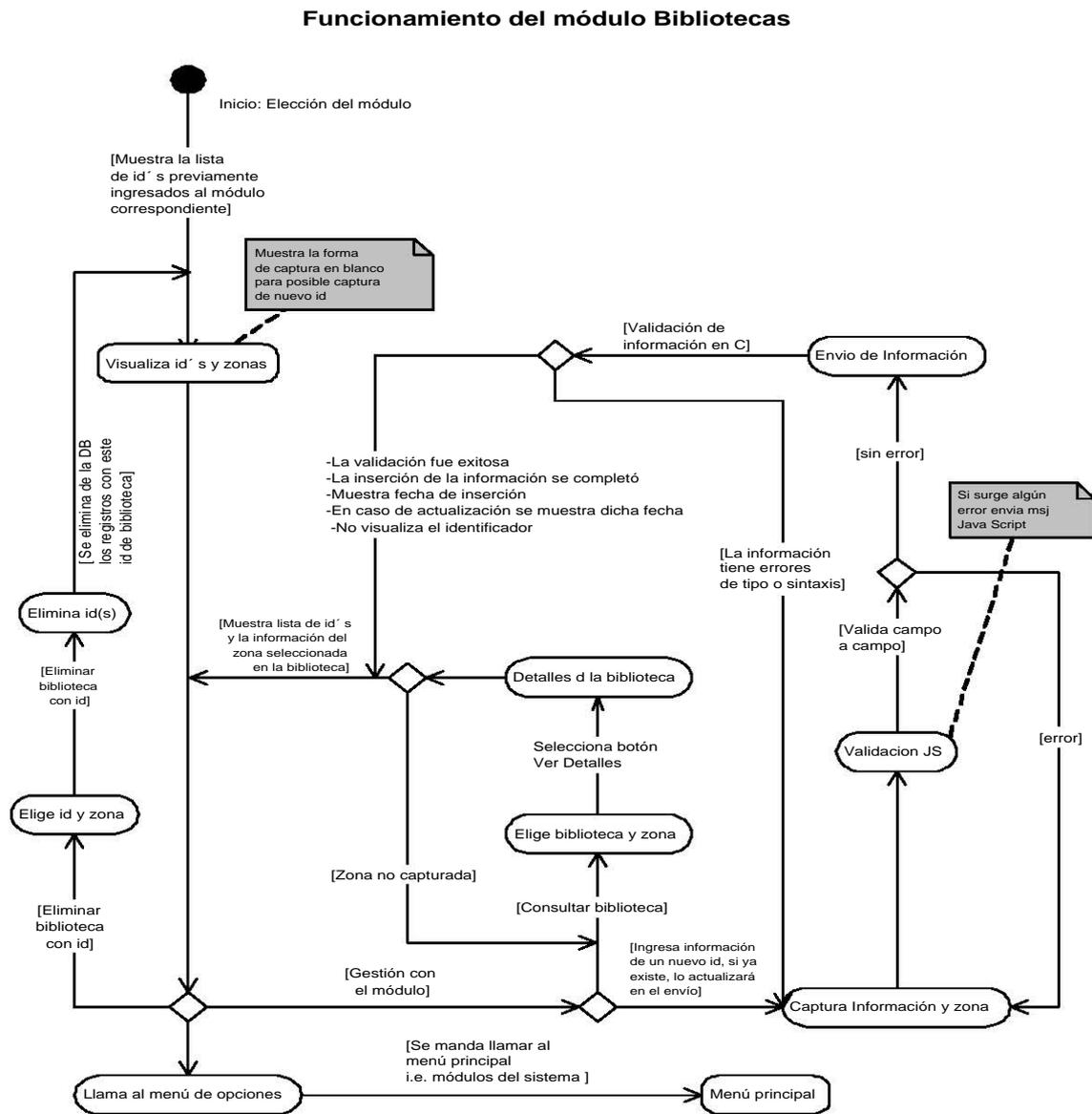


Figura 4.15: Funcionamiento del módulo Bibliotecas

El módulo Otras Áreas se compone a su vez de dos submódulos semánticamente relacionados, sin embargo funcionalmente distintos. El primero de ellos: Áreas Deportivas, se constituye por un manejo de características ingresadas y mostradas por renglones correspondientes. A diferencia de los otros módulos, tal como se puede observar en las imágenes de Áreas Deportivas, en esta parte se pueden visualizar todas las características de todas las áreas deportivas distinguido cada renglón por su correspondiente identificador. De esta misma manera se puede modificar y eliminar

cada renglón. El siguiente diagrama muestra el funcionamiento del submódulo.

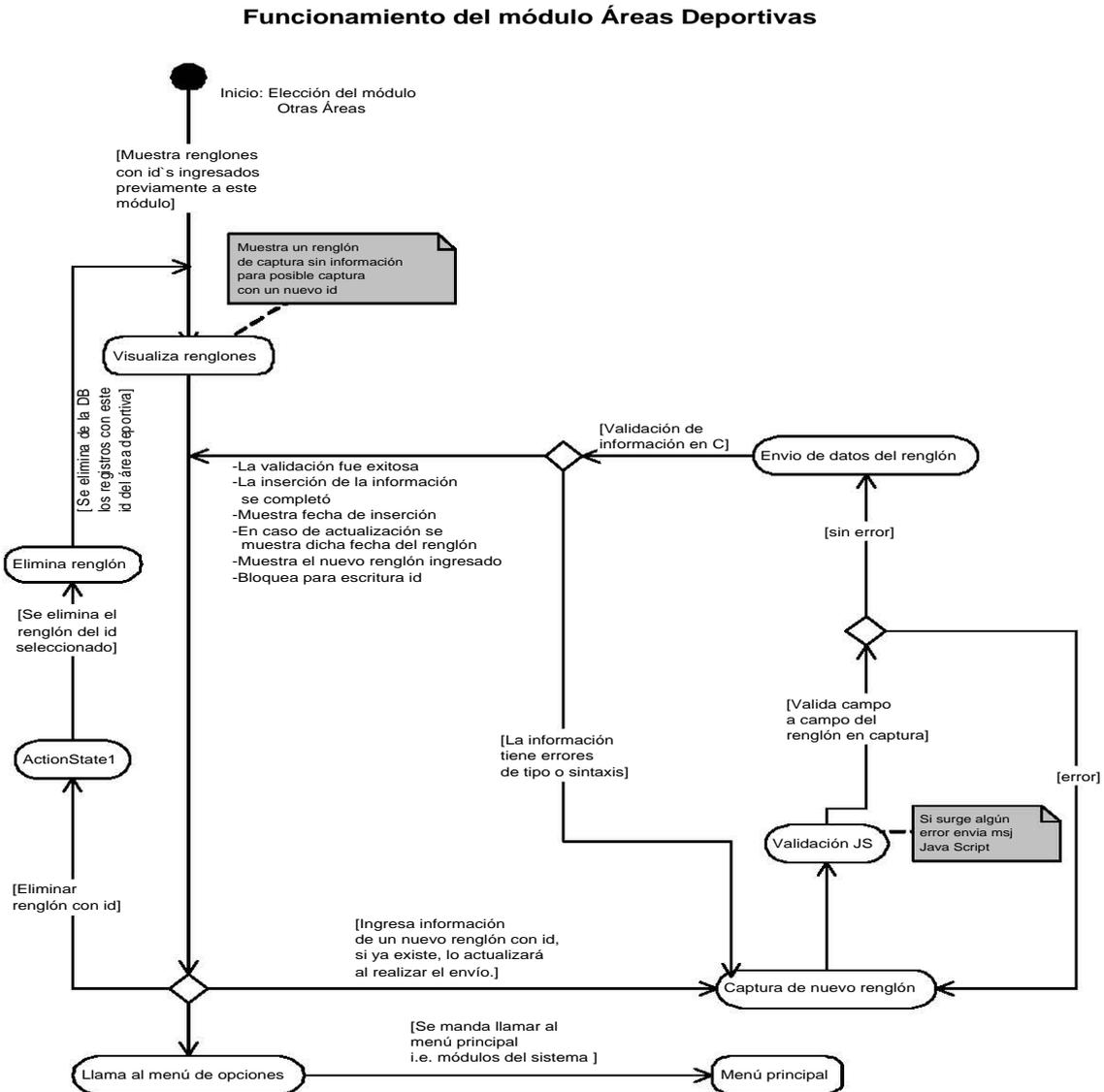


Figura 4.16: Funcionamiento del módulo de Áreas Deportivas.

El submódulo: Área de Esparcimiento y Dispersión, es un conjunto de características agrupadas en un registro único por cada plantel registrado. La razón de ello es que en esta área se expresan como su nombre lo indica todas las características de las áreas como: patios, entradas, salidas, etc. cuyos datos se requieren en medidas globales del plantel, es decir: si existen dos entradas a la sede, los metros cuadrados que conforman éstas se sumarán, dicha será el área de entradas global.

El siguiente módulo a describir es: Extintores, cuya organización y funcionamiento es similar al de Áreas Deportivas con la excepción de que cada renglón estará distinguido no por un identificador sino por el área correspondiente a la localización de el o los extintores, esto por requerir el número global de extintores por área, dicho razonamiento se aplicó pues resultaba ineficiente la distinción cada extintor con un identificadores. El funcionamiento de esta parte se visualiza a continuación.

Por último, la parte de Notas adicionales está enfocada a que de manera opcional cada plantel declare elementos extras, que no son tomados en cuenta para una supervisión. De esta manera se evita habilitar módulos que tengan un uso poco frecuente e innecesario con respecto a los requerimientos de la DGIRE.

Funcionamiento del módulo Esparcimiento y Dispersión

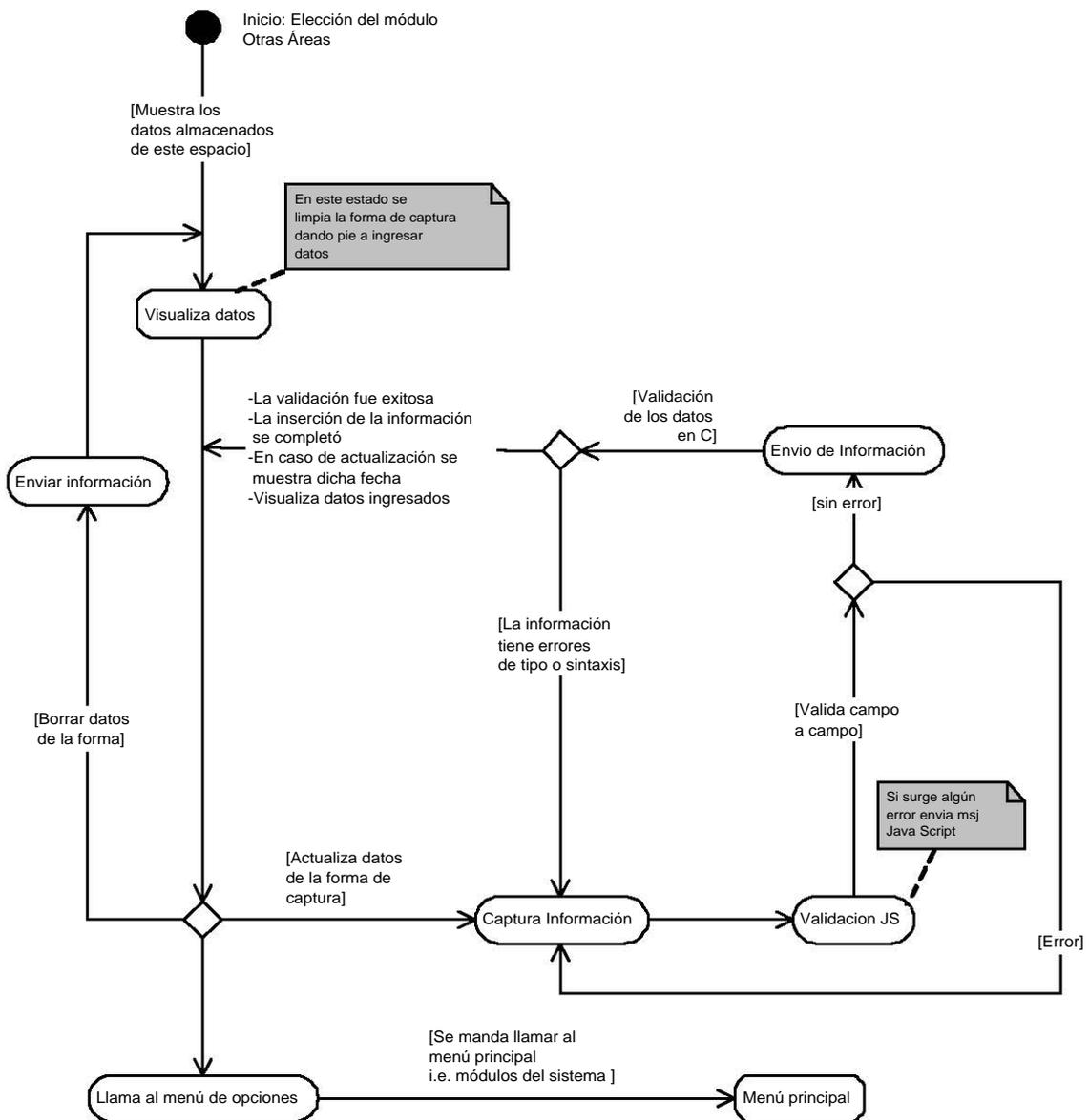


Figura 4.17: Funcionamiento del módulo de Áreas de dispersión

Funcionamiento del módulo Extintores

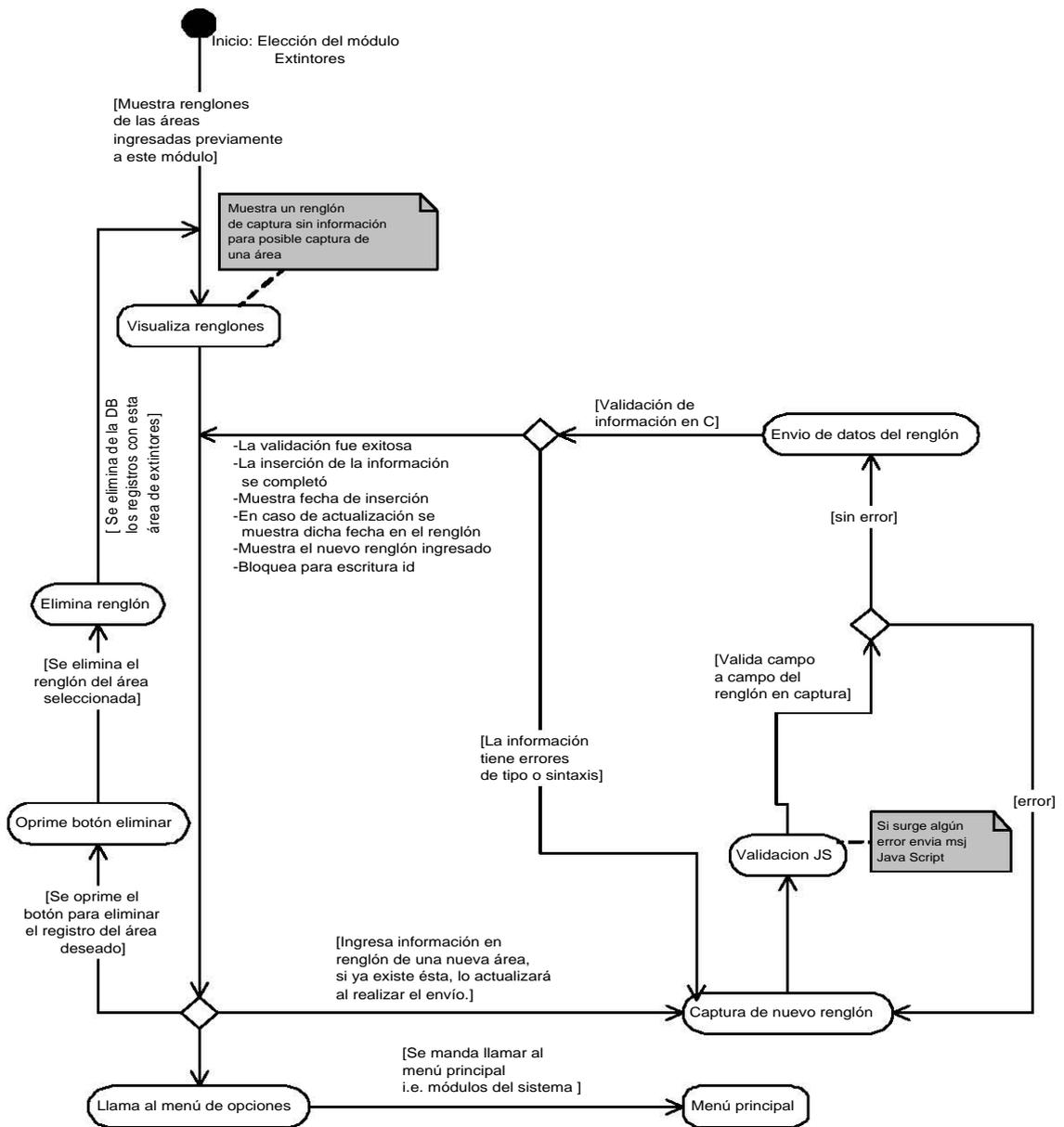


Figura 4.18: Funcionamiento del módulo de Extintores

Funcionamiento del módulo Notas Adicionales

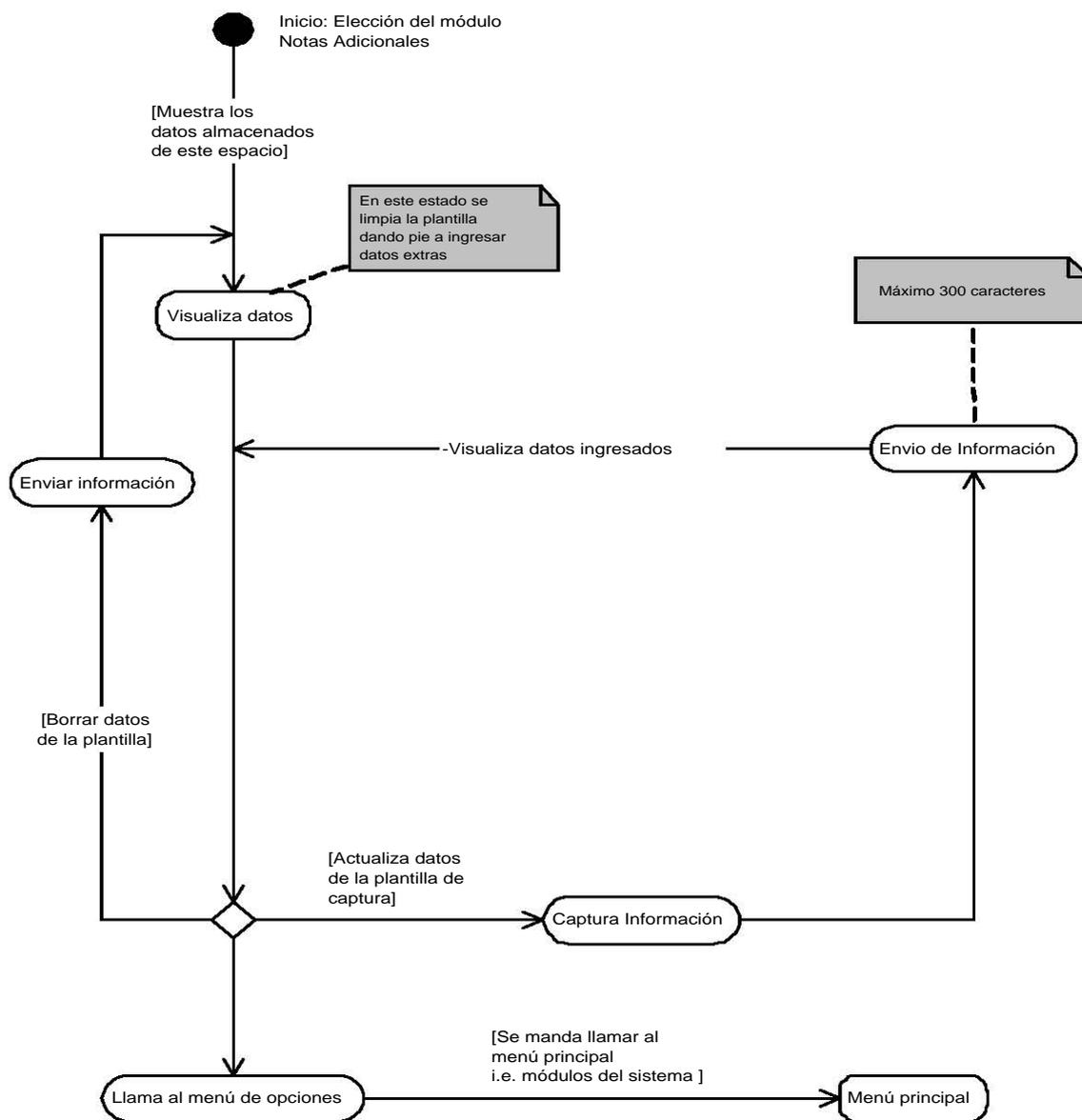


Figura 4.19: Funcionamiento del módulo de Notas Adicionales

Capítulo 5

Conclusiones

El sistema “Planta Física” surge de la necesidad de la Subdirección de Incorporación de la DGIRE de contar con una imagen detallada de cada una de las Instituciones Incorporadas; de manera que, dada una supervisión, ésta se pudiera auxiliar con la información necesaria para su correcta ejecución.

Para el diseño de la base de datos fueron tomados en cuenta todos los elementos necesarios que contribuyen al buen funcionamiento de una institución educativa, como se puede observar en la descripción ya realizada. En el diseño de las interfaces, se consideró que éstas fueran interactivas e intuitivas para los usuarios.

El proceso de desarrollo de software utilizado, fue base para que el sistema cumpliera con las siguientes características:

- La captura de requerimientos fue una fase esencial para que el sistema sea completo en términos de lo solicitado, y así, operar de manera funcional.
- El diseño de la interfaz es amigable; por tanto, el usuario puede entender los términos y funcionalidades de forma simple, proporcionándole así un manejo ágil y dinámico.
- La arquitectura de tres capas y dos niveles proporciona al usuario la accesibilidad solicitada. La centralización de la información en la base de datos de la DGIRE permite una operación con el sistema para los fines internos requeridos.

- En cuanto a la implementación, el software utilizado permitió contar con herramientas necesarias para desarrollar cada una de las funcionalidades del sistema; así también para realizar dicha implementación bajo la arquitectura de tres capas.
- El diseño de las tablas de la base de datos proporciona una idea clara del almacenamiento; no existe ambigüedad en las relaciones, por tanto el sistema arroja datos veraces, así también integra importantes volúmenes de información de módulos distintos sin necesidad de un número excesivo de tablas.

Ya en la puesta en operación se definieron tiempos límite en los que las instituciones cumplieran con la captura de la información. En esos tiempos se dieron todas las posibilidades de asesoría telefónica, por correo electrónico y personales, a fin de aclarar las dudas posibles. La operación del sistema resultó simple, ya que las asesorías fueron acerca de los elementos de Planta Física; es decir, se cumplió con el principio de que la operación del sistema fuera intuitiva para el usuario.

Por ahora, lo importante es la proyección del sistema y el uso de su información. El sistema está siendo utilizado para que las instituciones capturen información y la puedan consultar, en la DGIRE, su uso es exclusivo de consulta. Sin embargo, hace falta incluir esquemas de ayuda como podrían ser:

- Búsquedas.
- Reportes en general y por elementos(consultas no planeadas).

Para este último punto se analizó que podía hacerse uso del lenguaje *Query by Example* adaptándolo al web de manera tal que se muestren todos los campos de las tablas y se establezca una sintaxis de consulta. Así, el usuario podrá elegir la tabla, él o los campos de ella, las condiciones de consulta y relación entre las tablas elegidas. Obviamente ésta es una idea general, para implementarla es necesario un análisis mucho más profundo, tema que está fuera del alcance de esta Tesis.

La información almacenada por “Planta Física”, puede ser utilizada en un esquema donde se pueda combinar con elementos de administración escolar, como puede ser para el registro de grupos, esto es, en la definición de los grupos puede incluirse el

identificador del aula y, de forma automática, calcular la capacidad que tendrá cada grupo.

Podemos imaginar un encuentro de fútbol donde se consulte en la “Planta Física” las diferentes sedes de las instituciones que cuenten con canchas para este deporte y realizar dicho encuentro. Esto sería una aplicación más de “Planta Física”.

El sistema está funcionando y se están cumpliendo los objetivos planteados, es decir, es operado desde Internet, un medio eficiente y muy ágil. Se ha conseguido la descentralización, la desintermediación y la desagregación, entre otros aspectos.

Los resultados favorables de “Planta Física” son una invitación para realizar otras aplicaciones relativas a la Administración Escolar del Sistema Incorporado a la UNAM.

Bibliografía

- [1] Andrew S. Tanenbaum 1993 “**Sistemas Operativos Modernos**”, Prentice Hall, Amsterdam, 1era edición, español.
- [2] DGIRE “**Manual de disposiciones y Procedimientos para instituciones con estudios incorporados a la UNAM**” UNAM, 2004.
- [3] Douglas E. Comer “**Internetworking with TCP/IP, volume I of *Principles, Protocols and Architecture.***” Prentice Hall, 2nd. edition, 1991.
- [4] <http://es.wikipedia.org>.
- [5] <http://www.sybase.com>.
- [6] Ibargüengoitia Guadalupe, Oktaba Hanna “**Ingeniería de Software para Principiantes**”, victoria.fciencias.unam.mx 2005.
- [7] DGIRE Informe de Actividades DGIRE 2005, UNAM, 2005.
- [8] Oetiker Tobias & Partl “**The Not Short Introduction to L^AT_EX**” Tobias Oetiker and All the Contributors to LShort, version 4.14, 2004.
- [9] Ramírez P. Karla “**SISMO**”, Tesis Licenciatura (Licenciado en Ciencias de la Computacion)-UNAM, Facultad de Ciencias, 2002.
- [10] Rumbagh James & Jacobson Ivar & Boch Grady “**The Unified Language Reference Manual**” Addison Wesley, 2nd edition, 1999.
- [11] Suárez M. Olivia, “**INTERNET, INTERNET2 y su Impacto en la Educación a distancia**”, Tesina: Diplomado en Redes de Computadoras y Telecomunicaciones, 2002.

- [12] Sybase, Inc. Open Client DB-Library/C, Reference Manual, 1996.
 - [13] The Common Gateway Interface, <http://hohoo.ncsa.uiuc.edu/cgi/>.
 - [14] Viso G. Elisa, Curso Integral de Introducción a: Unix, X-Windows, \LaTeX , Emacs, Emacs-Lisp e Internet, Vínculos Matemáticos #9, 2001.
-