



**BENEMERITA UNIVERSIDAD AUTONOMA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN**

Tesis

Sistema Administrativo de una Papelería

Presenta:

Angélica Castillo Hernández

**Que para obtener el título de
Licenciado en Ciencias de la Computación**

Asesora:

M.E. Carmen Cerón Garnica

CONTENIDO

Introducción -----	5
Objetivo General-----	5
Objetivo Especifico-----	5
Capítulo 1. Marco Teórico-----	7
1.1 Ingeniería de software-----	7
1.1.1 Paradigmas de la ingeniería del software-----	7
1.1.2 Modelo clásico o secuencial-----	8
1.1.3 Construcción de prototipos-----	9
1.1.4 El modelo de espiral-----	10
1.1.5 Técnicas de cuarta Generación-----	11
1.2 Bases de datos-----	12
1.2.1 Sistema Gestión de Base de Datos-----	12
1.2.2 Modelos de Datos-----	12
1.2.3 Lenguajes de los SDGB-----	13
1.2.4 Clasificación de los sistemas de gestión de base de datos-----	13
1.2.5 Características de las bases de datos-----	14
1.2.6 Modelos relacionales-----	14
1.2.7 Bases de Datos Relacionales-----	15
1.2.8 Tipos de Bases de datos-----	15
1.2.9 Normalización-----	17
1.2.9.1 Grados de normalización-----	17
1.3 Ingeniería WEB-----	18
1.4 Modelo UML-----	20
1.4.1 El Proceso Unificado de Desarrollo de Software-----	25
1.5 Herramientas-----	25
1.5.1 WIN IIS25-----	25
1.5.2 PHP-----	26

1.5.3	MySQL-----	28
1.5.3.1	Historia de MySQL-----	29
1.5.3.2	Características principales-----	29
1.5.3.3	Ventajas-----	29
1.5.3.4	Desventajas-----	29
	Capítulo 2. Análisis del sistema-----	30
2.1	Planteamiento del problema-----	30
2.2	Casos de uso-----	30
2.2.1	Caso de uso Alta de Productos-----	32
2.2.2	Caso de uso Escoger Producto-----	33
2.2.3	Caso de uso Registro de Cliente-----	34
2.2.4	Caso de uso Envío de Productos-----	35
2.3	Diagramas de secuencia-----	36
2.3.1	Diagrama de secuencia de Alta de Producto-----	36
2.3.2	Diagrama de secuencia de Escoger Producto-----	37
2.3.3	Diagrama de secuencia de Registro de Cliente-----	38
2.3.4	Diagrama de secuencia de Envío de Producto-----	39
	Capítulo 3. Diseño del Sistema-----	40
3.1	Arquitectura del Sistema-----	40
3.2	Diagramas de Actividades-----	41
3.2.1	Diagrama de actividad para Alta de productos-----	41
3.2.2	Diagrama de actividad de Escoger Producto-----	41
3.2.3	Diagrama de actividad Registro de Cliente-----	42
3.2.4	Diagrama de actividad para Envío de Productos-----	42
3.3	Diseño del Sistema-----	43
3.3.1	Diagrama de bases de datos-----	43
3.3.2	Análisis de los datos-----	44
3.3.3	Normalización-----	45
3.3.4	Diccionario de datos-----	45
3.4	Diagrama de Navegación-----	47

3.5	Diseño de interfaz-----	48
3.5.1	Interfaz Inicio -----	48
3.5.2	Interfaz Bienvenida-----	48
3.5.3	Interfaz Promociones-----	49
	Capítulo 4. Desarrollo y Pruebas del sistema-----	50
4.1	Conectividad de la Base de Datos	50
4.2	Módulos	50
4.2.1	Tabla de Proveedores	50
4.2.2	Tabla Proveedor Producto-----	51
4.2.3	Tabla Pedido-----	51
4.2.4	Tabla Registro Cliente-----	52
4.2.5	Forma de Envío-----	52
4.2.6	Producto-----	53
4.2.7	Alta de Cliente-----	54
4.2.8	Compra -----	54
4.2.9	Captura de Proveedores-----	55
4.3.0	Login de Alta de Productos	56
4.3.1	Envío de Productos	56
	Conclusiones	57
	Referencias	58

Introducción

Todos los negocios tienen como punto de partida la superación día con día y llegar de a ser una gran empresa. Las grandes empresas papeleras en el estado de Puebla son muy pocas, y pequeñas papelerías hay demasiadas, todas ellas buscan el crecimiento interno, buscan el mejoramiento de los procesos administrativos ya que estos se realizan de manera manual, esto es, si desean realizar la compra de artículos deberán ir con el proveedor directamente y realizar el pedido, el manejo de inventario o almacenamiento de productos se realiza de manera también manual por medio de formatos escritos, el área administrativa se encarga de colocar precios a los artículos del almacén para poder sacar el producto a la venta. Los clientes al comprar un producto podrán observar el precio y realizar la compra si así lo desean, el pago del producto se realizará en efectivo.

Hay algunas tiendas que utilizan algún tipo de software para cubrir sus necesidades como por ejemplo Tiendas Tony, Escorpión, hiperlumen, Papelerías Lozano, entre otros todas ellas utilizan software diseñados especialmente para sus necesidades.

Objetivo General:

Automatizar y controlar el manejo de las compras de una papelería vía Web.

Objetivo Especifico:

- ◆ El cliente deberá tomar de una lista de productos clasificados por tipo, el producto que desee, de tal manera que al escoger el tipo y el número de productos los agregará al carrito de compras.
 - ◆ Podrá escoger listas de útiles ya realizadas por las escuelas y lo único que tendrán que escoger es el nombre de la escuela, grado y grupo e introducir los productos de la lista al carrito de compras
 - ◆ El cliente podrá registrarse para realizar la compra. Dicho registro constará de sus datos personales del cliente como nombre, dirección, c.p., teléfono e-mail.
-

- ◆ Podrá formalizar su pedido y observar los productos que ha puesto en el carrito, así como también su costo unitario, subtotal y total de su compra.
- ◆ Al dar aceptar en su compra, tendrá que indicar el lugar donde se van a enviar sus productos así que tendrá que llenar un formulario para indicar dicho lugar, según el lugar de envío tendrá que ser el costo del mismo.

Organización de la tesis

- **Capítulo 1**

En este capítulo se mostrarán las definiciones de términos, conceptos fundamentales que se involucren en la realización de este proyecto, Marco Teórico. Ingeniería del software. Así como se explicará los fundamentos de una base de datos, ventajas, desventajas, características, tipos de bases de datos en que consisten los Sistemas de gestión de una Base de Datos (SGBD) sus clasificaciones, su normalización, ingeniería WEB, modelo UML, sus objetivos, diagramas, las herramientas utilizadas WIN, IIS, PHP, MySQL,

- **Capítulo 2**

En el capítulo dos se establece el planteamiento del problema, los casos de uso y diagramas de secuencia de los que dependerá el sistema auxiliándose de la metodología UML para el desarrollo del análisis del sistema.

- **Capítulo 3**

En este capítulo se trabajará con la arquitectura y los diagramas de colaboraciones y actividades del sistema dándole continuidad a la metodología UML, la forma de Entidad-Relación así como su diagrama y los diccionarios de datos, y la normalización de las tablas que salgan resultantes.

- **Capítulo 4**

En este capítulo se mostrará el ambiente de desarrollo así como la implementación del sistema.

1. Marco Teórico

1.1 Ingeniería de software

Existen diversas definiciones acerca de la ingeniería del software, pero hasta ahora este término no ha sido incluido ni en el Diccionario de la Real Academia de la lengua Española, ni en la Real Academia de la Lengua Española de Ciencias. A continuación se mencionan algunas de las definiciones para esta disciplina

- La ingeniería de software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadoras y la documentación asociada requerida para desarrollar, operar (funcionar) y mantenerlos.
- La ingeniería del Software es una actividad de modelado. En el curso del desarrollo, los ingenieros de software constituyen modelos diferentes del sistema y del dominio de aplicación.
- La ingeniería de Software es una actividad para la solución de problemas. Se usan modelos para buscar una solución aceptable.
- La ingeniería de software es una actividad para adquisición de conocimiento. Se recopilan los datos, se organizan en información y se formalizan en conocimientos.

Un modelo es una representación abstracta de un sistema que nos permite responder preguntas acerca del sistema.

1.1.1 Paradigmas de la ingeniería de software

La ingeniería del software abarca un conjunto de elementos:

- ◆ Los métodos
- ◆ Herramientas
- ◆ Procedimientos.

Los **métodos** indican como construir técnicamente el software, las **herramientas** suministran un soporte automático para los métodos; los **procedimientos** son el pegamento que junta los métodos y las herramientas.

Estos pasos se denominan frecuentemente paradigmas del software, entre estos pasos se tienen los siguientes que son las más frecuentes en cuanto a su utilización:

- Clásico o modelo secuencial
- Diseño por prototipos
- Modelo de espiral
- Técnicas de 4ª generación

1.1.2 Modelo clásico o secuencial.

Esta aproximación ve el proceso de software como un conjunto de etapas sucesivas. Después de que se define una, se prueba y se procede a desarrollar la siguiente etapa. El modelo está formado por las siguientes etapas:

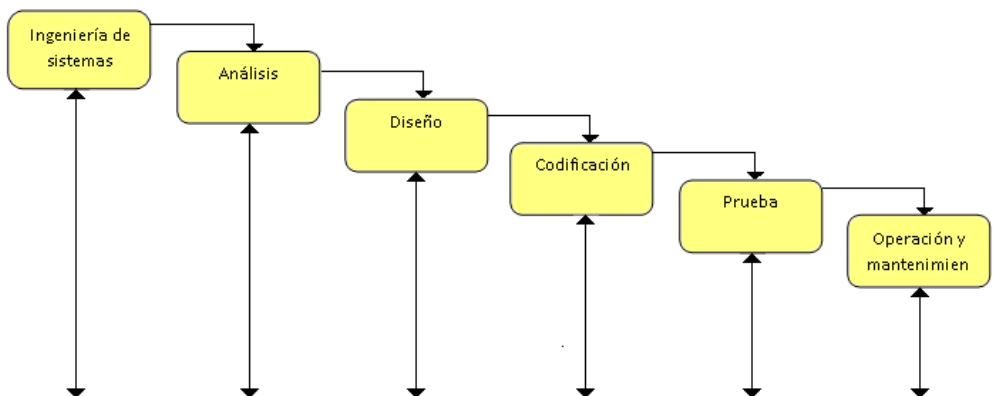


Figura 1.1 Ciclo de vida clásico

- **Ingeniería de sistemas.**- aquí se establecen los requisitos de todos los elementos del sistema y luego es asignado algún subconjunto de estos requisitos.

- **Análisis.-** Para comprender la naturaleza de los programas que hay que construir, se debe comprender el ámbito de la información del software, así como la función, el requerimiento y las interfaces requeridas.
- **Diseño.-** El proceso de diseño del sistema particiona los requisitos en sistemas de hardware y software, que establece la arquitectura completa del sistema.
- **Codificación y prueba.-** Las unidades del programa o programas se integran y prueban como un sistema completo para asegurar que los requerimientos del software se cumplan. Después de probar el sistema, libera y entrega al cliente.
- **Operación y mantenimiento.-** El software, indudablemente, sufrirá cambios después de que se entregue al cliente. Los cambios ocurrirán debido a que se hayan encontrado errores, a que el cambio solicitado debido a que se tiene un nuevo sistema operativo o dispositivo periférico, o debido a que el cliente requiera ampliaciones funcionales o del rendimiento.

1.1.3 Construcción de prototipos

La construcción de prototipos es un proceso que facilita al programador la creación de un modelo de software. El modelo tomará una de las siguientes formas.

- Un prototipo en papel o un modelo basado en PC que describa la interacción entre hombre-máquina, esto facilita al cliente la comprensión de cómo se producirá tal iteración.
- Un prototipo que implemente algunos subconjuntos de la función requerida del programa deseado.
- Un programa existente que ejecute parte o toda la función deseada, pero que tenga otras características que deberán ser mejoradas en el nuevo trabajo de desarrollo.

La construcción de prototipos inicia con la recolección de los requerimientos. El desarrollo y el cliente se reúnen y definen los objetivos globales para el software, identifican todos los requisitos conocidos y perfilan las áreas donde será necesaria una mayor definición.

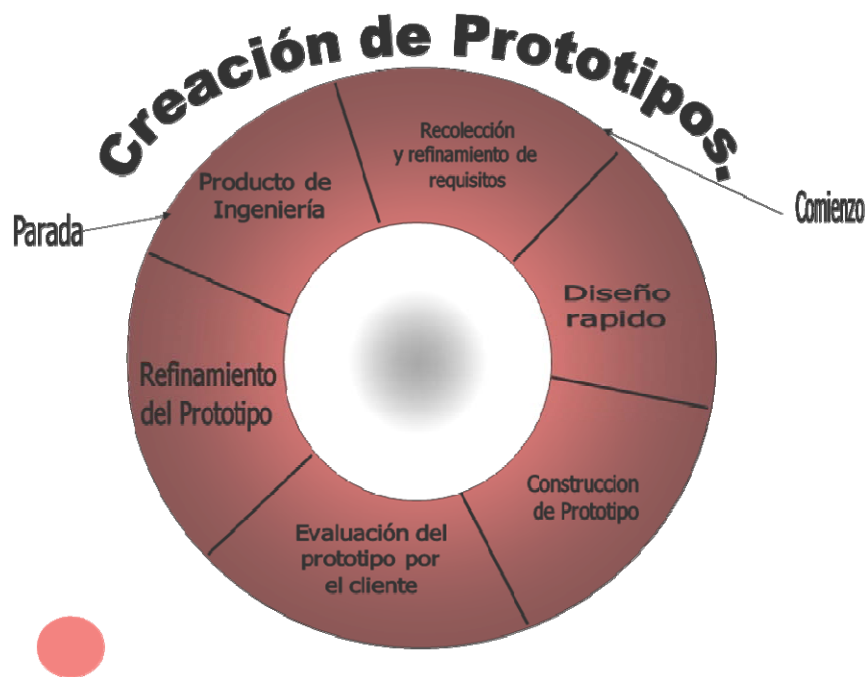


Figura 1.2 Creación de prototipos

1.1.4 El modelo de espiral

El modelo define cuatro actividades principales, que se presentan mediante la siguiente figura:

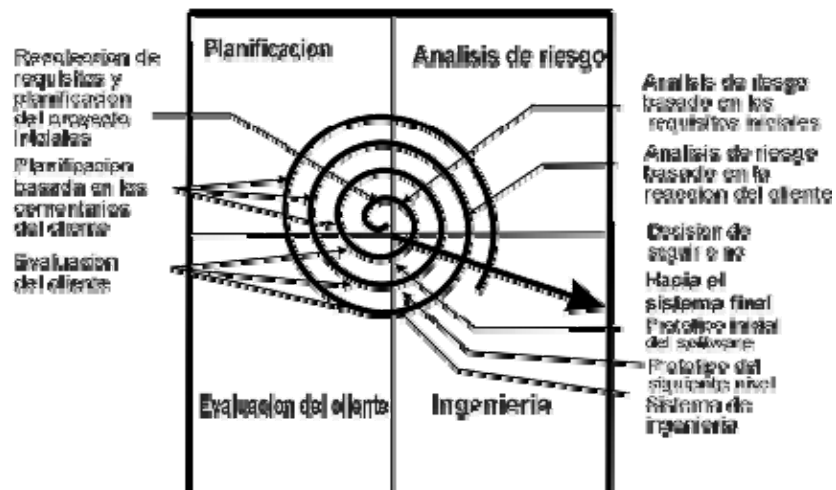


Figura 1.3 el modelo de espiral

- 1.- **Planificación:** Determinación de los objetivos, alternativas y restricciones.
- 2.- **Análisis de origen:** Análisis de alternativas e identificación y/o resolución de riesgo.
- 3.- **Ingeniería:** Desarrollo del producto del siguiente nivel.
- 4.- **Evaluación del cliente:** Valoración de los resultados de la ingeniería.

1.1.5 Técnicas de cuarta Generación.

Este paradigma se orienta hacia la posibilidad de especificar el software a un nivel más próximo al lenguaje natural, o en una notación que proporcione funciones significativas.

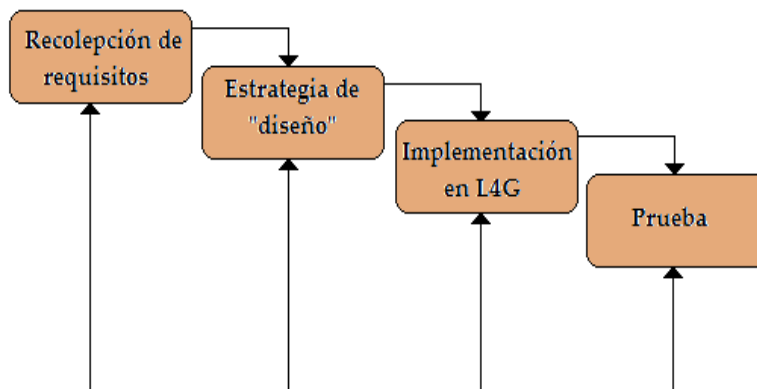


Figura 1.4 Técnicas de cuarta generación

Inicia con la recolección de requisitos. Idealmente el cliente los describe, estos son traducidos directamente a un prototipo operativo, usando un lenguaje de cuarta generación. La implementación mediante un lenguaje de cuarta generación, permite al desarrollador, centrarse en la representación de los resultados deseados, que es lo que se traduce automáticamente en el código fuente que produce dichos resultados.

Durante la implementación los desarrolladores traducen el modelo de objetos en código fuente.

1.2.- Bases de Datos

Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que posteriormente se pueda encontrar y utilizar fácilmente. Las bases de datos son creadas y manejadas por un Sistema Manejador de Bases de Datos (DBMS).

El término de bases de datos fue escuchado por primera vez en 1963, en un simposio celebrado en California, USA. Una base de datos se puede definir como un conjunto de información relacionada que se encuentra agrupada ó estructurada.

1.2.1 Sistema Gestión de Base de Datos

Un DBMS es un conjunto de funciones de software, donde cada una de éstas se encarga de realizar una tarea específica, además un DBMS es la interface entre el cliente y los datos, ya que mediante éste se realiza de manera eficiente la inserción, extracción y manipulación de la información en la base de datos.

Los objetivos de un Sistema de Gestión de base de Datos

- **Independencia.** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Seguridad.** Los SGBD deben garantizar que esta información se encuentra segura de permisos a clientes y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- **Tiempo de respuesta.** Lógicamente, es deseable minimizar el tiempo que el SGBD tarda en darnos la información solicitada y en almacenar los cambios realizados.

1.2.2 Modelos de Datos.

Los modelos de datos son conjuntos de conceptos que sirven para la estructura de la base de datos. Los modelos de datos pueden clasificarse dependiendo de los tipos de conceptos que ofrecen para describir la estructura de los datos: los modelos de datos de alto nivel o conceptuales, disponen de conceptos muy cercanos en el que la mayoría de los usuarios perciben los datos, los modelos de bajo nivel o físicos proporcionan conceptos que describen los detalles de cómo se almacenan los datos en el ordenador, entre estos dos extremos se encuentra el modelo lógico cuyos conceptos pueden ser entendidos por usuarios finales.

1.2.3 Lenguajes de los SDGB

Todos los SGBD ofrecen lenguajes e interfaces apropiadas para cada tipo de usuario: administradores, diseñadores, programadores de aplicaciones y usuarios finales. Los lenguajes van a permitir al administrador de la BD especificar los datos que componen la BD, su estructura, las relaciones que existen entre ellos, las reglas de integridad, los controles de acceso, las características de tipo físico y las vistas externas de los usuarios.

Un SDGB posee un compilador del **lenguaje de definición de datos (LDD)** cuya función consiste en procesar las sentencias del lenguaje para identificar las descripciones de los distintos elementos de los esquemas y almacenar la descripción del esquema en el catálogo o diccionario de datos. Cuando en un SGBD hay una clara separación entre los niveles conceptuales e interno, el lenguaje de definición de datos sólo sirve para especificar el esquema conceptual. Para especificar el esquema externo se utiliza un lenguaje de definición de datos de almacenamiento (LDA).

Una vez creados los esquemas de la base de datos, los usuarios necesitan un lenguaje que le permita manipular los datos es decir: realizar consultas, inserciones, eliminaciones y modificaciones a este tipo de lenguaje se le conoce como: **lenguaje de manipulación de datos (LMD)**. Hay dos tipos de lenguajes de LMD: los procedurales y los no procedurales.

Los procedurales el usuario especifica que datos se necesitan y como hay que obtenerlos. Los no procedurales se pueden utilizar de manera independiente para especificar operaciones complejas sobre la base de datos de forma concisa. A los LMD no procedurales también se les denomina declarativos. Las bases de datos relacionales utilizan un LMD no procedurales, como SQL.

1.2.4 Clasificación de los sistemas de gestión de base de datos

El criterio principal que se utiliza para clasificar los SGBD es el modelo lógico en que se basan. Los modelos lógicos empleados con mayor frecuencia en los SGBD comerciales actuales son el relacional, el de red y el jerárquico.

- El *modelo relacional* se basa en el concepto matemático denominado "relación", que gráficamente se puede representar como una tabla. En el modelo relacional, los datos y las relaciones existentes entre los datos se representan mediante estas relaciones matemáticas, cada una con un nombre que es único y con un conjunto de columnas.

- En el *modelo de red* los datos se representan como colecciones de registros y las relaciones entre los datos se representan mediante conjuntos, que son punteros en la implementación física. Los registros se organizan como un grafo: los registros son los nodos y los arcos son los conjuntos.
- El *modelo jerárquico* es un tipo de modelo de red con algunas restricciones, en el modelo jerárquico, cada nodo puede tener un sólo padre. Una base de datos jerárquica puede representarse mediante un árbol: en donde los registros son los nodos y los arcos son los conjuntos.

1.2.5 Características de las bases de datos

Entre las principales características de los sistemas de base de datos podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

1.2.6 Modelos relacionales

El modelo relacional se basa en dos ramas de las matemáticas: la teoría de conjuntos y la lógica. Al mismo tiempo, estas ramas de las matemáticas proporcionan los elementos básicos necesarios para crear una base de datos relacional con una buena estructura, y proporcionan las líneas que se utilizan para formular buenas metodologías de diseño.

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

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

- **Estructura de datos:** Es una forma de organizar un conjunto de datos elementales con el objetivo de facilitar su manipulación.
- **Integridad de datos.** Se refiere a la validez y la consistencia de los datos almacenados. Normalmente, la integridad se expresa mediante restricciones o reglas que no se pueden violar.
- **Manejo de datos.** Permiten una perfecta administración de la información que entra al sistema. Nos permite tener una mejor relación entre ellos, semántica asociada a los datos y restricciones de consistencia.

1.2.7 Bases de Datos Relacionales.

Edgar Frank Codd definió las bases del modelo relacionales a finales de los 60.

Trabajaba para IBM empresa que tardó un poco en implementar sus bases. Pocos años después el modelo se empezó a implementar cada vez más, hasta ser el modelo de bases de datos más popular.

Una base de datos relacional es una base de datos en donde todos los datos visibles al usuario están organizados estrictamente como tablas de valores, y en donde todas las operaciones de la base de datos operan sobre estas tablas.

Características principales de los "archivos" relacionales:

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

1.2.8 Tipos de Bases de datos

Modelo Jerárquico

Esta base de datos tiene como objetivo establecer una jerarquía de fichas, de manera que cada ficha puede contener a su vez listas de otras fichas, y así sucesivamente.

Una base de datos jerárquica está compuesta por una secuencia de bases de datos físicas, de manera que cada base de datos física se compone de todas las ocurrencias de un tipo de registro o ficha determinada.

Modelo en red.

Podemos considerar al modelo de bases de datos en red como de una potencia intermedia entre el jerárquico y el relacional que estudiaremos más adelante. Su estructura es parecida a la jerárquica.

Bases de datos analíticas

Estas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

Bases de datos operacionales

Estas son bases de datos más dinámicas, orientadas a almacenar información que es modificada con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta.

Tablas

Las bases de datos relacionales se basan en el uso de tablas (también se las llama **relaciones**). Las tablas se representan gráficamente como una estructura rectangular formada por filas y columnas. Cada columna almacena información sobre una propiedad determinada de la tabla (se le llama también **atributo**), nombre, dni, apellidos, edad,....

Cada fila posee una ocurrencia o ejemplar de la instancia o relación representada por la tabla (a las filas se las llama también **tuplas**).

Terminología relacional

- **Tupla.** Cada fila de la tabla (cada ejemplar que la tabla representa)
- **Atributo.** Cada columna de la tabla

- **Cardinalidad.** Número de tuplas de una tabla
- **Dominio.** Conjunto válido de valores representables por un atributo.

Tipos de tablas

- **Persistentes.** Sólo pueden ser borradas por los usuarios:
- **Base.** Independientes, se crean indicando su estructura y sus ejemplares.
- **Vistas.** Son tablas que sólo almacenan una definición de consulta, resultado de la cual se produce una tabla cuyos datos proceden de las bases o de otras vistas e instantánea. Si los datos de las tablas base cambian, los de la vista que utiliza esos datos también cambian.
- **Instantáneas.** Son vistas (creadas de la misma forma) que sí que almacenan los datos que muestra, además de la consulta que dio lugar a esa vista. Sólo modifican su resultado (actualizan los datos) siendo refrescadas por el sistema cada cierto tiempo.
- **Temporales.** Son tablas que se eliminan automáticamente por el sistema. Pueden ser de cualquiera de los tipos anterior.

Dominios

Los dominios suponen una gran mejora en este modelo ya que permiten especificar los posibles valores válidos para un atributo. Cada dominio incorpora su nombre y una definición del mismo.

Ejemplos de dominio:

- **Dirección:** 50 caracteres
- **Nacionalidad:** Español, Francés, Italiano,...

1.2.9 Normalización

Normalización es un conjunto de reglas que sirven para ayudar a los diseñadores a desarrollar un esquema que minimice los problemas de lógica. Cada regla está basada en la que le antecede. La normalización se adoptó porque el viejo estilo de poner todos los datos en un solo lugar, como un archivo o una tabla de la base de datos, era ineficiente y conducía a errores de lógica cuando se trataba de manipular los datos.

1.2.9.1 Grados de normalización

Existen básicamente tres niveles de normalización: Primera Forma Normal (1NF), Segunda Forma Normal (2NF) y Tercera Forma Normal (3NF). Cada una de estas formas tiene sus

propias reglas. Cuando una base de datos se conforma a un nivel, se considera normalizada a esa forma de normalización.

Primera forma normal (1FN)

La regla de la Primera Forma Normal establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas. Ésta es una regla muy fácil de seguir.

Segunda forma normal (2FN)

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

Tercera forma normal (3FN)

La regla de la Tercera Forma Normal señala que hay que eliminar y separar cualquier dato que no sea clave. El valor de esta columna debe depender de la clave. Todos los valores deben identificarse únicamente por la clave.

1.3 Ingeniería WEB

La ingeniería de la Web es la aplicación de metodologías sistemáticas, disciplinarias, cuantificables al desarrollo eficiente, y evolutivas de aplicables con alta calidad en la World Wide Web.

El desarrollo de sistemas y aplicaciones basados en Web representan una secuencia de acciones de ingeniería Web que comienzan con la identificación de las necesidades del negocio, sigue hacia una descripción de los objetivos de la Web.

En la actualidad la Web esta sufriendo grandes cambios, que han obligado a expertos en el tema a utilizar herramientas y técnicas basadas en la ingeniería del software, para poder garantizar el buen funcionamiento y administración de los sitios Web.

Ahora para garantizar el buen funcionamiento y mantenimiento de los sitios Web, este debe contar con cierto número de atributos y características, que en conjunto forman un concepto muy

importante, para alcanzar el éxito en cualquier organización y que se pueda considerar como servicio. Dicho concepto es la calidad, que con atributos como, usabilidad, navegabilidad, seguridad, mantenibilidad, entre otros, hace posible por un lado la eficiencia del artefacto Web y por otro la satisfacción del usuario final.

Cabe destacar que la ingeniería de la Web hace una diferencia entre un sitio Web y un aplicativo, ya que la ingeniería de la Web no se dedica a la construcción de sitios Web si no a la construcción de aplicativos Web, la principal característica que los distingue es que los sitios Web son sitios en la Web en donde se publica contenido generalmente estático o un muy bajo nivel de interactividad con el usuario, mientras que los aplicativos son lugares con alto contenido de interactividad y funcionalidades que bien podrían ser de un software convencional, el aplicativo Web más sencillo sería uno que contenga formularios y subiendo de nivel encontramos los que realizan conexiones con bases de datos remotas, y administradores de contenidos entre otras.

La ingeniería de la Web es multidisciplinar y aglutina contribuciones de diferentes áreas como : arquitectura de la información, ingeniería de hipermedia/hipertexto, diseño de interfaz de usuario, diseño gráfico y de presentación, diseño y análisis de sistemas, ingeniería de software, ingeniería de datos, testeo, modelado y simulación, despliegue de aplicaciones, operación de sistemas y gestión de proyectos.

Los sitios Web pueden ser categorizados de la siguiente forma:

- **Sólo estático** que se enfoca en la organización de la estructura y el contenido, en la forma como se va a presentar la información y que sea fácil de manejar para cualquier usuario.
- Sitio **estático** con formularios de entrada, éste sitio tiene las mismas características que el anterior, adicionándole que le permite a los usuarios la interacción por medio de cuestionarios, comentario y sugerencias.
- Sitio creado **dinámicamente** en este sitio los requerimientos son parecidos pero deben cumplir las necesidades de cada usuario de manera interactiva.

Cualquier proyecto de software se inicia por alguna necesidad de negocio; la necesidad de corregir una deficiencia en alguna aplicación existente; la necesidad de adaptar una aplicación existente a un modelo de negocio cambiante; la necesidad de extender funciones o la necesidad de crear un nuevo producto o servicio.

1.4 Modelo UML

Cualquier rama de ingeniería o arquitectura ha encontrado útil desde hace mucho tiempo la representación de los diseños de forma gráfica. Desde los inicios de la informática se han estado utilizando distintas formas para representar los diseños de una forma más bien personal o con algún modelo gráfico. La falta de estandarización en la manera de representar gráficamente un modelo impedía que los diseños gráficos realizados se pudieran compartir fácilmente entre distintos diseñadores.

Se necesitaba por tanto un lenguaje no sólo para comunicar las ideas a otros desarrolladores sino también para servir de apoyo en los procesos de análisis de un problema. Con este objetivo se creó el Lenguaje Unificado de Modelado (UML: Unified Modeling Language). UML se ha convertido en ese estándar tan ansiado para representar y modelar la información con la que se trabaja en las fases de análisis y, especialmente, de diseño.

UML es ante todo un lenguaje que proporciona un vocabulario y reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema. Este lenguaje nos indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones en:

- **Visualizar:** UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- **Especificar:** UML permite especificar cuáles son las características de un sistema antes de su construcción.
- **Construir:** A partir de los modelos especificados se pueden construir los sistemas diseñados.
- **Documentar:** Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Un modelo UML está compuesto por tres clases de bloques de construcción:

- **Elementos:** Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)

- **Relaciones:** relacionan los elementos entre sí.
- **Diagramas:** Son colecciones de elementos con sus relaciones.

Un **diagrama** es la representación gráfica de un conjunto de elementos con sus relaciones. En concreto, un diagrama ofrece una vista del sistema a modelar. Para poder representar correctamente un sistema, UML ofrece una amplia variedad de diagramas para visualizar el sistema desde varias perspectivas. UML incluye los siguientes diagramas:

- **Diagrama de casos de uso:** Un *caso de uso* describe, —desde el punto de vista de los actores—, un grupo de actividades de un sistema que produce un resultado concreto y tangible. Los casos de uso son descriptores de las interacciones típicas entre los usuarios de un sistema y ese mismo sistema. Representan el interfaz externo del sistema y especifican qué requisitos de funcionamiento debe tener este (recuerde, únicamente el qué, nunca el cómo).

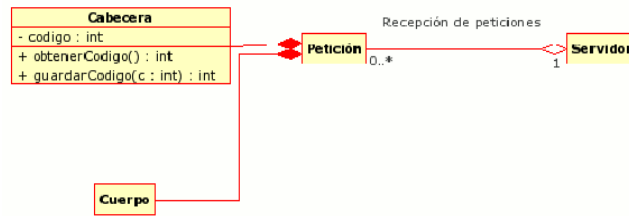
Cuando se trabaja con casos de uso, es importante tener presentes algunas sencillas reglas:

1. Cada caso de uso está relacionado como mínimo con un actor.
2. Cada caso de uso es un iniciador (es decir, un actor)
3. Cada caso de uso lleva a un resultado relevante (un resultado con “valor intrínseco”).

Actor

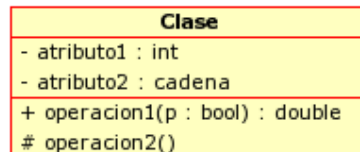
Los actores pueden ser gente real (por ejemplo, usuarios del sistema), otros ordenadores o eventos externos.

- **Diagrama de clases.** Los diagramas de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras. Se dice que los diagramas de clases son diagramas “estáticos” porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases “conocen” a qué otras clases o qué clases “son parte” de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.



Clase

Una clase define los atributos y los métodos de una serie de objetos. Todos los objetos de esta clase (instancias de esa clase) tienen el mismo comportamiento y el mismo conjunto de atributos.



Atributos

En UML, los atributos se muestran al menos con su nombre, y también pueden mostrar su tipo, valor inicial y otras propiedades.

Asociaciones

Una asociación representa una relación entre clases, y aporta la semántica común y la estructura de muchos tipos de “conexiones” entre objetos.

Las asociaciones son los mecanismos que permite a los objetos comunicarse entre sí. Describen la conexión entre diferentes clases (la conexión entre los objetos reales se denomina conexión de objetos o *enlace*).

Las asociaciones pueden tener un papel que especifica el propósito de la asociación y pueden ser unidireccionales o direccionales. Cada extremo de la asociación también tiene un valor de multiplicidad, que indica cuántos objetos de ese lado de la asociación están relacionados con un objeto del extremo contrario.

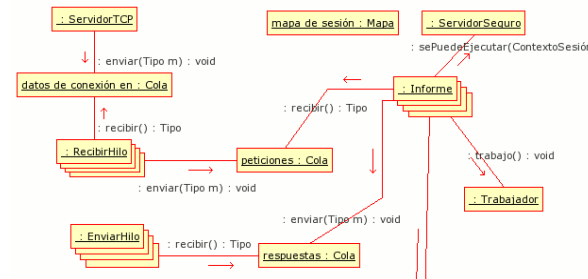
En UML, las asociaciones se representan por medio de líneas que conectan las clases participantes en la relación, y también pueden mostrar el papel y la multiplicidad de cada uno de los participantes. La multiplicidad se muestra como un rango [mín...máx] de valores no negativos, con un asterisco (*) representando el infinito en el lado máximo.



- Diagrama de objetos.
- Diagrama de secuencia.
- **Diagramas de secuencia**

Los diagramas de secuencia muestran el intercambio de mensajes (es decir la forma en que se invocan) en un momento dado. Los diagramas de secuencia ponen especial énfasis en el orden y el momento en que se envían los mensajes a los objetos.

En los diagramas de secuencia, los objetos están representados por líneas intermitentes verticales, con el nombre del objeto en la parte más alta. El eje de tiempo también es vertical, incrementándose hacia abajo, de forma que los mensajes son enviados de un objeto a otro en forma de flechas con los nombres de la operación y los parámetros.



- **Diagrama de actividades.**

Los diagramas de actividad describen la secuencia de las actividades en un sistema. Los diagramas de actividad son una forma especial de los diagramas de estado, que únicamente (o mayormente) contienen actividades.

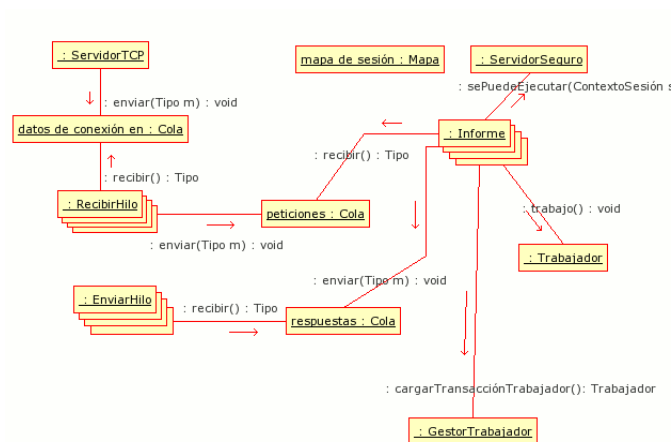
Los diagramas de actividad siempre están asociados a una *clase*, a una *operación* o a un *caso de uso*.

Los diagramas de actividad soportan actividades tanto secuenciales como paralelas.

• Diagrama de colaboración.

Los diagramas de colaboración muestran las interacciones que ocurren entre los objetos que participan en una situación determinada. Esta es más o menos la misma información que la mostrada por los diagramas de secuencia, pero destacando la forma en que las operaciones se producen en el tiempo, mientras que los diagramas de colaboración fijan el interés en las relaciones entre los objetos y su topología.

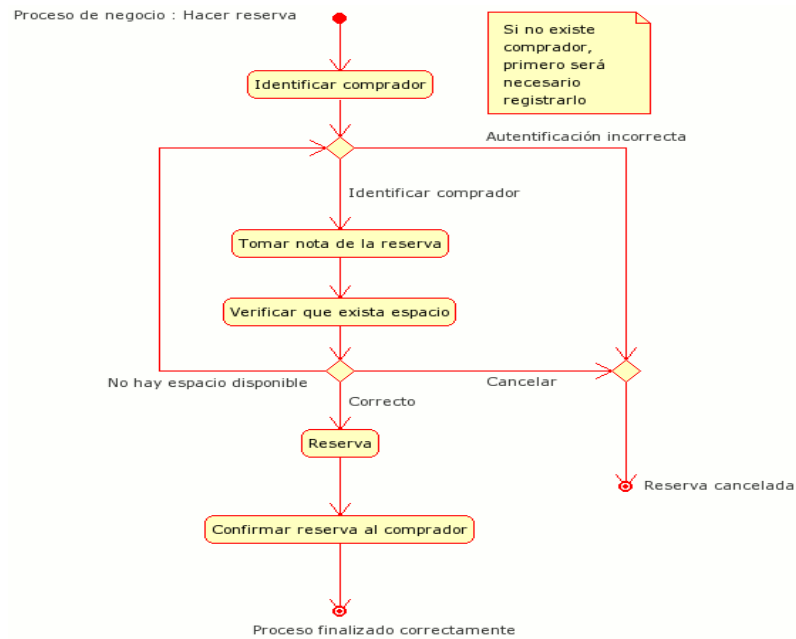
En los diagramas de colaboración los mensajes enviados de un objeto a otro se representan mediante flechas, mostrando el nombre del mensaje, los parámetros y la secuencia del mensaje.



• Diagrama de estados.

Los diagramas de estado muestran los diferentes estados de un objeto durante su vida, y los estímulos que provocan los cambios de estado en un objeto.

Los diagramas de estado ven a los objetos como *máquinas de estado* o autómatas finitos que pueden estar en un conjunto de estados finitos y que pueden cambiar su estado a través de un estímulo perteneciente a un conjunto finito.



1.4.1 El Proceso Unificado de Desarrollo de Software (RUP)

El Proceso Unificado es un proceso de software genérico que puede ser utilizado para una gran cantidad de tipos de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de competencia y diferentes tamaños de proyectos.

Provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de muy alta calidad que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecible.

1.5 Herramientas

1.5.1 WIN IIS

Los Servicios de Internet Information Services (IIS) proporcionan las herramientas y funciones necesarias para administrar de forma sencilla un servidor Web seguro. Si ha pensado alojar un sitio Web y FTP (File Transfer Protocol, Protocolo de transferencia de archivos) con IIS, configure el servidor como un servidor de aplicaciones.

IIS fue inicialmente lanzado como un conjunto de servicios basados en Internet para Windows NT 3.51. IIS 2.0 siguió agregando soporte para el sistema operativo Windows NT 4.0 e IIS 3.0 introdujo las Páginas activas de servidor, esta tecnología es una plataforma de scripting dinámico.

IIS 4.0 eliminó el soporte para el protocolo Gopher y fue puesto con Windows NT como un CD-ROM de "Paquete Opcional" separado.

La versión actual de IIS es la 6.0 para Windows Server 2003 e IIS 5.1 para Windows XP Professional. IIS 5.1 para Windows XP es una versión compacta del IIS que soporta sólo 10 conexiones simultáneas y sólo un sitio Web. IIS 6.0 ha agregado soporte para IPv6.

Windows Vista viene con IIS 7.0 preinstalado. No limitará el número de conexiones permitidas pero limitará el flujo de tareas basándose en las solicitudes activas concurrentes, mejorando la usabilidad y el rendimiento en escenarios punto-a-punto (peer-to-peer).

Las características agregadas en seguridad se aprovechan de las últimas tecnologías de cifrado y métodos de autenticación mediante certificados de cliente y servidor. Una de las formas que tiene IIS de asegurar los datos es mediante SSL (*Secure Sockets Layer*). Esto proporciona un método para transferir datos entre el cliente y el servidor de forma segura, permitiendo también que el servidor pueda comprobar al cliente antes de que inicie una sesión de usuario.

1.5.2 PHP

Todos los que nos hemos enfrentado con el diseño de páginas Web hemos echado de menos un poco más de dinamismo en ellas.

Representar una página repleta de gráficos y nada más, deja de ser suficiente para ciertas aplicaciones en Internet. Estas aplicaciones requieren de cierta interactividad con el usuario, y han sido muchas las tecnologías aplicadas a este fin.

HP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de la plataforma, rápido, con una gran librería de funciones y mucha documentación.

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 pagina WML.



Es un lenguaje opensource (software distribuido y desarrollado libremente), es interpretado, de alto nivel, enfocado al desarrollo de aplicaciones Web de contenido dinámico, incrustado sobre páginas HTML, de sintaxis sencilla, muy parecido a C, Java y Perl dentro de sus características se encuentran:

Es multiplataforma, al contar con una sintaxis parecida a C, Java y Perl, la mayoría de los usuarios con experiencia pueden utilizarlo sin mayor complicación, lenguaje que puede trabajar con programación estructurada, así como programación orientada a objetos, no requiere de la definición de variables, es un lenguaje que cuenta con un gran número de módulos o extensiones que lo potencializan.

PHP puede ser utilizado bajo la mayoría de los sistemas operativos actuales, llámese Unix, Linux, Windows, Mac OS, y también trabaja con la mayoría de servidores Web existentes tal es el caso de Apache, IIS, PWS y algunos más.

La seguridad, en este punto también es importante el hecho de que en muchas ocasiones PHP se encuentra instalado sobre servidores Unix o Linux, que son de sobra conocidos como más veloces y seguros que el sistema operativo donde se ejecuta las ASP, Windows NT o 2000. Además, PHP permite configurar el servidor de modo que se permita o rechacen diferentes usos, lo que puede hacer al lenguaje más o menos seguro dependiendo de las necesidades de cada cual.

Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. PHP se encuentra en su versión 4, que utiliza el motor Zend, desarrollado con mayor meditación para cubrir las necesidades de las aplicaciones Web actuales.

Algunas de las más importantes capacidades de PHP son: compatibilidad con las bases de datos más comunes, como MySQL, MSQL, Oracle, Informix, y ODBC, por ejemplo. Incluye funciones para el envío de correo electrónico, upload de archivos, crear dinámicamente en el

servidor imágenes en formato GIF, incluso animadas y una lista interminable de utilidades adicionales.

Trabaja con extensión DBX, también soporta el estándar abierto de conexiones de base de datos (ODBC).

Otra interesante área es la creación de aplicación de interfaz gráfica, sabemos que no es la principal potencialidad de PHP, pero se puede utilizar PHP-GTK (viene de la fusión del lenguaje de script PHP y la librería de objetos GTK+) para este fin, inclusive se puede escribir aplicaciones independientes de una plataforma.

1.5.3 MySQL

Es un gestor de base de datos relacionales, de código abierto, de los más populares en el mundo, el cual utiliza el estándar SQL, como lenguaje para acceder a las bases de datos, el desarrollo de MySQL obedeció al manejo de grandes bases de datos de una manera rápida y segura, lo cual ha logrado demostrar con éxito en los casos de producciones de alto rendimiento donde se le ha utilizado, actualmente nos ofrece una gran cantidad de funciones, dentro de las más importantes se encuentran la conectividad, la velocidad y la seguridad, lo cual hace a MySQL sea una excelente opción para el acceso a bases de datos en Internet.

Trabaja en un sistema cliente/servidor, que consiste en un servidor SQL multihilo ().

Fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca.

MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL.

lenguaje de programación que utiliza MySQL es Structured Query Language (**SQL**) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales.

1.5.3.1 Historia de MySQL

MySQL surgió alrededor de la década del 90, Michael Widenis comenzó a usar MSQL para conectar tablas usando sus propias rutinas de bajo nivel (ISAM). Tras unas primeras pruebas, llegó a la conclusión de que MySQL no era lo bastante flexible ni rápido para lo que necesitaba, por lo que tuvo que desarrollar nuevas funciones. Esto resultó en una interfaz SQL a su base de datos, totalmente compatible a MSQL.

El origen del nombre MySQL no se sabe con certeza de donde proviene, por un lado se dice que en sus librerías han llevado el prefijo “my” durante los diez últimos años, por otra parte, la hija de uno de los desarrolladores se llama My. Así que no está claramente definido cual de estas dos causas han dado lugar al nombre de este conocido gestor de bases de datos.

1.5.3.2 Características principales

Inicialmente, MySQL carecía de algunos elementos esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de esto, atrajo a los desarrolladores de páginas Web con contenido dinámico, debido a su simplicidad, de tal manera que los elementos faltantes fueron complementados por la vía de las aplicaciones que la utilizan. Poco a poco estos elementos faltantes, están siendo incorporados tanto por desarrolladores internos, como por desarrolladores de software libre.

1.5.3.3 Ventajas

Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.

Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.

Facilidad de configuración e instalación.

Soporta gran variedad de Sistemas Operativos

Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.

Conectividad y seguridad

1.5.3.4 Desventajas

Un gran porcentaje de las utilidades de MySQL no están documentadas.

No es intuitivo, como otros programas (ACCESS).

Capítulo 2. Análisis del Sistema

2.1 Planteamiento del Problema

El sistema a desarrollar tiene como objetivo automatizar el control de las compras de una papelería, el usuario podrá tomar de una lista de productos clasificados por tipo, el producto que desee, de tal manera que al escoger el mismo deberá indicar el número de productos que se está llevando y los agregará al carrito de compras. También podrá escoger listas de útiles ya establecidas por las escuelas indicadas y lo único que deberá hacer es escoger el nombre de la escuela, el grado, grupo y agregarla al carrito de compras. Después el usuario podrá registrarse para realizar la compra. Dicho registro del usuario será a través de sus datos personales, los cuales son: nombre, dirección, cp, teléfono, e-mail etc. Después tendrá que formalizar su pedido y podrá observar los productos que ha ido agregando al carrito, así como los subtotales, IVA y el total de su compra.

Al dar aceptar su compra tendrá que indicar el lugar donde se va a enviar sus productos, así que tendrá que llenar un formulario para indicar el lugar de envío.

Para el análisis y diseño del sistema se utilizará el proceso unificado de desarrollo de software. La captura de requerimientos se hará a través de los casos de uso y posteriormente se discutirá el diagrama general de clases de sistema. A continuación se discuten los diagramas de casos de uso.

2.2 Casos de uso

Los requerimientos del sistema son:

Los casos de uso plantean la acción que realizará el actor con el sistema durante la investigación en diferentes establecimientos de papelerías se recolectó la siguiente información para llevar el control detallado de la misma.

R1.- Proveedor

Este caso de uso tiene como objetivo llevar el control de los proveedores así como su información.

R2.- Producto

Este caso de uso tiene el objetivo de llevar un control de todos y cada uno de los productos que se esta manejando dentro de la papelería, así como también algunas características de ellos.

R3.- Pedido

Este caso de uso tiene el objetivo de llevar a cabo los pedidos que se realizaran en la papelería para controlar dichas compras.

R4.- Cliente

Este caso de uso el objetivo es llevara el registro de los clientes que desean realizar una compra en la papelería esto permitiría llevar un control de las personas que están comprando

R5.- Envío

Este caso de uso el objetivo es poder escoger la opción de envío, si desean que los productos se le lleven a un lugar o se le entregue la mercancía en el mismo lugar.

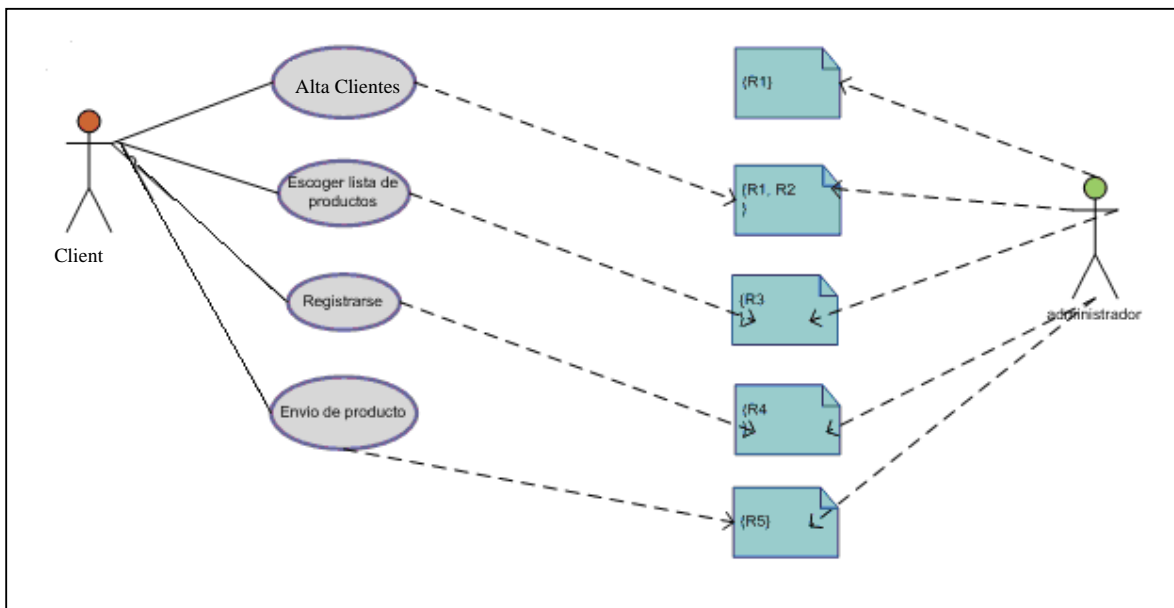


Fig. 2.1 Caso de uso

2.2.1 Caso de uso Alta de Productos

Nombre	Alta de productos
Objetivo	El Administrador del sistema WEB pueda dar de alta a nuevos productos
Actor	Administrador

FLUJO DE EVENTO	EVENTO	PROCESO DEL CASO
	1	El administrador escogerá la opción al alta de productos
	2	El sistema muestra la interna de introducir contraseña
	3	El sistema verifica la contraseña del usuario
	4	Si el administrador tiene acceso al sistema se muestra la página de alta de producto
	5	El administrador completa el formulario de alta de productos Clave del producto Nombre del producto Descripción Costo unitario
	6	El administrador acepta mediante el botón alta de producto
	7	El administrador decide dar de alta a otro producto o regresar a la opción principal

Estado de fin exitoso	El sistema indicará que la acción se realizo con éxito
Estado de fin fallido	El sistema advertirá que la acción de alta no se realizo con éxito
Variaciones	En caso de no tener acceso al sistema este mostrara un letrero de usuario no autorizado
Interfaz de cliente	Alta de productos

2.2.2 Caso de uso Escoger Producto

Nombre	Escoger producto
Objetivo	Que el cliente pueda escoger de una lista de producto los que vaya a comprar
Actor	Cliente

FLUJO DE EVENTO	EVENTO	PROCESO DEL CASO
	1	El cliente escoge la opción producto
	2	El sistema muestra la interfaz con la lista de productos
	3	El cliente escoge los productos a comprar
	4	El sistema marcara los productos
	5	El cliente agregara los productos al carrito
	6	Se abre el carrito de compra

Estado de fin exitoso	El sistema indicara que la acción de escoger producto fue terminado exitosamente
Estado de fin fallido	El sistema indicara que la acción de escoger el producto no fue exitosa
Variaciones	En caso de no querer agregar los productos al carrito saldrá de la interfaz escoger producto
Interfaz de usuario	Escoger producto

2.2.3 Caso de uso Registro de Cliente

Nombre	Registro de Clientes
Objetivo	El cliente llenara un formulario con sus datos personales para poder comprar productos
Actor	Cliente

FLUJO DE EVENTO	EVENTO	PROCESO DEL CASO
	1	El cliente escogerá la opción al registro de clientes
	2	El sistema muestra la interfaz de registro de clientes
	3	El sistema manda el enfoque al primer dato
	4	El cliente completará el formulario de registro de clientes Clave del cliente Nombre del cliente Apellido paterno Apellido materno Calle cliente Colonia cliente Ciudad cliente Teléfono cliente Mail cliente
	5	El cliente acepta mediante el botón registro de cliente
	6	El sistema verifica que todos los datos estén bien asignados
	7	El sistema guarda la información en la base de datos
	8	El sistema regresa a la pagina de inicio

Estado de fin exitoso	El sistema indicará que la acción de alta de cliente se realizo con éxito
Estado de fin fallido	El sistema advertirá que la acción de alta de cliente no se realizo con éxito
Variaciones	En caso de no tener los datos completos del formulario no dará de alta al cliente
Interfaz de cliente	Registro de clientes

2.2.4 Caso de uso Envío de Productos

Nombre	Envío de productos
Objetivo	El Administrador del sistema WEB pueda enviar los productos al lugar indicado por el cliente
Actor	Cliente

FLUJO DE EVENTO	EVENTO	PROCESO DEL CASO
	1	El cliente escogerá la opción envío de productos
	2	El sistema muestra la interfaz de envío de producto
	3	El sistema envía el enfoque al primer dato No de envío Lugar envío Dirección envío Calle Colonia Ciudad Código postal Forma pago
	4	El cliente acepta mediante el botón Enviar productos
	5	El sistema verifica que todos los datos estén bien asignados
	6	El sistema guarda la información en la base de datos
	7	El sistema regresa a la pagina de inicio

Estado de fin exitoso	El sistema indicará que la acción se realizo con éxito
Estado de fin fallido	El sistema advertirá que la acción de envío de productos no se realizo con éxito
Variaciones	En caso de no querer el envío saldrá de la interfaz envío de productos
Interfaz de cliente	Envío de productos

2.3 Diagrama de secuencia

Aquí se muestra los diagramas de secuencia que serán generados mediante los diferentes casos de uso que ya se describieron anterior mente.

2.3.1 Diagrama de secuencia de Alta de Producto

A qui se muestra la página de alta de producto, el administrador es el encargado de capturar los datos necesarios para introducirlos a la base de datos, los datos a capturar son: Clave del producto, Nombre del producto, Descripción, Costo unitario, al terminar la Captura se selecciona alta de registro con esto se conseguirá que se agregue lo capturado a la base de datos.

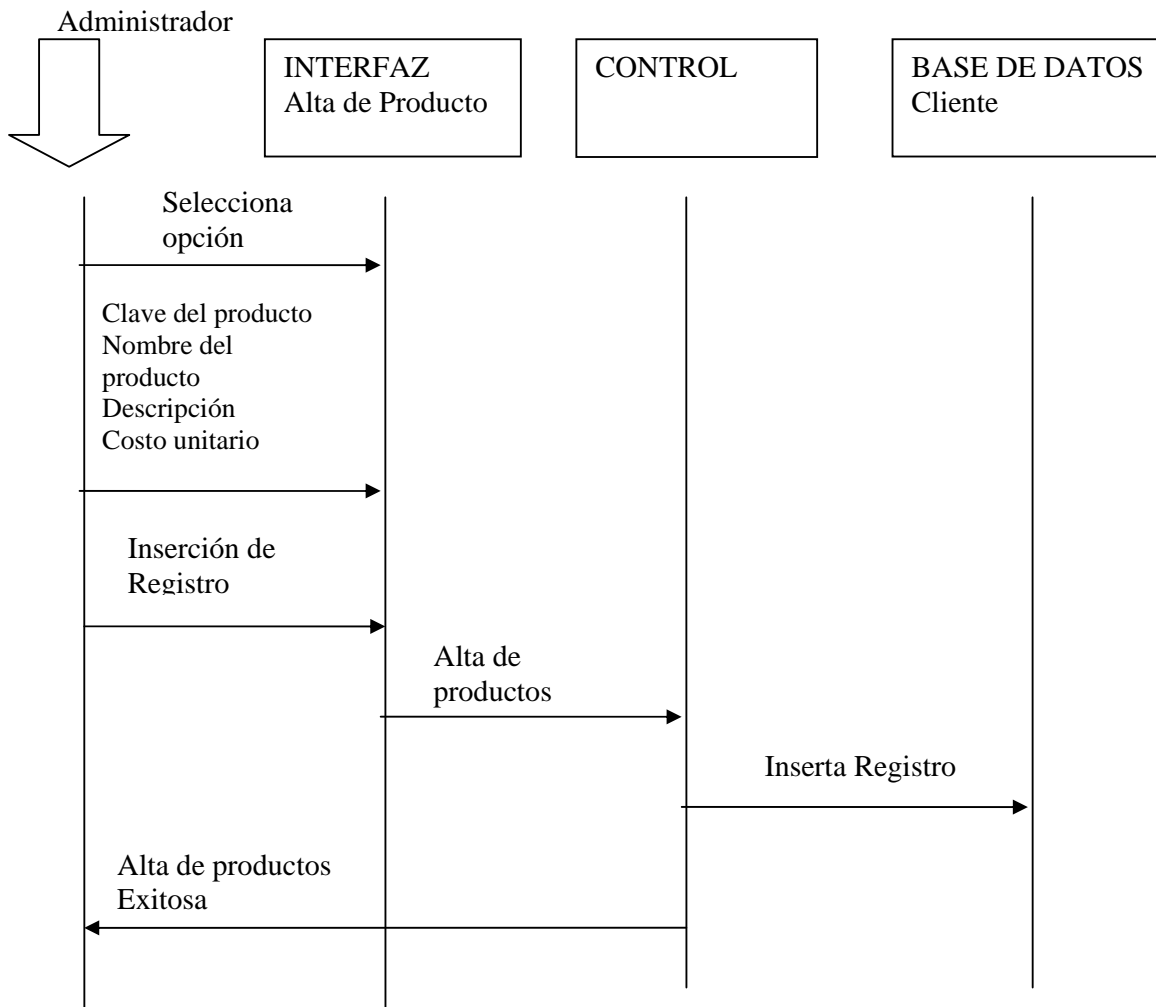


Fig. 2.2 Alta de de Producto

2.3.2 Diagrama de secuencia de Escoger Producto

Aquí se muestra una página donde el cliente puede escoger los productos que desea comprar, los datos que se solicitan son: clave del producto, nombre del producto, cantidad.

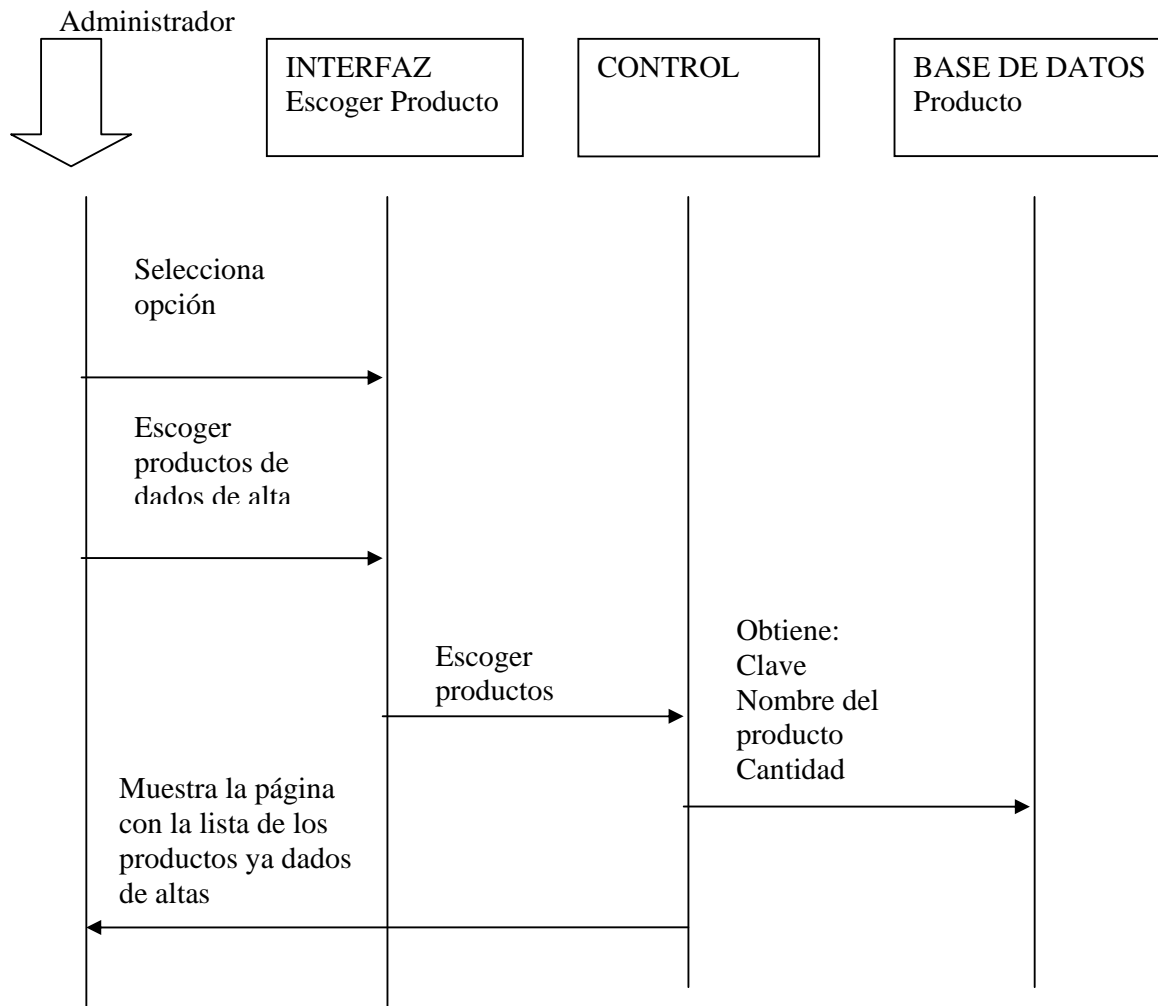


Fig. 2.3 Escoger Producto

2.3.3 Diagrama de secuencia de Registro de Clientes

Aquí se muestra una página donde el cliente puede escoger los productos que desea comprar, los datos que se solicitan son: clave del producto, nombre del producto, cantidad.

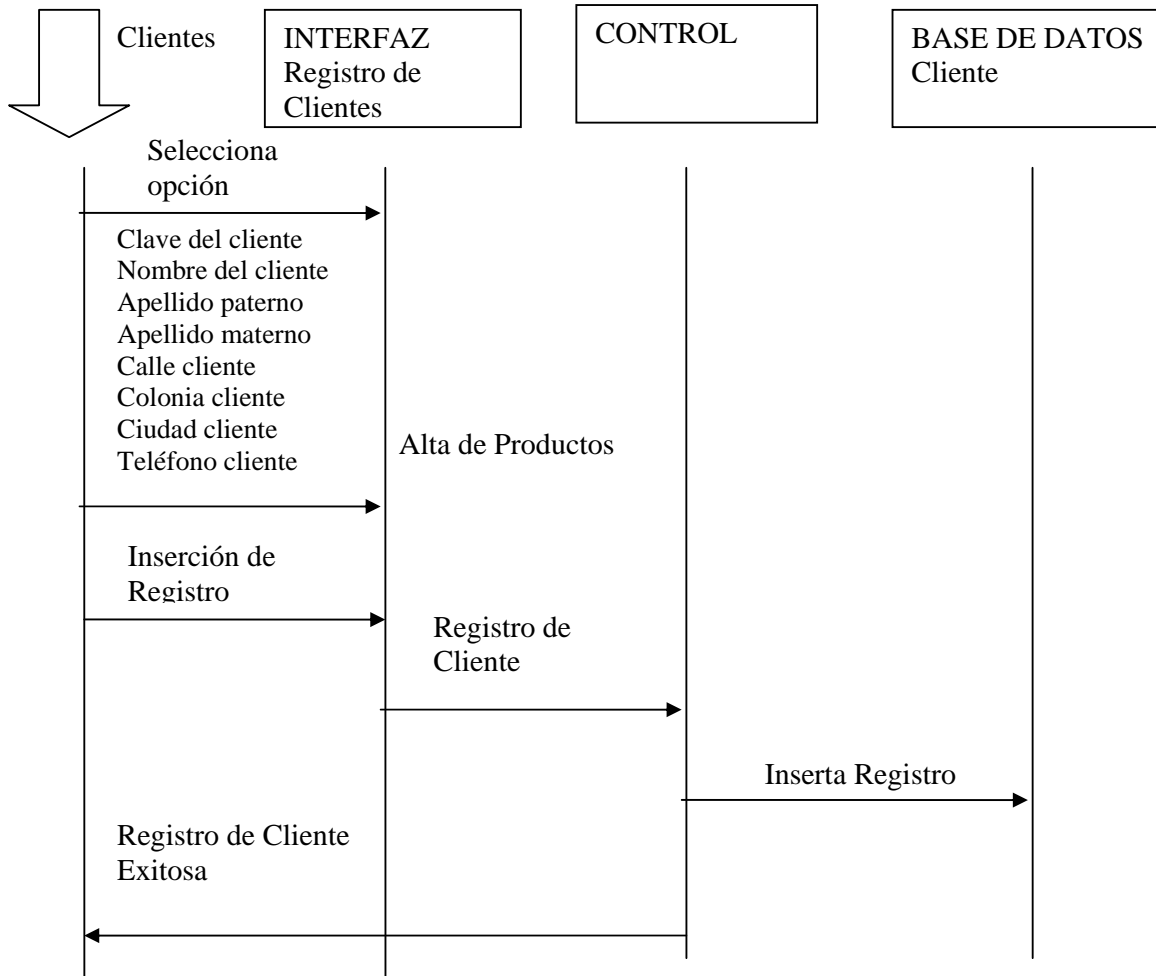


Fig. 2.4 Registro de Clientes

2.3.4 Diagrama de secuencia de Enviar productos

Aquí se muestra una página donde el cliente llena un formulario para realizar un pedido de envío a en domicilio que el escogió para ir a dejar su pedido, los datos que se solicitan son: No de envío, Lugar envío, Dirección envío, Calle, Colonia, Ciudad, Código postal, Forma pago.

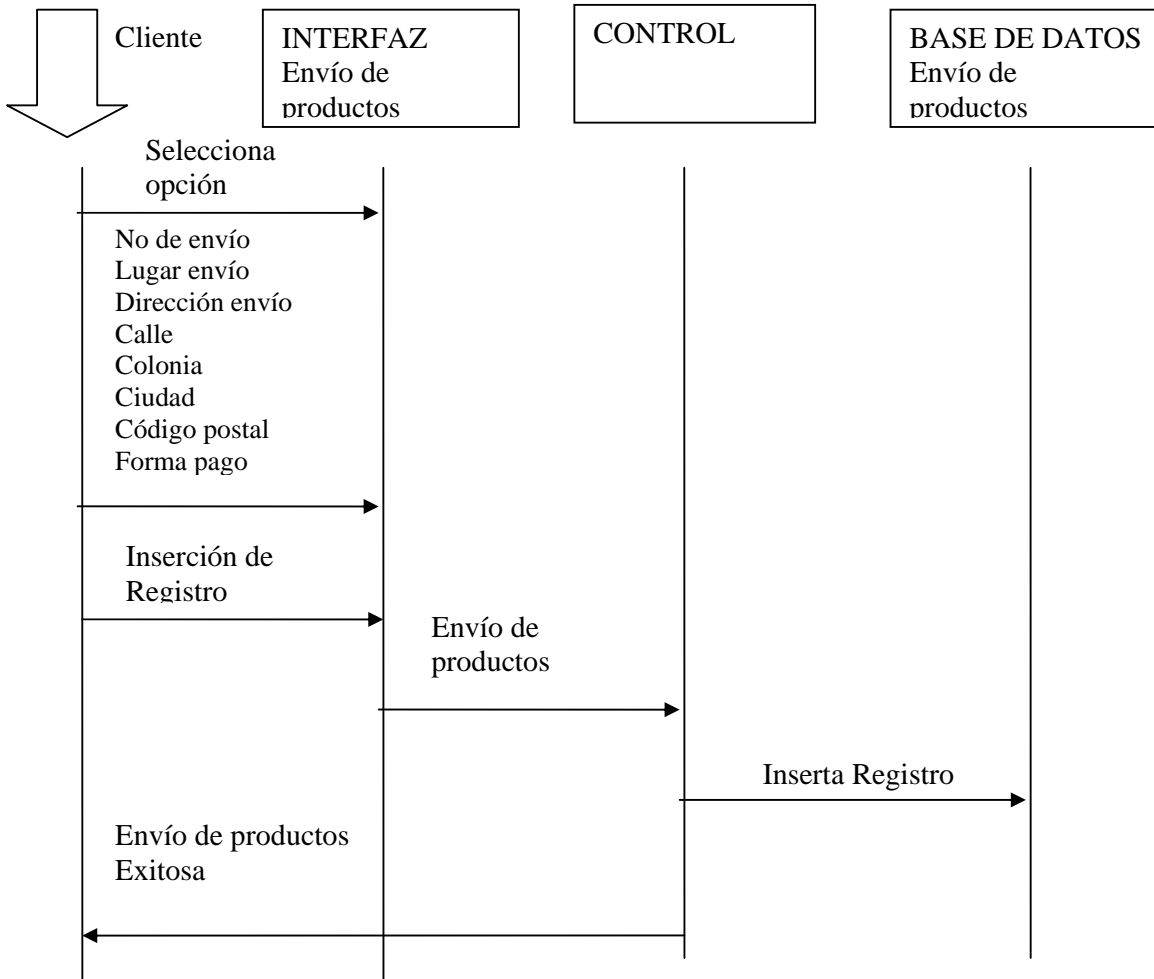


Fig. 2.5 Enviar productos

Capítulo 3 Diseño del Sistema alta de productos

3.1. Arquitectura del Sistema

La arquitectura que se usara para el sistema será cliente-servidor, donde el lado del cliente se estará ejecutando un browser o navegador HTML el cual constará de una interfaz gráfica y será el encargado de mostrar todas las pantallas con las cuales el cliente tendrá contacto con el sistema para su manejo. Cabe mencionar que el browser se encargará de todas las peticiones necesarias hacia el servidor.

Así mismo del lado del servidor se encontrará un Servidor Web el cual se encargará de atender las peticiones del cliente y éste a su vez tendrá acceso a las bases de datos analizando así el Servidor Web las peticiones de páginas y consultas de datos que se reciban.

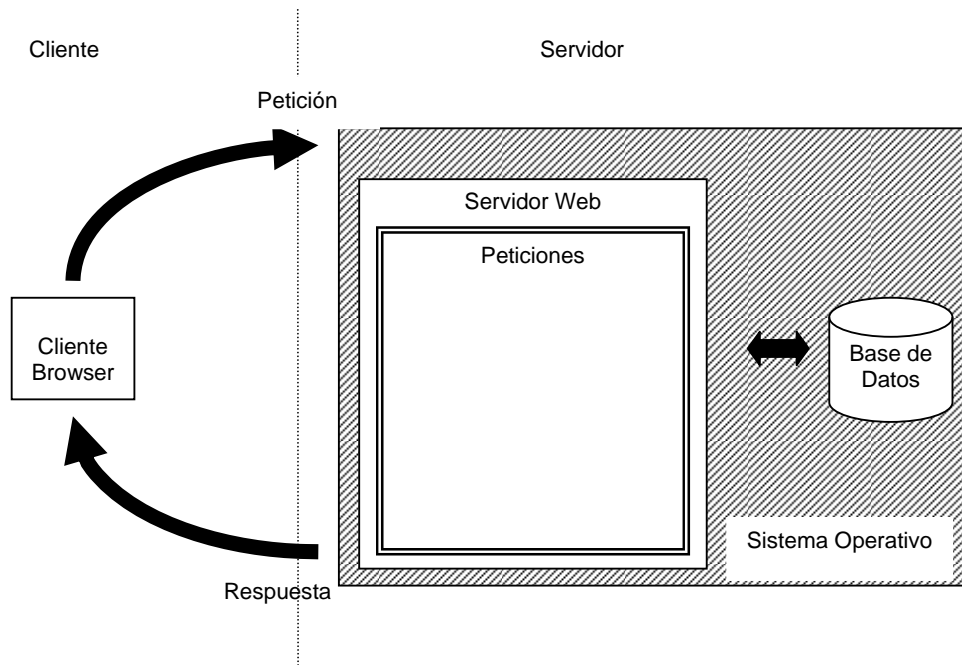
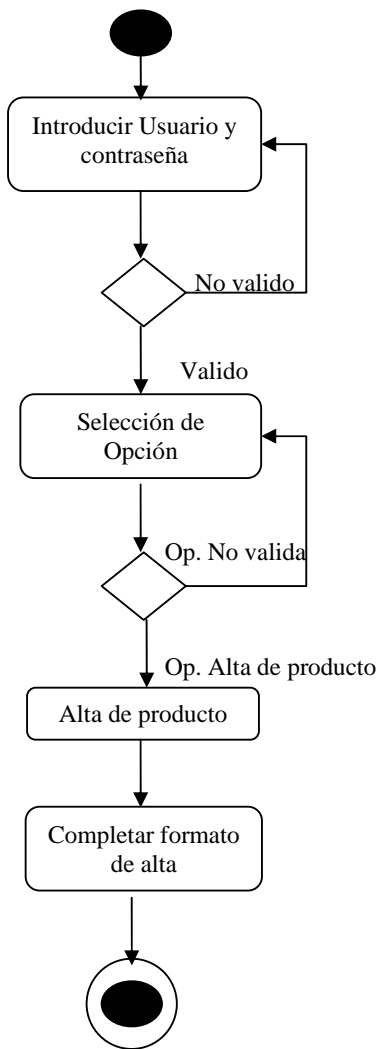


Figura 3.1 Esquema General del Sistema

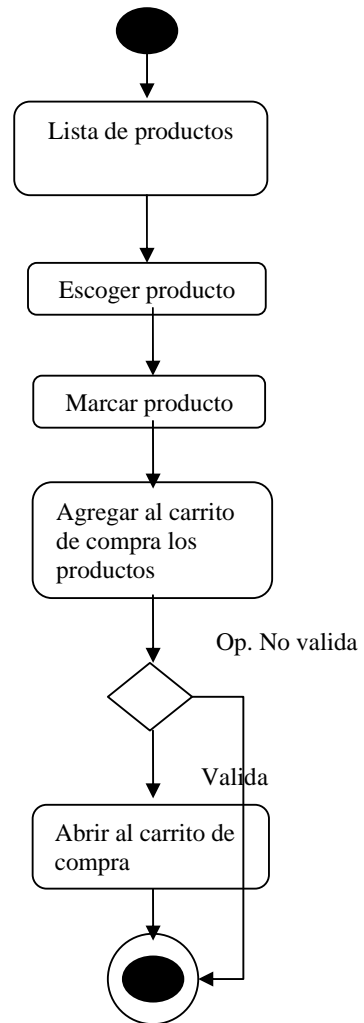
3.2 Diagramas de actividades

Los diagramas de actividades sirven para mostrar una visión simplificada de lo que ocurre durante una operación o proceso con el fin de expresar los puntos de decisión.

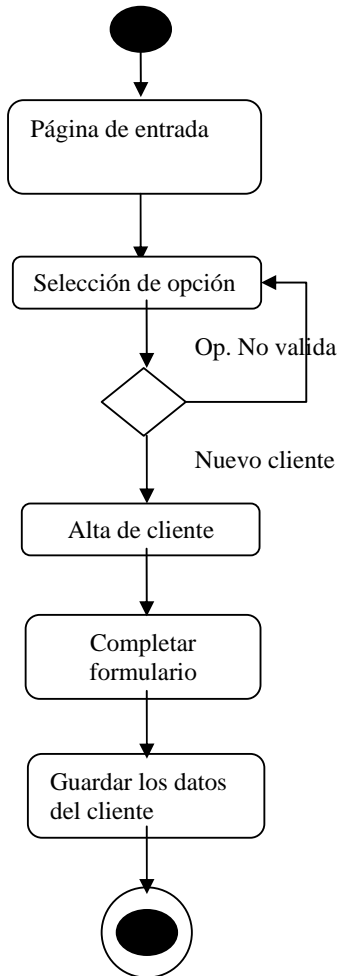
3.2.1 Diagrama de actividad para Alta de productos



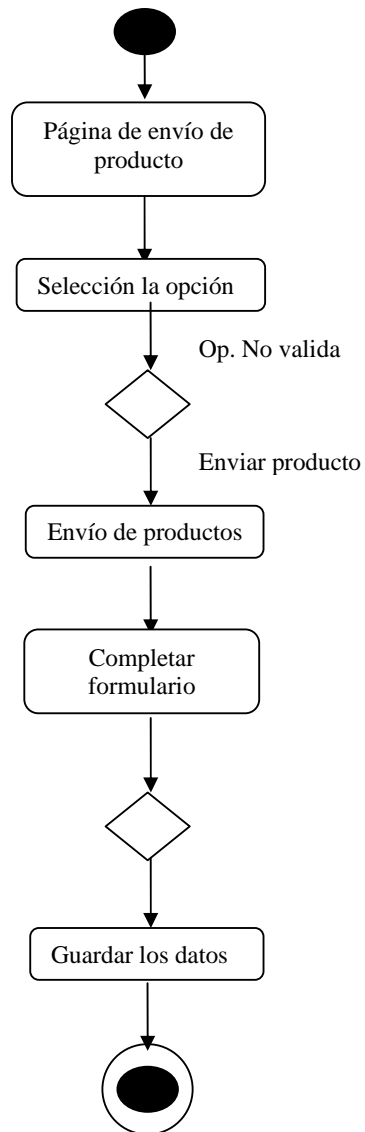
3.2.2 Diagrama de actividades de Escoger Producto



3.2.3 Registro de clientes

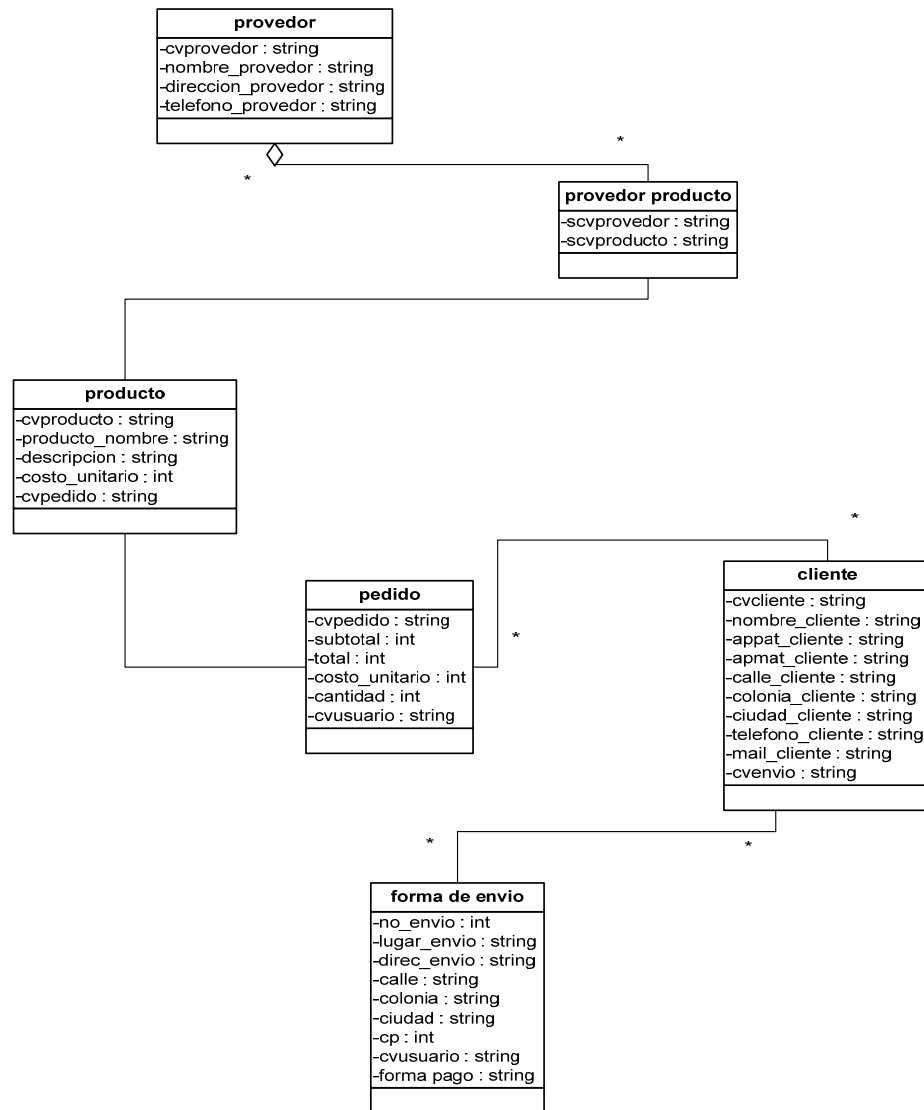


3.2.4 Envío de productos



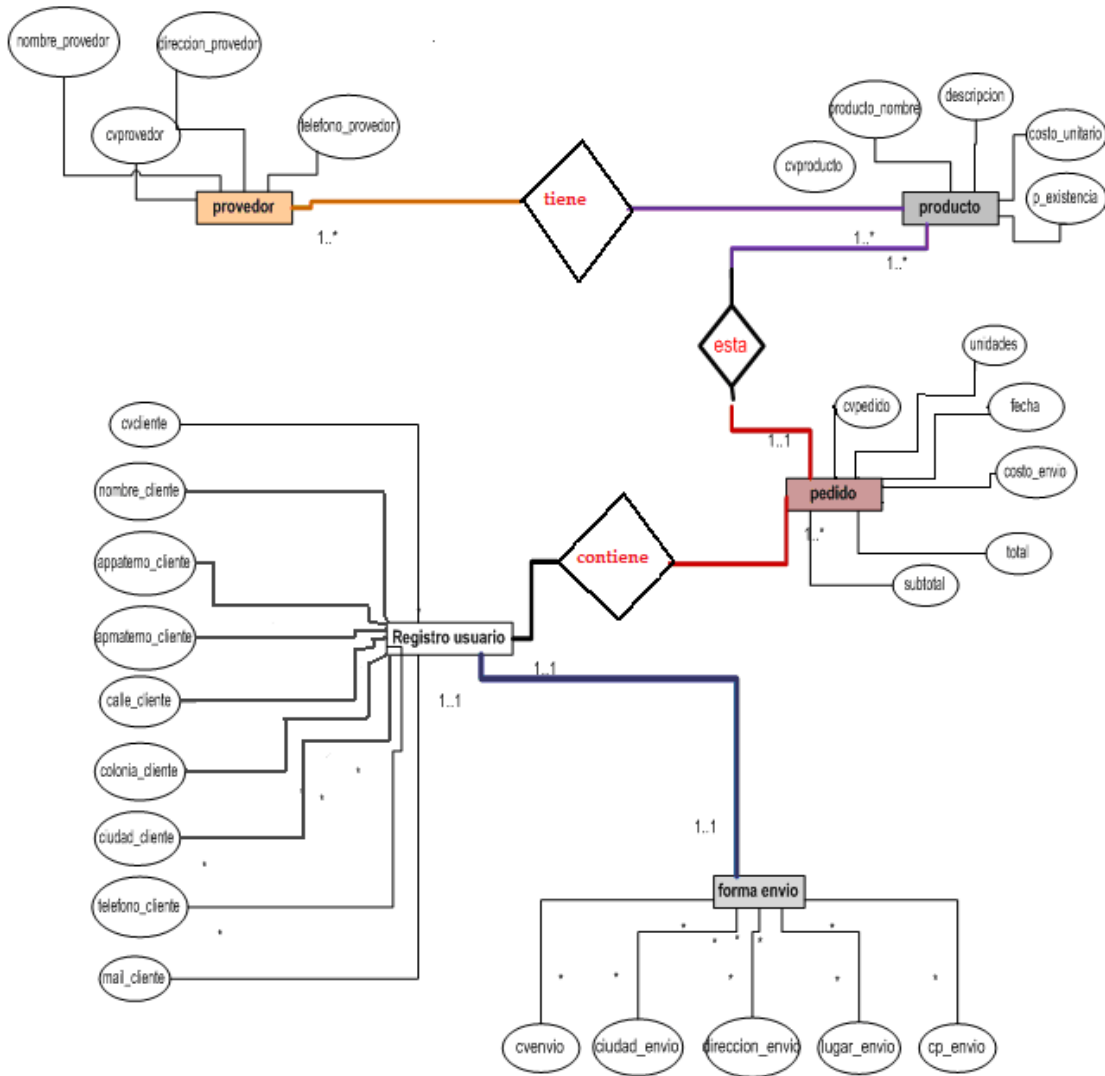
3.3 Diseño del Sistema

3.3.1 Diagrama de la base de datos



3.3.2 Análisis de los datos

Modelo entidad-relación



3.3.3 Normalización

Primera forma normal

Establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas.

Forma de envío

Pk

<u>no_envio</u>	<u>lugar_envio</u>	<u>direccion_envio</u>	<u>ciudad</u>	<u>cp</u>	scvusuario	forma:pago
-----------------	--------------------	------------------------	---------------	-----------	------------	------------

Segunda forma normal

Establece que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas. Una dependencia parcial es un término que describe a aquellos datos que no dependen de la clave de la tabla para identificarlos.

Tercera forma normal

Señala que hay que eliminar y separar cualquier dato que no sea clave. El valor de esta columna debe depender de la clave. Todos los valores deben identificarse únicamente por la clave.

3.3.3 Diccionario de datos

Un diccionario de datos muestra las características el tipo de datos que van a utilizarse en el sistema, así como también su longitud, así como la llave de cada tabl y la descripción de lo que representa. La opción nulas durante la captura nos permite decir si los datos del formulario no se introducen no alteran de manera significativa el proceso de la informa ión.

Proveedor

Campo	Tipo	Nulo	Predeterminado
cvproveedor	varchar(16)	Sí	
nombreprve	varchar(30)	Sí	
direccionprove	varchar(50)	Sí	
telprove	varchar(15)	Sí	

Proveedor_produ

Campo	Tipo	Nulo	Predeterminado
cvproducto	varchar(20)	Sí	
cvproveedor	varchar(20)	Sí	

Producto

Campo	Tipo	Nulo	Predeterminado
cvproducto	varchar(20)	Sí	
producnombre	varchar(50)	Sí	
descripcion	varchar(80)	Sí	
cantidad	int(11)	Sí	
consto_unitario	float	Sí	

Pedido

Campo	Tipo	Nulo	Predeterminado
cvpedido	varchar(20)	Sí	
cvproducto	varchar(20)	Sí	
producnombre	varchar(50)	Sí	
subtotal	float	Sí	
total	float	Sí	
costo_envio	float	Sí	
unidades_pedido	float	Sí	

Registro_cliente

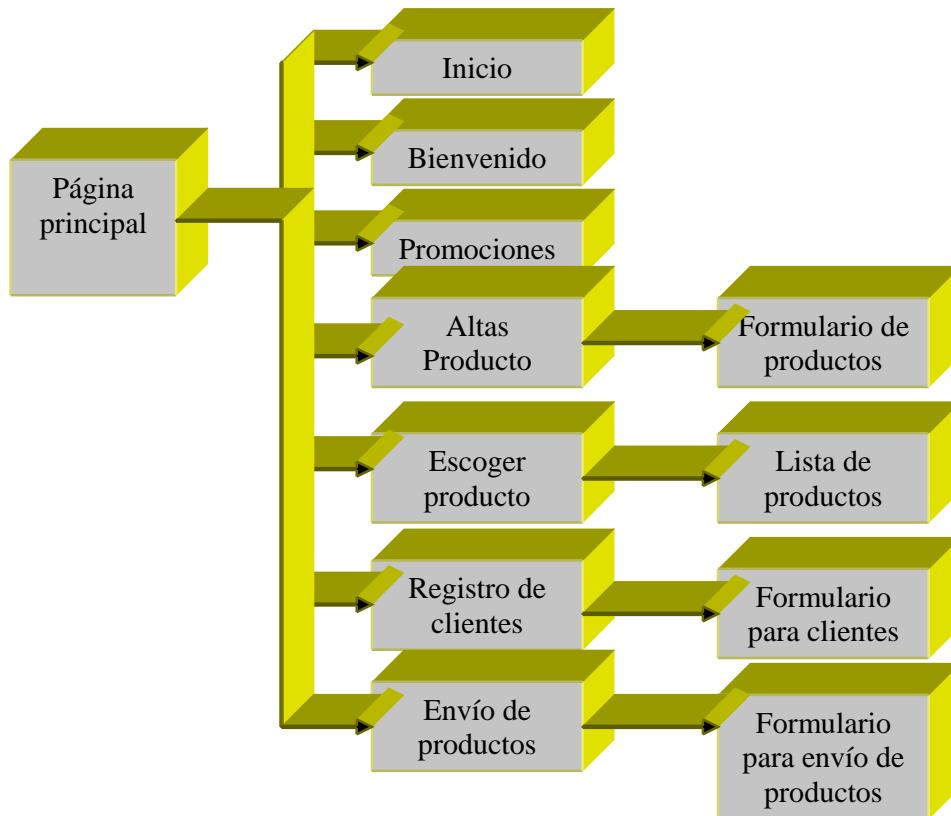
Campo	Tipo	Nulo	Predeterminado
cvusuario	varchar(20)	Sí	
nombrecliente	varchar(40)	Sí	
appaterno	varchar(20)	Sí	
apmaterno	varchar(20)	Sí	
calle	varchar(30)	Sí	
colonia	varchar(30)	Sí	
ciudad	varchar(20)	Sí	
tel_cliente	varchar(15)	Sí	
mail	varchar(30)	Sí	

num_envio	int(11)	Sí	
-----------	---------	----	--

Forma_envio

Campo	Tipo	Nulo	Predeterminado
no_envio	int(11)	Sí	
lugar_envio	varchar(30)	Sí	
colonia_envio	varchar(20)	Sí	
calle_envio	varchar(20)	Sí	
ciudad_envio	varchar(30)	Sí	
cp_envio	varchar(10)	Sí	
forma_envio	varchar(20)	Sí	
scvusuario	varchar(20)	Sí	

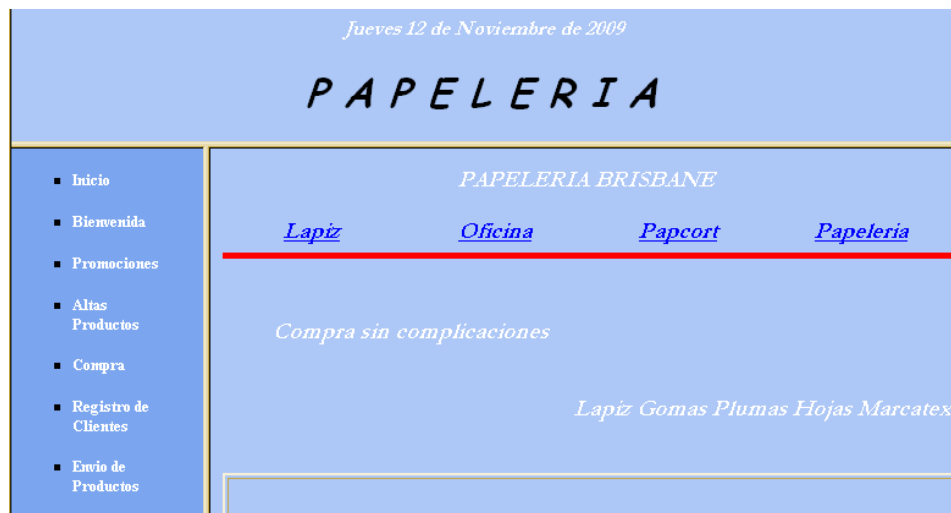
3.4 Diagrama de Navegación



3.5 Diseño de interfaz

3.5.1 Inicio

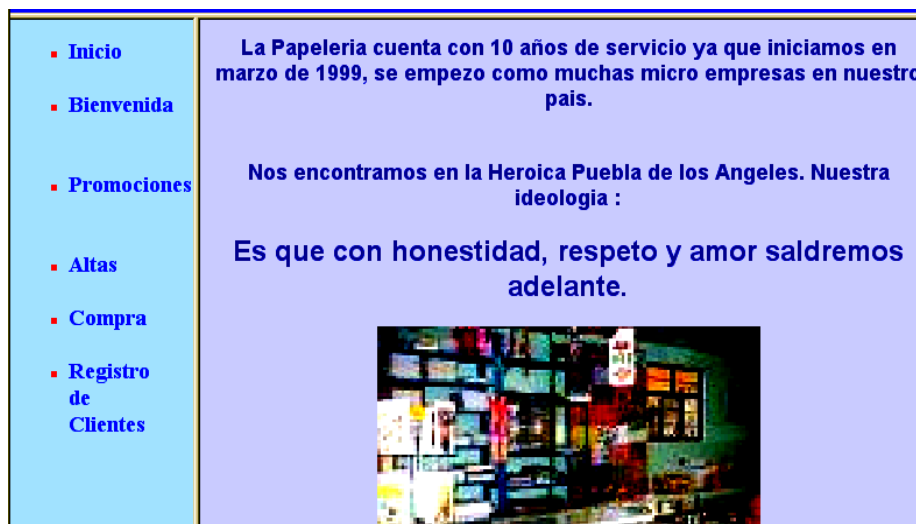
En esta pantalla se muestra una presentación del sistema como bienvenida y en su frame izquierdo se despliega un menú el cual tiene las opciones de Inicio, Bienvenida, Promociones, Alta de Productos, Compra, Registro de Cliente, Envío de Productos.



3.5.1 Inicio Página Principal

3.5.2 Bienvenida

Aquí se menciona una pequeña reseña sobre la creación e historia de la papelería, muestra una imagen sobre la misma.



3.5.2 Bienvenida

3.5.3 Promociones

En esta pantalla se muestran algunas Promociones que se tienen de temporadas.



3.5.3 Promociones

Capítulo 4. Desarrollo y Pruebas del sistema

4.1 Conectividad de la Base de Datos

Para realizar la conexión, se realizó un Script en PHP con los datos de la base de datos, y se utilizó una función propia del lenguaje para realizar la conexión con la base de datos.

Proceso para realizar la conexión:

```
<?php
$ip="localhost";
$usuario="root";
$password="password";
$conexion=mysql_connect($ip,$usuario,$password);
$seleccion_bd=mysql_select_db('papeleriaf',$conexion);
if((!$conexion) or (!$seleccion_bd))
{
die('conexion Fallida'.mysql_error());
}
else
{
echo("Conexion establecida ");
}
?>
```

4.2 Módulos

4.2.1 Tabla de proveedor

```
CREATE TABLE `proveedor` (
  `cvproveedor` VARCHAR( 16 ) NOT NULL ,
  `nombreprove` VARCHAR( 30 ) NOT NULL ,
  `direccionprove` VARCHAR( 50 ) NOT NULL ,
  `telprove` VARCHAR( 15 ) NOT NULL ,
  PRIMARY KEY ( `cvproveedor` )
) TYPE = innodb;
```

proveedor

Comentarios de la tabla: InnoDB free: 4096 kB

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>cvproveedor</u>	varchar(16)	Sí		
nombreprove	varchar(30)	Sí		
direccionprove	varchar(50)	Sí		
telprove	varchar(15)	Sí		

4.2.2 Tabla Proveedor Producto

```
CREATE TABLE `proveedor_prod` (  
  `cvproducto` VARCHAR( 20 ) NOT NULL ,  
  `cvproveedor` VARCHAR( 20 ) NOT NULL ,  
  PRIMARY KEY ( `cvproducto` )  
)TYPE=innodb;
```

proveedor_prod

Comentarios de la tabla: InnoDB free: 4096 kB

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>cvproducto</u>	varchar(20)	Sí		
cvproveedor	varchar(20)	Sí		

4.2.3 Tabla Pedido

```
CREATE TABLE `pedido` (  
  `cvpedido` VARCHAR( 20 ) NOT NULL ,  
  `cvproducto` VARCHAR( 20 ) NOT NULL ,  
  `producnombre` VARCHAR( 50 ) NOT NULL ,  
  `subtotal` FLOAT( 4 ) NOT NULL ,  
  `total` FLOAT( 4 ) NOT NULL ,  
  `costo_envio` FLOAT( 4 ) NOT NULL ,  
  `unidades_pedido` FLOAT( 4 ) NOT NULL ,  
  PRIMARY KEY ( `cvpedido` )  
) TYPE = innodb;
```

pedido

Comentarios de la tabla: InnoDB free: 4096 kB

Campo	Tipo	Nulo	Predeterminado	Comentarios
cvpedido	varchar(20)	Sí		
cvproducto	varchar(20)	Sí		
producnombre	varchar(50)	Sí		
subtotal	float	Sí		
total	float	Sí		
costo_envio	float	Sí		
unidades_pedido	float	Sí		

4.2.4 Tabla Registro cliente

```
CREATE TABLE `registro_cliente` (  
  `cvusuario` VARCHAR( 20 ) NOT NULL ,  
  `nombrecliente` VARCHAR( 40 ) NOT NULL ,  
  `appaterno` VARCHAR( 20 ) NOT NULL ,  
  `apmaterno` VARCHAR( 20 ) NOT NULL ,  
  `calle` VARCHAR( 30 ) NOT NULL ,  
  `colonia` VARCHAR( 30 ) NOT NULL ,  
  `ciudad` VARCHAR( 20 ) NOT NULL ,  
  `tel_cliente` VARCHAR( 15 ) NOT NULL ,  
  `mail` VARCHAR( 30 ) NOT NULL ,  
  `num_envio` INT NOT NULL ,  
  PRIMARY KEY ( `cvusuario` )  
 ) TYPE = innodb;
```

registro_cliente

Comentarios de la tabla: InnoDB free: 4096 kB

Campo	Tipo	Nulo	Predeterminado	Comentarios
cvusuario	varchar(20)	Sí		
nombrecliente	varchar(40)	Sí		
appaterno	varchar(20)	Sí		
apmaterno	varchar(20)	Sí		
calle	varchar(30)	Sí		
colonia	varchar(30)	Sí		
ciudad	varchar(20)	Sí		
tel_cliente	varchar(15)	Sí		
mail	varchar(30)	Sí		
num_envio	int(11)	Sí		

4.2.5 Forma de envío

```
CREATE TABLE `forma_envio` (  
  `no_envio` INT NOT NULL ,  
  `lugar_envio` VARCHAR( 30 ) NOT NULL ,  
  `colonia_envio` VARCHAR( 20 ) NOT NULL ,
```

```

`calle_envio` VARCHAR( 20 ) NOT NULL ,
`ciudad_envio` VARCHAR( 30 ) NOT NULL ,
`cp_envio` VARCHAR( 10 ) NOT NULL ,
`forma_envio` VARCHAR( 20 ) NOT NULL ,
`scvusuario` VARCHAR( 20 ) NOT NULL ,
PRIMARY KEY ( `no_envio` )
) TYPE = innodb;

```

forma_envio

Comentarios de la tabla: InnoDB free: 4096 kB

Campo	Tipo	Nulo	Predeterminado	Comentarios
no_envio	int(11)	Sí		
lugar_envio	varchar(30)	Sí		
colonia_envio	varchar(20)	Sí		
calle_envio	varchar(20)	Sí		
ciudad_envio	varchar(30)	Sí		
cp_envio	varchar(10)	Sí		
forma_envio	varchar(20)	Sí		
scvusuario	varchar(20)	Sí		

4.2.6 Producto

```

CREATE TABLE `producto` (
`cvproducto` VARCHAR( 20 ) NOT NULL ,
`producnombre` VARCHAR( 50 ) NOT NULL ,
`descripcion` VARCHAR( 80 ) NOT NULL ,
`cantidad` INT NOT NULL ,
`costo_unitario` FLOAT( 4 ) NOT NULL ,
PRIMARY KEY ( `cvproducto` )
) TYPE = innodb

```

producto

Comentarios de la tabla: InnoDB free: 4096 kB

Campo	Tipo	Nulo	Predeterminado	Comentarios
cvproducto	varchar(20)	Sí		Object: F:\angelica\ANGELICA (E)\ANGELICA (E)\java\jre1.5.0_16\lib\z\America\Araguaina.exe
producnombre	varchar(50)	Sí		Threat: BA/T/Autorun.B worm
descripcion	varchar(80)	Sí		Information: cleaned by deleting - quarantined
cantidad	int(11)	Sí		
consto_unitario	float	Sí		

4.2.7 Alta de cliente

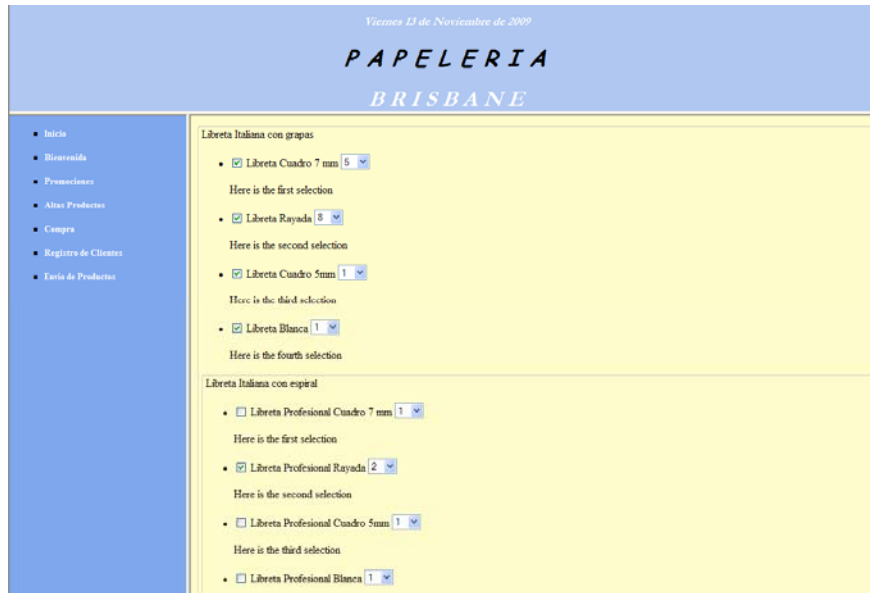
El usuario llena todos los campos del formulario al terminar, estos serán guardados en la base de datos al darle clic en el botón Alta Clientes.

The screenshot shows a web application interface for 'PAPELERIA BRISBANE'. The main header is blue with the text 'PAPELERIA' in a large, bold, black font and 'BRISBANE' in a smaller, white, italicized font below it. On the left side, there is a blue sidebar menu with white text and small square icons next to each item: 'Inicio', 'Bienvenida', 'Promociones', 'Altas Productos', 'Compra', 'Registro de Clientes', and 'Envío de Productos'. The main content area has a yellow background and is titled 'Alta de Clientes' in blue text. Below the title, there are eight input fields, each with a label to its left: 'Clave cliente', 'Apellido Paterno', 'Apellido Materno', 'Calle', 'Colonia', 'Ciudad', 'Mail', and 'Teléfono'. Each field is a simple white rectangle with a thin black border.

4.2.1 Alta de Clientes

4.2.8 Compra

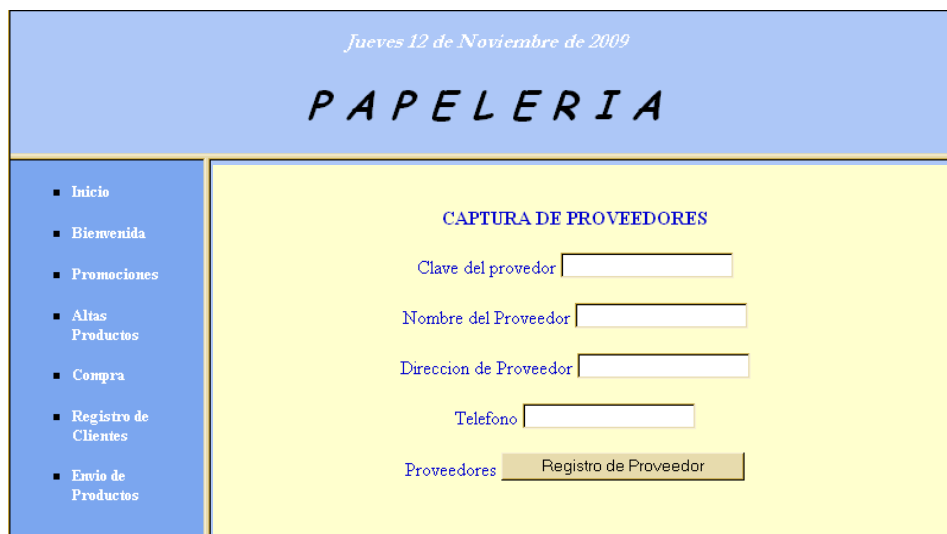
Todos los productos que se ofrecen para realizar una compra se muestran en una lista de productos; el cliente selecciona el producto y la cantidad del mismo, se acepta el pedido mediante un botón Compra para que así quede registrada la compra.



4.2.2 Compra de Productos

2.2.9 Captura de Proveedores

Se debe dar de alta a los diferentes proveedores, el formulario deberá ser llenado en su totalidad y dar clic en el botón Registro de Proveedor para que los datos se introduzcan en la base de datos. y quede registrado el proveedor.



4.2.3. Captura de Proveedores

4.3.0 Login de Alta de productos

El login de alta de producto valida la identificación del administrador ya registrado en caso de que no sea posible identificarse al primer intento el sistema le permitirá tener varias opciones para realizarlo, en caso de no identificarse correctamente el sistema no le permitirá acceder a alta de productos

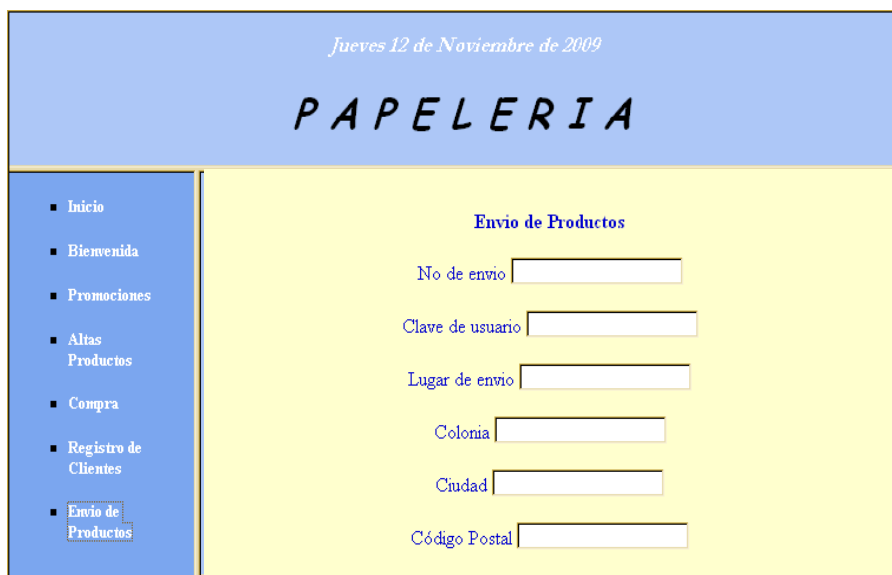


The screenshot shows the website header with 'PAPELERIA BRISBANE' in a blue banner. A left sidebar contains a menu with items: Inicio, Bienvenida, Promociones, Altas Productos, Compra, Registro de Clientes, and Envío de Productos. The main content area features a cartoon character holding a sign, a 'Uso Administrador: Login' label, and a login form titled 'Acceso Altas Productos'. The form includes 'Login' and 'Password' input fields and 'Accesar' and 'Limpiar' buttons.

4.2.4. Captura de Proveedores

2.3.1 Envío de Productos

En esta pantalla se muestran un formulario para que el cliente pueda llenar los datos para determinar donde se le va enviar el producto si así lo desea.



The screenshot shows the website header with the date 'Jueves 12 de Noviembre de 2009' and 'PAPELERIA' in a blue banner. The left sidebar menu is the same as in the previous screenshot. The main content area is titled 'Envío de Productos' and contains a form with the following fields: 'No de envío', 'Clave de usuario', 'Lugar de envío', 'Colonia', 'Ciudad', and 'Código Postal', each with an adjacent input box.

4.2.4. Envío de Productos

Conclusiones

El presente trabajo de tesis, deja como resultado un sistema funcional ya que es un sistema desarrollado con herramientas de software libre como: PHP, MYSQL, APACHE.

El sistema cuenta con la virtud de ejecutarse vía WEB, ya sea en una intranet o en Internet, con una interfaz amigable y sencilla para el cliente.

Se automatizó y se controló el manejo de compras de la papelería por medio de la WEB ya que teníamos la necesidad de realizar compras, ahora estas podrán realizarse en cualquier lugar por medio de una computadora y el Internet.

El cliente toma de una lista de productos clasificados por tipo el producto que desee, y escoge el número que desea comprar.

El cliente se registra para realizar su comprar, y realiza el llenado de un formulario en el sistema.

El cliente ya realizado su pedido tiene la opción de de que se le puedan mandar sus productos al domicilio que el indique.

Para modelar el problema y construir correctamente la base de datos utilice el modelo de base da datos relacional, ya que nos permite modelar problemas y administrar datos dinámicamente debido a su idea funcional.

Referencias

- ¿Que son las bases de datos?* (s.f.). Obtenido de www.maestrosdelweb.com/principiantes/%c2%bfque-son-las-bases-de-datos
- Bases de datos.* (s.f.). Obtenido de www3.uji.es/mmarques/f47/apun/node80.html
- Bases de datos: Modelo de datos.* (s.f.). Obtenido de <http://www.imgeek.net/?p=542>
- Diseño WEB.* (s.f.). Obtenido de www.usabilidadweb.com.ar/ingenieria_software.php
- Herramienta de la Ingenieria del software.*
- Ingenieria del Software.* (s.f.). Obtenido de www.angelfire.com/scifi/jzavala/apuntes/ingsoftware.html
- Manejo de bases de datos en PHP.* (s.f.). Obtenido de http://php.apsique.com/contenido/curso_php/manejo_base_de_datos
- Mellon's, C. *Software Engineering Intitute.*
- Modelo de base de datos.* (s.f.). Obtenido de [/www.imgeek.net/?p=542](http://www.imgeek.net/?p=542)
- MySQL.* (s.f.). Obtenido de www.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf
- Servidores Web.* (s.f.). Obtenido de www.jorgesanchez.net/bd/hherrera/servweb.html
- The Rational Edge . (s.f.). *E-MAGazine* .
- UML.* (s.f.). Obtenido de www.disca.upv.es/enheror/pdf/actaUML.pdf
- UML.* (s.f.). Obtenido de www.dcc.uchile.cl/psalinas/uml/modelo.html
- UML Elemental.* (s.f.). Obtenido de www.docs.kde.org/kde3/es/kdesdk/umbrello/uml-elements.html