



Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación.

**“APLICACIÓN DE BASE DE DATOS Y WEB
PARA EL CONTROL DE INVENTARIO
Y DIFUSIÓN DE UN ACUARIO”**

TESIS PROFESIONAL

Para obtener el Título de:

Licenciado en Ciencias de la Computación.

Presenta:

FELIX FLORES FLORES.

Dirigida por:

María del Rocío Boone Rojas.

Puebla, Puebla Otoño 2008

RESUMEN

En la actualidad la mayoría de los acuarios no se cuenta con alguna herramienta con la que facilite el control de su inventario. Lo cual genera a los usuarios responsables en las operaciones de ventas no permitir manejar directamente la información en cuestión y tener desconfianza hacia los datos de inventarios. Además dado el incremento en los productos la administración no tiene un adecuado control en los costos de inventarios, los costos de compras, costos de mermas y la utilidad que se obtiene así como el reporte de algunos productos importantes que se solicitan, entregar la información detallada de reportes en muchas ocasiones es una tarea imposible o incompleta.

En este trabajo de tesis se presenta el desarrollo e implementación de una aplicación de Base de Datos y Web, la cual apoya en las actividades del control de inventario, difusión y promoción de un acuario.

La aplicación de Base de Datos genera reportes sobre, las ventas, compras de productos y actualizaciones en los productos a si como también genera reportes de clientes de mayor concurrencia, y proveedores a los que se les compran más y menos productos, en la página Web se podrá consultar las categorías de productos, servicios, especies y promociones,

Para el desarrollo de esta aplicación se utilizaron herramientas como XAMPP que integra a APACHE, PHP, MySQL y PhpMyAdmin para instalar un servidor local y administrar bases de datos, estas herramientas son gratuitas y ofrecen opciones para desarrollar aplicaciones Web y bases de datos. El resultado de este trabajo es una aplicación con una interfaz fácil de utilizar para el usuario, la cual permite facilitar el control de los inventarios de los productos del acuario a si como sus diferentes servicios y promociones al cliente.

ÍNDICE

Resumen.....	I
Prefacio.....	IV

Capítulo 1.- Fundamentos de Bases de Datos

1.1.-Introducción.....	1
1.2.-Bases de Datos	1
1.3.-Objetivo de los sistemas de Bases de Datos.....	1
1.4.-Estructura de un sistema global.	2
1.5.-Abstracción de datos.	3
1.6- Modelos de datos.	3
1.6.1.-Modelo entidad-relación.....	4
1.6.2.-Modelo orientado a objetos.....	5
1.6.3.-Modelo relacional.	6
1.7.- Aplicaciones de Base de Datos en Web.	7
1.8.-¿Qué es una aplicación Web?	7
1.9.-Integración de Base de datos en el Web.....	7
1.10.- Categorización de Interfases WEB/DBMS.....	8
1.11.- Tecnologías para la integración de base de datos en el Web.....	9
1.11.1.-El Common Gateway Interface (CGI).....	9
1.11.2- Interfaz de Programación de Aplicaciones (API).....	10
1.11.3.-Interfaz de Programación de Aplicaciones del Servidor de Internet (ISAPI).....	10

Capítulo 2.- Proceso Unificado de Desarrollo de Software PUDS y UML

2.1.-¿Que es PUDS ?	11
2.2.-Proceso Unificado dirigido por casos de uso.	11
2.3.-Proceso de desarrollo centrado en la arquitectura.....	12
2.4.-Proceso Unificado es iterativo e incremental.	13
2.5.-Vida del Proceso Unificado.....	13
2.6.-¿Que es UML ?.....	14
2.7.-Áreas de trabajo en UML.....	15
2.8.-Proceso Unificado (Unified Process, UP).....	15
2.9.-Descripción de los diagramas de UML.....	17

Capítulo 3.- Análisis del Sistema

3.1.-Planteamiento del problema.....	27
3.2.-Objetivos generales.....	27
3.3.-Objetivos particulares.....	27
3.4.-Descripción de la información.....	28
3.5.-Restricciones.	28
3.6.-Diagramas de Casos de Uso.....	29
3.7.-Diagramas de Clases	33
3.8.-Diagrama de Secuencia	33

Capítulo 4.- Diseño de la Base de Datos

4.1.-Diagrama Entidad-Relación.....	37
4.2.-Diccionario de Datos.....	38
4.3.-Modelo Relacional.....	42
4.4.-Normalización.....	43
4.4.1.-Definición.....	43
4.4.2.-Formas Normales.....	43
4.4.3.-Normalización de la base de datos.....	44

Capítulo 5.- Implementación y Pruebas

5.1.-MySQL.....	48
5.2.-¿Qué es PHP ?.....	48
5.2.1.-Ventajas de PHP.....	49
5.3.-XAMPP.....	49
5.3.1.-Instalación de XAMPP.....	50
5.3.2.-Implementación De la Base De Datos.....	50
5.4.-Implementación De La Interfaz (PHP).....	52
5.4.1.-Página de Inicio.	52
5.4.2.-Interfaz de empleado.....	54
5.4.3.-Página de Inicio de Administrador.....	55

Conclusiones	61
--------------------	----

Bibliografía	62
--------------------	----

PREFACIO

El objetivo de esta tesis es desarrollar una aplicación de Base de Datos y web para apoyar las actividades de control de Inventarios, Difusión y Promoción de un Acuario cuyo desarrollo esta basado en el proceso unificado de desarrollo de software (Puds), Usando UML(The Unified Modeling Language) para visualizar, especificar, construir y documentar los artefactos del sistema de software. El modelo cliente-servidor, Mysql como administrador de la Base de Datos y Php como lenguaje de desarrollo.

El sistema generará reportes sobre, altas, bajas, consultas de productos y actualizaciones en los productos, el sistema apoyará en la toma de decisiones generando un reporte por ejemplo de la compra de algunos productos de temporada, o de mayor desplazamiento, y en dar a conocer una lista en la que se conozca sobre cuales son los proveedores a los que se les compran mas o menos productos.

Para propósito de divulgación el acuario ofrecerá una página Web por medio de la cual se publicarán las especies, los productos, servicios y las promociones disponibles.

A continuación se realizara un pequeño resumen por capítulo de esta tesis:

Capítulo 1. Se presenta los fundamentos de Bases de Datos, introducción, definición y objetivos, así como también las aplicaciones de Bases de Datos en Web.

Capítulo 2. En esta sección, trataremos el Proceso Unificado de Desarrollo de Software (Puds), y definición de UML(The Unified Modeling Language) así como la descripción de diagramas UML.

Capítulo 3. En esta sección trataremos lo correspondiente al análisis del sistema y la descripción de la información para la especificación de requisitos necesarios para la base de datos así como la descripción de los diagramas UML de la aplicación.

Capítulo 4. En este capítulo se mostraran los diagramas entidad-relación y las normalizaciones de las tablas así como también los pasos del diseño.

Capítulo 5. En este capítulo se realizará la descripción de una página ligada a una base de datos así como también, las pruebas necesarias del sistema, la descripción de las ventajas que ofrece PHP y MYSQL y la instalación de XAMPP como servidor local.

ÍNDICE DE FIGURAS Y TABLAS

Capítulo 1

Figura 1. Estructura general de un SGBD.....	3
Figura 1.1 Símbolos utilizados para representar un diagrama E-R.....	4
Figura 1.2 Ejemplo del modelo entidad-relación.....	5
Figura 1.3 Ejemplo de la clase cuenta.....	6
Figura 1.4 Ejemplo del modelo relacional.....	6

Capítulo 2

Figura 2. Modelo del Proceso Unificado.....	14
Figura 2.1. RUP.....	15
Figura 2.2 Fases en el Proceso Unificado.....	16
Figura 2.3 Diagrama de casos de uso.....	17
Figura 2.4 Relaciones de casos de uso.....	18
Figura 2.5 Asociación de Clases.....	19
Figura 2.6 Multiplicidad de Clases.....	19
Figura 2.7 Asociación Tripartita de Clases.....	20
Figura 2.8 Composición de Clases.....	20
Figura 2.9 Agregación de Clases.....	20
Figura 2.10 Generalización.....	21
Figura 2.11 Diagrama de Clases de una maquina de café.....	21
Figura 2.12 Diagrama de Objetos.....	22
Figura 2.13 Diagrama de Objetos con Atributos.....	22
Figura 2.14 Maquina de Estados, estados simples.....	23
Figura 2.15 Diagrama de Secuencia.....	23
Figura 2.16 Diagrama de Colaboración.....	24
Figura 2.17 Diagrama de Componentes.....	24
Figura 2.18 Diagrama de Actividades.....	25
Figura 2.19 Diagrama de Despliegue.....	26

Capítulo 3

Figura 3.1 Diagrama de Casos de uso para Administrador, Empleado del Acuario y Consultor Web.....	29
Figura 3.2 Diagrama de Casos de Uso para el Administrador.....	30
Figura 3.3 Diagrama de Casos de Uso para el Empleado de Acuario.....	30
Figura 3.4 Diagrama de Casos de Uso para el Consultor Web.....	31
Figura 3.5 Diagrama de Clases.....	33
Figura 3.6 Diagrama de Secuencia de alta de Productos para Administrador.....	34
Figura 3.7 Diagrama de Secuencia de baja de Productos para Administrador.....	34
Figura 3.8 Diagrama de Secuencia de Consulta de Productos para Administrador.....	35
Figura 3.9 Diagrama de Secuencia de Modificación de Servicios para Administrador.....	35

Capítulo 4

Figura 4.1 Diagrama Entidad-Relación.....	38
Figura 4.2 Muestra la tabla Acuario.....	44
Figura 4.3 Muestra la tabla Servicios.....	44
Figura 4.4 Muestra la tabla Clientes.....	45
Figura 4.5 Muestra la tabla Proveedores.....	45
Figura 4.6 Muestra la tabla Productos.....	45
Figura 4.7 Muestra la tabla Promociones.....	46
Figura 4.8 Muestra la tabla Especies.....	46
Figura 4.9 Muestra la tabla Peceras.....	46
Figura 4.10 Muestra la tabla Usuario.....	46
Figura 4.11 Muestra la tabla Poseen.....	47
Figura 4.12 Muestra la tabla Trabaja_en.....	47

Capítulo 5

Figura 5.1 Creación de una nueva base de datos desde phpMyAdmin.....	50
Figura 5.2 Creación de una tabla.....	51
Figura 5.3 Propiedades de entidad.....	51
Figura 5.4 Página de Inicio.....	52
Figura 5.5 Login y Contraseña.....	53
Figura 5.6 Pagina de inicio de empleado.....	54
Figura 5.7 Pagina de inicio de administrador.....	55
Figura 5.8 Registro de Productos.....	56
Figura 5.9 Búsqueda del dato a Modificar.....	59
Figura 5.10 Borrado de tuplas.....	60
Figura 5.11 Actualización de datos.....	60

Capítulo 1

Fundamentos de Bases de Datos

1.1.- Introducción.

Los sistemas de información existen desde las primeras civilizaciones. El concepto más esencial de sistema de información no ha variado desde los censos romanos, por poner un ejemplo. Los datos se recopilaban, se estructuraban, se centralizaban y se almacenaban convenientemente. El objetivo inmediato de este proceso era poder recuperar estos mismos datos u otros datos derivados de ellos en cualquier momento, sin necesidad de volverlos a recopilar, proceso que solía ser el más costoso o incluso irrepetible. El objetivo posterior de un sistema de información, no obstante, era proporcionar a los usuarios información fidedigna sobre el dominio que representaban, con el objetivo de tomar decisiones y realizar acciones más pertinentes que las que se realizarían sin dicha información.

Llamamos base de datos justamente a esta colección de datos recopilados y estructurados que existen durante un periodo de tiempo. Por ejemplo, un libro contable, debido a su estructura, se puede considerar una base de datos. Generalmente, un sistema de información consta de una o más base de datos, junto con los medios para almacenarlas y gestionarlas, sus usuarios y sus administradores.

Hoy en día, la administración de base de datos es completamente automatizada, mediante la utilización de aplicaciones específicas denominadas sistemas de gestión de base de datos (SGBD).

1.2.-Bases de Datos

Una base de datos es un conjunto de datos almacenados entre los que existen relaciones lógicas y ha sido diseñada para satisfacer los requerimientos de información de una empresa y organización. Los sistemas de base de datos se diseñan para gestionar grandes cantidades de información [1].

La base de datos es un gran almacén de información que se define una sola vez y que se utiliza al mismo tiempo por muchos usuarios. En lugar de trabajar con ficheros desconectados e información redundante, todos los datos se integran con una mínima cantidad de duplicidad. Además la base de datos no solo contiene de cómo esta organizada, también almacena una descripción de dichos datos. Esta descripción es lo que se denomina metadatos, se almacena en el diccionario de datos y es lo que permite que exista independencia de datos lógica-física.

1.3.-Objetivo de los sistemas de Bases de Datos.

Los sistemas de base de datos se diseñan para manejar grandes cantidades de información, la manipulación de los datos involucra tanto la definición de estructuras para el almacenamiento de la información como la provisión de mecanismos para la manipulación de la información, además un sistema de base de datos debe de tener implementados mecanismos de seguridad que

garanticen la integridad de la información, a pesar de caídas al sistema o de intentos de accesos no autorizados [2].

Un objetivo principal de un sistema de base de datos es proporcionar a los usuarios finales una visión abstracta de los datos, esto se logra escondiendo ciertos detalles de cómo se almacenan y mantienen los datos [2].

Los objetivos principales de un sistema de base de datos es disminuir los siguientes aspectos:

1. Redundancia e inconsistencia de los datos.
2. Dificultad para tener acceso a los datos.
3. Aislamiento de los datos.
4. Anomalías del acceso concurrente.
5. Problemas de seguridad.
6. Problemas de integridad.

1.4.-Estructura de un sistema global.

Un DBMS se divide en módulos que tratan cada una de las responsabilidades del sistema general. En la mayoría de los casos, el sistema operativo proporciona únicamente los servicios básicos, y el sistema debe partir de esa base. Así, el diseño de un DBMS debe incluir la consideración del interfaz entre la B.D. y el sistema operativo [2].

Los componentes funcionales de un DBMS incluyen:

- Gestor de archivos. Gestiona la asignación de espacio en la memoria del disco y de las estructuras de datos usadas para representar la información almacenada en el disco.
- Gestor de base de datos. Proporciona el interfaz entre los datos de bajo nivel almacenados y los programas de aplicación y las consultas al sistema.
- Procesador de consultas. Traduce sentencias en un lenguaje de consultas a instrucciones de bajo nivel que entiende el gestor de la BD.
- Precompilador de DML. Convierte las sentencias DML. Incorporadas en un programa de aplicación en llamadas normales a procedimientos del lenguaje principal. Debe interaccionar con el procesador del consultas para generar código apropiado.
- Compilador de DDL. Convierte sentencias DDL en un conjunto de tablas que contienen metadatos. Además se requiere varias estructuras de datos como parte de la implementación del sistema físico, como:
- Archivos de datos. Para almacenar la BD.
- Diccionario de datos. Que almacena metadatos sobre la estructura de la BD. Se usa ampliamente, por lo que debe ponerse mucho énfasis en el desarrollo de un buen diseño y

Una implementación eficiente del diccionario. La figura 1. Muestra la estructura general de un SGBD

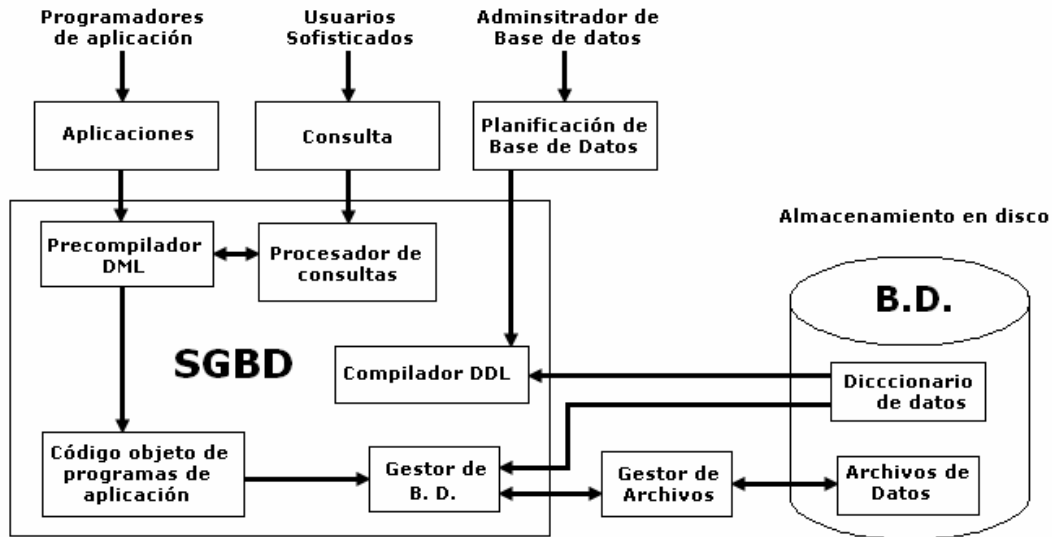


Figura 1. Estructura general de un SGBD

1.5.-Abstracción de datos.

Un SGBD (Sistema Gestor de Base de Datos) es una colección de archivos interrelacionados y un conjunto de programas que permiten a los usuarios acceder y modificar estos archivos. [1].

Uno de sus objetivos más importantes es proporcionar a los usuarios una visión abstracta de los datos, es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos, pero sin embargo se deben extraer eficientemente. Este requerimiento ha llevado al diseño de estructuras de datos complejas para la representación de datos en la base de datos. Se esconde esta complejidad a través de distintos niveles de abstracción para simplificar la interacción con el sistema [2].

1.6.-Modelos de datos

Para describir la estructura de una base de datos es necesario definir el concepto de modelo de datos. Un modelo de datos es un colección de herramientas conceptuales para describir datos, relacionales entre ellos, semántica asociada a los datos y restricciones de consistencia [1].

- Modelos lógicos basados en objetos. Utilizados para describir datos en el nivel conceptual y de visión. Se caracterizan por proporcionar una capacidad de estructuración flexible y permiten especificar restricciones de datos explícitamente (ejemplo: modelo entidad-relación, modelo orientado a objetos [2]).

- Modelos lógicos basados en registros. Utilizados para describir datos en el nivel conceptual y físico, se utiliza para especificar la estructura lógica de la base de datos. Estos modelos se llaman así por que la base de datos esta estructurada en registros de formato fijo de varios tipos [2].

1.6.1.-Modelo entidad-relación

El modelo de datos entidad-relación (E-R) se basa en una percepción de un mundo real que consiste en una colección de objetos básicos llamados entidades, y relaciones entre esos objetos. Una entidad es una “cosa” u “objeto” del mundo real que es distinguible de otros objetos por medio de un conjunto de atributos. Por ejemplo, los atributos número y saldo describen una cuenta particular [1].

Una relación es una asociación entre varias entidades. Por ejemplo, una relación ‘Cuenta Cliente asocia a un cliente con cada cuenta que posee (Fig 1.2). El conjunto de todas las entidades del mismo tipo y relaciones del mismo tipo se denomina conjunto de entidades y conjunto de relaciones. Además el modelo E-R representa ciertas restricciones a las que deben ajustarse los contenidos de una base de datos. Una restricción importante es la cardinalidad de asignación que expresa el número de entidades a las que se puede asociarse otra entidad mediante un conjunto de relación [1].

En la figura 1.1. se presenta los símbolos utilizados para representar un diagrama E-R.

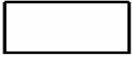



SÍMBOLO	SIGNIFICADO
	Se utiliza para representar una entidad.
	Se utiliza para representar un atributo.
	Se utiliza para representar las relaciones entre entidades
	Se utilizan para conectar atributos con entidades y entidades con relaciones.
Nota: Cada componente se etiqueta con el nombre de lo que representa.	

Figura 1.1 Símbolos utilizados para representar un diagrama E-R.

La estructura lógica global de una base de datos puede expresarse gráficamente por un diagrama E-R, como se muestra en la figura 1.2.

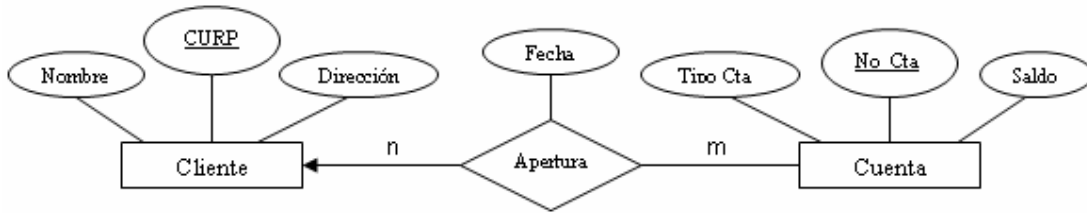


Figura. 1.2 Ejemplo del modelo entidad-relación.

1.6.2.-Modelo orientado a objetos.

Al igual que el modelo E-R, el modelo orientado a objetos se basa en una colección de objetos. Un objeto contiene valores acumulados en variables dentro de él, y estos valores son objetos por sí mismos. Así, los objetos contienen objetos a un nivel de anidamiento arbitrario. Un objeto también contiene partes de código que operan sobre el objeto, que se denominan métodos [2].

Los objetos que contienen los mismos tipos de valores y los mismos métodos se agrupan en clases. Una clase puede ser vista como una definición de tipo para los objetos. La única forma en la que un objeto puede acceder a los datos de otro objeto es invocando a un método de este otro objeto. Esto se llama envío de un mensaje al objeto. Así, la interfaz de llamada de los métodos de un objeto define su parte visible externamente, la parte interna del objeto (Las variables y el código de método) no son visibles externamente. El resultado es dos niveles de abstracción de datos [2].

A diferencia de las entidades en el modelo E-R, cada objeto tiene su propia identidad única independientemente de los valores que contiene. Así dos objetos que contienen los mismos

Valores son distintos. La distinción entre objetos individuales se mantiene en el nivel físico por medio de identificadores de objeto.

La figura 1.3 muestra un ejemplo de la clase cuenta.

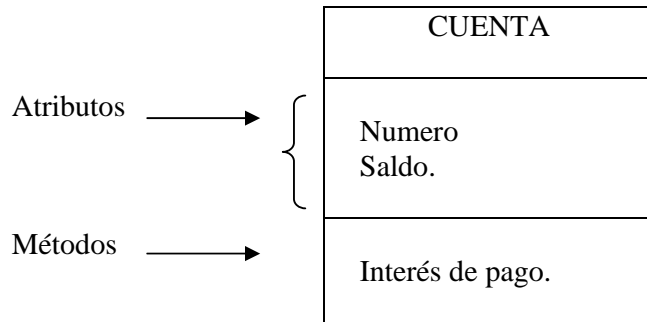


Figura 1.3 Ejemplo de la clase cuenta.

1.6.3.-Modelo relacional.

En el modelo relacional se usa una colección de tablas para representar tanto los datos como sus relaciones. En la figura 1.4. Se representa un ejemplo de una Base de Datos Relacional. Cada tabla tiene varias columnas, y cada columna tiene un nombre único, es un modelo basado en registros ya que la base de datos se estructura en registros que son de formato fijo y de varios tipos, en cada tabla contiene registros de un tipo dado [2].

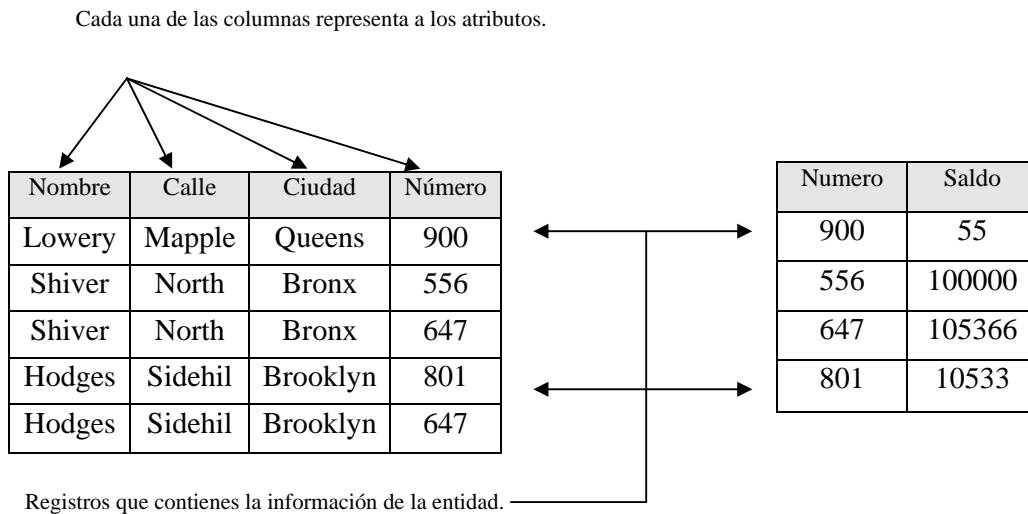


Figura 1.4 Ejemplo del modelo relacional.

1.7.- Aplicaciones de Base de Datos en Web.

1.8.- ¿Qué es una aplicación Web?

El término aplicación Web ha ido evolucionando desde un pequeño sitio Web a una robusta y entera aplicación. Ahora ya no es raro entender por una aplicación Web a cientos de usuarios simultáneos, distribuidos alrededor del mundo, todos conectados a la vez si se requiere. Este término varía de acuerdo a las personas, algunos creen que una aplicación Web es cualquier página que use java, otros consideran cualquier sistema que use un servidor Web. Aquí, una aplicación Web, será un servidor Web en el cual los usuarios introducen datos que afectan la condición del negocio. La diferencia entre una aplicación Web y un sitio Web radica en su uso. Una aplicación Web implementa la lógica del negocio, y usa cambios de estados del negocio [3].

Así las aplicaciones Web son sistemas de información donde una gran cantidad de datos volátiles, altamente estructurados, son consultados, procesados y actualizados mediante navegadores [7].

El diseño de su interfaz está condicionado por las necesidades de claridad y simplicidad. Debe tener una estructura que oriente a cada tipo de usuario en función de sus necesidades [7].

1.9.- Integración de Base de datos en el Web.

En la actualidad, muchas instituciones se han dado cuenta de la importancia que el Web tiene en el desarrollo de sus potencialidades, ya que con ello pueden lograr una mejor comunicación con personas o instituciones situadas en cualquier lugar del mundo. Gracias a la conexión con la red mundial Internet, poco a poco, cada individuo o institución va teniendo acceso a mayor cantidad de información de las diversas ramas de la ciencia con distintos formatos de almacenamiento. La mayor parte de información es presentada de forma estática a través de documentos HTML, lo cual limita el acceso a los distintos tipos de almacenamiento en que ésta pueda encontrarse [5].

Sin embargo, en la actualidad surge la posibilidad de utilizar aplicaciones que permitan acceder a información de forma dinámica, tal como a bases de datos, con contenidos y formatos muy diversos. Una de las ventajas de utilizar el Web para este fin, es que no hay restricciones en el sistema operativo que se debe usar, permitiendo la conexión entre sí, de las páginas Web desplegadas en un browser del Web que funciona en una plataforma, con servidores de bases de datos alojados en otra plataforma. Además, no hay necesidad de cambiar el formato o estructura de la información dentro de las bases de datos.

Para realizar una requisición de acceso desde el Web hasta una base de datos no sólo se necesita de un browser del Web y de un Servidor Web, sino también de un software de procesamiento (aplicación CGI), el cual es el programa que es llamado directamente desde un documento HTML en el cliente. Dicho programa lee la entrada de datos desde que provienen del cliente y toma cierta información de variables de ambiente. El método usado para el paso de datos está determinado por la llamada CGI [5].

Una vez se reciben los datos de entrada (sentencias SQL o piezas de ellas), el software de procesamiento los prepara para enviarlos a la interfaz en forma de SQL, y luego ésta procesa los resultados que se extraen de la base de datos.

La interfaz contiene las especificaciones de la base de datos necesarias para traducir las solicitudes enviadas desde el cliente, a un formato que sea reconocido por dicha base. Además, contiene toda la información, estructuras, variables y llamadas a funciones, necesarias para comunicarse con la base de datos [5].

El software de acceso usualmente es el software distribuido con la base de datos, el cual permite el acceso a la misma, a través de solicitudes con formato. Luego, el software de acceso recibe los resultados de la base de datos, aún los mensajes de error, y los pasa hacia la interfaz, y ésta a su vez, los pasa hasta el software de procesamiento.

Cualquier otro software (servidor HTTP, software de redes, etc.) agrega enlaces adicionales a este proceso de extracción de la información, ya que el software de procesamiento pasa los resultados hacia el servidor Web, y éste hasta el browser del Web (ya sea directamente o a través de una red) [5].

1.10.- Categorización de Interfaces WEB/DBMS.

Tradicionalmente en el Web se han utilizado documentos HTML estáticos para los cuales se creaban las posibles respuestas ante requisiciones del cliente. Este método requiere de un gran desarrollo de aplicaciones y de mantenimiento de las mismas. Al interactuar con las bases de datos, este proceso se complica aún más [5].

Como la necesidad de acceder a bases de datos desde el Web se ha incrementado, han sido creadas también interfaces que manipulan sus escritos para procesar la información, teniendo como punto común la ejecución de sentencias SQL para requerir datos a la base.

Ya han surgido aplicaciones de interfaz para la interacción de bases de datos con el Web. Los productos iniciales son simplemente modelos del ambiente cliente/servidor, con una capa adicional para crear resultados HTML que pueden ser vistos a través del Web, por medio de un procesamiento de los datos de la forma introducida por el cliente. Además, al usar estas interfaces se puede crear el programa principal de la aplicación. Como puede observarse, estas herramientas permiten construir poderosas aplicaciones en el Web, pero se requiere que programadores experimentados logren un desarrollo a gran escala. También, el mantenimiento de las mismas es significativamente más complejo y extenso [5].

Una de las estrategias más famosas para la creación de aplicaciones de interacción con el Web, es la de descargar del Web, aplicaciones o componentes funcionales que se ejecutarán dentro del browser. Con ellas se realizará un procesamiento complejo del lado del cliente, lo cual requiere un gran esfuerzo para crear las piezas de la aplicación. Estas estrategias poseen dos características principales: garantizan la seguridad tanto en los sistemas de distribución como en la comunicación que se establece con tales aplicaciones, a través de Internet.

También han aparecido bibliotecas que incluyen motores propios de servidor que corren de forma conjunta con el Servidor Web, lo cual facilita el desarrollo de nuevas aplicaciones.

Una aplicación que posibilita interconectar al Web con una base de datos tiene muchas ventajas, además de que las funciones que cumplen actualmente los Servidores Web y las herramientas de desarrollo de aplicaciones Web, hacen más fácil que nunca la construcción de aplicaciones más robustas. Tal vez el mayor beneficio del desarrollo de estas aplicaciones en el Web sea la habilidad de que sean para múltiples plataformas, sin el costo de distribuir múltiples versiones del software.

Cada una de las interfaces para comunicar al Web con bases de datos, ha sido creada basándose en una tecnología de integración especial, a través de procesos de interconexión especiales, que serán descritos en el siguiente apartado [5].

1.11.- Tecnologías para la integración de base de datos en el Web.

- Common Gateway Interfase (CGI).
- Interfaz de Programación de aplicaciones (API).
- Interfaz del Programador de Aplicaciones del Servidor de Internet (ISAPI).
- Java, JDBC y JavaScript.

Cuando se utiliza una interfaz para lograr la integración del Web con cierta base de datos, se puede verificar que los procesos seguidos varían, dependiendo de la tecnología que se esté utilizando [5].

1.11.1. - El Common Gateway Interface (CGI).

CGI es actualmente la solución que más se está utilizando para la creación de interfaces Web/DBMS. Fue probada por primera vez en el servidor NCSA.

Se ha comprobado que si el Servidor Web recibe un URL con una llave, para devolver un documento HTML como respuesta, tendrá que cargar el servicio (programa) que le indique las variables de ambiente y de la forma HTML. La mayoría de las veces dicha llave es el "cgi-bin". Entre las ventajas de la programación CGI, se tiene su sencillez, ya que es muy fácil de entender, además de ser un lenguaje de programación independiente, ya que los escritos CGI pueden elaborarse en varios lenguajes.

También es un estándar para usarse en todos los servidores Web, y funcionar bajo una arquitectura independiente, ya que ha sido creado para trabajar con cualquier arquitectura de servidor Web. Como la aplicación CGI se encuentra funcionando de forma independiente, no pone en peligro al servidor, en cuanto al cumplimiento de todas las tareas que éste se encuentre realizando, o al acceso del estado interno del mismo [5].

Pero el CGI presenta cierta desventaja en su eficiencia, debido a que el Servidor Web tiene que cargar el programa CGI y conectar y desconectar con la base de datos cada vez que se recibe

Una requisición. Además, no existe un registro de el estado del servidor, sino que todo hay que hacerlo manualmente.

1.11.2.-Interfaz de Programación de Aplicaciones (API).

Es un conjunto de rutinas, protocolos y herramientas para construir aplicaciones de interfaz. Una buena API hace más fácil el trabajo de desarrollo de un programa, ya que debe proveer todos los bloques para construirlo. El programador lo único que hace es poner todos los bloques juntos. API está diseñado especialmente para los programadores, ya que garantiza que todos los programas que utilizan API, tendrán interfaces similares. A si mismo, esto le facilita al usuario aprender la lógica de nuevos programas [5].

Cuando se realiza una requisición, el servidor llamará al API, brindando la ventaja de disponer de una mayor cantidad de servicios.

1.11.3.-Interfaz de Programación de Aplicaciones del Servidor de Internet (ISAPI).

Es la interfaz propuesta por Microsoft como una alternativa más rápida que el CGI, y ya está incluida en el Servidor Microsoft Internet Información (IIS). Así como los escritos CGI, los programas escritos usando ISAPI habilitan un usuario remoto para ejecutar un programa, busca información dentro de una base de datos, o intercambia información con otro software localizado en el servidor. Los programas escritos usando la interfaz ISAPI son compilados como bibliotecas de enlace dinámico (DLL - Dynamic Link Library), ya que son cargados por el servidor Web cuando éste se inicia. Dichos programas se vuelven residentes en memoria, por lo que se ejecutan mucho más rápido que las aplicaciones CGI, debido a que requieren menos tiempo de uso de CPU al no iniciar procesos separados [5].

Uno de los programas ISAPI más usados es el HTTPODBC.DLL que se usa para enviar y/o devolver información hacia y desde las bases de datos, a través de ODBC.

Además, ISAPI permite realizar un procesamiento previo de la solicitud y uno posterior de la respuesta, con lo cual manipula la solicitud/respuesta HTTP. Los filtros ISAPI pueden utilizarse para aplicaciones tales como autenticación, acceso o apertura de sesión.

Capítulo 2

Proceso Unificado de Desarrollo De Software PUDS y UML

2.1.-¿Que es PUDS ?

El proceso unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. El proceso unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto [8].

El Proceso Unificado esta basado en componentes, lo cual quiere decir que el sistema software en construcción esta formado por componentes software interconectados a través de interfaces bien definidas.

El Proceso Unificado utiliza el Lenguaje de Modelado (UML) para preparar todos los esquemas de un sistema software, de hecho, UML es una parte esencial del Proceso Unificado puesto que sus desarrollos fueron a la par [8].

2.2.-Proceso Unificado dirigido por casos de uso.

Para construir un sistema software con éxito debemos conocer lo que sus futuros usuarios necesitan y desean, con tal de proporcionar el servicio. Cuando nos referimos al termino usuario, no solo se hace referencia a los humanos, sino también puede referirse a aplicaciones, por lo que representara algo o alguien que interactúa con el sistema que se esta desarrollando. Un ejemplo de interacción seria una persona que utiliza un cajero automático. Inserta la tarjeta del usuario y sus contestaciones, el sistema lleva a cabo una secuencia de acciones que proporcionan al usuario un resultado importante.

Una interacción de este tipo es un caso de uso. Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Los casos de uso representan los requisitos funcionales. Todos los casos de uso juntos constituyen el modelo de casos de uso, el cual describe la funcionalidad total del sistema.

Sin embargo, los casos de uso no son sólo una herramienta para especificar los requisitos de un sistema. También guían su diseño, implementación y prueba; es decir, guían el proceso de desarrollo. Basándose en el modelo de casos de uso, los desarrolladores crean una serie de modelos de diseño e implementación que llevan a cabo los casos de uso. Los desarrolladores revisan cada uno de los sucesivos modelos para que sean conformes al modelo de casos de uso. Los ingenieros de prueba prueban la Implementación para garantizar que los componentes del modelo de implementación implementan correctamente los casos de uso. De este modo, los casos de uso no sólo inician el proceso de desarrollo sino que le proporcionan un hilo conductor. Dirigido por casos de uso quiere decir que el proceso de desarrollo sigue un hilo, avanza a través de una serie de flujos de trabajo que parten de los casos de uso [8].

Los casos de uso se especifican, diseñan y los casos de uso finales son la fuente a partir de la cual los ingenieros de prueba construyen sus casos de prueba.

Aunque los casos de uso guían el proceso, no se desarrollan aisladamente. Se desarrollan a la vez que la arquitectura del sistema. Es decir, los casos de uso guían la arquitectura del sistema y la arquitectura del sistema influye en la selección de los casos de uso. Por tanto la arquitectura del sistema como los casos de uso maduran según avanza el ciclo de desarrollo [8].

2.3.-Proceso de desarrollo centrado en la arquitectura.

El papel de la arquitectura software es parecido al papel que juega la arquitectura en la construcción de edificios. El edificio se contempla desde varios puntos de vista: estructura, servicios, conducción de la calefacción fontanería, electricidad, etc.

El concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades de la empresa, como las que perciben los usuarios y los inversores, y se refleja en los casos de uso. También se ve influida por muchos otros factores, como en la plataforma en la que tiene que funcionar el software, los bloques de construcción reutilizables de que se dispone, consideraciones de implantación, sistemas heredados, y requisitos no funcionales. La arquitectura es una vista del diseño completo con las características más resaltadas, dejando los detalles de lado. Debido a que lo que es significativo depende en parte de una valoración, que a su vez, se adquiere con la experiencia, el valor de una arquitectura depende de las personas que se hayan responsabilizado de su creación. No obstante, el proceso ayuda al arquitecto a centrarse en los objetivos adecuados, como la comprensibilidad, la capacidad de adaptación al cambio, y la reutilización. Los casos de uso deben encajar en la arquitectura cuando se llevan a cabo. Por otro lado, la arquitectura debe permitir el desarrollo de todos los casos de uso requeridos, ahora y en el futuro. Tanto la arquitectura como los casos de uso deben evolucionar a la par [8].

Los arquitectos moldean el sistema para darle forma. Así, la arquitectura la que debe Diseñarse para permitir que el sistema evolucione, no sólo en su desarrollo inicial, sino también a lo largo de las futuras generaciones. Para encontrar esa forma, los arquitectos deben trabajar sobre la comprensión general de las funciones clave, es decir, sobre los casos de uso clave del sistema. Estos casos de uso clave pueden suponer solamente entre el 5 y 10 por ciento de todos los casos de uso, pero los significativos, los que constituyen las funciones fundamentales del sistema. En resumen, podemos decir que el arquitecto:

- Crea un esquema en borrador de la arquitectura, comenzando por la parte de la arquitectura que no es específica de los casos de uso. Aunque esta parte de la arquitectura es independiente de los casos de uso, el arquitecto debe poseer una comprensión general de los caso de uso antes de comenzar la creación del esquema arquitectónico.
- El arquitecto trabaja con un subconjunto de casos de uso especificados, con aquellos que representen las funciones clave del sistema en desarrollo. Cada caso de uso seleccionado se especifica en detalle y se realiza en términos de subsistemas, clases y componentes.

- A medida que los casos de uso se especifican y maduran, se descubre más de la arquitectura. Esto a su vez, lleva a la maduración de más casos de uso. Este proceso se continúa hasta que la arquitectura es estable [8].

2.4.-Proceso Unificado es iterativo e incremental.

El desarrollo de un producto de software comercial supone un gran esfuerzo que puede durar entre varios meses hasta posiblemente un año más. Es práctico dividir el trabajo en partes más pequeñas o miniproyectos. Cada miniproyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto. Para una efectividad máxima, las iteraciones deben estar controladas; esto es, deben seleccionarse y ejecutarse de una forma planificada.

Los desarrolladores basan la selección de lo que se implementará en una iteración en dos factores. En primer lugar, la iteración trata un grupo de casos de uso que juntos amplían la utilidad del producto desarrollado hasta ahora. En segundo lugar, la iteración trata los riesgos más importantes. Las iteraciones sucesivas se construyen sobre los artefactos de desarrollo tal como quedaron al final de la última iteración. Al ser miniproyectos, comienzan con los casos de uso y continúan a través del trabajo de desarrollo subsiguiente, que termina convirtiendo en código ejecutable los casos de uso que se desarrollaban en la iteración. Por supuesto, un incremento no necesariamente es aditivo. Especialmente en las primeras fases del ciclo de vida, los desarrolladores pueden tener que reemplazar un diseño superficial por uno más detallado o sofisticado. En fases posteriores, los incrementos son típicamente aditivos [8].

En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean un diseño utilizando la arquitectura seleccionada como guía, implementan el diseño mediante componentes, y verifican que los componentes satisfacen los casos de uso. Si una iteración cumple con sus objetivos el desarrollo continúa con la siguiente iteración. Cuando una iteración no cumple sus objetivos, los desarrolladores deben revisar sus decisiones previas y probar con un nuevo enfoque.

Para alcanzar un mayor grado de economía en el desarrollo, un equipo de proyecto intentará seleccionar sólo las iteraciones requeridas para lograr el objetivo del proyecto. Intentará secuenciar las iteraciones en un orden lógico. Un proyecto con éxito se ejecutará de una forma directa, solo con pequeñas desviaciones del curso que los desarrolladores planificaron inicialmente. En la medida en que se añadan iteraciones o se altere el orden de las mismas por problemas inesperados, el proceso de desarrollo consumirá más esfuerzo y tiempo. Uno de los objetivos de la reducción del riesgo es minimizar los problemas inesperados [8].

2.5.-Vida del Proceso Unificado.

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo constituye con una versión del producto para los clientes.

Cada ciclo consta de cuatro fases: inicio, elaboración, construcción y transición. Cada fase se subdivide a su vez en iteraciones [10].

Cada ciclo produce una nueva versión del sistema, y cada versión es un producto preparado para su entrega. Consta de un cuerpo de código fue incluido en componentes que puede compilarse y ejecutarse, además de manuales y otros productos asociados. Sin embargo, el producto terminado no solo debe ejecutarse a las necesidades de los usuarios, sino también a las de todos los interesados, es decir, toda la gente que trabajará con el producto. El producto software debería ser algo más que el código máquina que se ejecutará.

El producto terminado incluye los requisitos, casos de uso, especificaciones no funcionales y casos de prueba. Incluye el modelo de la arquitectura y el modelo visual. Artefactos modelados con el Lenguaje Unificado de Modelado. Incluye todos los elementos que permiten a los interesados (clientes, usuarios, analistas, diseñadores, etc.) y directores especificar, diseñar, implementar, probar y utilizar un sistema. Estos elementos permiten a los usuarios utilizar y modificar el sistema de generación en generación En la figura 2. Se muestra el modelo del proceso unificado dirigido por casos de uso. [8].

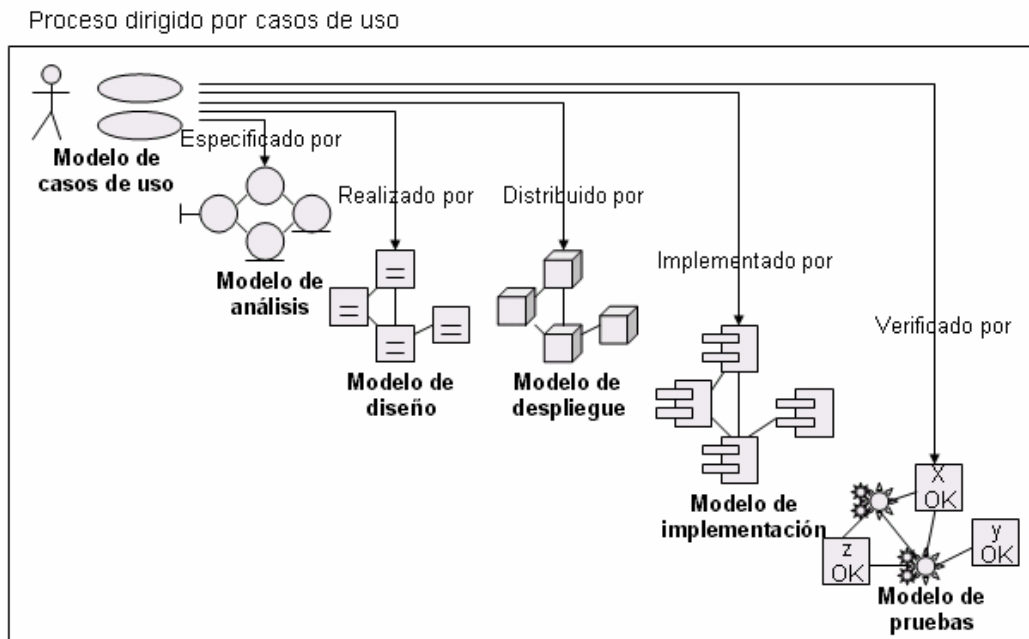


Figura 2. Modelo del Proceso Unificado.

2.6.-¿Que es UML?

UML quiere decir Unified Modeling Language, UML es un lenguaje estándar para visualizar, especificar, construir y documentar los artefactos de un sistema de software, UML puede usarse en las diferentes etapas del ciclo de vida del desarrollo en diferentes tecnologías de implementación [6].

2.7.-Áreas de trabajo en UML.

Los principales objetivos en el diseño de UML fueron éstos: obtener un lenguaje simple pero suficientemente expresivo, que permitiera modelar aplicaciones en cualquier dominio, obtener un lenguaje legible; puesto que sería un lenguaje utilizado por las personas; y permitir la generación automática de código.

Para combinar la simplicidad con la aplicabilidad a cualquier dominio, UML incorpora un conjunto de mecanismos de extensibilidad que permiten definir perfiles que lo adaptan a un dominio concreto (aplicaciones Web, CORBA (Common Object Request Broker Architecture), modelado de negocio, EJB (Enterprise Java Bean)).

La legibilidad se obtiene mediante diagramas visuales que presentan los modelos. La generación de código por parte de herramientas exige una definición formal de UML, que se consigue mediante un metamodelo expresado en un metalenguaje denominado MOF (Meta-Object Facility) [6].

2.8.-Proceso Unificado (Unified Process, UP)

Es una metodología para definir procesos de software basados en UML definido en Rational, la empresa de sus creadores que fue adquirida por IBM a finales de los 2002. En los últimos años se ha definido numerosos procesos que se ajustan en los principios del UP: procesos dirigidos por casos de uso, iterativos e incrementales, y centrados en la arquitectura. El proceso más extendido ha sido el RUP (Rational Unified Process), definido por Rational. La figura 2.1. Presenta las diferentes etapas de RUP. [6].

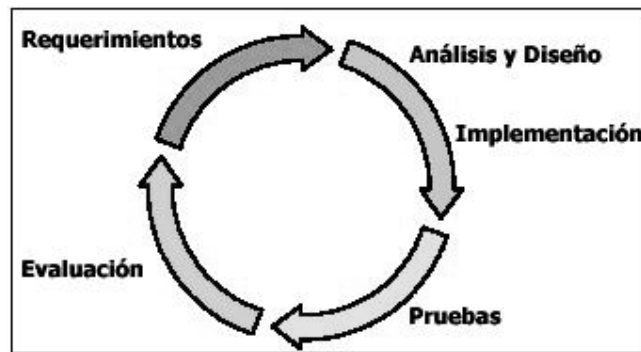


Figura 2.1 RUP.

Características del RUP.

- Iterativo. Refinamiento sucesivo
- Controlado, Gestión de requisitos y control de cambios
- Construcción de modelos
- Desarrollo de la arquitectura. Componentes de software

- Conducido por casos de uso
- Soporta técnicas OO. Uso de UML.
- Desarrollo de software basado en componentes
- Configurable
- Fomento al control de calidad

En la figura 2.2. Se presentan las diferentes fases en el proceso unificado.

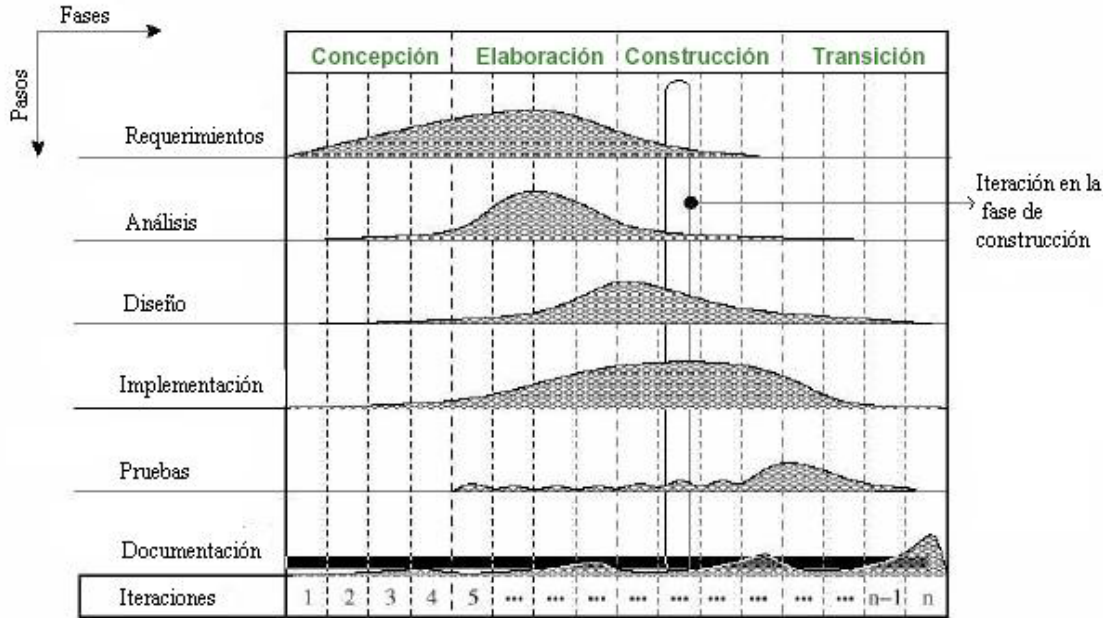


Figura 2.2 Fases en el Proceso Unificado.

2.9.-Descripción de los diagramas de UML.

En esta sección se presentará los principales diagramas en UML así como una breve descripción de cada uno de estos. Los cuales son: de casos de uso, de clases, de objetos, de estados de secuencia, de colaboración, de componentes, de actividad y de despliegue.

Diagrama de casos de Uso.- Un caso de uso es una secuencia de transacciones realizadas por el sistema que brinda un resultado de valor a un actor en particular. Un caso de uso es una unidad coherente de funcionalidad, externamente visible, proporcionada por una unidad de sistema y expresada por secuencias de mensajes intercambiados por el sistema y uno o más actores la figura 2.3 muestra un ejemplo.[9].

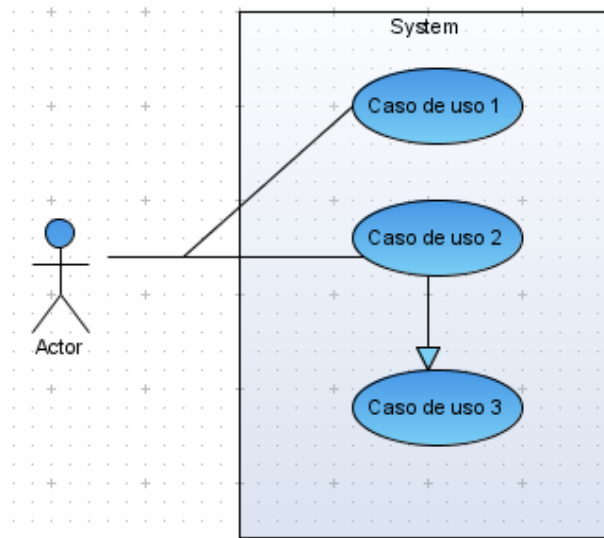


Figura 2.3 Diagrama de casos de uso.

El rectángulo representa los límites del sistema que contiene los casos de uso. Los actores se ubican fuera de los límites del sistema, los casos de uso se representan con óvalos, la etiqueta en el óvalo indica la función del sistema.

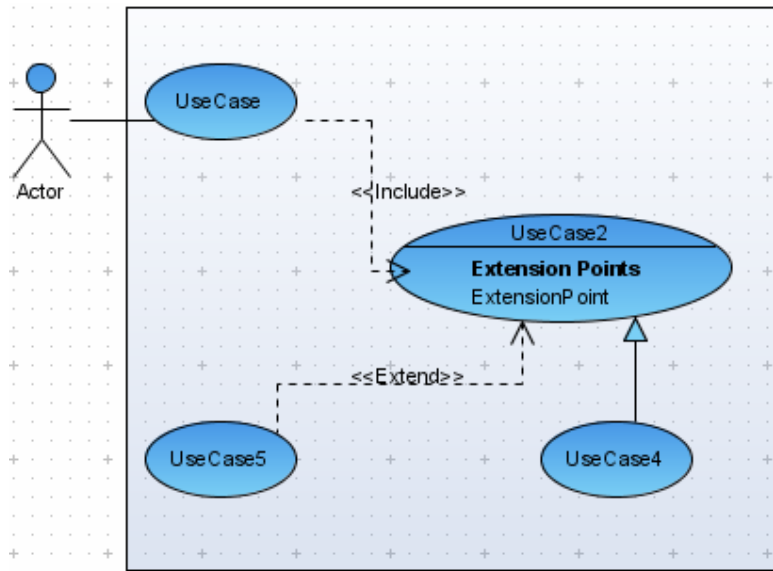


Figura 2.4 Relaciones de casos de uso.

Relaciones entre un actor y un caso de uso, se dibujan con una línea simple. Para relaciones entre casos de uso, se utilizan flechas etiquetadas “Include” o “Extend”. Una relación “Incluir” indica que un caso de uso es necesitado por otro para poder cumplir una tarea. Una relación “Extend” indica opciones alternativas para un cierto caso de uso.

Diagrama de clases.- Los diagramas de clases describen la estructura estática de un sistema. Las cosas que existen y que nos rodean se agrupan en categorías. Una clase es una categoría o grupo de cosas que tienen atributos (Propiedades) y acciones similares. Un rectángulo es el símbolo que representa la clase, y se divide en tres áreas. Un diagrama de clases esta formado por varios rectángulos de este tipo conectados por líneas que representan las asociaciones o maneras en que las clases se relacionan entre si [9].

- **Asociaciones.-** Las asociaciones son las que representan a las relaciones estáticas entre las clases. El nombre de la asociación va por sobre o por debajo de la línea que lo representa. Una flecha rellena indica la dirección de la relación. Los roles se ubican cerca del final de una asociación. Los roles representan la manera en que dos clases se ven entre ellas como muestra la figura 2.5.

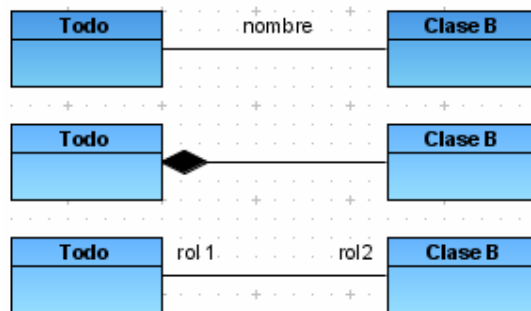


Figura 2.5 Asociación de Clases.

- **Multiplicidad.-** Las notaciones utilizadas para señalar la multiplicidad se colocan cerca del final de una asociación. Estos símbolos indican el número de instancias de una clase vinculadas a una de las instancias de la otra clase. Por ejemplo una empresa puede tener uno o más empleados, pero cada empleado trabaja para una sola empresa solamente cómo muestra la figura 2.6.

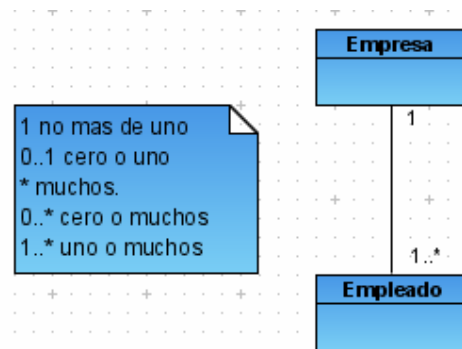


Figura 2.6 Multiplicidad de Clases.

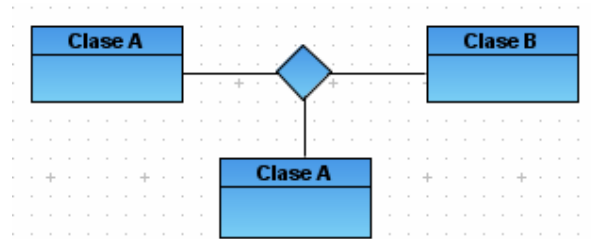


Figura 2.7 Asociación Tripartita de Clases.

- **Composición.-** Es un tipo especial de agregación que denota una fuerte posesión de la Clase “Todo” a la clase “Parte”. Se grafica con un rombo diamante relleno contra la clase que representa el todo, representado en la figura 2.8.

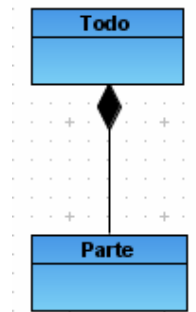


Figura 2.8 Composición de Clases.

- **Agregación.-** Es la relación en la que la Clase “Todo” juega un rol más importante que la clase “Parte”, pero las dos clases no son dependientes una de otra. Se grafica con un rombo diamante vacío contra la clase “Todo”(Ejemplo en la figura 2.9).

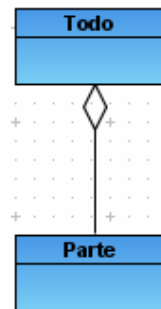


Figura 2.9 Agregación de Clases.

- **Generalización.-** Es otro nombre para herencia. Se refiere a una relación entre dos clases en donde una clase “Específica” es una versión especializada de la otra, o clase “General”(Ejemplo en la figura 2.10).

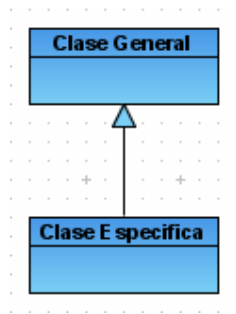


Figura 2.10 Generalización de Clases.

Un Diagrama de estructura inicial podría ser representado en la siguiente figura que muestra un rectángulo con compartimientos internos, siendo dichos rectángulos (clases) los elementos fundamentales del diagrama. Los compartimientos de una clase contienen su nombre, atributos y operaciones, en la figura encontramos las clases ingrediente, producto, máquina, depositoMonedas y depositoMonedasIguales [10].

El siguiente diagrama (ver la figura 2.11) modela el funcionamiento de una máquina de café. La clase principal es “máquina” asociada a un “producto”, que contiene “ingredientes” y una forma de pago en la clase “depositomonedas”.

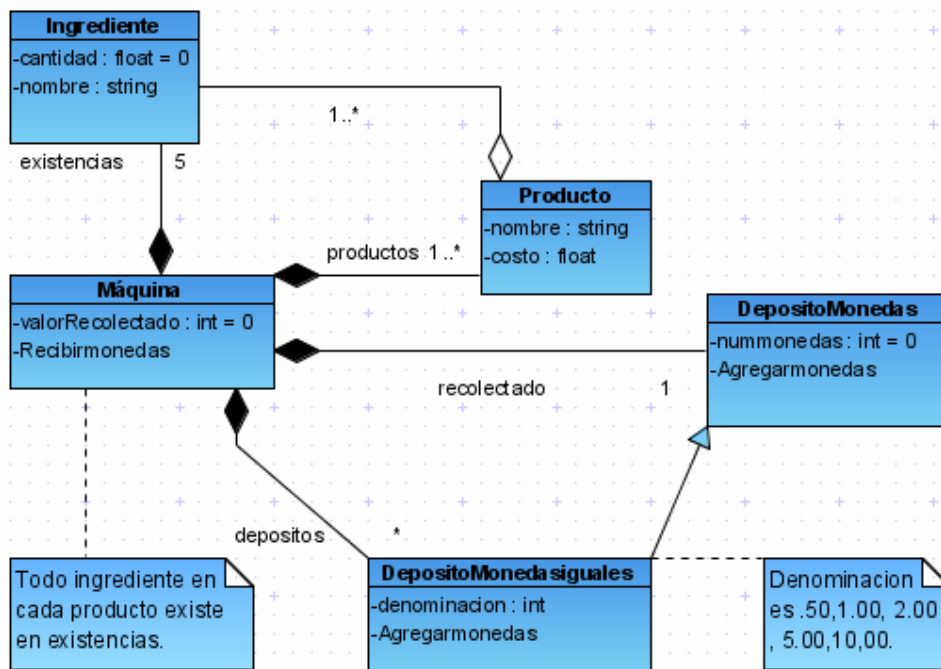


Figura 2.11 Diagrama de Clases.

Diagrama de Objetos.- Los diagramas de objetos están vinculados con los Diagramas de Clases. Un objeto es una instancia de una clase, por lo que un diagrama de objetos puede ser visto como una instancia de un diagrama de clases, Los diagramas de objetos describen la estructura estática de un sistema en un momento particular y son usados para probar la precisión de los diagramas de clases [9].

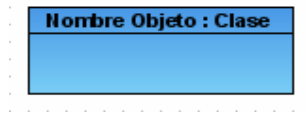


Figura 2.12 Diagrama de Objetos.

Cada objeto es representado como un rectángulo, que contiene el nombre del objeto y su clase subrayada y separadas por dos puntos. Como con las clases, los atributos se listan en un área inferior. Sin embargo, los atributos de los objetos deben tener un valor asignado.

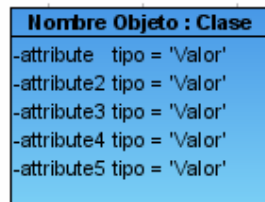


Figura 2.13 Diagrama de objetos con Atributos.

Diagrama de Estados.- Un diagrama de estados muestra los posibles estados en qué puede encontrarse un objeto y las transiciones qué pueden causar un cambio de estado. El estado de un objeto depende de la actividad que este llevando acabo o del alguna condición. Las transiciones son las líneas qué unen los diferentes estados. En ellas se representa la condición qué provoca el cambio, seguida de la acción oportuna separada por “/”. En un estado en qué el objeto esta pendiente de algún tipo de validación qué dependa de un proceso en curso, no es necesario evento externo alguno para qué se produzca la transición, ya qué esta ocurrirá cuando termine el proceso, en función del resultado de éste. En estos casos es conveniente, por claridad, incluir la condición qué de la que depende la transición (entre corchetes) [11].

Los estados inicial, a partir de qué se “entra” en la máquina de estados, y final, que indica qué la máquina de estados termina, no tienen otro significado adicional, son elementos ornamentales y se representan mediante un círculo negro y un círculo negro resaltado respectivamente. Los estados de un diagrama de estados pueden anidarse, de forma qué los estados relacionados pueden ser agrupados en un estado compuesto. Esto puede ser necesario cuando una actividad involucra sub-actividades asíncronas o concurrentes.[11]. Ejemplo de una maquina de estados figura 2.14.

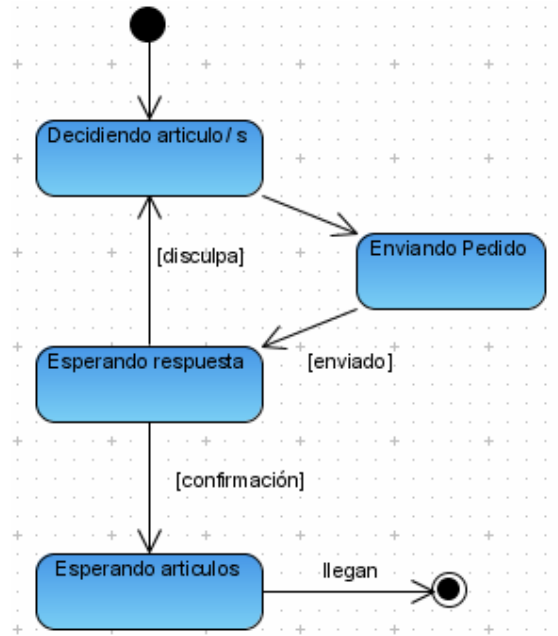


Figura 2.14 Máquina de Estados, estados simples.

Diagrama de Secuencia.- Los diagramas de secuencia describen cómo los objetos del sistema Colaboran. Se trata de un diagrama de interacción que detalla como las operaciones se llevan a cabo, qué mensajes son enviados y cuando, organizando todo en torno al tiempo. El tiempo avanza “hacia abajo” en el diagrama. Los objetos involucrados en la operación se listan de izquierda a derecha de acuerdo a su orden de participación dentro de la secuencia de mensajes, la figura 2.15 representa un ejemplo de un diagrama de secuencia de una tienda de préstamo de video [11].

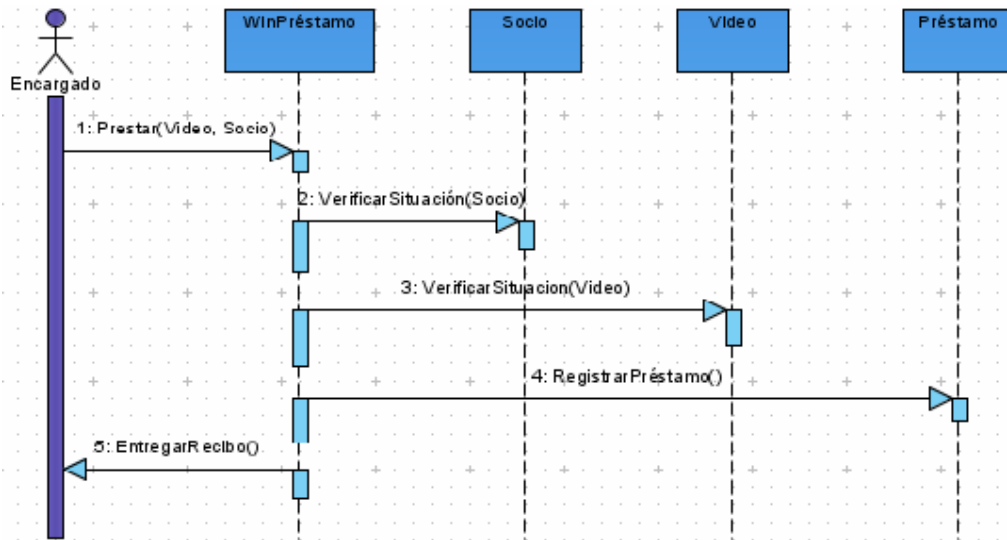


Figura 2.15 Diagrama de Secuencia.

Diagrama de Colaboración.- Son otro tipo de diagramas de interacción, que contiene la misma información que los de secuencia, solo que se centran en responsabilidades de cada objeto, en lugar de en el tiempo en que los mensajes son enviados representado en la figura 2.16. Cada mensaje de un diagrama de colaboración tiene un número de secuencia. El primer nivel de secuencia es 1, y los mensajes que son enviados durante la misma llamada a un método se numeran 1, 2 y así sucesivamente para tantos niveles como sea necesario [11].

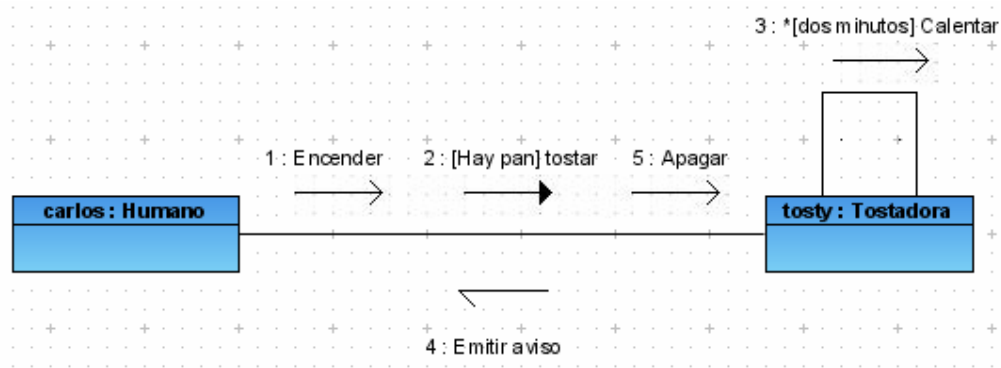


Figura 2.16 Diagrama de Colaboración.

Diagrama de Componentes.- Los componentes son módulos de código, así que los diagramas de componentes vienen a ser los análogos físicos a los diagramas de clases. Muestran como está organizado un conjunto de componentes y las dependencias que existen entre ellos ejemplo en la figura 2.17.

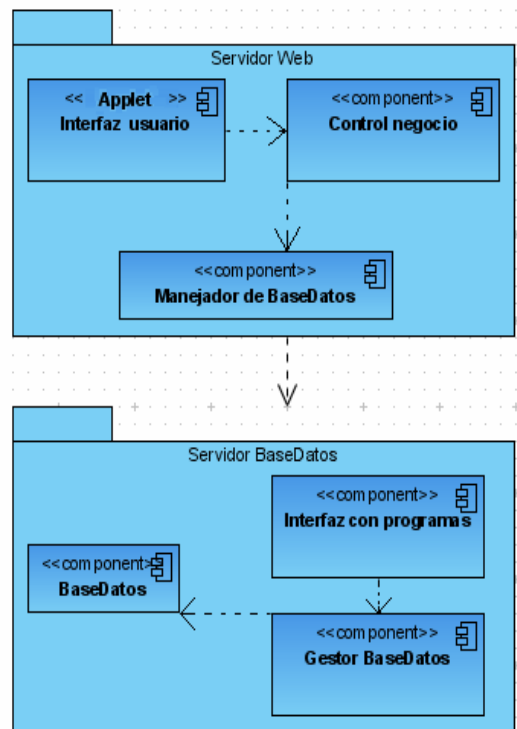


Figura 2.17 Diagrama de Componentes.

Diagrama de Actividad.- Son básicamente diagramas de flujo adornado que guardan mucha similitud con los diagramas de estados. Mientras que los diagramas de estados centran su atención en el proceso que está llevando a cabo un objeto, los diagramas de actividades muestran como las actividades fluyen y las dependencias entre ellas [11]. (Ver ejemplo en la figura 2.18.)

Los diagramas de actividades pueden dividirse en “calles” que determinan qué objeto es responsable de qué actividad. Las actividades vienen unidas por transiciones, que pueden separarse en ramas en función del resultado de una condición expresada entre corchetes. Cada rama muestra la condición que debe ser satisfecha para que un flujo opte por ese camino. Igualmente, las transiciones se pueden bifurcarse en dos o más actividades paralelas [11].

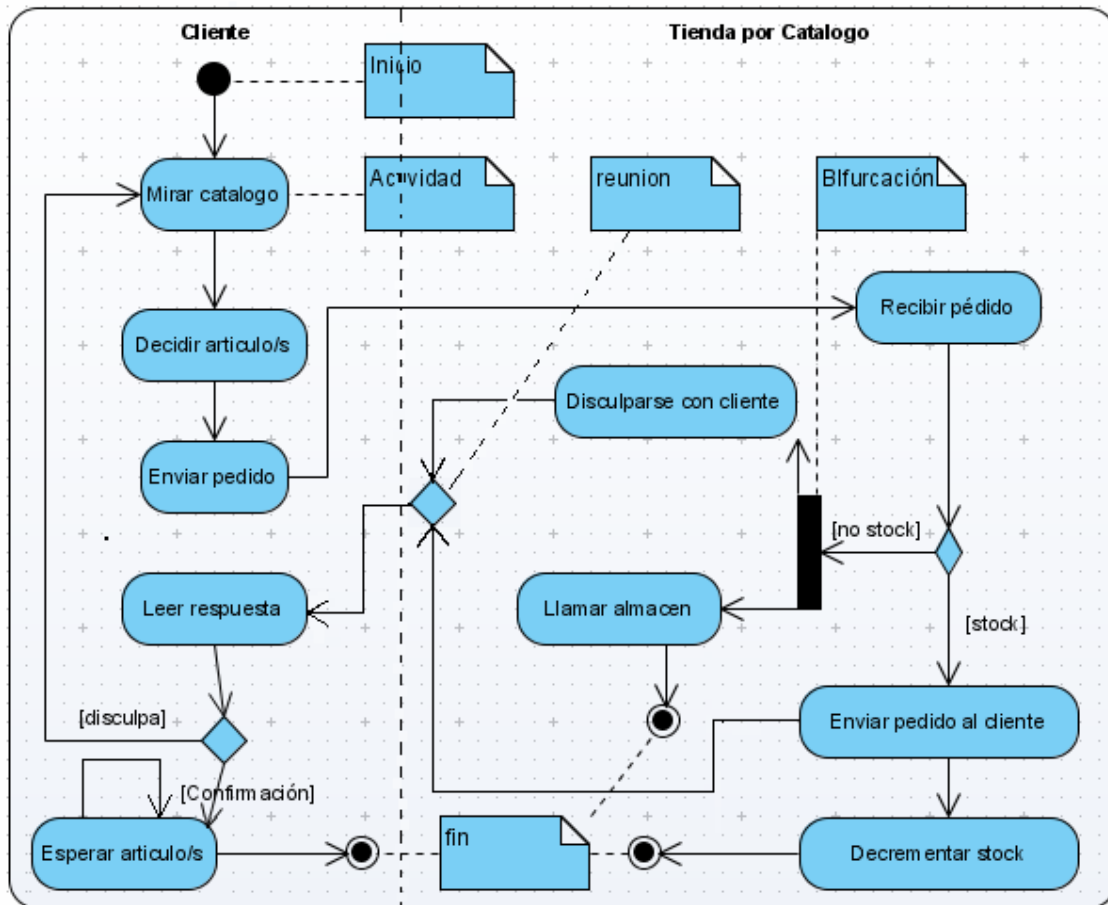


Figura 2.18 Diagrama de Actividades.

Diagrama de Despliegue.- Sirven para modelar la configuración hardware del sistema, mostrando qué nodos lo componen [11]. (Ver ejemplo en la figura 2.19).

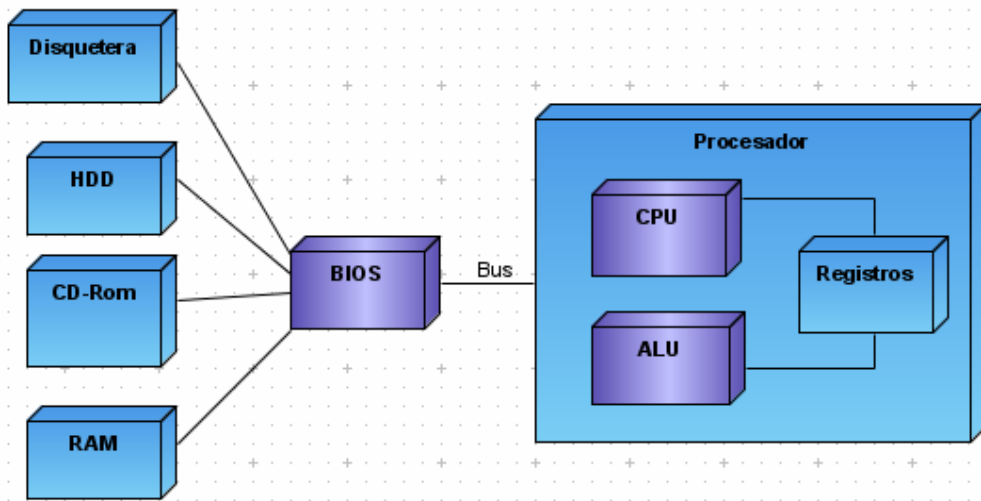


Figura 2.19 Diagrama de Despliegue.

Capítulo 3

Análisis del Sistema

En este capítulo se hablara del análisis de la aplicación Web, describiendo el planeamiento del problema, los objetivos generales, particulares, las restricciones del proyecto así como también se mostraran los diagramas de casos de uso principales, diagrama de secuencia dichos diagramas fueron elaborados con la herramienta de Visual Paradigm for UML, la cual facilita la construcción de diagramas así como la captura de requerimientos necesarios para la aplicación.

3.1.- Planteamiento del Problema.

En la actualidad la mayoría de los acuarios no se cuenta con alguna herramienta con la que facilite el control de su inventario. Lo cual genera a los usuarios responsables en las operaciones de ventas no permitir manejar directamente la información en cuestión y tener desconfianza hacia los datos de inventarios. Además dado el incremento en los productos la administración no tiene un adecuado control en los costos de inventarios, los costos de compras, costos de mermas y la utilidad que se tenía así como el reporte de algunos productos importantes que se solicitan, entregar la información detallada de reportes en muchas ocasiones es una tarea imposible o incompleta.

Por lo tanto se requiere de una aplicación de base de datos que permita facilitar el control y manejo de los datos de inventario, que tenga la capacidad de consultar y de generar reportes de los movimientos de los productos costeados de entradas y salidas.

Apoyar de manera directa al departamento de compras con reportes de compras, consultas de compras a proveedores, reportes de productos con mayor demanda.

Como apoyo al cliente se pretende que el pueda consultar desde la comodidad de su hogar y a través de Internet la página de Acuario en donde podrá encontrar las especies, productos, servicios y promociones disponibles.

3.2.- Objetivos Generales.

Desarrollar una aplicación de Base de Datos y Web para apoyar las actividades de control de inventario, difusión y promoción de un acuario.

3.3.- Objetivos Particulares.

- 1 Aplicar el Proceso Unificado de Desarrollo de Software (PUDS).
- 2 Usar la herramienta Visual Paradigm for UML Standard Edition, para visualizar, especificar y diseñar los artefactos de la aplicación.
- 3 Implantar la aplicación mediante el modelo Cliente-Servidor y MySQL como DBMS y PHP como lenguaje de desarrollo.
- 4 Desarrollar una aplicación Web para publicar las especies, productos, servicios y promociones disponibles.

Diseñar e implantar la BD necesaria para satisfacer las necesidades de información del acuario tales como:

- 1.- Entidades de productos.
- 2.- Clientes.
- 3.- Proveedores.
- 4.- Servicios.
- 5.- Promociones.
- 6.- Movimientos de productos.
- 7.- Precios.

3.4.- Descripción de la Información.

Todo el material que se obtuvo se presentará en los diagramas de casos de uso, diagramas de clases, diagramas de secuencia, los cuales se muestran en el apartado próximo.

3.5.- Restricciones.

Las restricciones para este proyecto son las siguientes:

-El administrador tendrá los permisos para dar de alta, borrar, modificar, buscar registros y obtener reportes, en la base de datos que contendrá la información de gestión de productos y gestión de personas así como encargarse del mantenimiento necesario y las actualizaciones para el sitio web.

-El empleado del Acuario podrá realizar operaciones en la aplicación como son altas, bajas, consultas, búsqueda de productos así como también la obtención de reportes; pero no podrá hacer ninguna modificación a la base datos.

-El consultor web podrá consultar desde Internet la página de acuario en donde podrá encontrar las especies, las diferentes categorías de productos, los servicios, y las promociones disponibles, no podrá tener acceso a la base de datos.

-Solo habrá un administrador para este proyecto.

-Los usuarios de la aplicación como administrador y empleado del acuario para tener acceso a la base de datos contarán con un nombre de usuario y un login para poder acceder a la información.

El sistema contará con los siguientes módulos:

- Altas.
- Bajas.
- Consultas.
- Modificaciones.
- Actualizaciones.
- Obtención de reportes.

3.6.- Diagrama de casos de uso.

La información sobre los requerimientos de la aplicación que se ha analizado se muestran a continuación en los diagramas de casos de uso que se incluyen en las figuras 3.1, 3.2, 3.3, 3.4.

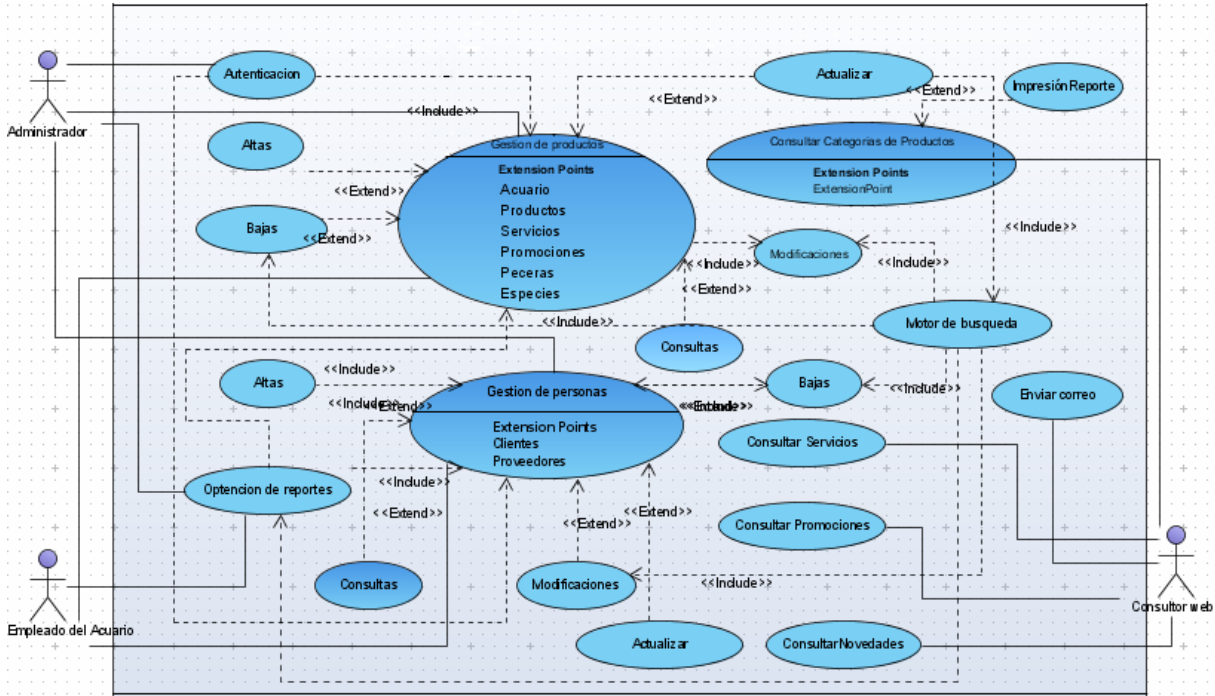


Figura 3.1 Diagrama de Casos de Uso para Administrador, Empleado del Acuario y Consultor Web.

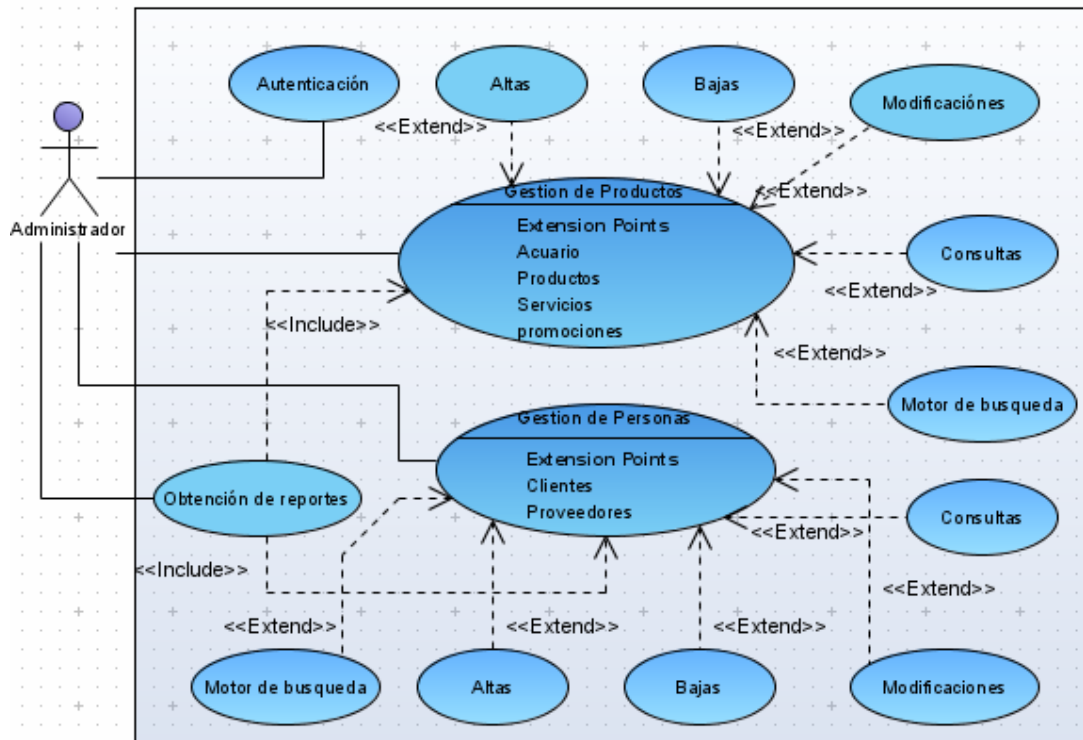


Figura 3.2 Diagrama de Casos de Uso para el Administrador.

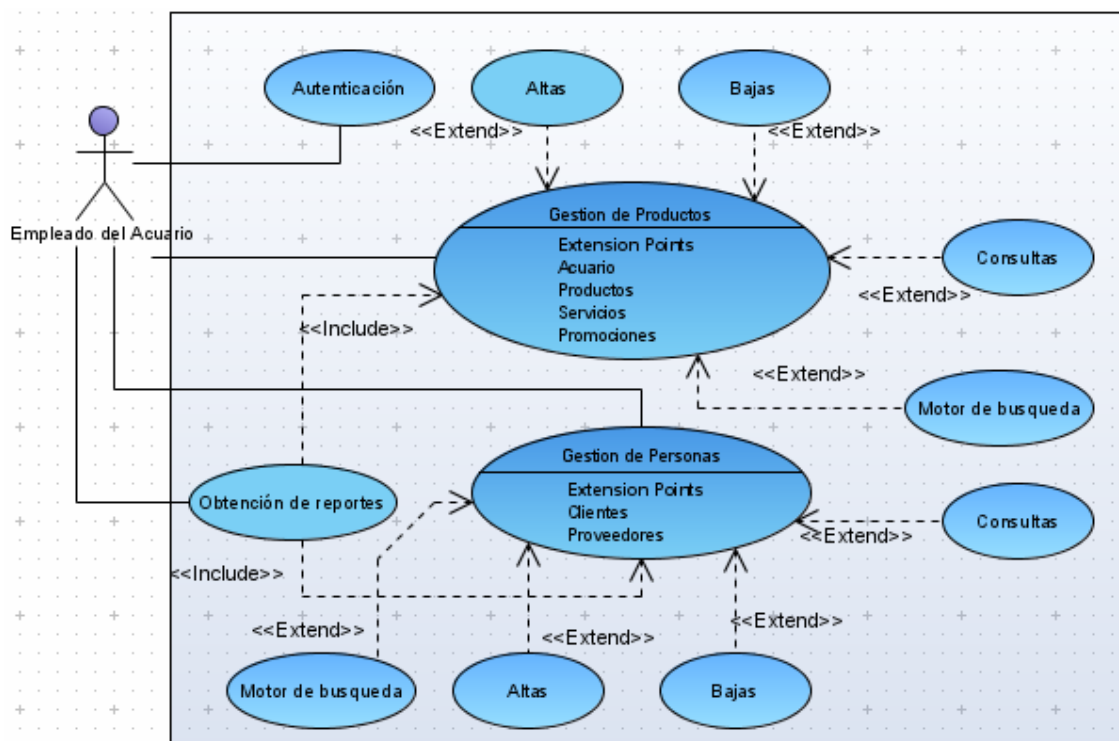


Figura 3.3 Diagrama de Casos de Uso para el Empleado del Acuario.

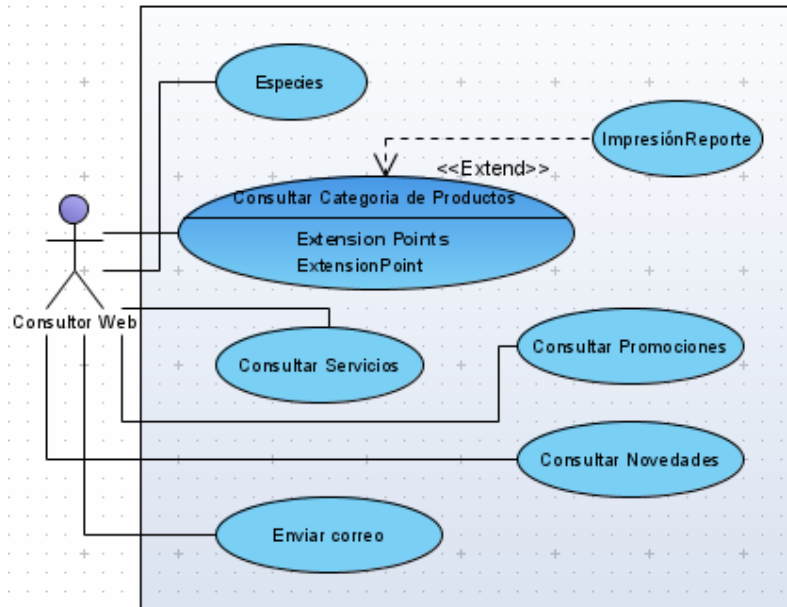


Figura 3.4 Diagrama de Casos de Uso para el Consultor Web.

A continuación describiremos detalladamente los casos de uso, del diagrama antes mencionado.

Actores:

1.- Administrador.- El administrador podrá tener acceso a la aplicación por medio de login para realizar las operaciones en la base de datos como altas, bajas, consultas, modificaciones, y búsqueda de gestión de productos y de gestión de personal, tendrá la posibilidad de dar los permisos correspondientes a los empleados así como también para la obtención de reportes la figura 3.2 nos muestra el actor administrador.

2.- Empleado del Acuario.- Es el que tendrá que entrar a la aplicación por medio de login para realizar operaciones en la aplicación de altas, bajas, consultas, búsquedas y obtener reportes sobre las entidades de gestión de productos y gestión de personal siempre y cuando el administrador le otorgue permisos necesarios la figura 3.3 nos muestra los casos de uso para el empleado del acuario.

3.- Consultor Web.- Es cualquier investigador web que realiza consultas a la página con la finalidad de obtener información acerca de consulta de categorías de productos, consulta de servicios, consulta de promociones a si como te mandar correo para pedir alguna información la figura 3.4 nos muestra los casos de uso para el consultor web.

Los casos de uso para cada actor se representan en las tablas siguientes:

Administrador

Tipo de opción	Operación	Entidades
Login.	Autenticación	Usuario
Gestión de Productos.	Alta	Acuario, Productos, Servicios, Promociones, peceras, Especies.
Gestión de Personas.	Alta.	Cientes, Proveedores.
Gestión de Productos.	Baja.	Acuario, Productos, Servicios, Promociones, peceras, Especies.
Gestión de Personas.	Baja.	Cientes, Proveedores.
Gestión de Productos.	Consulta	Acuario, Productos, Servicios, Promociones, peceras, Especies.
Gestión de Personas.	Consulta.	Cientes, Proveedores.
Gestión de Productos.	Actualización.	Acuario, Productos, Servicios, Promociones, peceras, Especies.
Gestión de Personas.	Actualización.	Cientes, Proveedores.
Gestión de Productos.	Modificación.	Acuario, Productos, Servicios, Promociones, peceras, Especies.
Gestión de Personas.	Modificación.	Cientes, Proveedores.
Gestión de Productos.	Búsqueda.	Productos, Especies..
Gestión de Personas.	Búsqueda.	Cientes, Proveedores.
Gestión de Productos.	Obtención de reportes.	Acuario, Productos, Servicios, Promociones, peceras, Especies.
Gestión de Personas.	Obtención de reportes.	Cientes, Proveedores.

Empleado de Acuario

Tipo de opción	Operación	Entidades
Login.	Autenticación	Usuario
Gestión de Productos.	Alta	Acuario, Productos, Servicios, Promociones, peceras, Especies.
Gestión de Personas.	Alta.	Cientes, Proveedores.
Gestión de Productos.	Baja.	Acuario, Productos, Servicios, Promociones, peceras, Especies.
Gestión de Personas.	Baja.	Cientes, Proveedores.
Gestión de Productos.	Consulta	Acuario, Productos, Servicios, Promociones, peceras, Especies.
Gestión de Personas.	Consulta.	Cientes, Proveedores.
Gestión de Productos.	Actualización.	Acuario, Productos, Servicios, Promociones, peceras, Especies.
Gestión de Personas.	Actualización.	Cientes, Proveedores.
Gestión de Productos.	Búsqueda.	Productos, Especies..
Gestión de Personas.	Búsqueda.	Cientes, Proveedores.
Gestión de Productos.	Obtención de reportes.	Acuario, Productos, Servicios, Promociones, peceras, Especies.
Gestión de Personas.	Obtención de reportes.	Cientes, Proveedores.

Consultor Web

Tipo de opción	Operación	Entidades
Consultar Categoría de Productos.	Consulta.	Productos.
Consultar Servicios.	Consulta.	Promociones, peceras, Especies.
Consultar Promociones.	Consulta.	Precios, Descuentos.
Consultar Novedades.	Consulta.	Vitaminas, Libros, Plantas.
Consular Especies.	Consulta.	Tipo de especie.
Imprimir datos.	Imprimir.	Acuario, Productos, Servicios, Promociones, Peceras, Especies.
Enviar Correo.	Enviar-e-mail	Acuario.

3.7.- Diagramas de Clases.

En la figura 3.5 representa el diagrama de clases en el cual se han localizado las clases y sus relaciones, mostrando una vista de diseño estático en el modelado del sistema.

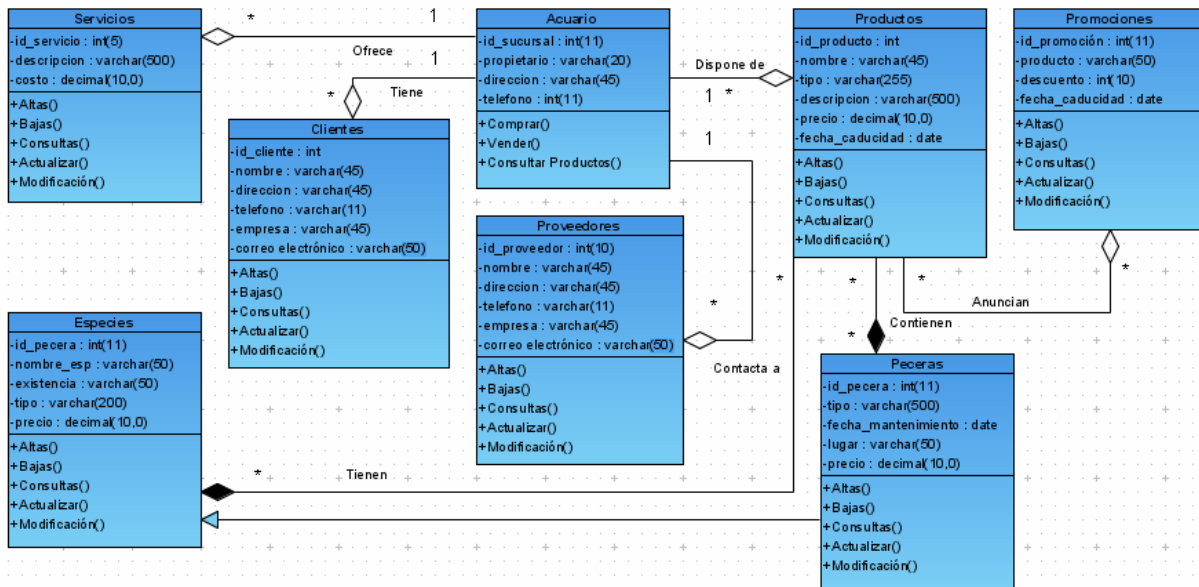


Figura 3.5 Diagrama de Clases.

3.8.-Diagrama de Secuencia.

En el siguiente apartado se muestran las figuras correspondientes a casos de diagramas de secuencia más importantes, qué hacen referencia a los casos de uso a su vez los diagramas para las entidades de gestión de productos y en gestión de personas son similares en el modulo de altas en la base de datos como se muestra en la figura 3.6.

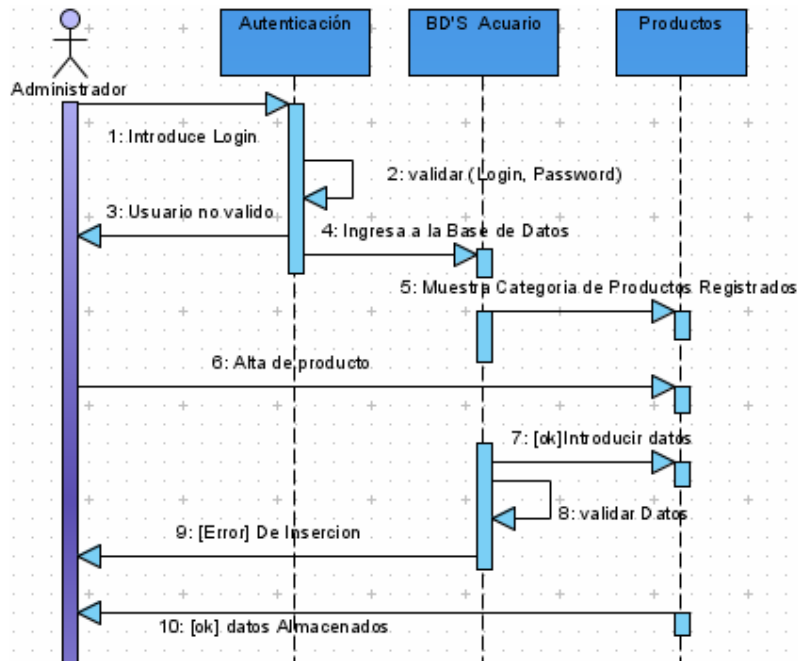


Figura 3.6 Diagrama de Secuencia de Alta de Productos para Administrador.

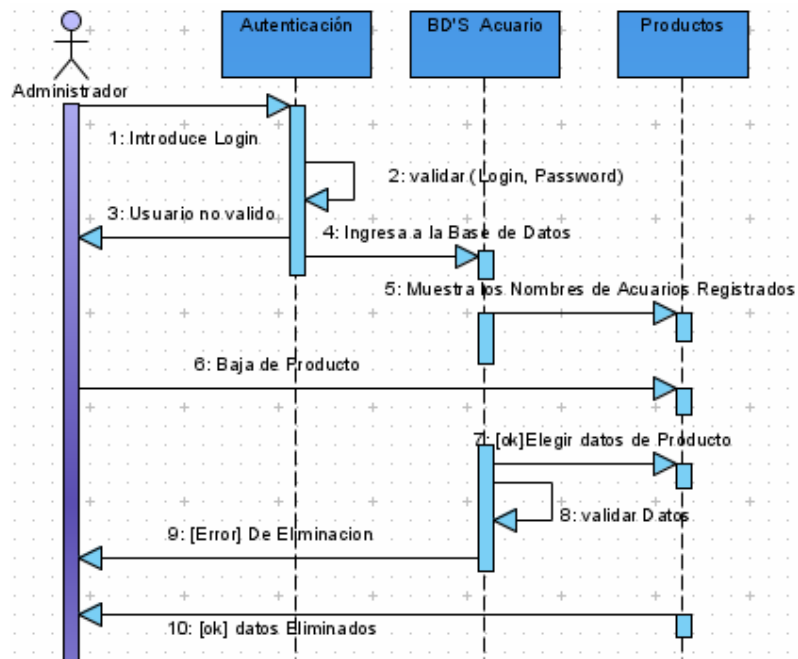


Figura 3.7 Diagrama de Secuencia de Baja de Productos para Administrador.

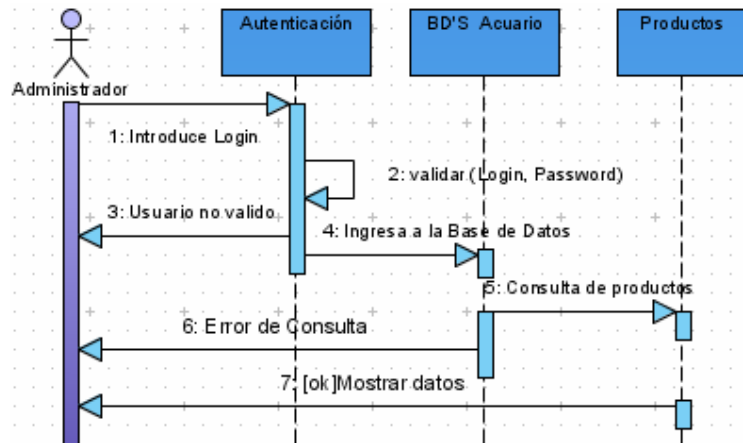


Figura 3.8 Diagrama de Secuencia de Consulta de Productos para Administrador.

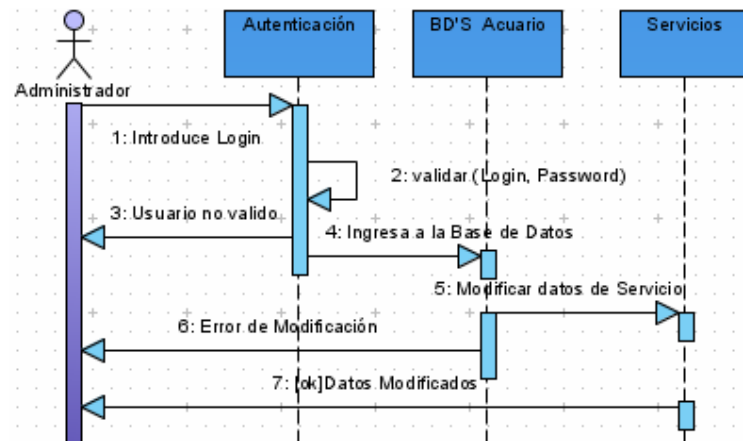


Figura 3.9 Diagrama de Secuencia de Modificación de Servicios para Administrador.

Nota: Los diagramas de secuencia para empleado son similares a los anteriores.

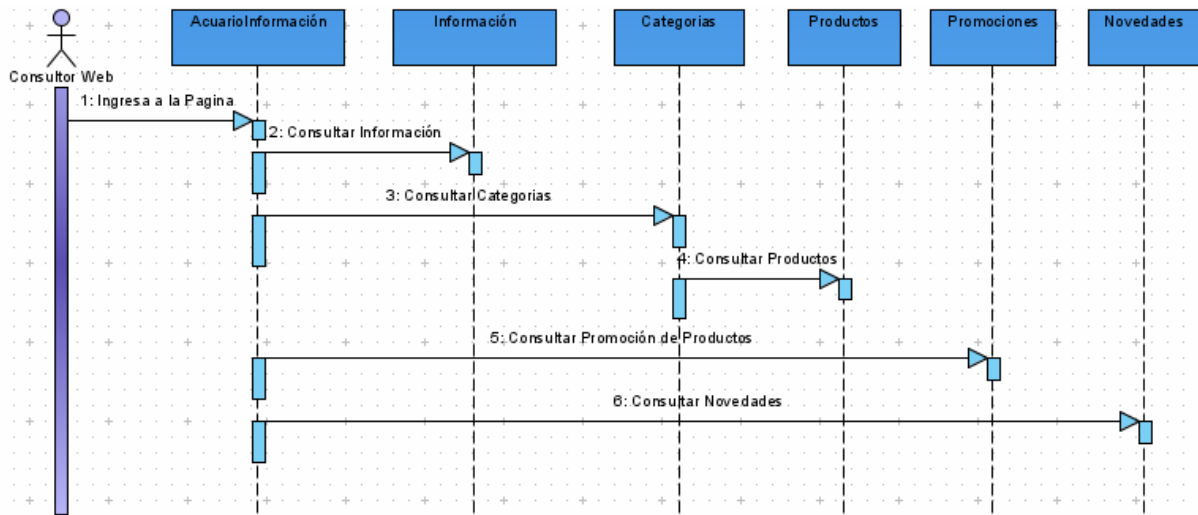


Figura 3.10 Diagrama de Secuencia de Consultor Web.

El consultor web ingresara a la página <http://acuarioinformacion.mypresonline.com> la cual proporcionara información de envíos y devoluciones como formas de pago y diferentes categorías de productos para su acuario y también las promociones que tengan los productos así como también las Novedades, se hace mención que la pagina esta en línea ya que se subió el proyecto vía FTP a un hosting que proporciona servicios gratuitos para el alojamiento web proporcionando herramientas como PHP, MySQL con 300MB de espacio disponible y la creación de una base de datos la pagina es la siguiente <http://www.leadhoster.com/>.

Capítulo 4

Diseño de la Base de Datos

En este capítulo se mostrará el diseño de la base de datos de la aplicación en la cual abarcaremos el diagrama entidad-relación, Diccionario de datos, Modelo relacional, y se justifica que la base de datos está en las primeras formas normales.

4.1.- Diagrama Entidad-Relación.

Una vez tomados los datos correspondientes, se procede a hacer el diagrama Entidad-Relación, en el cual mostraremos la relación entre las tablas de acuario, clientes, especies, productos, promociones, proveedores, servicios y usuarios.

La primera relación que se toma, es la de la entidad de acuario con los servicios, la cual tiene cardinalidad de uno a muchos (1: m), ya que un acuario puede tener muchos servicios, y muchos servicios solo tienen un acuario la relación entre ambas tablas se llamará **ofrece**.

La segunda relación que se presenta es acuario con clientes, la cual tiene cardinalidad de (1: m), ya que un acuario tiene muchos clientes, y muchos clientes tienen un acuario la relación sería **tiene**.

La tercera relación es la de acuario con peceras, la cual tiene cardinalidad de (1: m), puesto que un acuario dispone de muchas peceras, y solo un acuario puede disponer de peceras con ciertas características la relación de entre ambas tablas se llamará **dispone_de**.

La cuarta relación entre proveedores y productos es de cardinalidad de (n: m), ya que un proveedor puede poseer uno o más productos, y muchos productos pueden poseer distintos proveedores la relación entre ambas tablas se llamará **proveen**.

La siguiente relación que se observa es la de productos y promociones de (1: m), ya que un producto puede anunciar una o más promociones, pero la promoción la tiene solo un producto la relación entre las dos tablas se llamará **anuncian**.

La próxima relación es acuario con proveedores es de (1: m), ya que un acuario contacta a diversos proveedores, pero el proveedor tiene solo un acuario al cual abastecer de productos específicos la relación entre las dos tablas se llamará **contacta_a**.

La relación, es la de peceras con las especies la cardinalidad es de (1: m), en esta relación una pecera puede contener solo un tipo de especie, y una pecera puede tener más de una especie del mismo tipo la relación entre las dos tablas se llamará **contienen**.

La última relación, es de usuario y acuario, esta tabla tiene dos tipos de usuario uno es el administrador del sistema y otro es el empleado del acuario, la cardinalidad es de (n, m), puesto que un acuario puede tener registrado a más de un usuario (empleado), y muchos usuarios (empleados) pueden estar registrados en diferentes sucursales de acuario la relación de las dos tablas se llamará **trabaja_en**.

Se observa que hay una relación N a M, se genera dos tablas mas, las cuales tendrán los campos llave de ambas tablas relacionadas que es de proveedores a productos y la de usuario y acuario.

Una vez que se describieron las tablas con sus relaciones se procede a realizar el diagrama.

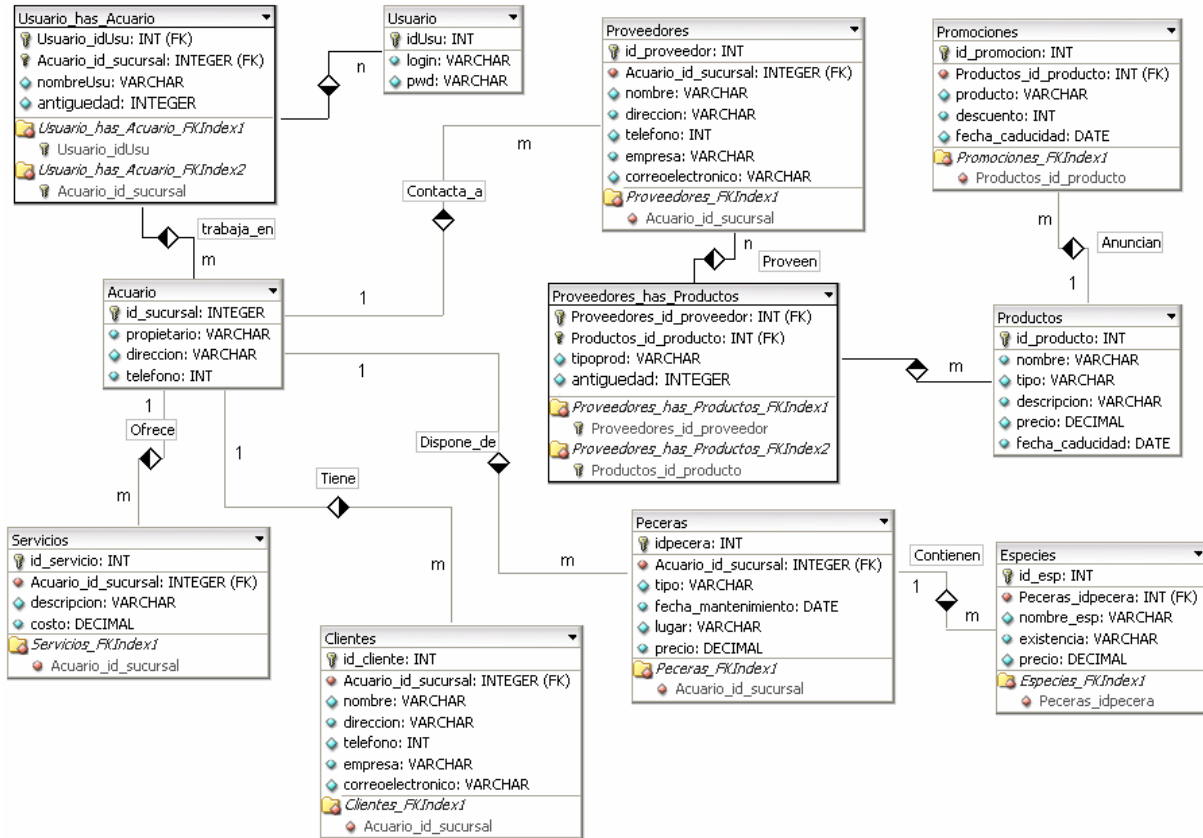


Figura 4.1 Diagrama Entidad-Relación.

4.2.- Diccionario de Datos

El diccionario de datos contiene los valores de cada atributo de las tablas, a continuación se describirá cada uno.

ACUARIO

Atributo	Descripción	Tipo dato	Longitud	Null
Id_sucursal	Clave del acuario	Int	11	No
Propietario	Propietario _ de _ acuario	Varchar	50	No
Dirección	Dirección de acuario	Varchar	40	No
Teléfono	Teléfono de acuario.	Int	11	Si

La entidad ACUARIO se relaciona directamente con SERVICIOS, ya que el acuario ofrece diferentes servicios.

SERVICIOS

Atributo	Descripción	Tipo_dato	Longitud	Null
Id_servicio	Clave del servicio	Int	5	No
Descripción	Descripción del servicio	Varchar	400	No
Costo	Precio del serv.	Decimal	(10,0)	No

SERVICIOS va relacionada con Acuario, ya que cada servicio puede ser diferente en cada acuario.

CLIENTES

Atributo	Descripción	Tipo_dato	Longitud	Null
Id_cliente	Clave del cliente.	Int	11	No
Nombre	Nombre del cliente.	Varchar	50	No
Dirección	Dirección de acuario	Varchar	40	No
Teléfono	Teléfono de acuario.	Int	11	No
Correo electrónico	E-mail de cliente.	Varchar	50	No

El acuario tiene clientes, ya que un acuario tiene muchos clientes y muchos clientes adquieren sus productos en un acuario que satisface sus necesidades, como se muestra la tabla contiene el id de cliente, el nombre, la dirección y el teléfono, y el correo.

PROVEEDORES

Atributo	Descripción	Tipo_dato	Longitud	Null
Id_proveedor	Clave del proveedor	Int	5	No
Nombre	Nombre del prov.	Varchar	50	No
Dirección	Dirección actual del proveedor.	Varchar	40	No
Teléfono.	Teléfono del prov.	Int	15	No
Empresa	Empresa de prov.	Varchar	50	No
Correo electrónico	E-mail de proveedor.	Varchar	50	No

Dentro de la tabla Proveedores se agregaron 6 atributos importantes como el id del proveedor el nombre, la dirección el teléfono la empresa y el correo, ya que el acuario contacta a distintos proveedores.

PRODUCTOS

Atributo	Descripción	Tipo_dato	Longitud	Null
Id _ producto	Clave del producto	Int	11	No
Nombre	Nombre del producto.	Varchar	50	No
Tipo	Perecedero y no perecedero.	Varchar	100	No
Descripción	Características del producto.	Varchar	500	No
Precio	Precio de producto	Decimal	(10,0)	No
Fecha_caducidad.	Caducidad de producto.	Date.		No

La tabla productos contiene los atributos más importantes como son: id del producto el nombre, la fecha de caducidad, el precio y el tipo, los acuarios disponen de diferentes tipos de productos ya que pueden ser perecederos y no perecederos.

PROMOCIONES

Atributo	Descripción	Tipo_dato	Longitud	Null
Id _ promoción.	Clave de promoción	Int	11	No
Producto.	Nombre del producto.	Varchar	50	No
Descuento.	Descuento de producto.	Int	10	No
Fecha_caducidad.	Caducidad de promoción.	Date		No

Las promociones se relacionan con los productos, ya que cada producto contiene muchas promociones, la tabla anterior muestra los atributos más importantes de las promociones como el descuento y la fecha en la cual se vence la promoción.

ESPECIES

Atributo	Descripción	Tipo_dato	Longitud	Null
Nombre_esp	Nombre de la especie.	Varchar	50	No
Existencia.	Existencias de especies.	Varchar	50	No
Id_pecera.	Clave de pecera	Int	11	No
Tipo.	Tipo de especie.	Varchar	200	No
Precio.	Precio.	Decimal	(10,0)	No

La tabla de especies contiene atributos importantes como son: el precio, la existencia el tipo, el nombre, la tabla especies se relaciona con la tabla peceras ya que cada pecera contiene diferente tipo de especie.

PECERAS

Atributo	Descripción	Tipo_dato	Longitud	Null
Idpecera	Clave de pecera	Int	11	No
Tipo	Tipo y tamaño de pecera.	Varchar	500	No
Fecha_mantenimiento	Fecha de mantenimiento.	Date		No
Lugar	Lugar en donde se realizara en mant.	Varchar	50	No
Precio	Costo de pecera	Decimal	(10,0)	No

La tabla de peceras se relaciona con la tabla de especies, ya que cada pecera posee diferentes características como son el tamaño, la forma el color, en la tabla peceras contiene los atributos más importantes como el precio, el tipo y la fecha de mantenimiento.

USUARIO

Atributo	Descripción	Tipo_dato	Longitud	Null
IdUsu	Clave de admin. y usuario.	Int	10	No
Login	Login de admin. y usuario.	Varchar	20	No
Pwd	Password de admin. y usuario.	Varchar	32	No

La tabla de usuario contiene atributos de login y password para la autenticación de administrador y la de empleado de acuario, esta tabla a su vez va relacionada con la de acuario, ya que serán muchos empleados que trabajen en distintas sucursales.

PROVEEN

Atributo	Descripción	Tipo_dato	Longitud	Null
Id_proveedor.	Llave foránea.	Int	5	No
Id_producto.	Llave foránea.	Int	11	No
tipoprod	tipo de producto	Varchar	32	No
antigüedad	Años de antigüedad	Int	5	No

TRABAJA_EN

Atributo	Descripción	Tipo_dato	Longitud	Null
Id Usu	Llave foránea.	Int	10	No
Id sucursal.	Llave foránea	Int	11	No
nombreUsu	Nombre del usuario	Varchar	32	No
antigüedad	Años de antigüedad	Int	4	No

De las relaciones N M se generaron dos tablas la de proveedores con la de productos y la usuario con la de acuario que contienen las dos llaves primarias de cada una, lo cual pasan a ser llaves ajenas en la nuevas tablas.

4.3.- Modelo Relacional.

Las tablas que se formaron del modelo entidad relación son las siguientes:

ACUARIO			
<u>Id sucursal</u>	Propietario	Dirección	Teléfono

SERVICIOS		
<u>Id servicio</u>	Descripción	Costo

CLIENTES				
<u>Id cliente</u>	Nombre	Dirección	Teléfono	Correo electrónico

PROVEEDORES					
<u>Id proveedor</u>	Nombre	Dirección	Teléfono	Empresa	Correo electrónico

PRODUCTOS					
<u>Id producto</u>	Nombre	Tipo	Descripción	Precio	Fecha caducidad

PROMOCIONES			
<u>Id Promoción</u>	Producto	Descuento	Fecha caducidad

ESPECIES				
<u>Id esp</u>	Nombre esp	Existencia	Tipo	Precio

PECERAS				
Idpecera	Tipo	Fecha_mantenimiento	Lugar	Precio

USUARIO		
IdUsu	Login	Pwd

PROVEEN			
Id_proveedorINT (FK)	Id_producto INT (FK)	tipoprod	antigüedad

TRABAJA_EN			
Id Usu INT (FK)	Id sucursal INT (FK)	nombreUsu	antigüedad

4.4.- Normalización.

4.4.1.- Definición.

Normalización es un proceso que clasifica relaciones, objetos, formas de relación y demás elementos en grupos, con base a las características que cada uno posee. Si se identifican ciertas reglas, se aplica una categoría; si se definen otras reglas, se aplicara otra categoría [16].

Estamos interesados en particular en la clasificación de las relaciones BDR, la forma de efectuar esto es a través de los tipos de dependencias que podemos determinar dentro de la relación. Cuando las reglas de clasificación sean más y más restrictivas, diremos que la relación está en una forma más elevada. La relación que está en la forma normal más elevada posible es que mejor se adapta a nuestras necesidades debido a que optimiza las condiciones que son de importancia para nosotros:

- La cantidad de espacio requerido para almacenar los datos es la menor posible;
- La facilidad para actualizar la relación es la mayor posible;
- La explicación de la base de datos es la más sencilla posible.

4.4.2.- Formas Normales.

Primera Forma Normal (1FN)

Sea un conjunto de atributo perteneciente (\in) a la relación R, en donde R está en la primera forma normal si todos los atributos $\alpha[n]$ son atómicos, es decir no pueden seguir dividiéndose [12].

Una columna no puede tener múltiples valores. Los datos son atómicos (Si a cada valor X le pertenece un valor Y, entonces a cada valor de Y le pertenece un valor de X).

Segunda Forma Normal (2FN)

Dependencia completa. Una relación está en 2FN si está en 1FN y sus atributos no principales dependen de forma completa de la clave principal. Toda columna que no sea clave debe depender por completo de la clave primaria. Los atributos dependen de la clave. Varía la clave y varían los atributos. Dependencia completa. Sus atributos no principales dependen de forma completa de la clave principal [12].

Tercera Forma Normal (3FN)

Si la relación está en segunda forma normal y todo atributo no primo es implicado por la clave primaria en una secuencia no transitiva. Se eliminan las dependencias transitivas. Todos los atributos no llave dependen de manera no transitiva de la llave primaria (no deben depender de ninguna otra columna que no sea llave). [12].

4.4.3.- Normalización de la base de datos.

Tomando en cuenta las definiciones anteriores, justificaremos la normalización de las tablas de la base de datos:

Tabla acuario. Esta tabla está en la primera forma normal, puesto que no se repite algún atributo dentro de ella, como la forma normal lo dice, cada elemento es atómico. Por otra parte, tampoco tiene múltiples valores cada uno de estos atributos. Con esta justificación, se da por hecho que la tabla está en primera forma normal (1FN).

Analizando la tabla también esté en segunda forma normal, ya que, además de estar en primera forma normal, todos los atributos dependen de la llave primaria y también se encuentra en tercera forma por que no existen dependencias transitivas todos los atributos no llave dependen de manera no transitiva de la llave primaria, no dependen de ninguna otra columna que no sea llave.

ACUARIO			
<u>Id sucursal</u>	Propietario	Dirección	Teléfono

Figura 4.2 Muestra la tabla Acuario.

Tabla servicios. Esta tabla está en primera forma normal, puesto que no se repite algún atributo dentro de ella, como la forma normal lo dice, cada elemento es atómico, por otra parte, tampoco tiene múltiples valores cada uno de estos atributos. Con esta justificación, se da por hecho que la tabla está en primera forma normal (1FN).

Analizando la tabla, también está en segunda forma normal, ya que además de estar en primera forma normal, todos los atributos dependen de la llave primaria, es decir no hay algún atributo que dependa de alguna otra llave, pues no tiene alguna llave foránea.

SERVICIOS	
<u>Id_servicio</u>	Descripción

Figura 4.3 Muestra la tabla Servicios.

Tabla clientes. Esta tabla muestra cuatro atributos de los cuales no se repite ninguno dentro de ella, cada elemento es atómico, por otra parte tampoco tiene múltiples valores y ninguno de sus atributos es llave foránea, por lo cual se da por hecho que la tabla esté en primera forma normal.

CLIENTES			
<u>Id_cliente</u>	Nombre	Dirección	Teléfono

Figura 4.4 Muestra la tabla Clientes.

Analizando la tabla, también está en segunda forma normal, ya que además de estar en primera forma normal, todos sus atributos dependen de la llave primaria, por tal razón la tabla está en segunda forma normal (2FN).

Tabla proveedores. De igual manera la tabla cumple con la primera forma normal, puesto que ningún campo se repite, y cada dato es atómico como lo dice la regla.

También está en segunda forma normal, puesto que, a pesar de tener una llave foránea, ningún atributo depende de ella.

Como todos los atributos dependen de la llave primaria, y ninguno de ellos depende de la llave foránea no puede haber ninguna dependencia transitiva, por esta razón también está en tercera forma normal y de esta manera la tabla queda normalizada.

PROVEEDORES				
<u>Id_proveedor</u>	Nombre	Dirección	Teléfono	Empresa

Figura 4.5 Muestra la tabla Proveedores.

Tabla productos. La tabla productos está en primera forma normal, puesto que no se repite ningún atributo dentro de ella, son totalmente independientes.

Ninguno de ellos es llave foránea, solo hay una llave primaria y todos los demás atributos dependen de ella, por esas razones la tabla está en segunda forma normal (2FN).

PRODUCTOS					
<u>Id_producto</u>	Nombre	Tipo	Descripción	Precio	Fecha caducidad

Figura 4.6 Muestra la tabla Productos.

Tabla promociones. Como en las tablas anteriores, esta tabla está en primera forma normal, puesto que ninguno de sus atributos es repetido, como ya se había mencionado.

También se encuentra en segunda forma normal, puesto que cumple con lo que dicta la segunda y así también queda en tercera forma por que no existen dependencias transitivas.

PROMOCIONES			
<u>Id_Promoción</u>	Producto	Descuento	Fecha_caducidad

Figura 4.7 Muestra la tabla Promociones.

Tabla especies. Analizando la tabla nos damos cuenta de que ningún atributo esta repetido, esto concuerda con la primera forma normal, además de que ninguno de sus atributos tiene múltiples valores, por esa razón la tabla se encuentra en primera forma normal.

Además como en las tablas pasadas todos sus atributos dependen de la llave primaria, así que también se encuentra en segunda forma normal.

ESPECIES				
<u>Id_pecera</u>	Nombre_esp	Existencia	Tipo	Precio

Figura 4.8 Muestra la tabla Especies.

Tabla peceras. La tabla peceras está en primera forma normal ya que ninguno de sus atributos es repetido, cada elemento es atómico, por otra parte tampoco tiene múltiples valores cada uno de esos atributos, con esta justificación se da por hecho que la tabla esté en primera forma normal (1FN).

La tabla también esta en segunda forma normal, ya que además de esta en primera forma, todos sus atributos dependen de la llave primaria, es decir no hay algún atributo que depende de alguna otra llave.

PECERAS				
<u>Id_pecera</u>	Tipo	Fecha_mantenimiento	Lugar	Precio

Figura 4.9 Muestra la tabla Peceras.

Tabla usuario. La tabla usuario tiene solo tres atributos y esta en primera forma normal, ya que ninguno de ellos se repite, por tal razón cumple con la regla de la primera forma normal.

Analizando la tabla también se encuentra en segunda forma normal, ya que todos sus atributos dependen de la llave primaria, y no hay algún atributo que dependa de alguna otra llave, pues no tiene alguna llave foránea.

USUARIO		
<u>IdUsu</u>	Login	Pwd

Figura 4.10 Muestra la tabla Usuario.

Tabla proveen. Como solo son 2 llaves foráneas, lleva dos atributos que son tipoprod y antigüedad del cliente, tampoco dependen de alguna otra llave, y mucho menos hay dependencias transitivas, por esas razones la tabla esta en primera, segunda, y tercera forma normal, la tabla poseen esta normalizada hasta la tercera forma normal (Fig 3.7).

PROVEEN			
Id_proveedorINT (FK)	Id_producto INT (FK)	tipoprod	antigüedad

Figura 4.11 Muestra la tabla Proveen.

Tabla trabaja_en. De igual manera que la tabla de la figura 4.11, esta tabla tiene 2 llaves foráneas y dos atributos. Sus valores son atómicos, no dependen cada una de otra llave, y tampoco hay dependencias transitivas. Así que esta tabla cumple con la tres formas normales que se toman en cuenta para esta base de datos, por tanto concluyo que no causara problemas en adelante.

TRABAJA_EN			
Id_Usu INT (FK)	Id_sucursal INT (FK)	nombreUsu	antigüedad

Figura 4.12 Muestra la tabla Trabaja_en.

Capítulo 5

Implementación y Pruebas.

En este capítulo se presentan los aspectos de implantación y pruebas de la aplicación web, que a su vez esta ligada a una base de datos en la cual se utilizaron herramientas como XAMPP que integra a APACHE, PHP, MySQL y PhpMyAdmin para instalar un servidor local, y administrar bases de datos, estas herramientas son gratuitas y ofrecen opciones para desarrollar aplicaciones web y base de datos, se mostraran además los pasos para crear la base de datos, así como la interfaz (página principal), para manipular la base de datos.

5.1.- MySQL

MySQL es un sistema de gestión de base de datos relacional, con licencia GLP de GNU. Su diseño multihilo le permite soportar gran carga de forma muy eficiente. Este gestor de bases de datos es, uno de los más usados en Internet, debido a su gran rapidez y facilidad de uso [13].

Características de MySQL

Las principales características de este gestor de bases de datos son las siguientes:

1. Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
2. Soporta gran cantidad de tipos de datos para las columnas.
3. Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
4. Gran portabilidad entre sistemas.
5. Soporta hasta 32 índices por tabla.
6. Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

5.2.- ¿Qué es PHP?

Es un lenguaje de programación interpretado usado normalmente para la creación de páginas web dinámicas. PHP es un (acrónimo de "PHP: Hypertext Preprocessor") [14].

El lenguaje PHP es un lenguaje de programación de estilo clásico, con esto quiero decir que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones. No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C, para aquellos que conocen estos lenguajes.

Pero a diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML pero igualmente podría ser una página WML.

Como funcionalidades primordiales en PHP se consideran:

- Funciones de correo electrónico: envío de correos individual o por grupos, parametrizando toda una serie de aspectos tales como el e-mail de procedencia, asunto, destinatario.
- Gestión de bases de datos: El lenguaje PHP ofrece interfaces para el acceso a la mayoría de las bases de datos posibles en sistemas Microsoft.
- Gestión de archivos: mediante operaciones de creación, borrado, modificación, además de ofrecer transferencia de archivos.
- Tratamiento de imágenes: mediante funciones de automatización de formato, envío de lotes de imágenes y funciones de graficado.

5.2.1.- Ventajas de PHP

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL
- Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Permite crear los formularios para la Web.
- Biblioteca nativa de funciones sumamente amplia e incluida
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

5.3.- XAMPP

Es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los interpretes para lenguajes de script: PHP y Perl. El nombre proviene del acronimo de **X** (para cualquiera de los diferentes sistemas operativos), Actualmente XAMPP esta disponible para Microsoft Windows y, GNU/Linux, Solaris, y MacOS X [15].

5.3.1.- Instalación de XAMPP

Para instalar XAMPP, solo tenemos que descargarlo de Internet, actualmente la versión 1.4.13. Ejecutar el fichero llamado xampp-win32-1.6.3a-installer, para la instalación y posteriormente dar clic en el ícono que aparece para abrir el panel de control y inicializar Apache y MySQL.



Las herramientas que proporciona XAMPP son las siguientes:

- Apache HTTPD 2.2.8
- MySQL 5.0
- PHP 5.2
- phpMyAdmin 2.11.4
- FileZilla FTP Server 0.9.25
- Mercury Mail Transport System 4.52

5.3.2.- Implementación de la Base de Datos

Para la implementación de la base de datos sin la necesidad de entrar a la consola de MySQL, se procede a utilizar phpMyAdmin lo cual facilita la creación de la base de datos con sus tablas correspondientes (Ver figura 5.1).



Figura 5.1 Creación de una nueva base de datos desde phpMyAdmin.

Una vez creada nuestra base de datos, continuamos con la creación de las tablas en las cuales vamos almacenar la información correspondiente para cada entidad que se requiera.

Crear nueva tabla en la base de datos **sucursal_a1**

Nombre: Número de campos:

Figura 5.2 Creación de una tabla.

Después de crear la tabla, se seleccionan los campos que lleva la misma, y los tipos de valor que llevara cada campo, es decir se dan las propiedades que la tabla requiere, una vez hecho esto el programa muestra un reporte de los valores que tiene cada tabla (Ver figura 5.3).

phpMyAdmin Servidor: localhost ▶ Base de datos: sucursal_a1 ▶ Tabla: proveedores

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/> id_proveedor	int(10)			No		auto_increment	[Edit] [Delete] [Add] [Refresh]
<input type="checkbox"/> nombre	varchar(50)	latin1_general_ci		No			[Edit] [Delete] [Add] [Refresh]
<input type="checkbox"/> direccion	varchar(40)	latin1_general_ci		No			[Edit] [Delete] [Add] [Refresh]
<input type="checkbox"/> telefono	int(15)			No			[Edit] [Delete] [Add] [Refresh]
<input type="checkbox"/> empresa	varchar(50)	latin1_general_ci		No			[Edit] [Delete] [Add] [Refresh]
<input type="checkbox"/> correoelectronico	varchar(50)	latin1_general_ci		No			[Edit] [Delete] [Add] [Refresh]

↑ Marcar todos/as / Desmarcar todos Para los elementos que están marcados: [Edit] [Delete] [Add] [Refresh]

Vista de impresión Planteamiento de la estructura de tabla

Añadir 1 campo(s) Al final de la tabla Al comienzo de la tabla Después de id_proveedor Continuar

Índices:				Espacio utilizado	
Nombre de la clave	Tipo	Cardinalidad	Acción	Tipo	Uso
PRIMARY	PRIMARY	11	[Edit] [Delete]	Datos	572 Bytes
nombre	FULLTEXT	Ninguna	[Edit] [Delete]	índice	3,072 Bytes
				Total	3,644 Bytes

Crear un índice en 1 columna(s) Continuar

Estadísticas de la fila	
Enunciado	Valor
Formato	dinámico/a
Cotejamiento	latin1_general_ci
Filas	11
Longitud de la fila	52

Figura 5.3 Propiedades de entidad.

5.4.- Implementación de la Interfaz (PHP)

5.4.1.- Página de Inicio

Como interfaz para ver el diseño, he utilizado el software Macromedia DreamWeaver, el cual ofrece muy buenas herramientas para el diseño de páginas web, además de tener vista de diseño y vista de programador dependiendo de lo que se vaya hacer.

La interfaz de inicio la cual contiene información de productos del acuario así como también, la vinculación de la base de datos, con identificación (login) para administrador y empleado.



Figura 5.4 Página de Inicio.

En la figura siguiente muestra la interfaz de login para usuarios administrador y empleado.

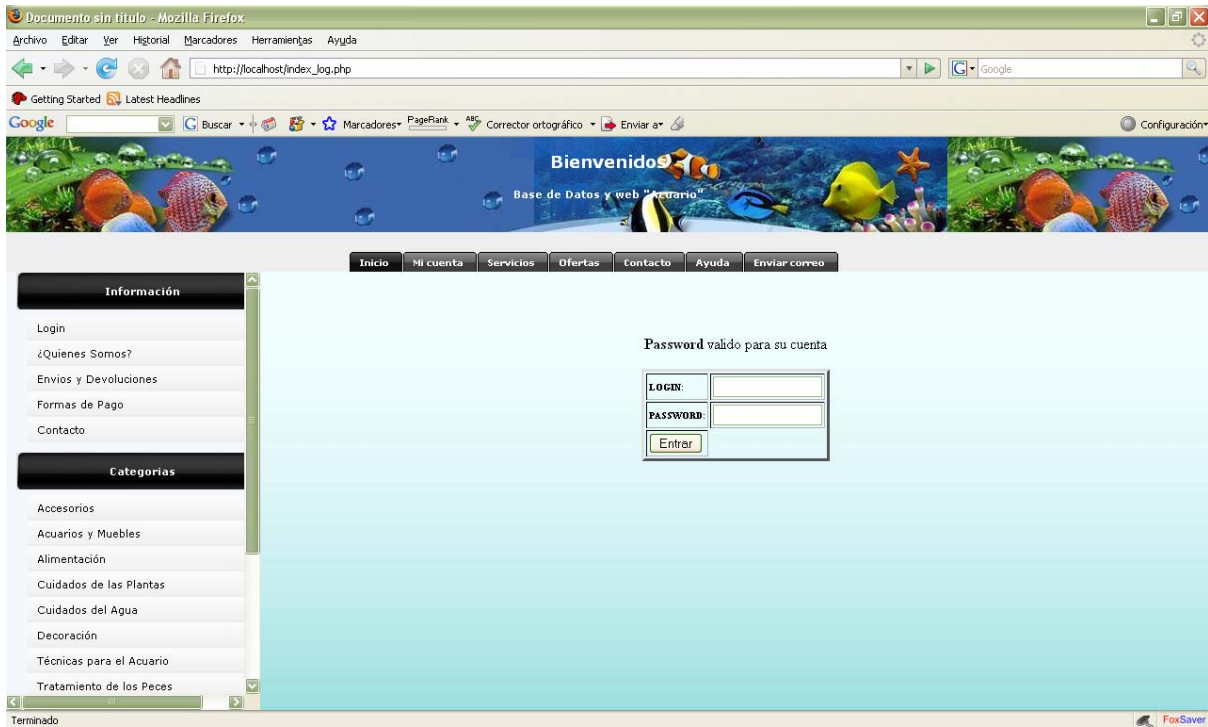


Figura 5.5 Login y Contraseña

El código para este pequeño módulo es el siguiente:

```
<?php
$_log = trim($_POST['log']);
$_pas = trim($_POST['pas']);

include_once("conex.php");

$Q = "select idUsu,pwd from usuario where login='".$_._log.'";";
$R = mysql_query($Q);

$n = mysql_num_rows($R);

if ($n==0){
    header('Location: index_log.php');
    exit();
}
else{
    $D = mysql_fetch_row($R);
    $penc = $D[1];
    $psusu = MD5($_pas);
    if (strcmp($penc,$psusu)!=0){
        header('Location: index_pas.php ');
        exit();
    }

    if($D[2]=='i'){
        header('Location: index_ina.php ');
        exit();
    }
}
```

```

session_start();
$stip = $D[3];
$_SESSION['gtipo'] = $stip;
$_SESSION['gidusu'] = $_log;
include_once("menu1_bda.php");

switch($stip){
case 'a' : // hablamos del administrador
    header('Location: menu1_bda.php ');
    break;

case 'i' : // hablamos de un empleado
    header('Location: index2.php ');
    break;

//case 'd' : header('Location: index3.php '); break;
}
exit();
}
?>

```

Al iniciar con login correcto nos direcciona a menu1_bda.php lo cual se puede acceder a la base de datos y realizar las operaciones según se requiera, en caso contrario se denegara el acceso pidiendo nuevamente la contraseña correcta.

5.4.2.- Interfaz de empleado

Una vez ingresando el Id y contraseña correctos, entrará a la página índice para empleado, la cual se muestran las opciones con las que cuenta el empleado.

-Insertar, Borrar, Consultar, Buscar en las tablas.

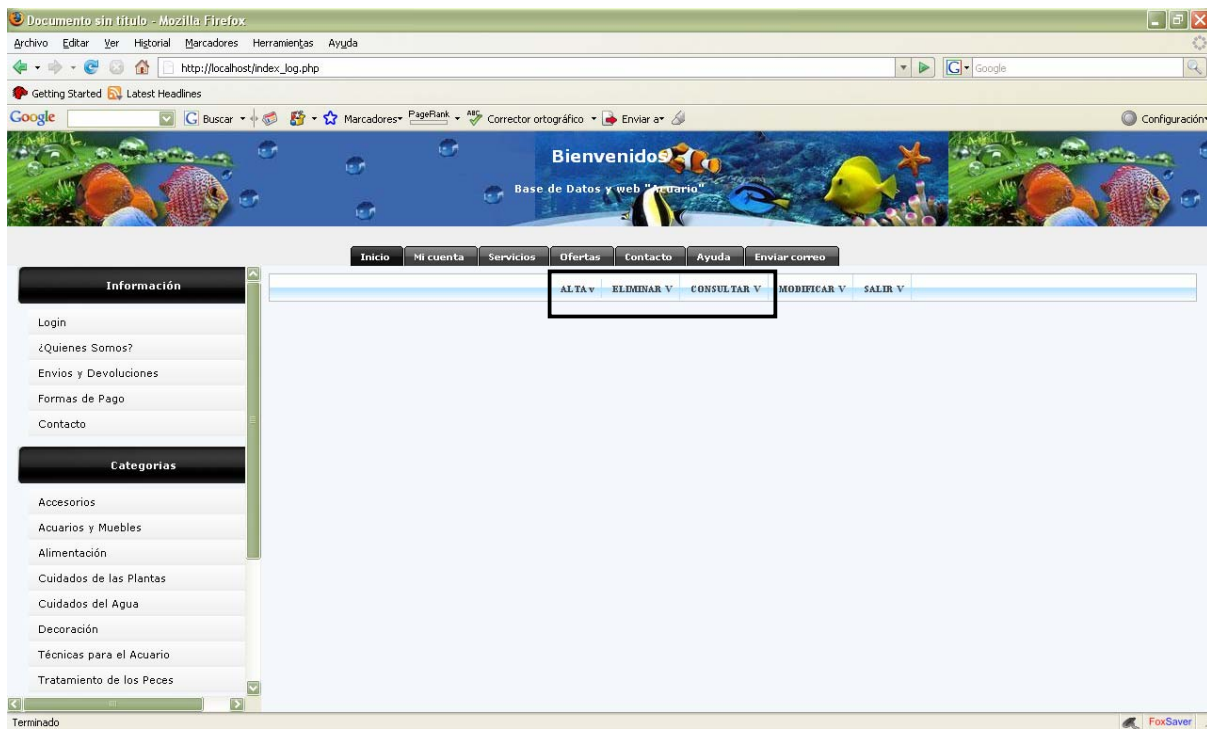


Figura 5.6 Página de inicio de empleado.

5.4.3.- Página de Inicio de Administrador

En caso de ser administrador entra al índice de Administrador, donde se le ofrecen todas las opciones para trabajar con las tablas de la base de datos.

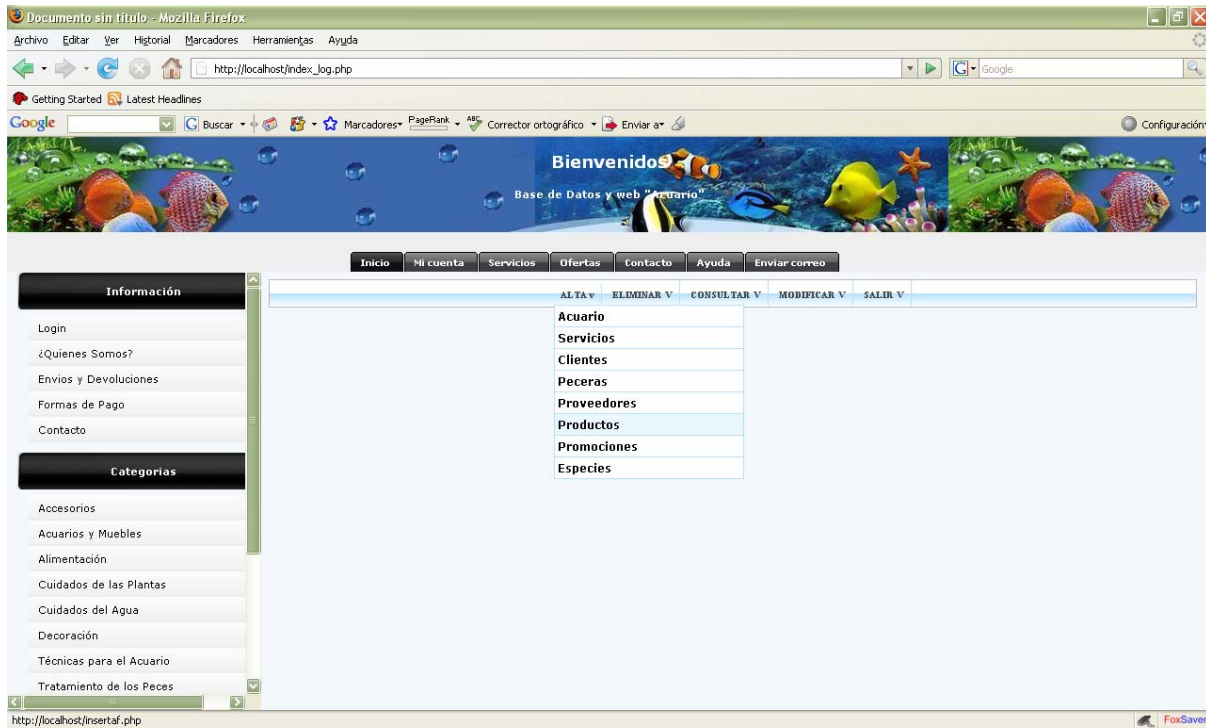


Figura 5.7 Página de inicio de administrador.

La página muestra la base de datos con las entidades con las cuales el administrador puede realizar operaciones como inserción, borrar, consultar, modificar y buscar.

En la siguiente pantalla mostrará los campos para insertar productos para el administrador.

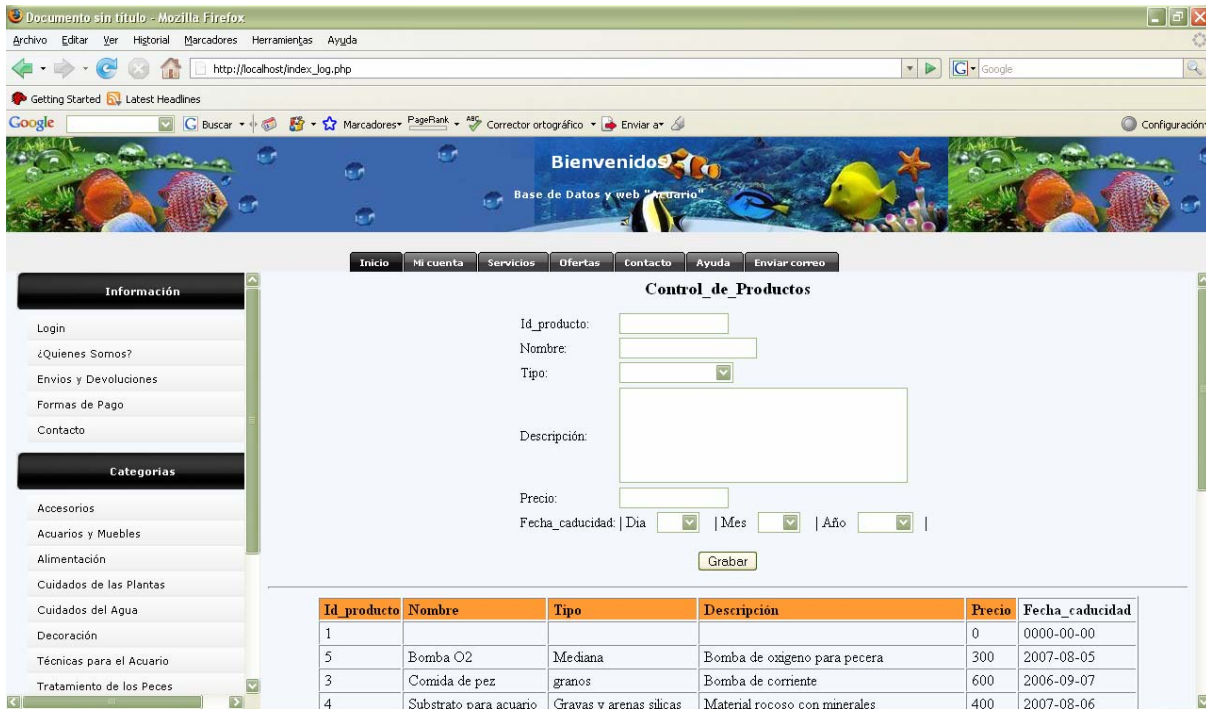


Figura 5.8 Registro de Productos.

Código para el módulo de insertar productos:

```

<html>
<head>
  <title>Np</title>
  <style type="text/css">
<!--
.Estilo1 {
  color: #000000;
  font-size: 18px;
}
body {
  background-color: #F4F7FB;
}
.Estilo4 {font-size: 14px}
-->
</style>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"></head>
<body>
<H1 align="center" class="Estilo1">Control_de_Productos</H1>
<FORM NAME="insertaf" ACTION="insertar10f.php" METHOD="POST">
  <div align="center">
  <TABLE bordercolor="#333333">

  <TR>
  <TD><span class="Estilo4">Id_producto:</span></TD>
  <TD><INPUT TYPE="text" NAME="id_producto" SIZE="15" MAXLENGTH="30"></TD>
  </TR>
  <TR>
  <TD><span class="Estilo4">Nombre:</span></TD>
  <TD><INPUT TYPE="text" NAME="nombre" SIZE="20" MAXLENGTH="30"></TD>
  </TR>

  <TR>
  <TD><span class="Estilo4">Tipo:</span></TD>

```


Si el administrador quiere modificar algún registro, sólo se selecciona el criterio de búsqueda que generalmente son llaves primarias cómo se representa en la figura 5.9.

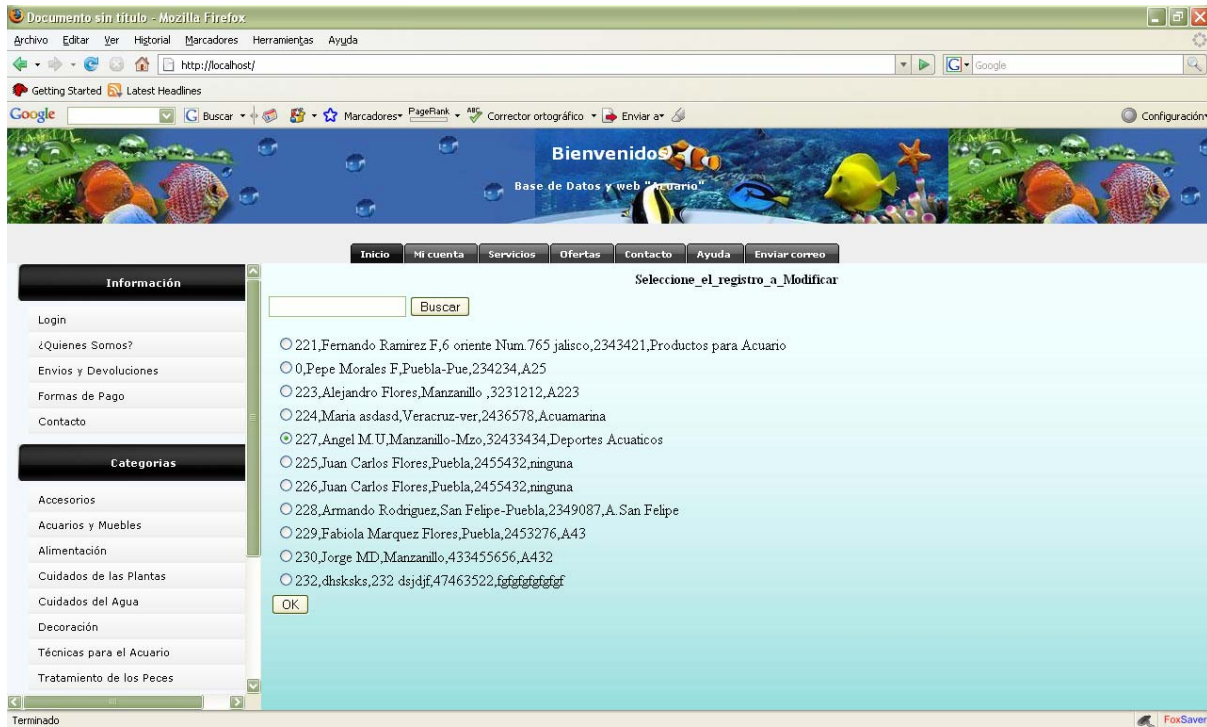


Figura 5.9 Búsqueda del dato a Modificar.

Una vez localizada la tupla a modificar se ingresan los nuevos campos, y se manda la actualización. En la figura 5.10 se puede observar los campos que se tienen para modificar las tuplas deseadas.

En la siguiente página muestra cómo borrar alguna tupla en la tabla de “proveedores”.

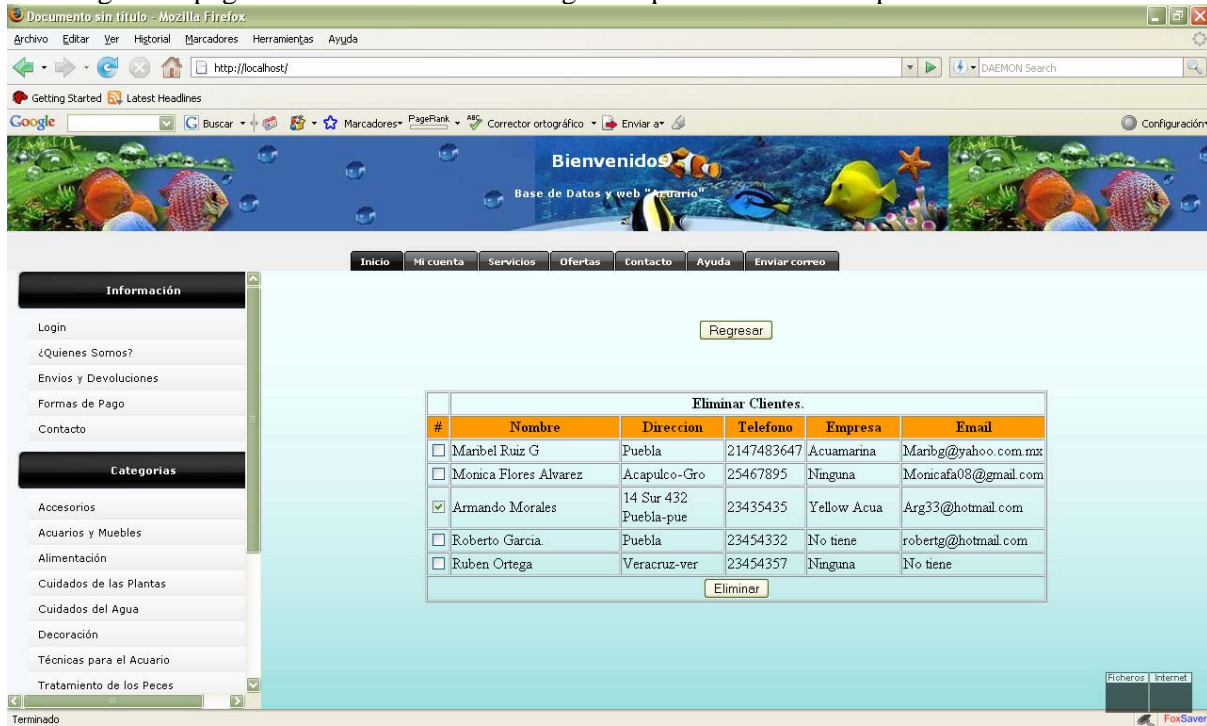


Figura 5.10 Borrado de tuplas.

En este caso se selecciona el registro a eliminar y se da clic en el botón eliminar, el administrador y el empleado sólo tienen que dar clic en las tuplas que quieran eliminar, y automáticamente se dará de baja en la base de datos, pueden eliminar dos o más registros a la vez (Ver figura 5.10).

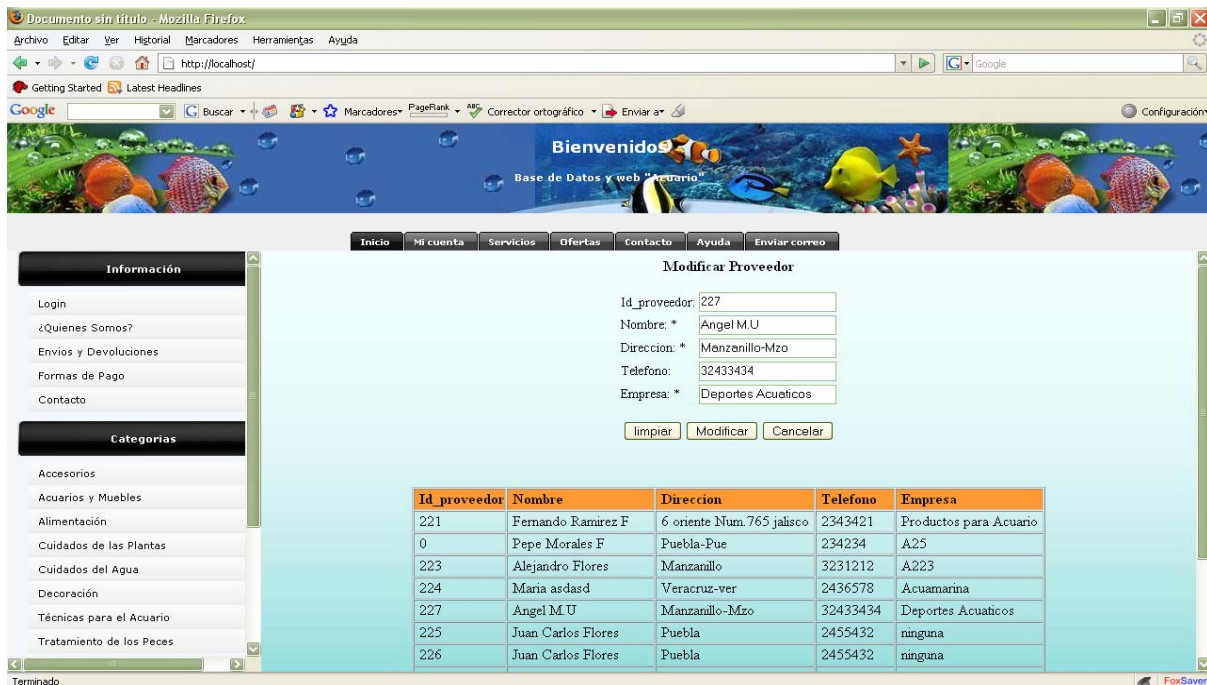


Figura 5.11 Actualización de datos.

Conclusiones.

Como primer punto se desarrolló la aplicación de Base de datos y Web, para apoyar a un Acuario en su administración y en su divulgación. Contribuye al manejo adecuado de la información y el acceso a los datos de forma fácil, confiable y segura y podrá ponerse en función en el momento que se decida; Con lo que se ha cumplido con las especificaciones establecidas para esta etapa del proyecto.

La página esta ligada a una base de datos desarrollada en MySQL 5.0 como sistema administrador de base de datos, que es una buena alternativa, ya que cuenta con licencia pública general (GNU), y con libre acceso a su código fuente, proporciona rapidez y es capaz de almacenar grandes cantidades de datos, cuenta con la opción de protección de contraseña, la cual es flexible y segura y además posee gran integración con Php para el desarrollo de interfaces, lo cual permitió obtener los resultados deseados dentro del sistema.

Otro punto importante es que aprendí a manejar ambas herramientas las cuales son de mucha utilidad, y son utilizadas por muchas empresas lo que me permite contar con elementos para desarrollarme en el ámbito profesional.

Bibliografía.

- [1] Abraham Silberschatz, Henry F. Korth, S. Sudarshan
“Fundamentos de Bases de Datos”.
Mc. Graw Hill, Quinta Edición, 2006, México
- [2] <http://atenea.udistrital.edu.co/profesores/jdimate/basedatos1/>
- [3] Alberto López Aguilar.
“Desarrollo de Aplicaciones Web usando UML”
Asesor: Dr. Abraham Sánchez López.
Tesis de Licenciatura 2005, BUAP, Fac.de Cs. de la Computación, Pue.Mexico
- [4] Comunidad hispana de MySQL. Artículo: Paquetes de instalación
Apache/PHP/MySQL.
<http://www.mysql-hispano.org/page.php?id=17>
- [5] Guía Didáctica para la integración de bases de datos en Internet.
<http://www.uca.edu.sv/investigacion/bdweb/indice.html#contenido>
- [6] García Molina Jesús J.1, Moreira Ana2, Rossi Gustavo3. “Presentación UML:
el lenguaje estándar para el modelado de software”, Facultad de Informática,
Universidad Nacional de La Plata (Argentina), Marzo-Abril 2004
<http://www.ati.es/novatica/2004/168/168-4.pdf>
- [7] García De Mateos Juan Jimeno, Herrera González Patricia, Pérez Luján M^a
Carmen. “Metodologías de desarrollo de aplicaciones Web”, 2003/2004
<http://alarcos.inf-cr.uclm.es/doc/aplicabdd/DASBD-UWE.pdf>
- [8] J.Rumbaugh, I.Jacobson , G.Booch.
“El Proceso Unificado de Desarrollo de Software”
Addison-Wesley, 1999.
- [9] http://www.webbress.com.ar/web/sitios/web_proyectoisi/download/diagramas_del_uml.pdf
- [10] <http://www-gris.det.uvigo.es/~avilas/UML/node31.html>
- [11] http://www.conganat.org/Seis/inforsalud04/2004_Inforsalud_TutorialUML-UP.doc
- [12] <http://es.tldp.org/Tutoriales/NOTAS-CURSO-BBDD/notas-curso-BD.pdf>
- [13] http://www.netpecos.org/docs/mysql_postgres/x57.html
- [14] <http://es.wikipedia.org/wiki/.php>
- [15] <http://es.wikipedia.org/wiki/XAMPP>
- [16] https://www.ibercom.com/soporte/index.php?_m=knowledgebase&_a=viewarticle&kbarticleid=1449