



Benemérita Universidad Autónoma de Puebla



Facultad de Ciencias de la Computación
Licenciatura en Ciencias de la Computación

“Registro de Inversión Extranjera con Dispositivos Móviles”

PARA OBTENER EL TÍTULO DE:
Licenciado en Ciencias de la Computación

Presenta:
Erika Delgado Hernández

Asesor de Tesis:
MC. José Esteban Torres León.

ÍNDICE

INTRODUCCIÓN	4
CAPÍTULO 1 Marco Teorico	5
1.1 Definición de una Base de Datos.....	5
1.2 Características.....	7
1.3 Ventajas de las Bases de Datos	7
1.4 Tipos de Bases de Datos.....	9
1.5 Sistema manejador de Bases de Datos (DBMS)	10
1.6 Esquema de Bases de Datos	11
1.7 Usuarios de las Bases de Datos	12
1.8 Niveles de abstracción.....	12
1.9 Modelos de datos.....	13
1.10 La Ingenieria de Software.....	17
1.11 Ciclo de vida del Software en el proceso unificado	18
1.12 Tipos de modelo del ciclo de vida del Software.....	25
1.12.1 Fases del Modelo.....	28
CAPÍTULO 2 HERRAMIENTAS A UTILZIAR	30
2.1 Dispositivo Móvil.....	31
2.2 Categoría de los Dispositivos Móviles.....	31
2.3 Dispositivos Móviles en el trabajo.....	31
2.4 Historia de la PALM	32
2.5 PALM	32
2.6 Aplicaciones incluidas en la PALM OS.....	33
2.6.1 Libreta de Direcciones.....	33
2.6.2 Calculadora.....	33
2.6.3 Calendario.....	34
2.6.4 Gastos	34
2.6.5 Libreta de Notas	34
2.6.6 Notas.....	34
2.6.7 Tareas	34
2.7 Comunicaciones de la PALM OS.....	35
2.7.1 Infrarrojos	35
2.7.2 Bluetooth	35
2.8 Acces Point.....	35
2.9 Significado de WAP	36
2.10 PHP.....	36
CAPÍTULO 3 ANALISIS DE REQUERIMIENTOS	38
3.1 Descripción del Problema.....	39

3.2	Objetivo	39
3.3	Objetivos Específicos:	40
3.4	Restricciones:	40
3.5	Descripción de la información.....	40
3.5.1	Diagramas de Casos de Uso.	40
3.5.2	Composición.....	41
3.6	Registro de Inversión Extranjera con Dispositivos Móviles	42
3.7	Diagramas de Secuencia del Sistema.	44
CAPÍTULO 4 DISEÑO LÓGICO DEL SISTEMA		47
4.1	Diseño Conceptual.....	48
4.2	Diseño Lógico	48
4.3	Modelo Entidad Relación	49
4.4	Estructura Física de la Base de Datos.....	52
CAPÍTULO 5 IMPLEMENTACIÓN DEL SISTEMA		56
5.1	Funcionamiento del Sistema “Registro de Inversión Extranjera con Dispositivos Móviles”	57
5.2	Registro al sistema de Inversionistas.....	59
5.3	Validación de Datos.....	61
5.3.1	Registro de Datos de la Empresa y Representante	61
5.3.2	Registro de Inversión Nueva	61
5.3.3	Validación de datos de la Empresa.....	61
5.3.4	Validación de Datos del Representante	62
5.3.5	Validación de Usuario y Contraseña	63
5.3.6	Validación de usuario cuando inicia sesión.....	64
5.4	Pantalla de Administrador	67
CAPÍTULO 6 PRUEBAS Y RESULTADOS		69
6.1	Ingreso al sistema través de un Dispositivo Móvil.....	70
6.1.2	Registro de una empresa a través de un Dispositivo Móvil	71
6.2	Registro de una Empresa	71
6.3	Resultados obtenidos del registro de las empresas a través de una palm y una computadora portatil.....	74
Conclusiones.....		77
Perspectivas		78
Anexos.....		79
Bibliografía.....		83

INTRODUCCIÓN

Actualmente, el registro de proyectos en diversas instituciones públicas o privadas se hace de manera tradicional: llenar documentos a papel o desde una terminal por Internet.

Las tecnologías móviles como celulares, PDAs, PALM'S entre otros, son un mercado nuevo y se pueden usar en muchas aplicaciones.

En particular, en el gobierno del estado de Puebla, se manejan proyectos de inversión extranjera. Los inversionistas son diversos y de diferentes lugares del mundo por lo que la mayoría de los inversionistas utiliza tecnología móvil. Esto motivó para realizar el sistema de inversión extranjera utilizando dicha tecnología.

Por lo anterior se desprende el objetivo del presente proyecto, que es desarrollar un sistema de inversión extranjera, accesado por dispositivos móviles y terminales normales y que genere diversos documentos.

El trabajo realizado requiere el conocimiento y uso de herramientas de bases de datos, páginas Web, conexión a dispositivos móviles, etcétera.

El trabajo realizado se presenta de manera escrita y organizada en capítulos, los cuales conforman al presente documento y se describen a continuación de manera breve.

En el primer capítulo se da un panorama general de bases de datos. Se hace hincapié a comprender y entender conceptos que nos servirán para la realización del proyecto y una breve explicación acerca de la metodología de ingeniería de software.

En el capítulo 2 se definen algunos conceptos y descripciones de las herramientas a utilizar para la realización del proyecto.

En el capítulo 3 se realiza el diseño de la base de datos comenzando por el diseño conceptual y el diseño lógico se muestra la arquitectura del sistema así como cada módulo se seguirá para el desarrollo del proyecto.

En el capítulo 4 se realiza la arquitectura que tendrá el sistema, tanto en el diseño de base de la base de datos como el diseño y desarrollo de la página Web.

En el capítulo 5 se presenta la implementación del sistema tanto para un inversionista, como un empleado, así como para el administrador del sistema.

En el capítulo 6 se presentan las pruebas realizadas y resultados obtenidos.

Finalmente, se presentan las conclusiones y perspectivas del trabajo.

CAPÍTULO 1

MARCO TEÓRICO

En este capítulo se definen los conceptos teóricos que corresponde a la metodología de ingeniería de software utilizada, así como los conceptos y definiciones de lo que es una base de datos que va a ser utilizada y mencionada en el sistema de “Registro de Inversión Extranjera con Dispositivos Móviles”.

Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente.

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.

Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más **columnas** y **filas**. Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla, cada fila de la tabla conforma un registro.

1.1 DEFINICIÓN DE UNA BASE DE DATOS

También conocida como Database, la base de datos es un conjunto de información que está almacenada en forma sistemática, de manera tal que los datos que la conforman pueden ser utilizados en forma fragmentada cuando sea necesario.

Los datos almacenados pueden ser muy diversos: nombres, números telefónicos, direcciones, años, etc. Todo depende de la finalidad para la que sea armada la base.

Actualmente, en todos los quehaceres cotidianos se utiliza una base de datos: cajeros automáticos, catálogos de bibliotecas o librerías, páginas amarillas, listado de medicamentos e incluso los mismos buscadores de Internet. Todo cuenta con una base de datos a la cual recurrir para consultar su información y mantenerla actualizada.

Tradicionalmente, se suelen organizar según campos, registros o archivos. El campo es una fracción única de información; el registro, un sistema de campos y el archivo, un conjunto de registros.

Existen tanto bases estáticas como dinámicas. Las primeras son sólo de lectura y que generalmente se utilizan para almacenar datos históricos que podrán ser utilizados a lo largo del tiempo para, por ejemplo, realizar proyecciones. Y las dinámicas son las que contienen información que puede ser modificada tanto para actualizar los datos que la integran como para agregar nuevos.

También se pueden clasificar las bases según su contenido, en donde podemos encontrar bases de datos bibliográficos, de texto completo, directorios, de imágenes etc. Otra forma en que se clasifican es según su modelo de administración de datos (la forma en que se describen, se almacenan y se recuperan los datos). Por ejemplo bases de datos jerárquicos, de red, relacional o de relaciones, deductivos o lógicos, distributivos, etc.

1.2 CARACTERÍSTICAS

Entre las principales características de los sistemas de base de datos se puede mencionar:

1. **Independencia lógica y física de los datos:** se refiere a la capacidad de modificar una definición de esquema en un nivel de la arquitectura sin que esta modificación afecte al nivel inmediatamente superior. Para ello un registro externo en un esquema externo no tiene por qué ser igual a su registro correspondiente en el esquema conceptual.
2. **Redundancia mínima:** se trata de usar la base de datos como repositorio común de datos para distintas aplicaciones.
3. **Acceso concurrente por parte de múltiples usuarios:** control de concurrencia mediante técnicas de bloqueo o cerrado de datos accedidos.
4. **Distribución espacial de los datos:** la independencia lógica y física facilita la posibilidad de sistemas de bases de datos distribuidas. Los datos pueden encontrarse en otra habitación, otro edificio e incluso otro país. El usuario no tiene por qué preocuparse de la localización espacial de los datos a los que accede.
5. **Integridad de los datos:** se refiere a las medidas de seguridad que impiden que se introduzcan datos erróneos. Esto puede suceder tanto por motivos físicos (defectos de hardware, actualización incompleta debido a causas externas), como de operación (introducción de datos incoherentes).
6. **Consultas complejas optimizadas:** la optimización de consultas permite la rápida ejecución de las mismas.
7. **Seguridad de acceso y auditoría:** se refiere al derecho de acceso a los datos contenidos en la base de datos por parte de personas y organismos. El sistema de auditoría mantiene el control de acceso a la base de datos, con el objeto de saber qué o quién realizó una determinada modificación y en qué momento.
8. **Respaldo y recuperación:** se refiere a la capacidad de un sistema de base de datos de recuperar su estado en un momento previo a la pérdida de datos.
9. **Acceso a través de lenguajes de programación estándar:** se refiere a la posibilidad ya mencionada de acceder a los datos de una base de datos mediante lenguajes de programación ajenos al sistema de base de datos propiamente dicho[2].

1.3 VENTAJAS DE LAS BASES DE DATOS

Control sobre la redundancia de datos:

Los sistemas de archivos almacenan varias copias de los mismos datos en archivos distintos. Esto hace que se desperdicie espacio de almacenamiento, además de provocar la falta de consistencia de datos.

En los sistemas de bases de datos todos estos archivos están integrados, por lo que no se almacenan varias copias de los mismos datos. Sin embargo, en una base de datos no se puede eliminar la redundancia completamente, ya que en ocasiones es necesaria para modelar las relaciones entre los datos.

Consistencia de datos:

Eliminando o controlando las redundancias de datos se reduce en gran medida el riesgo de que haya inconsistencias. Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez, y está disponible para todos los usuarios inmediatamente. Si un dato está duplicado y el sistema conoce esta redundancia, el propio sistema puede encargarse de garantizar que todas las copias se mantienen consistentes

Compartición de datos:

En los sistemas de archivos, los archivos pertenecen a las personas o a los departamentos que los utilizan. Pero en los sistemas de bases de datos, la base de datos pertenece a la empresa y puede ser compartida por todos los usuarios que estén autorizados.

Mantenimiento de estándares:

Gracias a la integración es más fácil respetar los estándares necesarios, tanto los establecidos a nivel de la empresa como los nacionales e internacionales. Estos estándares pueden establecerse sobre el formato de los datos para facilitar su intercambio, pueden ser estándares de documentación, procedimientos de actualización y también reglas de acceso.

Mejora en la integridad de datos:

La integridad de la base 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. Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD quien se debe encargar de mantenerlas.

Mejora en la seguridad:

La seguridad de la base de datos es la protección de la base de datos frente a usuarios no autorizados. Sin unas buenas medidas de seguridad, la integración de datos en los sistemas de bases de datos hace que éstos sean más vulnerables que en los sistemas de archivo.

Mejora en la accesibilidad a los datos:

Muchos SGBD proporcionan lenguajes de consultas o generadores de informes que permiten al usuario hacer cualquier tipo de consulta sobre los datos, sin que sea necesario que un programador escriba una aplicación que realice tal tarea.

Mejora en la productividad:

El SGBD proporciona muchas de las funciones estándar que el programador necesita escribir en un sistema de archivo. A nivel básico, el SGBD proporciona todas las rutinas de manejo de ficheros típicas de los programas de aplicación.

El hecho de disponer de estas funciones permite al programador centrarse mejor en la función específica requerida por los usuarios, sin tener que preocuparse de los detalles de implementación de bajo nivel.

Mejora en el mantenimiento:

En los sistemas de archivos, las descripciones de los datos se encuentran inmersas en los programas de aplicación que los manejan.

Esto hace que los programas sean dependientes de los datos, de modo que un cambio en su estructura, o un cambio en el modo en que se almacena en disco, requiere cambios importantes en los programas cuyos datos se ven afectados.

Sin embargo, los SGBD separan las descripciones de los datos de las aplicaciones. Esto es lo que se conoce como independencia de datos, gracias a la cual se simplifica el mantenimiento de las aplicaciones que acceden a la base de datos.

Aumento de la concurrencia:

En algunos sistemas de archivos, si hay varios usuarios que pueden acceder simultáneamente a un mismo archivo, es posible que el acceso interfiera entre ellos de modo que se pierda información o se pierda la integridad. La mayoría de los SGBD gestionan el acceso concurrente a la base de datos y garantizan que no ocurran problemas de este tipo.

Mejora en los servicios de copias de seguridad:

Muchos sistemas de archivo dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos.

En este caso, todo el trabajo realizado sobre los datos desde que se hizo la última copia de seguridad se pierde y se tiene que volver a realizar. Sin embargo, los SGBD actuales funcionan de modo que se minimiza la cantidad de trabajo perdido cuando se produce un fallo.

1.4 TIPOS DE BASES DE DATOS

Entre los diferentes tipos de base de datos, se pueden encontrar los siguientes:

- **MySQL:** es una base de datos con licencia GPL basada en un servidor. Se caracteriza por su rapidez. No es recomendable usar para grandes volúmenes de datos.

- **PostgreSql y Oracle:** Son sistemas de base de datos poderosos. Administra muy bien grandes cantidades de datos, y suelen ser utilizadas en intranets y sistemas de gran calibre.
- **Access:** Es una base de datos desarrollada por Microsoft. Esta base de datos, debe ser creada bajo el programa Access, el cual crea un archivo .mdb con la estructura ya explicada.
- **Microsoft SQL Server:** es una base de datos más potente que Access desarrollada por Microsoft. Se utiliza para manejar grandes volúmenes de informaciones.

1.5 SISTEMA MANEJADOR DE BASES DE DATOS (DBMS)

El sistema manejador de bases de datos es la porción más importante del software de un sistema de base de datos. Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica.

El objetivo primordial de un sistema manejador base de datos es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer, almacenar y manipular información de la base de datos. Todas las peticiones de acceso a la base, se manejan centralizadamente por medio del DBMS, como se muestra en la Figura. 1.1; por lo que este paquete funciona como interfaz entre los usuarios y la base de datos.

Las funciones principales de un DBMS son:

- Crear y organizar la Base de datos.
- Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos puedan ser accedados rápidamente.
- Manejar los datos de acuerdo a las peticiones de los usuarios.
- Registrar el uso de las bases de datos.
- **Interacción con el manejador de archivos.** Esto a través de las sentencias en DML al comando del sistema de archivos. Así el Manejador de base de datos es el responsable del verdadero almacenamiento de los datos.
- **Respaldo y recuperación.** Consiste en contar con mecanismos implantados que permitan la recuperación fácilmente de los datos en caso de ocurrir fallas en el sistema de base de datos.
- **Control de concurrencia.** Consiste en controlar la interacción entre los usuarios concurrentes para no afectar la inconsistencia de los datos.
- **Seguridad e integridad.** Consiste en contar con mecanismos que permitan el control de la consistencia de los datos evitando que estos se vean perjudicados por cambios no autorizados o previstos.

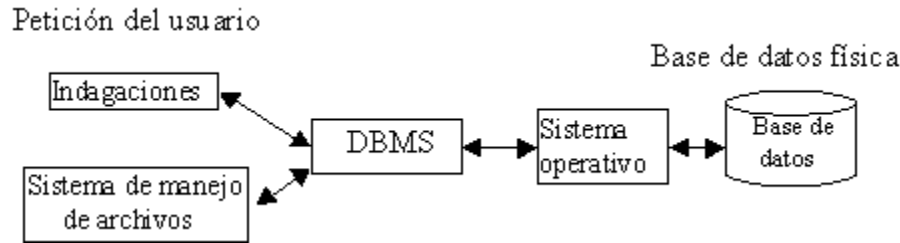


Figura 1.1 Peticiones de acceso por medio del DBMS

El DBMS interpreta las peticiones de entrada/salida del usuario y las manda al sistema operativo para la transferencia de datos entre la unidad de memoria secundaria y la memoria principal.

En sí, un sistema manejador de base de datos es el corazón de la base de datos ya que se encarga del control total de los posibles aspectos que la puedan afectar.

1.6 ESQUEMA DE BASE DE DATOS

Es la estructura que conforma la base de datos, se especifica por medio de un conjunto de definiciones que se expresa mediante un lenguaje especial llamado lenguaje de definición de datos. (DDL).

DDL (Data Definition language): Lenguaje de Definición de Datos. Por medio de este el DBMS identifica las descripciones de los elementos de los esquemas y almacena la descripción del esquema en el catálogo del DBMS. Por medio de este el DBMS especifica el esquema conceptual e interno (Base de datos Almacenada).

En la estructura que conforma una base de datos se enuncian otros lenguajes que permiten trabajar sobre la base de datos. Estos lenguajes estándar son:

- **SDL (Store Definition language):** Lenguaje de definición de almacenamiento. Es utilizado por el DBMS para especificar el esquema interno que corresponde a la Base de Datos Almacenada.
- **VDL (View Definition language):** Lenguaje de Definición de Vistas. Es utilizado por el DBMS para especificar las vistas del usuario y sus correspondencias con el esquema conceptual.

En las Bases de Datos Relacionales, el SQL, representa una combinación de los anteriores.

- **DML (Data Manipulation language):** Lenguaje de Manipulación de Datos. Permite la manipulación de las operaciones de Inserción, Eliminación y Modificación.
 - Tipos de DML's:

- De alto Nivel o No por procedimientos: SQL.
- De bajo Nivel o por procedimientos.

1.7 USUARIOS DE LAS BASES DE DATOS

Podemos definir a los usuarios como toda persona que tenga todo tipo de contacto con el sistema de base de datos desde que este se diseña, elabora, termina y se usa.

Los usuarios que accesan una base de datos pueden clasificarse como:

- **Programadores de aplicaciones.** Los profesionales en computación que interactúan con el sistema por medio de llamadas en DML (Lenguaje de Manipulación de Datos), las cuales están incorporadas en un programa escrito en un lenguaje de programación (Por ejemplo, COBOL, PL/I, Pascal, C, etc.)
- **Usuarios sofisticados.** Los usuarios sofisticados interactúan con el sistema sin escribir programas. En cambio escriben sus preguntas en un lenguaje de consultas de base de datos.
- **Usuarios especializados.** Algunos usuarios sofisticados escriben aplicaciones de base de datos especializadas que no encajan en el marco tradicional de procesamiento de datos.
- **Usuarios ingenuos.** Los usuarios no sofisticados interactúan con el sistema invocando a uno de los programas de aplicación permanentes que se han escrito anteriormente en el sistema de base de datos, podemos mencionar al usuario ingenuo como el usuario final que utiliza el sistema de base de datos sin saber nada del diseño interno del mismo por ejemplo: un cajero.

1.8 NIVELES DE ABSTRACCIÓN.

Existen diferentes niveles de abstracción para simplificar la interacción de los usuarios con el sistema; Interno, conceptual y externo, específicamente el de almacenamiento físico, el del usuario y el del programador como se muestra en la Figura. 1.2.

Los niveles de abstracción se dividen en:

- *Nivel Físico*
- *Nivel Lógico*
- *Novel de Vistas*

* *NIVEL FISICO*

El nivel más bajo de abstracción describe como se almacenan realmente los datos. En el nivel físico se describen en detalle las estructuras de datos complejas de bajo nivel.

* *NIVEL LÓGICO*

El siguiente nivel más alto de abstracción describe que los datos se almacenan en la base de datos y que relaciones existen entre esos datos. La base de datos completa se describe así en términos de un número pequeño de estructuras relativamente simples en el nivel físico, los usuarios del nivel lógico no necesitan preocuparse de esta complejidad.

Los administradores de base de datos, que deben decidir la información que se mantiene en la base de datos, usan el nivel lógico de abstracción [17].

* *NIVEL DE VISTAS*

El nivel más alto de abstracción describe solo parte de la base de datos completa. A pesar del uso de estructuras más simples en el nivel lógico, queda algo de complejidad, debido a la variedad de información almacenada en una gran base de datos.

Muchos usuarios del sistema de base de datos no necesitan todo esta información. En su lugar, tales usuarios necesitan acceder solo a una parte de la base de datos.

Para que su interacción con el sistema se simplifique, se define la abstracción a nivel de vistas. El sistema puede proporcionar muchas vistas para la misma base de datos.

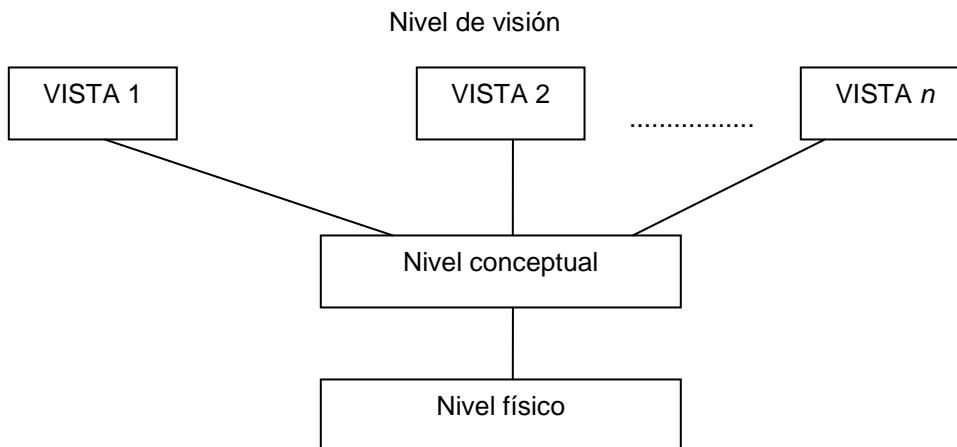


Figura 1.2 Arquitectura de tres niveles

1.9 MODELOS DE DATOS

Para introducirnos en este tema, se empezará definiendo lo que es un modelo.

Modelo: Es una representación de la realidad que contiene las características generales de algo que se va a realizar. En base de datos, es una colección de herramientas conceptuales para describir los datos con lo que vamos a trabajar. Podemos representar (en algunos casos de forma gráfica):

- *los propios datos*
- *las relaciones entre ellos*
- *la semántica de los datos*
- *las restricciones de consistencia de los datos*

Modelos de Datos

Es una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, semántica asociada a los datos y restricciones de consistencia.

Los modelos de datos se dividen en tres grupos:

- A) Modelos lógicos basados en objetos.
- B) Modelos lógicos basados en registros.
- C) Modelos físicos de datos.

A) MODELOS LÓGICOS BASADOS EN OBJETOS

Se usan para describir datos en los niveles conceptual y de visión, es decir, con este modelo representamos los datos de tal forma como nosotros los captamos en el mundo real, tienen una capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente. Existen diferentes modelos de este tipo, pero el más utilizado por su sencillez y eficiencia es el *modelo Entidad-Relación*.

- Modelo Entidad-Relación

Denominado por sus siglas como: E-R; Este modelo representa a la realidad a través de *entidades*, que son objetos que existen y que se distinguen de otros por sus características, por ejemplo: un alumno se distingue de otro por sus características particulares como lo es el nombre, o el número de control asignado al entrar a una institución educativa, así mismo, un empleado, una materia, etc. Las entidades pueden ser de dos tipos:

- **Tangibles:** Son todos aquellos objetos físicos que podemos ver, tocar o sentir.
- **Intangibles:** Todos aquellos eventos u objetos conceptuales que no podemos ver, aun sabiendo que existen, por ejemplo: la entidad materia, sabemos que existe, sin embargo, no la podemos visualizar o tocar.

Las características de las entidades en base de datos se llaman *atributos*, por ejemplo el nombre, dirección teléfono, grado, grupo, etc. son atributos de la entidad alumno; Clave, número de seguro social, departamento, etc., son atributos de la entidad empleado. A su vez una entidad se puede asociar o relacionar con más entidades a través de *relaciones*.

El modelado entidad-relación es una *técnica* para el modelado de datos utilizando diagramas entidad relación. No es la única técnica pero sí la más utilizada.

Brevemente consiste en los siguientes pasos:

1. Se parte de una descripción textual del problema o sistema de información a automatizar (los requisitos).
2. Se hace una lista de los sustantivos y verbos que aparecen.
3. Los sustantivos son posibles entidades o atributos.
4. Los verbos son posibles relaciones.
5. Analizando las frases se determina la cardinalidad de las relaciones y otros detalles.
6. Se elabora el diagrama (o diagramas) entidad-relación.
7. Se completa el modelo con listas de atributos y una descripción de otras restricciones que no se pueden reflejar en el diagrama.

Dado lo rudimentario de esta técnica se necesita cierto entrenamiento y experiencia para lograr buenos modelos de datos.

B) MODELOS LÓGICOS BASADOS EN REGISTROS

Se utilizan para describir datos en los niveles conceptual y físico. Estos modelos utilizan registros e instancias para representar la realidad, así como las relaciones que existen entre estos registros (ligas) o apuntadores. A diferencia de los modelos de datos basados en objetos, se usan para especificar la estructura lógica global de la base de datos y para proporcionar una descripción a nivel más alto de la implementación.

Los tres modelos de datos más ampliamente aceptados son:

- 1) *Modelo Relacional*
- 2) *Modelo de Red*
- 3) *Modelo Jerárquico*

1) Modelo Relacional: En este modelo se representan los datos y las relaciones entre estos, a través de una colección de tablas, en las cuales los renglones (tuplas) equivalen a los de cada uno de los registros que contendrá la base de datos y las columnas corresponden a las características (atributos) de cada registro localizado en la tupla.

Las ventajas de este modelo están:

1. Garantiza herramientas para evitar la duplicidad de registros, a través de campos claves o llaves.
2. Garantiza la integridad referencial: Así al eliminar un registro elimina todos los registros relacionados dependientes.
3. Favorece la normalización por ser más comprensible y aplicable.

2) Modelo de Red: Este modelo representa los datos mediante colecciones de registros, sus relaciones se representan por medio de ligas o enlaces, los cuales pueden verse como punteros. Los registros se organizan en un conjunto de gráficas arbitrarias.

El modelo de datos en red, representa las entidades en forma de nodos de un grafo, y las interrelaciones se representan mediante arcos que unen dichos nodos, como se muestra en la Figura. 1.3. En principio esta representación no impone restricción alguna acerca del tipo y el número de arcos que puede haber, con lo que se pueden modelar estructuras de datos tan complejas como sea necesario.

Para definir el modelo de red general con cierta formalización, lo haríamos como un conjunto finito de tipos de entidades:

$\{E_1, E_2, \dots, E_n\}$, Con sus respectivas propiedades o atributos: $\{A_{11}, A_{12}, \dots, A_{1k}, \dots, A_{n1}, A_{n2}, \dots, A_{nm}\}$, y un conjunto finito de tipos de interrelaciones: $\{I_{hj,k}, \dots, n, \}$

La anterior notación representa la interrelación entre los elementos j, k, \dots, n , que a su vez pueden ser entidades o interrelaciones, y el superíndice h permite diferenciar dos interrelaciones distintas entre los mismos elementos, ya que se refiere al nombre de la interrelación.

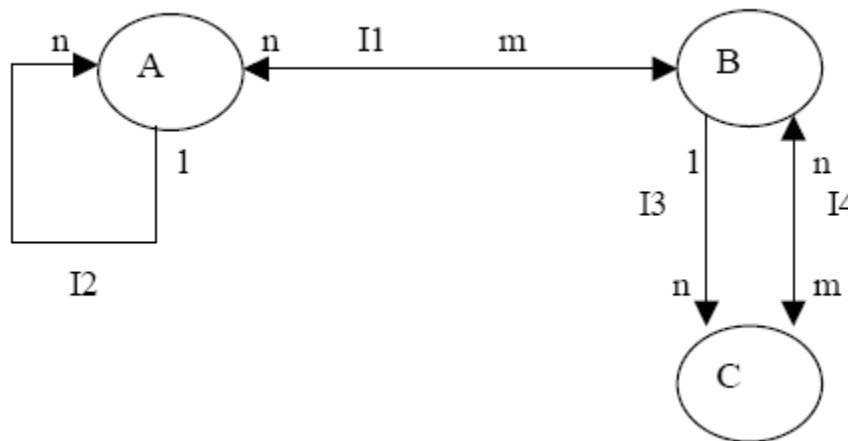


Figura 1.3 Ejemplo modelo de Red

3) Modelo Jerárquico: Es similar al modelo de red en cuanto a las relaciones y datos, ya que estos se representan por medio de registros y sus ligas. La diferencia radica en que están organizados por conjuntos de árboles en lugar de gráficas arbitrarias.

Como se mencionó, los **datos** se almacenan en la forma de **registros**, el equivalente a las *filas* del modelo relacional. Cada registro consta de un conjunto de **campos**, el equivalente a las *columnas* del modelo relacional. Un conjunto de registros con los mismos campos se denomina **archivos (record type, en inglés)**, el equivalente a las *tablas* del modelo relacional.

El modelo jerárquico facilita relaciones *padre-hijo*, es decir, relaciones 1:N (de uno a varios) del modelo relacional. Pero a diferencia de éste último, las relaciones son

unidireccionales. En justicia, dichas relaciones son *hijo-padre*, pero no *padre-hijo*. Por ejemplo, el registro de un empleado (nodo *hijo*) puede relacionarse con el registro de su departamento (nodo *padre*), pero no al contrario. Esto implica que solamente se puede consultar la base de datos desde los nodos *hoja* hacia el nodo *raíz*. La consulta en el sentido contrario requiere una búsqueda secuencial por todos los registros de la base de datos (por ejemplo, para consultar todos los empleados de un departamento). En las bases de datos jerárquicas no existen índices que faciliten esta tarea.

C) MODELOS FISICOS DE DATOS

Se usan para describir a los datos en el nivel más bajo, aunque existen muy pocos modelos de este tipo, básicamente capturan aspectos de la implementación de los sistemas de base de datos. Existen dos clasificaciones de este tipo que son:

- Modelo unificador
- Memoria de elementos.

Con el paso del tiempo la información que se va acumulando y desechando en la base de datos, ocasiona que esta cambie.

Se denomina:

- **Instancia:** Al estado que presenta una base de datos en un tiempo dado. Véalo como una fotografía que tomamos de la base de datos en un tiempo t , después de que transcurre el tiempo t la base de datos ya no es la misma.
- **Esquema:** Es la descripción lógica de la base de datos, proporciona los nombres de las entidades y sus atributos especificando las relaciones que existen entre ellos. Es un banco en el que se inscriben los valores que irán formando cada uno de los atributos. El esquema no cambia los que varían son los datos y con esto se tiene una nueva instancia.
- **Independencia de los datos:** Se refiere a la protección contra los programas de aplicación que puedan originar modificaciones cuando se altera la organización física o lógica de la base de datos.
- **Independencia física de datos:** Es la capacidad de modificar el esquema físico sin provocar que se vuelvan a escribir los programas de aplicación.
- **Independencia lógica de datos:** Capacidad de modificar el esquema conceptual sin provocar que se vuelvan a escribir los programas de aplicación.

1.10 LA INGENIERIA DE SOFTWARE

Software es la suma total de los programas de computadora, procedimientos, reglas, documentación asociada y los datos que pertenecen a un sistema de cómputo. En este contexto, la Ingeniería de Software es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software, que en palabras más llanas, se considera que la **Ingeniería de Software** es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en

costo o económicas) a los problemas de desarrollo de software, es decir, permite elaborar consistentemente productos correctos, utilizables y costo-efectivos.

El **proceso de ingeniería de software** se define como "un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad" El **proceso de desarrollo de software** "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo".

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el **ciclo de vida del software** que comprende cuatro grandes fases: concepción, elaboración, construcción y transición. La concepción define el alcance del proyecto y desarrolla un caso de negocio. La elaboración define un plan del proyecto, especifica las características y fundamenta la arquitectura. La construcción crea el producto y la transición transfiere el producto a los usuarios.

Actualmente se encuentra en una etapa de madurez el enfoque Orientado a Objetos (OO) como paradigma del desarrollo de sistemas de información. El Object Management Group (OMG) es un consorcio a nivel internacional que integra a los principales representantes de la industria de la tecnología de información OO.

El OMG tiene como objetivo central la promoción, fortalecimiento e impulso de la industria OO. El OMG propone y adopta por consenso especificaciones entorno a la tecnología OO.

Una de las especificaciones más importantes es la adopción en 1998 del Lenguaje de Modelado Unificado o UML (del inglés *Unified Modeling Language*) como un estándar, que junto con el Proceso Unificado están consolidando la tecnología OO [3].

1.11 CICLO DE VIDA DEL SOFTWARE EN EL PROCESO UNIFICADO

Las *fases del ciclo de vida del software* son: concepción, elaboración, construcción y transición, como se muestra en la Figura. 1.5.

- La **concepción** es definir el alcance del proyecto y definir el caso de uso.
- La **elaboración** es proyectar un plan, definir las características y cimentar la arquitectura.
- La **construcción** es crear el producto y la transición es transferir el producto a sus usuarios.

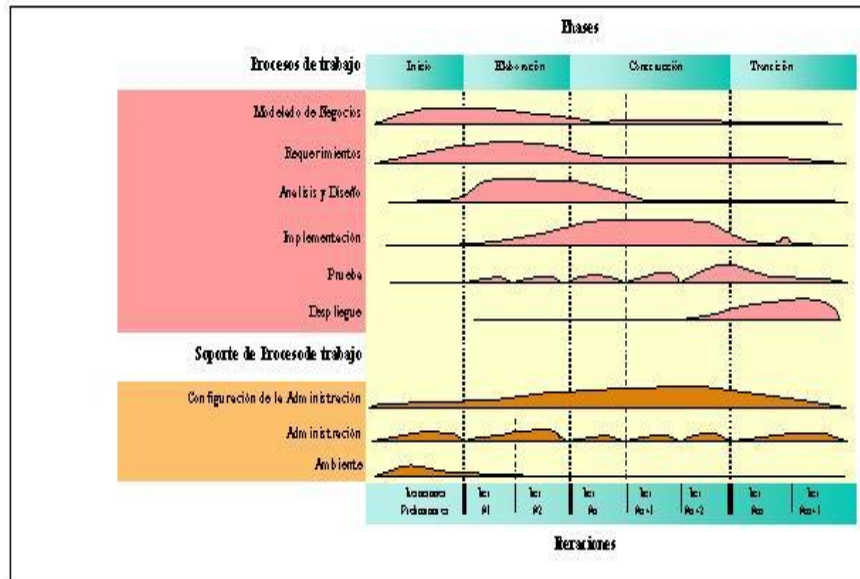


Figura 1.5 Ciclo de Vida del software

El diseño de software se realiza a tres niveles:

- I. Diseño Conceptual**
- II. Diseño Lógico**
- III. Diseño Físico**

I. Diseño Conceptual

El **diseño conceptual** se considera como un análisis de actividades y consiste en la solución de negocios para el usuario y se expresa con los casos de uso. El diseño conceptual es la solución del equipo de proyecto del negocio y consiste de las siguientes tareas:

- Identificar los usuarios y sus roles
- Obtener datos de los usuarios
- Evaluar la información
- Documentar los escenarios de uso
- Validar con los usuarios
- Validar contra la arquitectura de la empresa

Una forma de obtener estos requerimientos es construir una matriz usuarios-actividades de negocios, realizar entrevistas, encuestas y/o visitas a los usuarios, de tal manera que se obtenga quién, qué, cuándo, dónde y por qué de la solución.

II. Diseño Lógico

El **diseño lógico** traduce los escenarios de uso creados en el diseño conceptual en un conjunto de *objetos de negocio y sus servicios*. El diseño lógico se convierte en parte en la

especificación funcional que se usa en el diseño físico. El diseño lógico es independiente de la tecnología. El diseño lógico refina, organiza y detalla la solución de negocios y define formalmente las reglas y políticas específicas de negocios.

Un **objeto de negocios** es la encapsulación de un servicio que abstrae las cualidades esenciales de algo de interés.

Un **servicio** es una unidad con capacidad de cómputo. Un servicio debe satisfacer lo siguiente:

- Ser seguro, lo que equivale a un uso correcto y con autorización
- Ser válido, que tareas o reglas se pueden aplicar
- Manejar excepciones, informando al cliente
- Contar con un catálogo de servicios que constituye un repositorio de servicios.

Los objetos de negocio deben verificarse y probarse de tal manera que asegure que los módulos operen como unidades completas de trabajo. Las tareas de verificación incluyen:

- Una verificación independiente:
 - Pre y post condiciones
 - Lógica y funcionalidad individual
- Una verificación dependiente:
 - Verificación de dependencias
 - Que operan como una unidad específica de trabajo

El diseño lógico comprende las siguientes **tareas**:

- Identificar y definir los objetos de negocio y sus servicios
- Definir las interfaces
- Identificar las dependencias entre objetos
- Validar contra los escenarios de uso
- Comparar con la arquitectura de la empresa
- Revisar y refinar tanto como sea necesario

Para definir los objetos de negocios y sus servicios se puede usar la técnica de análisis nombre-verbo de los escenarios de uso. También se puede emplear la técnica sujeto-verbo-objeto directo. En estas técnicas los sujetos y el objeto directo son los candidatos a objetos de negocio y los verbos activos son los candidatos a servicios.

Una **interfaz** tiene las siguientes partes:

- Nombre
- Precondiciones, lo que debe estar presente antes de ejecutarse
- Postcondiciones, estado final
- Capacidad o funcionalidad (MySQL, pseudocódigo, función matemática)
- Dependencias

La tarea de **identificar las dependencias entre objetos** permite identificar eventos, sucesos o condiciones que permitan la realización de tareas de negocios coordinadamente o transaccionalmente. Para ello se debe considerar lo siguiente:

- Identificar los eventos disparadores (triggers)
- Determinar cualquier dependencia (existencial o funcional)
- Determinar cualquier problema de consistencia o secuencia
- Identificar cualquier regulación de tiempo crítica
- Considerar algún problema organizacional (transacciones)
- Identificar y auditar los requerimientos de control
- Determinar lugares y dependencias a través de la ubicación
- Determinar cuando el servicio que controla la transacción es dependiente de los servicios contenidos en otros objetos de negocio

La **validación del modelo lógico** debe ser tal que éste sea:

- Completo – debe representar todos los escenarios de uso.
- Correcto – el comportamiento lógico debe corresponder con el comportamiento conceptual, y
- Claro – los objetos de negocio y servicios no deben ser ambiguos

En el diseño lógico conceptualmente se divide en tres niveles de servicios con el fin de que la aplicación resulte flexible ante los cambios de requerimientos y/o de tecnología cambiando únicamente la capa o capas necesarias. Los tres niveles son: servicios de usuario, servicios de negocio y servicios de datos.

Los **servicios de usuario** (*user services*) controlan la interacción. Un servicio de usuario son personas, aplicaciones, otros servicios o la combinación de éstos. Generalmente involucra una interfaz gráfica de usuario (GUI) o puede ser no visual (mensajes o funciones), maneja todos los aspectos de la interacción con la aplicación. El objetivo central es minimizar el esfuerzo de conocimiento requerido para interpretar la información. Un servicio de usuario incluye un contenido (qué se necesita comunicar al usuario) y una forma (cómo se comunica el contenido) cuando es necesaria la comunicación.

Los **servicios de negocio** (*business services*) convierten datos recibidos de los servicios de datos y de usuario en información (datos + regla de negocio) y pueden usar otros servicios de negocio para completar su tarea.

Las tareas de los servicios de negocio son:

- Dar formato a los datos
- Obtener y mover datos desde y hasta los servicios de datos
- Transformar los datos en información
- Validar los datos inmediatamente en el contexto o en forma diferida una vez terminada la transacción.

Los **servicios de datos** (data services) son los servicios de bajo nivel que apoyan los servicios de negocio y son de una amplia gama de categorías como las siguientes:

- Declaración del esquema y su evolución (estructuras de datos, tipos, acceso indexado, SQL, APIs)
- Respaldo y recuperación (recuperación de datos si un evento falla)
- Búsqueda y Lectura (búsquedas, compilación, optimización y ejecución de solicitudes, formación de un conjunto de resultados)
- Inserción, actualización y borrado (procesar modificaciones consistentemente transaccional). Una **transacción** es atómica (ocurre o no), consistente (preserva integridad), aislada (otras transacciones ocurren antes o después) y durable (una vez completada, ésta sobrevive).
- Bloqueo (permite al acceso concurrente a los datos)
- Validación de datos (verifica la integridad del dominio, triggers y gateways para verificar el estado de los datos antes de aceptarlos, manejo de errores)
- Seguridad (acceso seguro a los objetos, operaciones, permisos a usuario y grupos y servicios)
- Administración de la conexión (mecanismos básicos para establecer una sesión de los servicios de datos). Establecer una conexión involucra: una identificación, la colocación y provisión de datos, tiempo de sesión, el tipo de interacción (conversacional, transaccional, multiusuario, monousuario).
- Distribución de datos (Distribuye información, a múltiples unidades de recuperación, bases de datos heterogéneas, según la topologías de la red).

III. Diseño Físico

El **diseño físico** traduce el diseño lógico en una solución implementable y costo-efectiva o económica.

El componente es la unidad de construcción elemental del diseño físico. Las **características de un componente** son:

- Se define según cómo interactúa con otros
- Encapsula sus funciones y sus datos
- Es reusable a través de las aplicaciones
- Puede verse como una caja negra
- Puede contener otros componentes

En el diseño físico se debe cuidar el nivel de granularidad (un componente puede ser tan grande o tan pequeño según su funcionalidad, es decir, del tamaño tal que pueda proveer de una funcionalidad compleja pero de control genérico) y la agregación y contención (un componente puede rehusar utilizando técnicas de agregación y contención, sin duplicar código) Figura 1.6.

El diseño físico debe involucrar:

- El diseño para distribución – debe minimizarse la cantidad de datos que pasan como parámetros entre los componentes y éstos deben enviarse de manera segura por la red.
- El diseño para multitarea – debe diseñarse en términos de la administración concurrente de dos o más tareas distintas por una computadora y el *multithreading* o múltiples hilos de un mismo proceso)
- El diseño para uso concurrente – el desempeño de un componente remoto depende de si está corriendo mientras recibe una solicitud.
- El diseño con el manejo de errores y prueba de eventos:
 - Validando los parámetros- a la entrada antes de continuar con cualquier proceso.
 - Protegiendo recursos críticos –manejar excepciones para evitar la falla o terminación sin cerrar archivos, liberar objetos sincronizados o memoria.
 - Protegiendo datos importantes – contar con una excepción a la mitad de la actuación en las bases de datos.
 - Debugging – crear una versión para limpiar errores.
 - Protección integral de transacciones de negocios – los errores deben regresarse al componente que llama

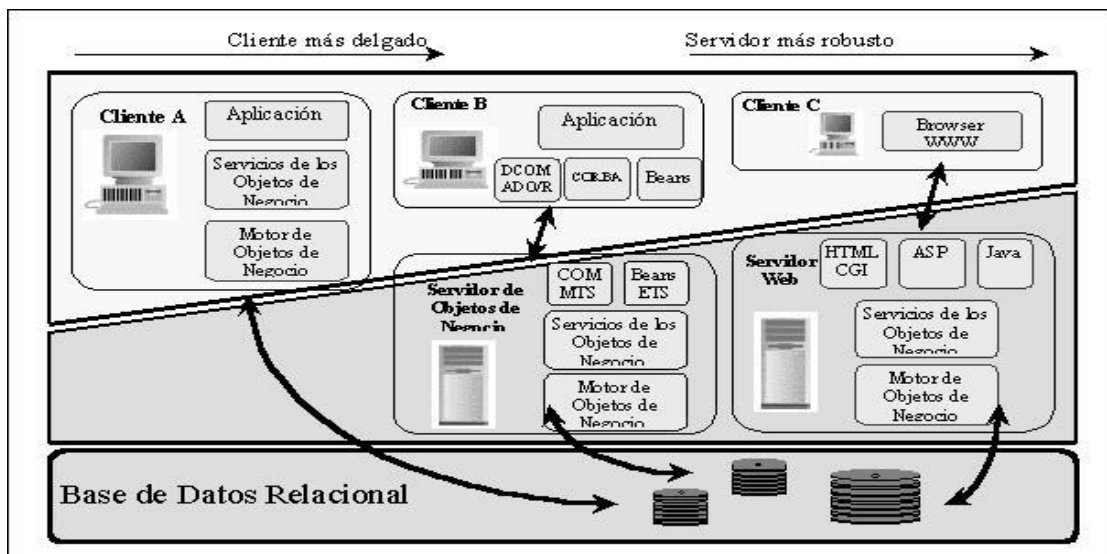


Figura 1.6 Diseño Físico

El diseño físico comprende las siguientes tareas:

- Definir los componentes
- Refinar el empaquetamiento y distribución de componentes
- Especificar las interfaces de los componentes
- Distribuir los componentes en la red
- Distribuir los repositorios físicos de datos
- Examinar la tolerancia a fallas y la recuperación de errores
- Validar el diseño físico

De las tareas anteriores la más importante es la distribución de los datos que pueden ser centralizados, una partición, un extracto o una réplica.

Los **datos centralizados** equivalen a una base de datos maestra ubicada en un lugar central. No hay copias de los datos.

Una **partición de datos** es una segmentación de la base de datos maestra. Es útil cuando los datos se pueden fragmentar fácilmente y actualizarse en un sitio local con cambios frecuentes. No hay sobre posición entre particiones. En una partición horizontal cada hilera existe en una sola base de datos. En una partición vertical cada columna es contenida en una y solo una base de datos.

Un **extracto de datos** es una copia de toda o una porción de la base de datos maestra. No se permite la actualización. Se usa un *timestamp* o etiqueta de tiempo para indicar qué tan viejos son los datos.

Una **réplica de datos** es un fragmento de la bases de datos maestra que se puede actualizar. Una réplica de datos es cuando el sitio de actualización cambia a un sitio local. No se permiten actualizaciones en la base de datos réplica y en la base de datos maestra a la vez, por lo que debe de haber sincronización entre ambas.

El diseño físico está íntimamente ligado a una alternativa tecnológica. Ante la acelerada evolución tecnológica es importante considerar los estándares del momento y las tendencias ya que una mala decisión implicará un costo enorme (en dinero y en tiempo) al actualizarse a otra plataforma distinta.

La tendencia actual en la arquitectura cliente/servidor es crear el back-end como un servidor robusto multitareas y *multithreading* y el front-end como un cliente muy delgado que no acapare al servidor comunicándose entre sí en una plataforma internet con protocolos estándar en redes heterogéneas[18].

1.12 TIPOS DE MODELO DEL CICLO DE VIDA DEL SOFTWARE

Las principales diferencias entre distintos modelos de ciclo de vida están en:

- El **alcance** del ciclo dependiendo de hasta dónde llegue el proyecto correspondiente. Un proyecto puede comprender un simple estudio de viabilidad del desarrollo de un producto, o su desarrollo completo o, llevando la cosa al extremo, toda la historia del producto con su desarrollo, fabricación, y modificaciones posteriores hasta su retirada del mercado.
- Las **características** (contenidos) de las fases en que dividen el ciclo. Esto puede depender del propio tema al que se refiere el proyecto (no son lo mismo las tareas que deben realizarse para proyectar un avión que un puente), o de la organización (interés de reflejar en la división en fases aspectos de la división interna o externa del trabajo).
- La **estructura** de la sucesión de las fases que puede ser lineal, con prototipado, o en espiral.

Ciclo de Vida Lineal

Es el más utilizado, siempre que es posible, precisamente por ser el más sencillo. Consiste en descomponer la actividad global del proyecto en fases que se suceden de manera lineal, es decir, cada una **se realiza una sola vez**, cada una se realiza **tras la anterior y antes que la siguiente**. Con un ciclo lineal es fácil dividir las tareas entre equipos sucesivos, y prever los tiempos (sumando los de cada fase) como se muestra en la Figura.1.7.

Requiere que la actividad del proyecto pueda descomponerse de manera que una fase no necesite resultados de las siguientes (realimentación), aunque pueden admitirse ciertos supuestos de realimentación correctiva. Desde el punto de vista de la gestión (para decisiones de planificación), requiere también que se sepa bien de antemano lo que va a ocurrir en cada fase antes de empezarla.

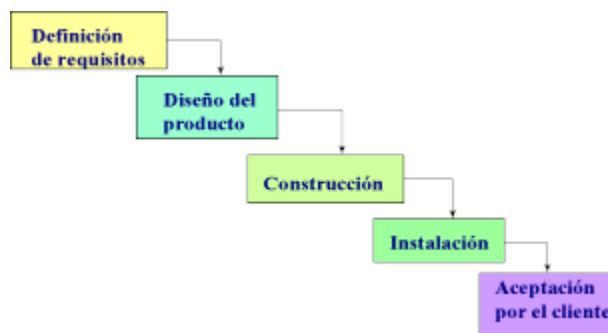


Figura 1.7 Ejemplo de ciclo lineal para un proyecto de construcción

Ciclo de Vida Prototipado

A menudo ocurre en desarrollos de productos con innovaciones importantes, o cuando se prevé la utilización de tecnologías nuevas o poco probadas, que las incertidumbres sobre los resultados realmente alcanzables, o las ignorancias sobre el comportamiento de las tecnologías, impiden iniciar un proyecto lineal con especificaciones cerradas.

Si no se conoce exactamente cómo desarrollar un determinado producto o cuáles son las especificaciones de forma precisa, suele recurrirse a definir especificaciones iniciales para hacer un **prototipo**, o sea, un **producto parcial** (no hace falta que contenga funciones que se consideren triviales o suficientemente probadas) y **provisional** (no se va a fabricar realmente para clientes, por lo que tiene menos restricciones de coste y/o prestaciones). Este tipo de procedimiento es muy utilizado en desarrollo avanzado como se muestra en la Figura.1.8.

La experiencia del desarrollo del prototipo y su evaluación deben permitir la **definición de las especificaciones más completas** y seguras para el producto definitivo. A diferencia del modelo lineal, puede decirse que el ciclo de vida con prototipado repite las fases de definición, diseño y construcción dos veces: para el prototipo y para el producto real.



Figura 1.8 Ciclo de Vida Prototipado

Ciclo de Vida en Espiral

El ciclo de vida en espiral puede considerarse como una generalización del anterior para los casos en que **no basta con una sola evaluación de un prototipo** para asegurar la desaparición de incertidumbres y/o ignorancias. El propio producto a lo largo de su desarrollo puede así considerarse como una sucesión de prototipos que progresan hasta llegar a alcanzar el estado deseado. En cada ciclo (espirales) las especificaciones del producto se van resolviendo paulatinamente como se observa en la Figura.1.9.

A menudo la **fuerza de incertidumbres es el propio cliente**, que aunque sepa en términos generales lo que quiere, no es capaz de definirlo en todos sus aspectos sin ver como unos influyen en otros. En estos casos la evaluación de los resultados por el cliente no puede esperar a la entrega final y puede ser necesaria repetidas veces.

El esquema del ciclo de vida para estos casos puede representarse por un bucle en espiral, donde los cuadrantes son, habitualmente, fases de **especificación, diseño, realización y**

evaluación (conceptos y términos análogos). En cada vuelta el producto gana en “*madurez*” (aproximación al final deseado) hasta que en una vuelta la evaluación lo apruebe y el bucle pueda abandonarse.

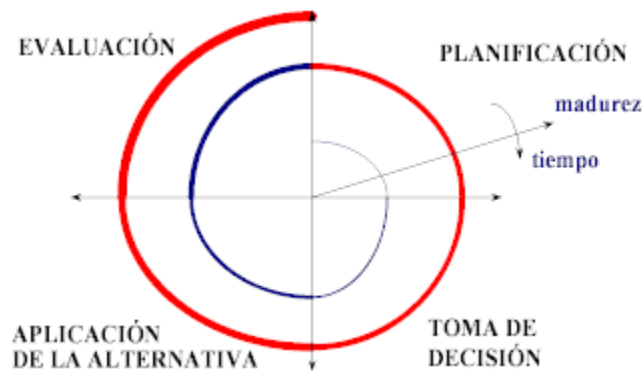


Figura 1.9 Modelo Espiral

Modelo Cascada

Es el enfoque metodológico que ordena rigurosamente las etapas del **ciclo de vida del software**, de forma tal que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior.

Un ejemplo de una metodología de desarrollo en cascada es:

1. Análisis de requisitos
2. Diseño del Sistema
3. Diseño del Programa
4. Codificación
5. Pruebas
6. Implantación
7. Mantenimiento

De esta forma, cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costes del desarrollo. La palabra *cascada* sugiere, mediante la metáfora de la fuerza de la gravedad, el esfuerzo necesario para introducir un cambio en las fases más avanzadas de un proyecto.

Si bien ha sido ampliamente criticado desde el ámbito académico y la industria, sigue siendo el paradigma más seguido al día de hoy.

1.12.1 Fases del Modelo

1.- Análisis de requerimientos.

Se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir. De esta fase surge una memoria llamada SRD (documento de especificación de requisitos), que contiene la especificación completa de lo que debe hacer el sistema sin entrar en detalles internos.

Es importante señalar que en esta etapa se deben **consensuar** todo lo que se requiere del sistema y será aquello lo que seguirá en las siguientes etapas, no pudiéndose requerir nuevos resultados a mitad del proceso de elaboración del software.

2.- Diseño del Sistema.

Se descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Como resultado surge el SDD (Documento de Diseño del Software), que contiene la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras.

3.- Diseño del Programa.

Es la fase en donde se realizan los algoritmos necesarios para el cumplimiento de los requerimientos del usuario así como también los análisis necesarios para saber que herramientas usar en la etapa de Codificación.

4.- Codificación.

Es la fase de programación o implementación propiamente dicha. Aquí se implementa el código fuente, haciendo uso de prototipos así como pruebas y ensayos para corregir errores. Dependiendo del lenguaje de programación y su versión se crean las librerías y componentes reutilizables dentro del mismo proyecto para hacer que la programación sea un proceso mucho más rápido.

5.- Pruebas.

Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente antes de ser puesto en explotación.

6.- Implantación.

El software obtenido se pone en producción. Se implantan los niveles software y hardware que componen el proyecto. La implantación es la fase con más duración y con más cambios en el ciclo de elaboración de un proyecto. Es una de las fases finales del proyecto.

7.- Mantenimiento.

Durante la explotación del sistema software pueden surgir cambios, bien para corregir errores o bien para introducir mejoras. Todo ello se recoge en los documentos de cambios.

Para la realización de este proyecto se utilizara el modelo de cascada ya que este modelo del ciclo de vida de software es el mejor y el adecuado ya que de adapta a las necesidades de tiempo y costo de este proyecto.

Además se utilizara PHP4 para la gestión de la base de datos ya que ofrece una portabilidad bajo cualquier sistema operativo.

PHP ofrece una mayor rapidez a comparación con otros productos, ya que usa tecnologías open source y una gestión de memoria más acertada, esta es la razón por la cual se utilizara PHP, además, tiene soporte para conectarse a una gran variedad de base de datos como: MySQL, PostgreSQL, Oracle, HyperWave entre otros.

El manejador o DBMS que se utilizará para el desarrollo del proyecto será MySQL, ya que MySQL es un manejador que puede competir con sus famosas contrapartes comerciales: Oracle, DB2, Informix, Sybase.

El motivo por el cual se opta por usar MySQL en lugar de otro manejador es la siguiente:

- Es gratuito
- Es extensible
- Es robusto
- Es rápido
- Es fácil de usar y administrar
- Tiene una arquitectura cliente-servidor
- Se integra con cualquier lenguaje de programación

CAPÍTULO 2.

HERRAMIENTAS A UTILIZAR

En este capítulo se definirán algunos conceptos y descripción de herramientas que van a ser utilizadas en el sistema de “Registro de Inversión Extranjera con Dispositivos Móviles”.

2.1 DISPOSITIVO MÓVIL.

Los dispositivos móviles (también conocidos como computadora de mano, Palmtop o simplemente handheld) son aparatos de pequeño tamaño, con algunas capacidades de procesamiento, móviles o no, con conexión permanente o intermitente a una red, con memoria limitada, diseñados específicamente para una función, pero que pueden llevar a cabo otras funciones más generales.

2.2 CATEGORIA DE LOS DISPOSITIVOS MÓVILES.

Dado el variado número de niveles de funcionalidad asociado con dispositivos móviles, en el 2005, T38 y DuPont Global Mobility Innovation Team propusieron los siguientes estándares para la definición de dispositivos móviles:

- **Dispositivo Móvil de Datos Limitados (*Limited Data Mobile Device*):** dispositivos que tienen una pantalla pequeña, principalmente basada en pantalla de tipo texto con servicios de datos generalmente limitados a SMS y acceso WAP. Un típico ejemplo de este tipo de dispositivos son los teléfonos móviles.
- **Dispositivo Móvil de Datos Básicos (*Basic Data Mobile Device*):** dispositivos que tienen una pantalla de mediano tamaño, (entre 120 x 120 y 240 x 240 pixeles), menú o navegación basada en íconos por medio de una "rueda" o cursor, y que ofrecen acceso a e-mails, lista de direcciones, SMS, y un navegador web básico. Un típico ejemplo de este tipo de dispositivos son las BlackBerry y los Teléfonos Inteligentes.
- **Dispositivo Móvil de Datos Mejorados (*Enhanced Data Mobile Device*):** dispositivos que tienen pantallas de medianas a grandes (por encima de los 240 x 120 pixeles), navegación de tipo *stylus*, y que ofrecen las mismas características que el "Dispositivo Móvil de Datos Básicos" (*Basic Data Mobile Devices*) más aplicaciones nativas como aplicaciones de Microsoft Office Mobile (Word, Excel, PowerPoint) y aplicaciones corporativas usuales, en versión móvil, como Sap, portales intranet, etc. Este tipo de dispositivos incluyen los Sistemas Operativos como Windows Mobile 2003 o versión 5, como en las Pocket PC.

2.3 DISPOSITIVOS MÓVILES EN EL TRABAJO.

Los Handhelds se han convertido en equipos robustos para su utilización en la gestión de situaciones empresariales, como por ejemplo, para la grabación de información estando en la calle. Actualmente son usados para archivar una variedad de tareas y para incrementar la eficiencia, como ser, la digitalización de notas, gestión de archivos, capturas de firmas, gestión y scaneo de partes de código de barras, etc. Los dispositivos portátiles usados en el trabajo, han moldeado a través del tiempo una variedad de factores y de formas, incluyendo teléfonos inteligentes en el extremo inferior, Dispositivos portátiles, PDA, PC Ultra Móviles, Tablet PC, e incluso computadoras portátiles.

2.4 HISTORIA DE LA PALM

El primer PDA de la historia se remonta al no muy famoso NEWTON de Apple Computing el cual es una agenda electrónica que funcionaba con un lápiz (Stylus) con el cual se escribe en la pantalla y esta reconoce lo que la persona escribe. No tuvo mucho éxito por su costo y tamaño (se aproxima al de una HandHeld).

Luego se lanzaron las PILOT 1000 y PILOT 5000 estas fueron creadas por Steven Hopkins, estas no poseían Back Light y solo tenían 250kb y 512kb respectivamente. Para este momento las PILOT no fueron muy famosas. Luego se lanzo al mercado la nueva línea de PalmPilot la cual estaba compuesta por los modelos PERSONAL (500kb) y PROFESIONAL (1mb y TCP/IP), a partir de estas el mundo conoció el verdadero poder de un asistente personal.

Las ventajas primordiales de las PalmPilot en relación a la competencia fue su sistema operativo de muy fácil uso (PalmOS 2.0), su pequeño tamaño, su ligero peso, larga duración de las baterías, el poder escribir en la pantalla sin necesidad de un teclado y algo muy importante que no tenían las agendas personales, comunicación directa con su computadora personal (HotSync Technology), en el caso de la PalmPilot Professional esta sincroniza directamente con el programa de correo electrónico que usted tenga en su PC para que luego usted pueda leer estos y escribir nuevos mensajes, esto es algo que ninguna agenda electrónica en el mercado podía hacer. Por esto es que solo en los Estados Unidos se hayan vendido más de un millón.

En febrero de 1998 se lanzo al mercado la PALM III la cual conservaba las ventajas de la PalmPilot Professional como lo son el sistema operativo, el uso de pilas AAA y por supuesto la sincronización con el PC y Back Light, superando al modelo anterior con muchas innovaciones como lo son su forma ergonómica, Stylus de metal, el doble de memoria, un sistema operativo mejorado (PalmOS 3.0), y una característica muy especial, el puerto Infrarrojo (IR port) con el cual se le encontraron nuevas funciones a las Palm.

Luego tenemos a la PALM IIIx la cual es la misma PALM III pero con el doble de memoria (4mb.), un puerto libre para expansiones y una pantalla mejorada.

La PALM VII es el último y más avanzado modelo de PDA's existente, tiene la capacidad de realizar cualquier acción de manera inalámbrica, es decir, navegar por Internet, enviar y recibir e-mails, enviar faxes, chatear, jugar, etc. Ya el año 2000 se pueden adquirir a las nuevas PALMIIIxe y la PALMIIIc el c significa COLOR. La PALMIIIxe es la misma PalmIIIx pero con 8mb de memoria.

2.5 PALM

Una PALM es un dispositivo móvil, o computadora de bolsillo (por su tamaño), que contiene diferentes utilidades. Algunas de ellas vienen directamente de fábrica, como puede ser, agenda (Date Book), calculadora, libreta de direcciones (Address Book) y un anotador de ideas (Memo Pad). Y otras pueden ser bajadas desde Internet, donde hay más de 10.000 aplicaciones orientadas a diferentes áreas de interés (juegos, programas, etc.).

Esta **minicomputadora** cuenta con diferentes elementos:

- **Microprocesador (cerebro):** capaz de resolver cálculos complejos
- **Memoria:** encargada de almacenar los datos.
- **Lápiz:** se utiliza para ingresar los datos, los cuales se deben ingresar en imprenta letra por letra. Este procedimiento se evita con la adquisición de un teclado.
- **Dispositivo infrarrojo:** sirve para intercambiar información con otras Palms (sin ningún tipo de cables) o con otros dispositivos de este tipo, como impresoras, cámaras, etc.
- **Cradle:** conecta la PALM a cualquier PC, permite bajar los correos electrónicos contenidos en la computadora. Hay PALMS que vienen preparadas para conectarse a Internet sin necesidad de módem, en las cuales la conexión se realiza mediante una antena (PALM VII). Además de brindarnos una vía de conexión con nuestras PC, los Cradles hacen una copia de seguridad de todo lo que tengas almacenado en la PALM, dentro de la PC.

También podemos pasar información, provista por diferentes sitios de Internet, o noticias a nuestra PALM a través de los canales de AvantGo.

Hay muchas aplicaciones que nos podrían ser de mucha ayuda para nuestro trabajo, como por ejemplo: aplicaciones de cálculos de estructuras para arquitectos e ingenieros, guía detallada de medicamentos o resumen de las leyes para abogados, etc [16].

2.6 APLICACIONES INCLUIDAS CON EL PALM OS.

2.6.1 Libreta de Direcciones

El programa Libreta de Direcciones del Palm almacena información personal, en cualquiera de las categorías definidas por el usuario. Se muestran las entradas y salidas se ordenan por los apellidos, y nombre (esto sólo se puede cambiar a Compañía, Apellidos). Hay cinco campos para el teléfono o correo electrónico, cada uno de los que se puede designar a Trabajo, a Casa, Fax u Otras, Correo Electrónico, Principal, Buscapersonas o Móvil (los nombres de los campos no se pueden cambiar).

2.6.2 Calculadora

La Calculadora convierte el ordenador de bolsillo en una calculadora estándar de 4 funciones con botones de tres tonos morados y azules que contrastan con los dos botones rojos para borrar. Incluye teclas de raíz cuadrada y de porcentaje y tiene memoria.

También tiene una opción para mostrar un historial de los cálculos realizados, como muchas calculadoras que se usaban antes.

2.6.3 Calendario

El Calendario muestra un horario diario o semanal, o una vista mensual simple (los nuevos tiene una vista mensual más elegante). El horario diario tiene una línea cada hora, entre dos horas del día que el usuario puede elegir. Al hacer clic en una línea vacía se crea una cita nueva. Las líneas vacías se llenan con las citas correspondientes, y la hora en que empiezan y su duración se muestran en el margen de la izquierda.

El sistema operativo puede anunciar una cita con una alarma, en el momento indicado, minutos, horas o días antes. Estas alarmas suenan incluso cuando la unidad está apagada.

Las citas se pueden repetir en un número especificado de días, semanas, meses o años y pueden contener notas.

2.6.4 Gastos

La aplicación Gastos permite a un usuario seguir los gastos comerciales habituales. El ordenador de bolsillo no realiza ningún cálculo para obtener el total. El usuario tiene que sincronizar con un ordenador de escritorio y ver los datos de los gastos en una hoja de cálculo en las que se incluyen las plantillas para Microsoft Excel. Debido a esta desventaja, muchos usuarios consideran esta aplicación poco útil.

2.6.5 Libreta de Notas

Las Notas de Texto permiten escribir notas de hasta 4000 caracteres, clasificadas en categorías configurables por el usuario. Las notas se pueden ordenar alfabéticamente o manualmente (que permiten al usuario escoger el orden de las notas). Las Notas de Texto sólo pueden incluir texto, no dibujos. Por esto, el texto en las Notas se ha de introducir utilizando el alfabeto Graffiti.

2.6.6 Notas

En las Notas se pueden hacer dibujos y notas manuscritas. Se pueden ingresar hasta 10 palabras por página y limitan el escrito a ciertos bytes, si la escritura es pulcra. De lo contrario, es mejor poner texto en la Libreta de Notas. Hay tres medidas de lápices de dibujo, más una goma de borrar. Es posible dibujar un mapa muy simple.

2.6.7 Tareas

También denominada lista de tareas. Es el lugar adecuado por crear recordatorios personales y priorizar las cosas que tienes que hacer. Cada elemento de la lista de tareas también puede tener: una prioridad, categorías (por organizar y agrupar las tareas en grupos lógicos), adjuntar una Nota (por añadir una descripción o una aclaración de la tarea). Las tareas se pueden ordenar por: fecha, prioridad o categoría.

2.7 COMUNICACIONES PALM OS.

2.7.1 Infrarrojos

Una de las características de las máquinas con Palm OS es que se venden de serie con un puerto infrarrojo, es decir que se pueden comunicar con otros dispositivos que incluyan un puerto infrarrojo, como por ejemplo algunos modelos de impresora, ordenadores portátiles y modelos de teléfono móviles así como también a las redes de computadoras con las que pueda establecer comunicación por medio de Wi-Fi.

A partir del PalmOS 3.0, que salió con el Palm III, el sistema operativo incluye apoyo por comunicaciones por infrarrojo, de acuerdo con la norma IrDA.

Una de las limitaciones de la comunicación por infrarrojos, es que no puede haber obstáculos entre los dispositivos que se comunican. Para superar este inconveniente, más adelante han aparecido nuevas tecnologías como por ejemplo el Bluetooth, y el Internet Wi-Fi.

2.7.2 Bluetooth

El Bluetooth es un sistema de conexión inalámbrica de corto alcance (aproximadamente 10 metros).

En los últimos tiempos ha proliferado tanto la conexión entre los pequeños dispositivos y, al mismo tiempo, éstos han ido reduciendo tanto su volumen, que no es extraño que la muchedumbre de cables que usamos para conectarlos ocupe casi el mismo espacio, si no más, que los mismos dispositivos. La tecnología Bluetooth ha venido a salvarnos de esta absurda situación.

A partir del PalmOS 5.0, que salió con el Palm Tungsten T, el sistema operativo incluye apoyo para comunicaciones por Bluetooth, de acuerdo con el estándar Bluetooth v1.1, que corresponde al 802.15.1 IEEE, que usa la misma gama de frecuencias (2.4 Ghz) que el IEEE 802.11b, aun cuando no se tienen que confundir.

2.8 Access Point.

Un **punto de acceso inalámbrico (WAP o AP** por sus siglas en inglés: Wireless Access Point) en redes de computadoras es un dispositivo que interconecta dispositivos de comunicación inalámbrica para formar una red inalámbrica. Normalmente un WAP también puede conectarse a una red cableada, y puede transmitir datos entre los dispositivos conectados a la red cable y los dispositivos inalámbricos. Muchos WAPs pueden conectarse entre sí para formar una red aún mayor, permitiendo realizar "roaming". (Por otro lado, una red donde los dispositivos cliente se administran a sí mismos - sin la necesidad de un punto de acceso - se convierte en una red **ad-hoc**). Los puntos de acceso inalámbricos tienen direcciones IP asignadas, para poder ser configurados.

Son los encargados de crear la red, están siempre a la espera de nuevos clientes a los que dar servicios. El punto de acceso recibe la información, la almacena y la transmite entre la WLAN (Wireless LAN) y la LAN cableada.

Un único punto de acceso puede soportar un pequeño grupo de usuarios y puede funcionar en un rango de al menos treinta metros y hasta varios cientos. Este o su antena son normalmente colocados en alto pero podría colocarse en cualquier lugar en que se obtenga la cobertura de radio deseada.

El usuario final accede a la red WLAN a través de adaptadores. Estos proporcionan una interfaz entre el sistema de operación de red del cliente (NOS: Network Operating System) y las ondas, mediante una antena inalámbrica.

2.9 SIGNIFICADO DE WAP.

WAP son las siglas de **Wireless Application Protocol** (protocolo de aplicaciones inalámbricas), un estándar seguro que permite que los usuarios accedan a información de forma instantánea a través de dispositivos inalámbricos como PDAs, teléfonos móviles, buscas, walkie-talkies y teléfonos inteligentes (smartphones).

El estándar WAP soporta la mayoría de las redes inalámbricas, incluyendo CDPD, CDMA, GSM, PDC, PHS, TDMA, FLEX, ReFLEX, iDEN, TETRA, DECT, DataTAC y Mobitex y es soportado por todos los sistemas operativos.

Los WAP que utilizan pantallas y tienen acceso a Internet utilizan lo que se llama micro navegadores, navegadores con archivos de pequeño tamaño, que se pueden adaptar a las restricciones de memorias pequeñas y baja anchura de banda que tienen los dispositivos que utilizan este estándar.

Aunque WAP soporta HTML y XML, el lenguaje WML (una aplicación XML) se ha ideado específicamente para pantallas pequeñas y con navegación con una mano sin teclado. WAP también soporta WMLScript, similar a JavaScript, pero con demandas mínimas de memoria y de energía de la CPU, ya que no contiene muchas de las funciones innecesarias encontradas en otros lenguajes.

El Open Mobile Alliance (OMA), es el consorcio que se ocupa de la definición de diversas normas relacionadas con las comunicaciones móviles, entre ellas las normas WAP [15].

2.10 PHP

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

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el

servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

Una vez que se conoce el concepto de lenguaje de programación de scripts del lado del servidor se puede hablar de PHP. PHP se escribe dentro del código HTML, lo que lo hace realmente fácil de utilizar, al igual que ocurre con el popular ASP de Microsoft, pero con algunas ventajas como su gratuidad, independencia de plataforma, rapidez y seguridad. Cualquiera puede descargar a través de la página principal de PHP www.php.net y de manera gratuita, un módulo que hace que nuestro servidor web comprenda los scripts realizados en este lenguaje. Es independiente de plataforma, puesto que existe un módulo de PHP para casi cualquier servidor web. Esto hace que cualquier sistema pueda ser compatible con el lenguaje y significa una ventaja importante, ya que permite portar el sitio desarrollado en PHP de un sistema a otro sin prácticamente ningún trabajo.

PHP, en el caso de estar montado sobre un servidor Linux u Unix, es más rápido que ASP, dado que se ejecuta en un único espacio de memoria y esto evita las comunicaciones entre componentes COM que se realizan entre todas las tecnologías implicadas en una página ASP.

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. Actualmente 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.

Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones web gracias a la extensa librería de funciones con la que está dotado. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, por poner dos ejemplos.

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 [14].

CAPÍTULO 3.

ANÁLISIS DE REQUERIMIENTOS

En este capítulo se realiza el análisis del sistema. El análisis se realiza a partir de los diagramas de flujo, concluyendo con las especificaciones concretas del sistema de bases de datos.

3.1 DESCRIPCION DEL PROBLEMA.

En la actualidad los gobiernos de diferentes estados de la República Mexicana promueven la inversión extranjera para tener un crecimiento social, económico y laboral.

Los inversionistas viendo la importancia de la propuesta de los diferentes gobiernos de la República Mexicana para la inversión extranjera y el mejoramiento del mismo, desean participar y registrarse en dicha propuesta; aunque uno de los inconvenientes es el como registrar sus proyectos o propuestas.

Su experiencia en el registro de los proyectos no ha sido buena ya que por país o incluso dentro de una misma ciudad tienen que llenar formatos y de diferentes tipos para poder registrar su proyecto de inversión.

Otro inconveniente en el caso de un inversionista que ya ha sido registrado, es que, si desea proponer un nuevo proyecto tendría que llenar los mismos formatos como si registrara un primer proyecto, cabe mencionar que el llenado de los diferentes formatos es de manera manual, los cuales tienen que ser enviados por correo ordinario.

Aunado a esta problemática es que el departamento encargado de administrar la información que envían los inversionistas de sus proyectos es archivado manualmente, es decir, reciben las propuestas en hojas y estos datos recibidos de los inversionistas se deben actualizar cada mes.

Esto implica mucho trabajo y administración de toda la información de los diferentes formatos, y peor si el gobernador requiere la información de los inversionistas para alguna reunión o conferencia, sería muy difícil tener la información en tiempo y forma.

La tecnología actual puede facilitar el acceso y administración de la información al departamento encargado de la misma, ya que en la actualidad hay más de doscientos inversionistas y el llevar la administración manualmente ya no es recomendable.

Por lo tanto considerando los puntos anteriores, se propone realizar un formato de registro de inversión extranjera digital, el cual sea sencillo de llenar, fácil de entender y usar, que contenga la información respectiva para registrar al inversionista utilizando tecnología de punta, esto es lo que da importancia al trabajo propuesto.

3.2 Objetivo

Por los antecedentes anteriores, se pueden citar los siguientes objetivos:

Objetivo General:

- Desarrollar un sistema que permita registrar proyectos de inversión extranjera con dispositivos móviles.

3.3 Objetivos Específicos:

- Diseñar y desarrollar la aplicación API para la PALM con SuperWaba.
- Diseñar y desarrollar la página WEB con PHP.
- Diseñar y desarrollar la base de datos con MySQL.
- Diseñar y desarrollar la comunicación entre la base de datos y la página WEB.
- Instalación y puesta en marcha del sistema.
- Realizar pruebas.

3.4 Restricciones:

- El administrador es el único que podrá dar de alta al personal que se encargará de eliminar la información de los inversionistas.
- Los inversionistas se podrán registrar desde la Web pero no podrán hacer cambios en la información que estén registrando, es decir, hacer modificaciones de información que haya sido enviada
- Solamente el inversionista se podrá registrar una sola vez, otorgándole su nombre de usuario y contraseña para verificar información.
- Únicamente existirá un usuario con privilegios de administrador.

3.5 DESCRIPCION DE LA INFORMACIÓN

La descripción de los datos y su procesamiento se especificará en los diagramas de caso de uso.

3.5.1 DIAGRAMAS DE CASOS DE USOS.

Un diagrama de casos de uso (Use Case Diagram) es una representación gráfica de parte o el total de los actores y casos de uso del sistema, incluyendo sus interacciones. Todo sistema tiene como mínimo un diagrama Main Use Case, que es una representación gráfica del entorno del sistema (actores) y su funcionalidad principal (casos de uso).

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario, por tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar.

Se pueden usar durante las siguientes fases del desarrollo:

- Captura de requisitos
- Planificación de iteraciones de desarrollo
- Validación del sistema

Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente.

3.5.2 Composición

- **Actor:** Un actor representa quien o que inicia una acción dentro del sistema, en otras palabras, es simplemente un rol que es llevado a cabo por una persona o cosa. Un Actor en un diagrama Uso-Caso es representado por una figura en forma de persona.
- **Uso-Caso:** El uso-caso en sí es representado por un ovalo que describe la funcionalidad a *grosso modo* que se requiere por el sistema.
- **Comunicación:** Este elemento representa la relación que existe entre un Uso-Caso y un Actor, dicho elemento es representado simplemente por una línea recta que se extiende de la figura del actor hacia el ovalo del uso-caso.
- **Limite de Sistema (System Boundry):** Empleado para delimitar los límites del sistema, y representado por un rectángulo con color de fondo distintivo.
- **Generalización:** Una generalización indica que un uso-caso (ovalo) es un caso especial de otro caso, en otros términos, representa una relación padre-hijo, donde el hijo puede ser suplido directamente por el padre en cualquier momento. Este elemento es representado por una línea con flecha que se extiende del uso-caso hijo hacia el uso caso padre (general).
- **Inclusión:** Una inclusión es utilizada para indicar que un uso-caso (ovalo) depende de otro caso, dicho de otra manera, significa que la funcionalidad de determinado caso se requiere para realizar las tareas de otro. Este elemento es representado por una línea punteada con flecha y comentario <<include>> que se extiende del uso-caso base hacia el uso caso de inclusión.
- **Extensión:** Una extensión representa una variación de un uso-caso a otro, aunque similar a una **generalización**, una extensión representa una dependencia específica, mientras una generalización no implica que los usos-casos dependen uno del otro. Este elemento es representado por una línea punteada con flecha y comentario <<extend>> que origina del uso-caso base hacia el uso caso de extensión.

Un diagrama de casos de uso muestra, por tanto, los distintos requisitos funcionales que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuarios u otras aplicaciones)

3.6 Registro de Inversión Extranjera con Dispositivos Móviles

Existirán en el sistema tres tipos de usuarios administrador, inversionista y los empleados que están en el departamento encargado de la inversión del estado de Puebla.

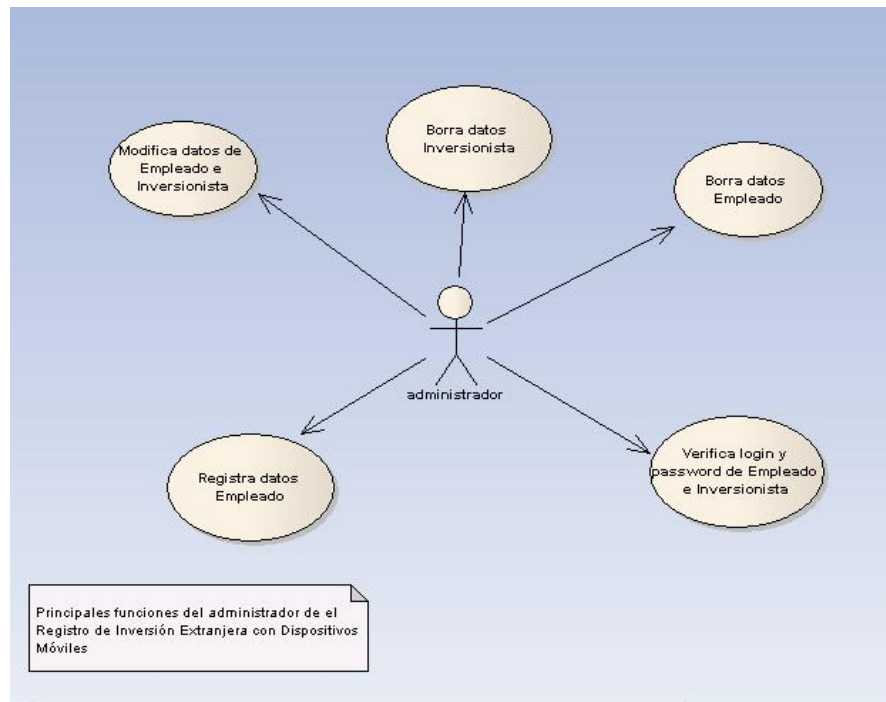


Figura 3.1 Caso de Uso de funciones principales del administrador del “Registro de Inversión Extranjera”

Como se puede observar en la Figura. 3.1 el administrador tendrá las siguientes funciones en el sistema

- Modifica los datos contenidos en la base de datos
- Borra a inversionistas que ya no estén interesados en continuar con un proyecto en el estado
- Borra a los empleados que ya no estarán verificando la información de los inversionistas
- Registra empleados encargados a dar seguimiento a los proyectos que sean propuestos
- Verifica contraseñas y login tanto de empleados como de inversionistas

Cabe señalar que el administrador será el único que podrá hacer algún cambio en la información de algún inversionista registrado así como el registro de los empleados que podrán visualizar dicha información.

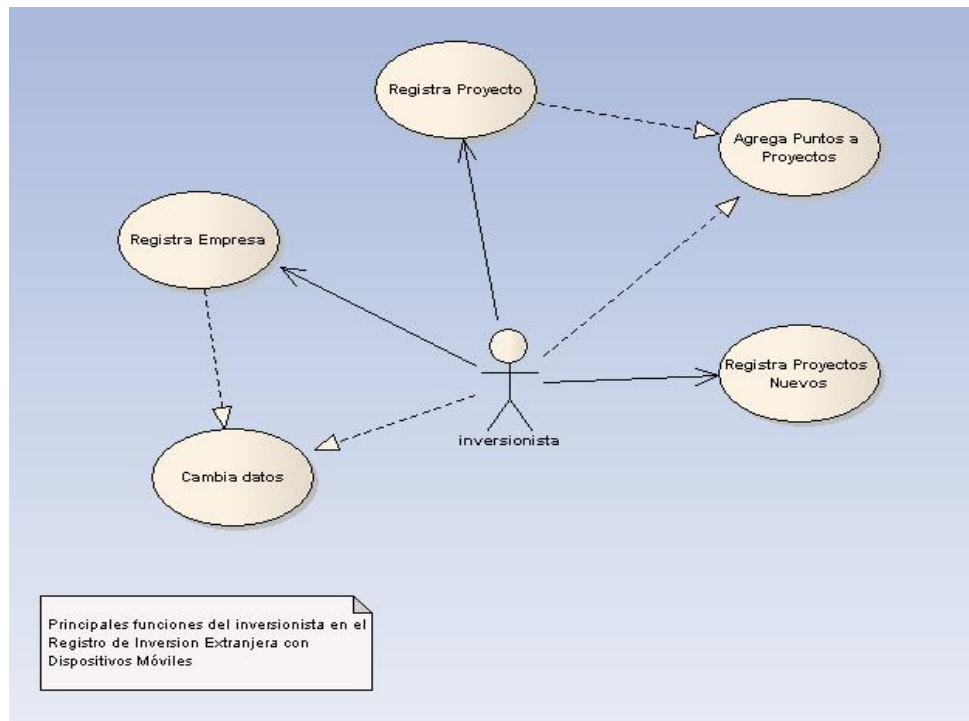


Figura 3.2 Caso de Uso de funciones principales del inversionista en el “Registro de Inversión Extranjera”

En la Figura3.2 se puede observar las funciones del Inversionista, las cuales son:

- Registra los datos de la Empresa a la cual pertenece
- Puede cambiar datos relacionados con el registro de datos de la Empresa
- Registra proyectos
- Agrega puntos a proyectos que ya hayan sido registrados
- Registra proyectos nuevos

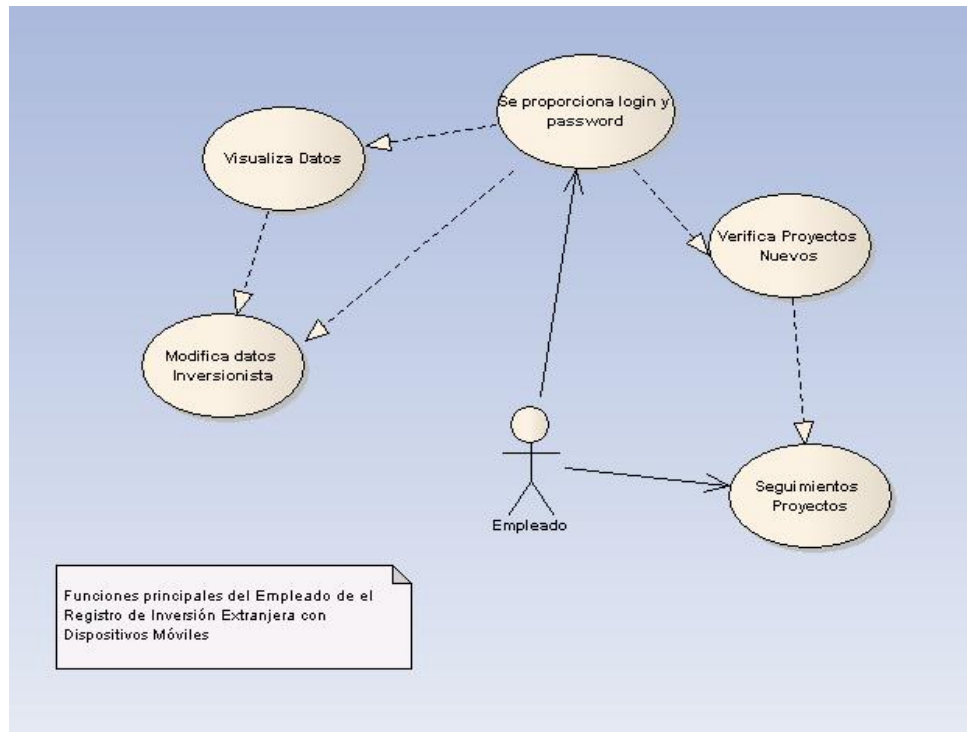


Figura 3.3 Caso de Uso de funciones principales de el empleado en el “Registro de Inversión Extranjera”

En la Figura 3.3 se puede observar las funciones del Empleado, las cuales son:

- Modifica datos del Inversionista
- Visualiza datos
- Verifica Proyectos puestos en marcha y da seguimiento a los mismos.
- Verifica Proyectos Nuevos.

3.7 DIAGRAMAS DE SECUENCIA DEL SISTEMA.

Los diagramas de secuencia están incluidos en la notación UML. Ellos ilustran las interacciones de los actores y las operaciones iniciadas por ellos.

Un **diagrama de secuencia del sistema** es un gráfico que muestra, para un escenario particular de un caso de uso, los eventos que generan actores externos, su orden y los eventos entre sistemas. Todos los sistemas son tratados como cajas negras; el énfasis de los diagramas está en los eventos que cruzan la frontera del sistema, desde los actores al sistema.

Un diagrama de secuencia de un sistema ilustra eventos desde los actores a los sistemas. Su creación tiene lugar durante la fase de análisis de un ciclo de desarrollo y es dependiente del desarrollo previo de los casos de uso.

Antes de realizar el diseño lógico de cómo trabajará una aplicación de software debe investigarse y definir su comportamiento como una “caja negra”, como se mencionó anteriormente.

El Comportamiento de un Sistema es la descripción de **qué** hace el sistema sin explicar cómo lo hace. Una parte de esa descripción es un **diagrama de secuencia del sistema**.

Nombre:	Registrar empleado y modifica datos inversionista
Autor:	Erika Delgado Hernández
Fecha:	17/Noviembre/2008
Descripción:	Se registra empleado para que pueda acceder a la información que requiera de los inversionistas registrados así como modificar su información.
Autores:	Administrador
Precondiciones:	Tanto inversionista como empleado debieron haber proporcionado su usuario y contraseña para identificarse en el sistema.
Flujo normal:	<ol style="list-style-type: none"> 1. El empleado que verifique información deberá pertenecer al departamento que se encarga de la inversión en el estado de Puebla. 2. Sólo el administrador podrá borrar la información de empresas que no desean seguir con proyecto de inversión en el estado de Puebla.
Flujo alternativo:	<ol style="list-style-type: none"> 2. Si el inversionista o el empleado no recuerda su usuario y/o contraseña, se deberá notificar al administrador para que se le proporcione una nueva contraseña.
Poscondiciones:	<ol style="list-style-type: none"> 1. El registro finalizó con éxito.

Tabla 2.1 documento UML para registrar empleado y modificar datos.

Nombre:	Funciones Principales del Inversionista
Autor:	Erika Delgado Hernández
Fecha:	17/Noviembre/2008
Descripción:	Se registra inversionista, una vez registrado podrá acceder al sistema para proponer el proyecto de inversión así como agregar puntos nuevos a proyectos que ya se hayan puesto en marcha.
Autores:	Inversionista
Precondiciones:	El inversionista para poder acceder al sistema debe haberse registrado.
Flujo normal:	<ol style="list-style-type: none"> 1. El Inversionista una vez registrado podrá registrar su proyecto. 2. Solo el administrador podrá borrar la información de empresas que no desean seguir con proyecto de inversión en el estado de Puebla. 3. Sólo el administrador podrá borrar los proyectos que se hayan registrado.

<p>Flujo alternativo:</p> <p>2. Si el inversionista no recuerda usuario y/o contraseña, se deberá notificar al administrador para que se le proporcione una nueva contraseña.</p>
<p>Poscondiciones:</p> <p>1. El registro finalizó con éxito.</p>

Tabla 2.2 documento UML de funciones principales de un Inversionista.

Nombre:	Funciones Principales de un Empleado
Autor:	Erika Delgado Hernández
Fecha:	17/Noviembre/2008
Descripción: Se registra a un empleado, se le proporcionará un usuario y contraseña, el empleado podrá visualizar los datos de los inversionistas así como verificar los proyectos puestos en marcha.	
Autores: Empleado	
Precondiciones: Para visualizar y modificar datos, el empleado debió haber sido registrado por el administrador.	
<p>Flujo normal:</p> <ol style="list-style-type: none"> 1. El empleado una vez registrado podrá visualizar los datos de los inversionistas registrados. 2. Sólo podrá hacer modificaciones a los datos de inversionistas y a proyectos registrados. 3. Sólo el administrador podrá borrar los proyectos que se hayan registrado. 	
<p>Flujo alternativo:</p> <p>2. Si el empleado no recuerda su usuario y/o contraseña, se deberá notificar al administrador para que se le proporcione una nueva contraseña.</p>	
<p>Poscondiciones:</p> <p>1. El registro finalizó con éxito.</p>	

Tabla 2.3 documento UML de funciones principales de un Empleado.

CAPÍTULO 4.

DISEÑO LÓGICO DEL SISTEMA

En este Capítulo se presenta el diseño de la base de datos comenzando por el diseño conceptual seguido por el diseño lógico. Se muestra la estructura del sistema así como cada etapa del desarrollo del mismo.

4.1 DISEÑO CONCEPTUAL.

En esta etapa se debe construir un esquema de la información que se usa en la empresa, independientemente de cualquier consideración física. A este esquema se le denomina *esquema conceptual*. Al construir el esquema, los diseñadores descubren la semántica (significado) de los datos de la empresa: encuentran entidades, atributos y relaciones. El objetivo es comprender:

- La perspectiva que cada usuario tiene de los datos.
- La naturaleza de los datos, independientemente de su representación física.
- El uso de los datos a través de las áreas de aplicación.

El esquema conceptual se puede utilizar para que el diseñador transmita a la empresa lo que ha entendido sobre la información que ésta maneja. Para ello, ambas partes deben estar familiarizadas con la notación utilizada en el esquema.

El esquema conceptual se construye utilizando la información que se encuentra en la especificación de los requisitos de usuario. El diseño conceptual es completamente independiente de los aspectos de implementación, como puede ser el SGBD que se vaya a usar, los programas de aplicación, los lenguajes de programación, el hardware disponible o cualquier otra consideración física. Durante todo el proceso de desarrollo del esquema conceptual éste se prueba y se valida con los requisitos de los usuarios. El esquema conceptual es una fuente de información para el diseño lógico de la base de datos.

4.2 DISEÑO LÓGICO

El diseño lógico es el proceso de construir un esquema de la información que utiliza la empresa, basándose en un modelo de base de datos específico, independiente del SGBD concreto que se vaya a utilizar y de cualquier otra consideración física.

En esta etapa, se transforma el esquema conceptual en un esquema lógico que utilizará las estructuras de datos del modelo de base de datos en el que se basa el SGBD que se vaya a utilizar, como puede ser el modelo relacional, el modelo de red, el modelo jerárquico o el modelo orientado a objetos. Conforme se va desarrollando el esquema lógico, éste se va probando y validando con los requisitos de usuario.

El esquema lógico es una fuente de información para el diseño físico. Además, juega un papel importante durante la etapa de mantenimiento del sistema, ya que permite que los futuros cambios que se realicen sobre los programas de aplicación o sobre los datos, se representen correctamente en la base de datos.

Tanto el diseño conceptual, como el diseño lógico, son procesos iterativos, tienen un punto de inicio y se van refinando continuamente. Ambos se deben ver como un proceso de aprendizaje en el que el diseñador va comprendiendo el funcionamiento de la empresa y el

significado de los datos que maneja. El diseño conceptual y el diseño lógico son etapas clave para conseguir un sistema que funcione correctamente. Si el esquema no es una representación fiel de la empresa, será difícil, sino imposible, definir todas las vistas de usuario (esquemas externos), o mantener la integridad de la base de datos. También puede ser difícil definir la implementación física o el mantener unas prestaciones aceptables del sistema. Además, hay que tener en cuenta que la capacidad de ajustarse a futuros cambios es un sello que identifica a los buenos diseños de bases de datos[1].

4.3 MODELO ENTIDAD RELACIÓN.

El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real consistente en objetos básicos llamados entidades y relaciones entre los objetos.

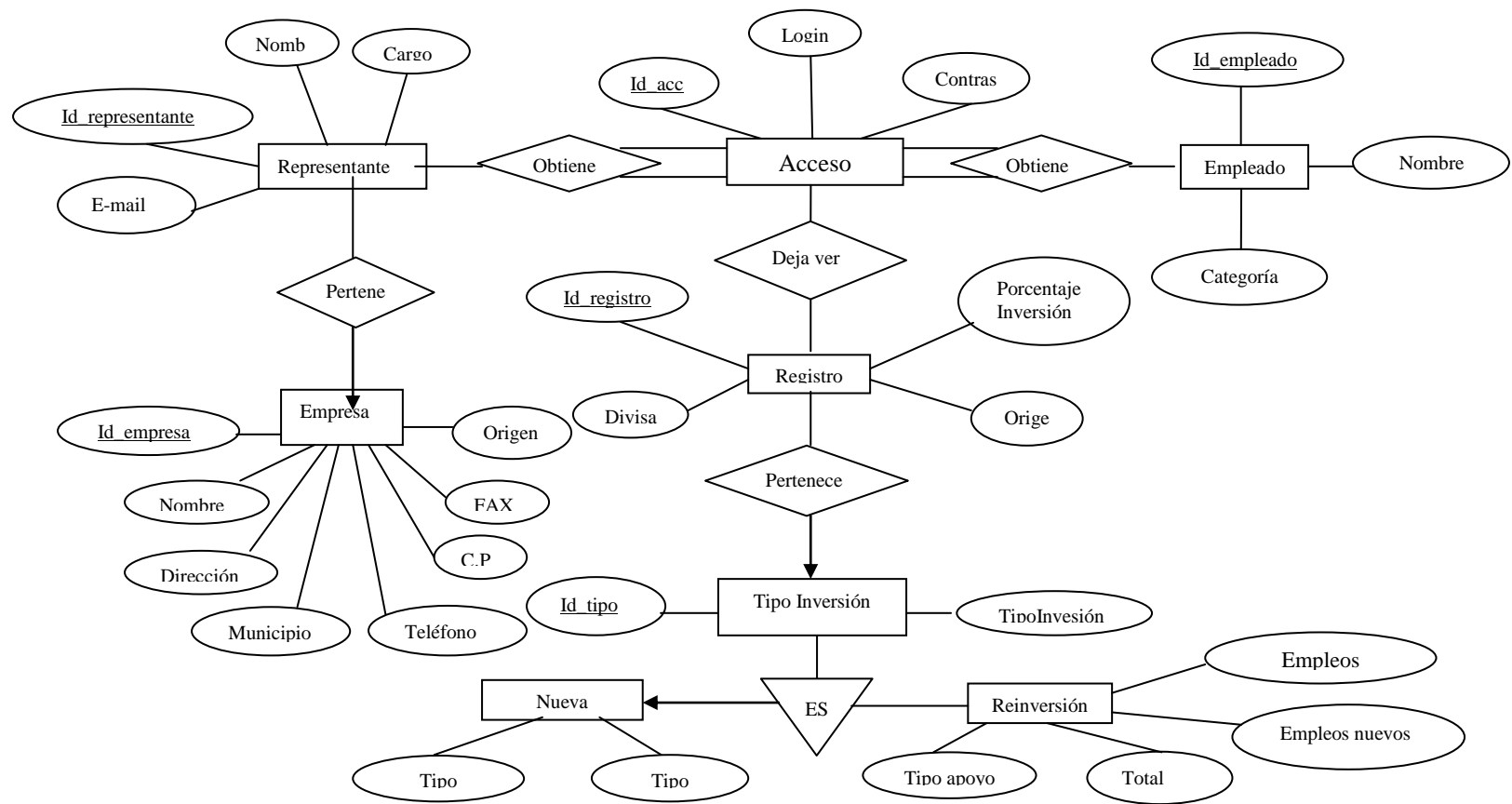
Se desarrolló para facilitar el diseño de bases de datos permitiendo la especificación de un esquema de la empresa que representa la estructura lógica completa de una base de datos.

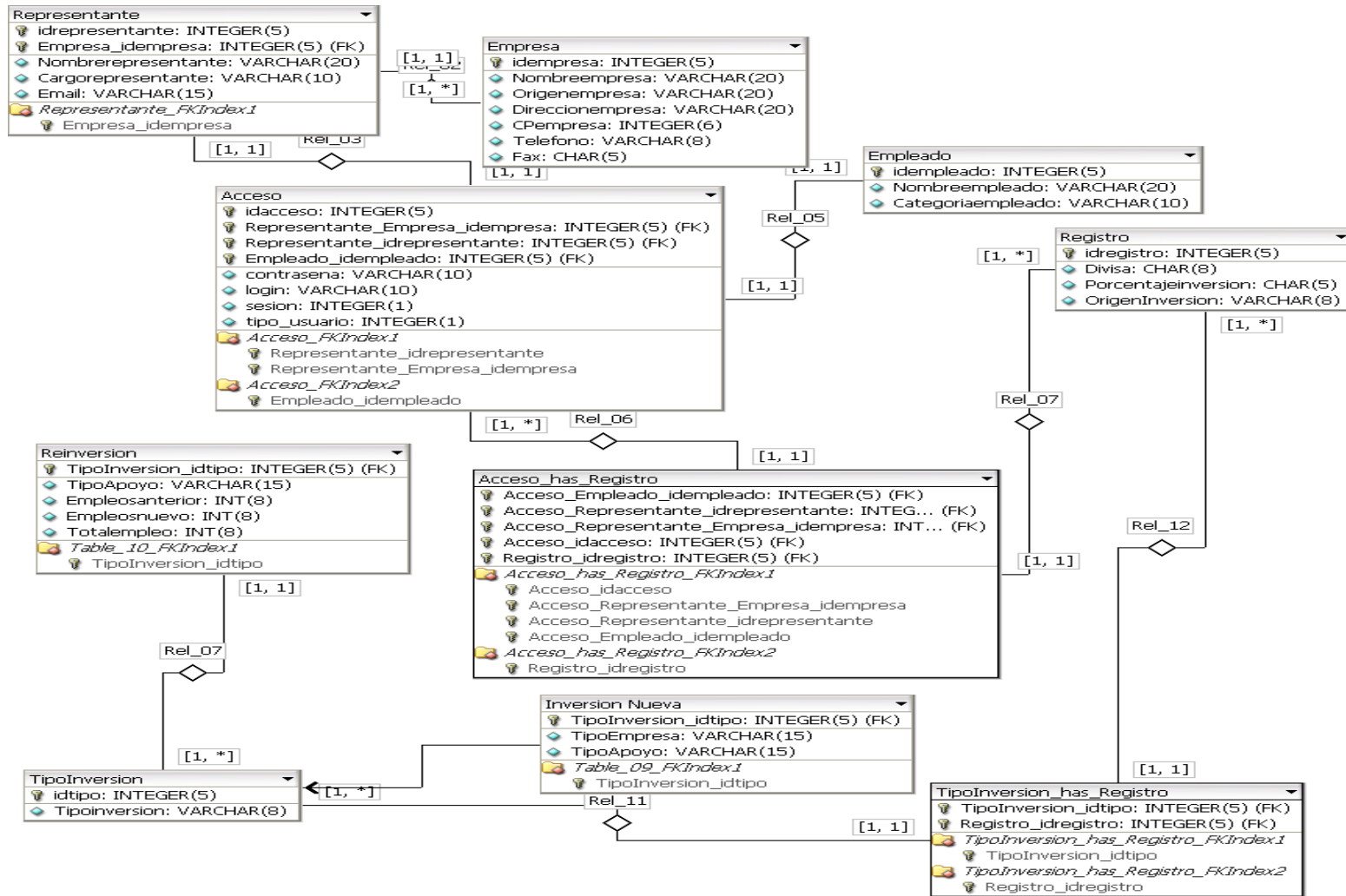
El modelo de datos E-R es uno de los diferentes modelos de datos semánticos; el aspecto semántico del modelo yace en la representación de significado de los datos.

El modelo E-R es extremadamente útil para hacer corresponder los significados e interacciones de las empresas del mundo real con un esquema conceptual. Debido a esta utilidad, muchas herramientas de diseño de bases de datos se basan en los conceptos del modelo E-R.

El modelo entidad relación consiste en los siguientes pasos:

1. Se parte de una descripción textual del problema o sistema de información a automatizar (los requisitos).
2. Se hace una lista de los sustantivos y verbos que aparecen.
3. Los sustantivos son posibles entidades o atributos.
4. Los verbos son posibles relaciones.
5. Analizando las frases se determina la cardinalidad de las relaciones y otros detalles.
6. Se elabora el diagrama (o diagramas) entidad-relación.
7. Se completa el modelo con listas de atributos y una descripción de otras restricciones que no se pueden reflejar en el diagrama





4.4 ESTRUCTURA FÍSICA DE LA BASE DE DATOS.

El nivel físico es el nivel más bajo de abstracción, en este nivel se muestra a detalle cómo se almacenan los datos, en este caso la estructura física muestra como se guardan los datos en MySQL, con la estructura de las tablas.

```
CREATE TABLE Empresa (  
  idempresa INTEGER(5) UNSIGNED NOT NULL AUTO_INCREMENT,  
  Nombreempresa VARCHAR(20) NULL,  
  Origenempresa VARCHAR(20) NULL,  
  Direccionempresa VARCHAR(20) NULL,  
  CPempresa INTEGER(6) UNSIGNED NULL,  
  Telefono VARCHAR(8) NULL,  
  Fax CHAR(5) NULL,  
  PRIMARY KEY(idempresa)  
);
```

```
CREATE TABLE Registro (  
  idregistro INTEGER(5) UNSIGNED NOT NULL AUTO_INCREMENT,  
  Divisa CHAR(8) NULL,  
  Porcentajeinversion CHAR(5) NULL,  
  OrigenInversion VARCHAR(8) NULL,  
  PRIMARY KEY(idregistro)  
);
```

```
CREATE TABLE Empleado (  
  idempleado INTEGER(5) UNSIGNED NOT NULL AUTO_INCREMENT,  
  Nombreempleado VARCHAR(50) NULL,  
  Categoriaempleado VARCHAR(50) NULL,  
  Email VARCHAR(50) NULL,  
  PRIMARY KEY(idempleado)  
);
```

```
CREATE TABLE TipoInversion (  
    idtipo INTEGER(5) UNSIGNED NOT NULL AUTO_INCREMENT,  
    Tipoinversion VARCHAR(8) NULL,  
    PRIMARY KEY(idtipo)  
);
```

```
CREATE TABLE Reinversion (  
    TipoInversion_idtipo INTEGER(5) UNSIGNED NOT NULL,  
    TipoApoyo VARCHAR(15) NULL,  
    Empleosanterior INT(8) NULL,  
    Empleosnuevo INT(8) NULL,  
    Totalempleo INT(8) NULL,  
    nombreReinversion VARCHAR(50) NULL,  
    PRIMARY KEY(TipoInversion_idtipo),  
    INDEX Table_10_FKIndex1(TipoInversion_idtipo)  
);
```

```
CREATE TABLE Representante (  
    idrepresentante INTEGER(5) UNSIGNED NOT NULL AUTO_INCREMENT,  
    Empresa_idempresa INTEGER(5) UNSIGNED NOT NULL,  
    Nombrepresentante VARCHAR(20) NULL,  
    Cargorepresentante VARCHAR(10) NULL,  
    Email VARCHAR(15) NULL,  
    PRIMARY KEY(idrepresentante, Empresa_idempresa),  
    INDEX Representante_FKIndex1(Empresa_idempresa)  
);
```

```
CREATE TABLE Inversion Nueva (  
    TipoInversion_idtipo INTEGER(5) UNSIGNED NOT NULL,  
    TipoEmpresa VARCHAR(15) NULL,  
    TipoApoyo VARCHAR(15) NULL,  
    nombreInversion VARCHAR(50) NULL,  
    PRIMARY KEY(TipoInversion_idtipo),
```

```
INDEX Table_09_FKIndex1(TipoInversion_idtipo)
);
```

```
CREATE TABLE TipoInversion_has_Registro (
  TipoInversion_idtipo INTEGER(5) UNSIGNED NOT NULL,
  Registro_idregistro INTEGER(5) UNSIGNED NOT NULL,
  PRIMARY KEY(TipoInversion_idtipo, Registro_idregistro),
  INDEX TipoInversion_has_Registro_FKIndex1(TipoInversion_idtipo),
  INDEX TipoInversion_has_Registro_FKIndex2(Registro_idregistro)
);
```

```
CREATE TABLE Acceso (
  idacceso INTEGER(5) UNSIGNED NOT NULL AUTO_INCREMENT,
  Representante_Empresa_idempresa INTEGER(5) UNSIGNED NOT NULL,
  Representante_idrepresentante INTEGER(5) UNSIGNED NOT NULL,
  Empleado_idempleado INTEGER(5) UNSIGNED NOT NULL,
  contraseña VARCHAR(10) NULL,
  login VARCHAR(10) NULL,
  sesion INTEGER(1) UNSIGNED NULL,
  tipo_usuario INTEGER(1) UNSIGNED NULL,
  PRIMARY KEY(idacceso, Representante_Empresa_idempresa,
  Representante_idrepresentante, Empleado_idempleado),
  INDEX Acceso_FKIndex1(Representante_idrepresentante,
  Representante_Empresa_idempresa),
  INDEX Acceso_FKIndex2(Empleado_idempleado)
);
```

```
CREATE TABLE Acceso_has_Registro (
  Acceso_Empleado_idempleado INTEGER(5) UNSIGNED NOT NULL,
  Acceso_Representante_idrepresentante INTEGER(5) UNSIGNED NOT NULL,
  Acceso_Representante_Empresa_idempresa INTEGER(5) UNSIGNED NOT NULL,
  Acceso_idacceso INTEGER(5) UNSIGNED NOT NULL,
  Registro_idregistro INTEGER(5) UNSIGNED NOT NULL,
```

```
PRIMARY KEY(Acceso_Empleado_idempleado,  
Acceso_Representante_idrepresentante, Acceso_Representante_Empresa_idempresa,  
Acceso_idacceso, Registro_idregistro),  
INDEX Acceso_has_Registro_FKIndex1(Acceso_idacceso,  
Acceso_Representante_Empresa_idempresa, Acceso_Representante_idrepresentante,  
Acceso_Empleado_idempleado),  
INDEX Acceso_has_Registro_FKIndex2(Registro_idregistro)  
);
```

CAPÍTULO 5.

IMPLEMENTACIÓN DEL SISTEMA

En este capítulo se presenta la implementación del sistema, se describe paso a paso el funcionamiento del mismo.

5.1 FUNCIONAMIENTO DEL SISTEMA “REGISTRO DE INVERSIÓN EXTRANJERA CON DISPOSITIVOS MÓVILES”.

En la Figura 5.1 se presenta la interfaz de la base de datos del sistema de Registro de Inversión Extranjera implementada con MySQL. En ella se distinguen los siguientes campos [11]:

Acceso: Permite almacenar los datos de acceso al sistema, tanto de inversionistas como de Empleados.

Empleado: Permite almacenar los datos de los Empleados del Departamento de inversión.

Empresa: Permite almacenar los datos de la empresa que desea realizar una inversión.

Representante: Permite almacenar los datos del representante de la empresa que se registro para realizar una inversión.

Registro: Permite almacenar los datos correspondientes a la información general de la empresa.

Tipo de Inversión: Permite almacenar el tipo de inversión que realizara cada empresa registrada.

Inversión Nueva: Permite almacenar los nombres de proyectos y tipo de apoyo de empresas que registran por primera vez un proyecto.

Reinversión: Permite almacenar los nombres de proyectos y tipo de apoyo de empresas que ya han registrado un proyecto y requieren que se de seguimiento al mismo.

Tabla	Acción	Registros	Tipo	Cotejamiento	Tamaño	Residuo
<input type="checkbox"/> acceso	[Icons]	15	MyISAM	utf8_general_ci	4.5 KB	24
<input type="checkbox"/> acceso_has_registro	[Icons]	0	MyISAM	utf8_general_ci	1.0 KB	
<input type="checkbox"/> empleado	[Icons]	8	MyISAM	utf8_general_ci	2.3 KB	24
<input type="checkbox"/> empresa	[Icons]	10	MyISAM	utf8_general_ci	2.8 KB	156
<input type="checkbox"/> inversionnueva	[Icons]	1	MyISAM	utf8_general_ci	2.0 KB	
<input type="checkbox"/> registro	[Icons]	14	MyISAM	utf8_general_ci	2.3 KB	
<input type="checkbox"/> reinversion	[Icons]	1	MyISAM	utf8_general_ci	2.0 KB	
<input type="checkbox"/> representante	[Icons]	13	MyISAM	utf8_general_ci	3.5 KB	
<input type="checkbox"/> tipoinversion	[Icons]	2	MyISAM	utf8_general_ci	2.0 KB	
<input type="checkbox"/> tipoinversion_has_registro	[Icons]	0	MyISAM	utf8_general_ci	1.0 KB	
10 tabla(s)	Número de filas	64	MyISAM	utf8_general_ci	23.5 KB	204

Figura. 5.1 Base de datos implementada en PHP MyAdmin.

Para la implementación del sistema se utilizó Dreamweaver. En la Figura 5.2 se muestra una de tantas pantallas y es de bienvenida al usuario o principal.

Descripción de las áreas de la pantalla. En la pantalla o interfaz se identifican un área de imágenes, para dar a conocer parte de la ciudad de Puebla. Un área con cinco íconos que presentan información de la Institución como misión, visión, valores y dos del sistema como inicio y sugerencias, cabe mencionar que estas dos áreas se repiten en todas las pantallas o interfaces. Un área o encabezado que presenta la bienvenida al usuario acompañado de texto. Un área de enlaces a otras páginas como son las secretarías de: Cultura y Deporte, Economía, Empleo, Gobernación Hacienda. Finalmente un área para recibir del usuario datos de usuario y contraseña, esto le permitirá ingresar al sistema. Estas áreas varían por pantalla y en su momento se describirán.

Funcionalidad. El inversionista, al ingresar datos de usuario y contraseña, la página le permitirá ir a la página de registro de proyecto o de seguimiento del mismo.



Figura. 5.2 Pantalla Inicial del Sistema

5.2 REGISTRO AL SISTEMA DE INVERSIONISTAS.

En la Figura 5.3 y 5.4 en estas interfaces el usuario podrá registrar los datos de su empresa así como los datos del representante.

Descripción de las áreas de la pantalla: Un área en el que el inversionista podrá registrar datos de la empresa así como los datos de su representante, requisitos indispensables para poder proponer un proyecto de inversión.

Funcionalidad. El inversionista, al ingresar datos de su empresa, la página le permitirá avanzar para seguir con el registro siempre que se hayan ingresado los datos en los campos.

Inicio ... Misión ... Visión ... Fichas ... Sugerencias ...

Datos de la Empresa

Nombre de la Empresa

País de Origen de la Empresa

Calle y número

Colonia

C.P.

Teléfono(s)

Fax

Enviar datos Limpicar

Figura. 5.3 Pantalla de registro del Inversionista.

Inicio ... Misión ... Visión ... Fichas ... Sugerencias ...

Datos de la Empresa

Nombre del Representante

Cargo del Representante

Email

Enviar datos Limpicar Atras

Figura.5.4 Pantalla de registro de Representante

Una vez que el inversionista ha llenado los campos requeridos para registrar los datos de la empresa y del representante, el inversionista procederá a ingresar su usuario y contraseña como se muestra en la Figura.5.5.

Descripción de las áreas de la pantalla: Un área en el que el inversionista podrá registrar datos de usuario y contraseña, datos con los cuales el inversionista tendrá acceso al sistema.

Funcionalidad. El inversionista, al ingresar datos de usuario y contraseña, la página le permitirá avanzar para seguir con el registro siempre que se hayan ingresado los datos en los campos.

...: Inicio: Misión: Visión: Valores: Sugerencias ...

Datos de la Empresa

Usuario(max 10)

Password(max 10)

Confirma

...:Links de Interes:...

Inegi

Secretaria de la Cultura y el Deporte

Secretaria de Economia

Secretaria de Empleo

Figura. 5.5 Registro de usuario y contraseña

5.3 VALIDACIÓN DE DATOS.

Para que el registro de los inversionistas se almacene en la base de datos de Inversión, se validaron dichos datos. Es decir, en una tabla se tienen palabras claves por cada inversionista. Si un inversionista da sus datos personales de manera incorrecta, al compararse estos con los almacenados, no podrá ingresar al sistema.

Se utilizó PHP y MySQL para comunicar al usuario con la base de datos y realizar una comparación de datos del usuario con los reservados, respectivamente como se muestra en el siguiente código [8]:

```
mysql_connect('localhost','root','**')or die ('Ha fallado la conexión: '.mysql_error());
mysql_select_db('inversion')or die ('Error al seleccionar la Base de Datos:
'.mysql_error());
```

5.3.1 Registro de Datos de la Empresa y Representante

Para el registro de los datos de la empresa y del representante se valida que el inversionista haya introducido todos los datos requeridos y en caso que el inversionista no haya llenado algún campo, el sistema enviará un aviso de campo vacío como, esto se muestra en el siguiente código [9]:

```
$nombre = $_POST["nombre"];
$origen = $_POST["origen"];
$calle = $_POST["calle"];
$colonia = $_POST["colonia"];
$cp = $_POST["cp"];
$telefono = $_POST["telefono"];
$fax = $_POST["fax"];

if($nombre == NULL | $origen == NULL | $calle == NULL | $colonia == NULL | $cp ==
NULL | $telefono == NULL | $fax == NULL){
    echo '
        <p>&nbsp;</p>
        <p>&nbsp;</p>
        <p>&nbsp;</p>
        <p align="center"><span class="style5"><a href="reg1.php" class="style3
style1 style6"> <font color="#336633" size="4" face="Arial, Helvetica, sans-
serif">Existe un campo vacío,de click para continuar</font></a></span></p> '
    }
}
```

5.3.2 Validación de datos de la Empresa

Primero se realiza la conexión al servidor, pasando como parámetros el nombre del servidor, usuario y contraseña, después de que la conexión haya sido exitosa, se realiza la conexión a la base de datos, pasando como parámetros el nombre de la base de datos como se muestra:

```
mysql_connect('localhost','root','**')or die ('Ha fallado la conexión: '.mysql_error());
mysql_select_db('inversion')or die ('Error al seleccionar la Base de Datos:
'.mysql_error());
```

Una vez que se haya conectado al servidor y la base de datos, se asigna el valor que tengan los cuadros de texto a unas variables como se muestra:

```
$nombre = $_POST["nombre"];
$origen = $_POST["origen"];
$calle = $_POST["calle"];
$colonia = $_POST["colonia"];
$cp = $_POST["cp"];
$telefono = $_POST["telefono"];
$fax = $_POST["fax"];
```

Una vez que se asignó valores a las variables, se verifica que cada variable tenga un valor diferente de NULL, esto con la finalidad que no se inserten campos vacíos a la tabla Empresa, en caso de que exista una variable con el valor NULL se mostrara una pantalla con el mensaje de que existe un campo vacío, si las variables no contiene un valor NULL entonces se insertara los datos de los campos de texto en la tabla Empresa, como se muestra a continuación:

```

if($nombre == NULL | $origen == NULL | $calle == NULL | $colonia == NULL | $cp ==
NULL | $telefono == NULL | $fax == NULL){
    echo '
        <p>&nbsp;</p>
        <p>&nbsp;</p>
        <p>&nbsp;</p>
        <p align="center"><span class="style5"><a href="reg1.php" class="style3
style1 style6"><font color="#336633" size="4" face="Arial, Helvetica, sans-serif">Existe
un campo vacío, de click para continuar</font></a></span></p> ' ;
    }
else{
mysql_query("insert into empresa
(nombreempresa,origenempresa,calle_numero,colonia,cpempresa,telefono,fax) values
('$nombre','$origen','$calle','$colonia','$cp','$telefono','$fax')");
}

```

5.3.3 Validación de Datos del Representante

Una vez que se haya conectado al servidor y la base de datos, se asigna el valor que tengan los cuadros de texto a unas variables como se muestra:

```

$nombre = $_POST['nombre'];
$cargo = $_POST['cargo'];
$email = $_POST['email'];

```

Asignado los valores a las variables, se verifica que cada variable tenga un valor diferente de NULL, esto con la finalidad que no se inserten campos vacíos a la tabla Representante, en caso de que exista una variable con el valor NULL se mostrara una pantalla con el mensaje de que existe un campo vacío, si las variables no contiene un valor NULL entonces se insertara los datos de los campos de texto en la tabla Representante, como se muestra a continuación:

```

if($nombre == NULL | $cargo == NULL | $email == NULL){
    echo '
        <p>&nbsp;</p>
        <p>&nbsp;</p>
        <p>&nbsp;</p>
        <p align="center"><span class="style5"><a href="regRepresentante1.php"
class="style3 style1 style6"><font color="#336633" size="4" face="Arial, Helvetica, sans-
serif">Existe un campo vacío,de click para continuar</font></a></span></p> ' ;
    }else{
mysql_query("insert into
representante(empresa_idempresa,nombrerepresentante,cargorepresentante,email) values
('idempresa','$nombre','$cargo','$email')");
}

```

5.3.4 Validación de Usuario y Contraseña

Se asigna los valores que contengan los cuadros de texto a unas variables, esto con la finalidad de realizar las validaciones correspondientes, como se muestra:

```
$usuario = $_POST["login"];
$contra = $_POST["password"];
$contra2 = $_POST["password2"];
```

Asignados los valores a las variables, se verifica que cada variable tenga un valor diferente de NULL, esto con la finalidad que no se inserten campos vacíos a la tabla Acceso, en caso de que exista una variable con el valor NULL se mostrara una pantalla con el mensaje de que existe un campo vacío, si las variables no contiene un valor NULL entonces se insertara los datos de los campos de texto en la tabla Acceso, como se muestra a continuación:

```
if($usuario==NULL | $contra == NULL | $contra2 == NULL){
    echo '<p align="center"><span class="style5"><a href="usuario.php" class="style3
style1 style6"><font color="#336633" size="4" face="Arial, Helvetica, sans-serif">Existe
un campo vacío,de click para continuar</font></a></span></p> ';
}
}
```

Después de que se valida que no exista un campo vacío, se verifican que las contraseñas sean las correctas, esto con la finalidad que se ingrese un valor correcto a la tabla Acceso, ya que en esa contraseña ingresaran al sistema

```
if ($contra != $contra2){
    //echo "lo sentimos, pero las contraseñas no coinciden";
    echo '<p align="center"><span class="style5"><a href="usuario.php" class="style3
style1 style6"><font color="#336633" size="4" face="Arial, Helvetica, sans-
serif">Lo sentimos,pero las contraseñas no coinciden,de click para
continuar</font></a></span></p> ';
}
}
```

Si las contraseñas con correctas y no existe un campo vacío, ser verifica que el usuario que eligió para ingresar al sistema no exista en la tabla de Acceso como se muestra a continuación:

```
$validar = mysql_query("SELECT usuario FROM acceso WHERE usuario =
'$usuario'");
$existe = mysql_num_rows($validar);

if ( $existe > 0){
    //echo "usuario en uso";
    echo '<p align="center"><span class="style5"><a href="usuario.php" class="style3
style1 style6"><font color="#336633" size="4" face="Arial, Helvetica, sans-serif">Lo
sentimos,usuario en uso,de click para continuar</font></a></span></p> ';
}
}
```

Si el usuario no existe, si las contraseñas coinciden y no existe campos vacíos se inserta la información en la tabla Acceso como se muestra:

```
mysql_query("INSERT INTO acceso(password,usuario,sesion,tipo_usuario) values
('$usuario','$contra','2','2')");
```

```
echo ' <p align="center"><span class="style5"><a href="Dinversion.php" class="style3
style1 style6"><font color="#336633" size="4" face="Arial, Helvetica, sans-serif">Da
click para continuar con el Registro</font></a></span></p> ';
```

5.3.5 Validación de usuario cuando inicia sesión.

Para iniciar sesión en el sistema, cuando el inversionista de clic en “Entrar” se validará que el usuario no haya ingresado algún carácter no valido, como se muestra:

```
function quitar($mensaje)
{
    $nopermitidos = array('","', '\\', '<', '>', '\\');
    $mensaje = str_replace($nopermitidos, '', $mensaje);
    return $mensaje;
}
```

Una vez que el inversionista haya ingresado su usuario y contraseña, el sistema validará que el usuario y contraseña que ingreso este en la base de datos, una vez validad e ingresado al sistema el usuario procederá a ingresar un proyecto nuevo o que se de seguimiento al proyecto registrado, como se muestra:

```
if(trim($HTTP_POST_VARS["usuario"]) != "" &&
trim($HTTP_POST_VARS["password"]) != "")
{
    $usuario = strtolower(htmlentities($HTTP_POST_VARS["usuario"], ENT_QUOTES));
    $password = $HTTP_POST_VARS["password"];
    $result = mysql_query('SELECT password, usuario FROM acceso WHERE
usuario='.$usuario.'');
    if($row = mysql_fetch_array($result)){
        if($row["password"] == $password){
            $_SESSION["k_username"] = $row['usuario'];

            echo ' <p align="center">&nbsp;</p>
            <p align="center"><font color="#336633" size="4" face="Arial,
Helvetica, sans-serif">Bienvenido al Sistema de Registro de Inversi&oacute;n Extranjera,
de click para especificar el Tipo de Inversión requerido </font></p>
            <p align="center">&nbsp;</p>
            <p align="center"><a href="tipoInv.php" style="text-decoration:none"><font
color="#669900" size="5" face="Arial, Helvetica, sans-serif"> Continuar
</font></a></p>;

        }else{
            echo '
            <p align="center" ><font color="#669900" size="5"
face="Arial, Helvetica, sans-serif">Lo sentimos pero su contraseña es incorrecta y no
puede ingresar al Sistema</font></p>;
            echo ' <p align="center">&nbsp;</p>
```

```

        <p align="center"><a href="pantallaB.php" style="text-decoration:none"><font color="#669900" size="5" face="Arial, Helvetica, sans-serif">
Continuar </font></a></p>;

    }
    }
mysql_free_result($result);
}else{
    echo '<p align="center" ><font color="#669900" size="5" face="Arial, Helvetica,
sans-serif">Lo sentimos pero debe ingresar un usuario y contraseña para ingresar al
Sistema</font></p>;
        echo '<p align="center">&nbsp;</p>
        <p align="center"><a href="pantallaB.php" style="text-decoration:none"><font color="#669900" size="5" face="Arial, Helvetica, sans-serif">
Continuar </font></a></p>;
    }
}

```

5.4 Registro de Inversión Nueva

Una vez que el inversionista haya registrado los datos de empresa, representante, usuario y contraseña, procederá a registrar el tipo de inversión y el apoyo que requiere Figura. 5.6.

Descripción de las áreas de la pantalla: Un área en el que el inversionista podrá registrar el tipo de inversión y apoyo que requiere, datos con los cuales el inversionista podrá registrar lo que necesita para llevar a cabo un proyecto de inversión.

Funcionalidad. El inversionista, al ingresar datos de Inversión Nueva, la página le permitirá ingresar los datos del proyecto de inversión nueva y finalizar el registro, siempre que se hayan ingresado los datos en los campos.



Figura.5.6 Registro de Inversión Nueva

Si el inversionista registrado desea continuar con la inversión que registro como nueva, entonces procederá a registrar la inversión como reinversión Figura.5.7

Descripción de las áreas de la pantalla: Un área en el que el inversionista podrá registrar el proyecto como uno de reinversión, datos con los cuales el inversionista podrá registrar lo que necesita para llevar a cabo un proyecto de inversión.

Funcionalidad. El inversionista, al ingresar datos de Reinversión, la página le permitirá ingresar los datos del proyecto de Reinversión y finalizar el registro, siempre que se hayan ingresado los datos en los campos.

...: Inicio: Misión: Visión: Valores: Sugerencias ...

Reinversión

Empleos Generados:

Empleos permanentes hasta antes de realizar la inversión

Empleos nuevos generados por la inversión realizada

Total de empleos permanentes

Tipo de Apoyo Especifico:

Seguimiento de Proyecto

Capacitación de Personal

...: Links de Interes:...

Inegi

Secretaría de la Cultura y el Deporte

Secretaría de Economía

Secretaría de Empleo

Figura.5.7 Registro de Reinversión

5.5 PANTALLA DE ADMINISTRADOR

En la pantalla de administrador se pide que ingrese el usuario y contraseña, esto con la finalidad de tener un mejor control de información de inversionistas como empleados Figura.5.8.

Descripción de las áreas de la pantalla: Un área en el que el administrador como el empleado, podrá registrar el tipo de inversión y apoyo que requiere, datos con los cuales el inversionista podrá registrar lo que necesita para llevar a cabo un proyecto de inversión.

Funcionalidad. El inversionista, al ingresar datos de Inversión Nueva, la página le permitirá ingresar los datos del proyecto de inversión nueva y finalizar el registro, siempre que se hayan ingresado los datos en los campos.

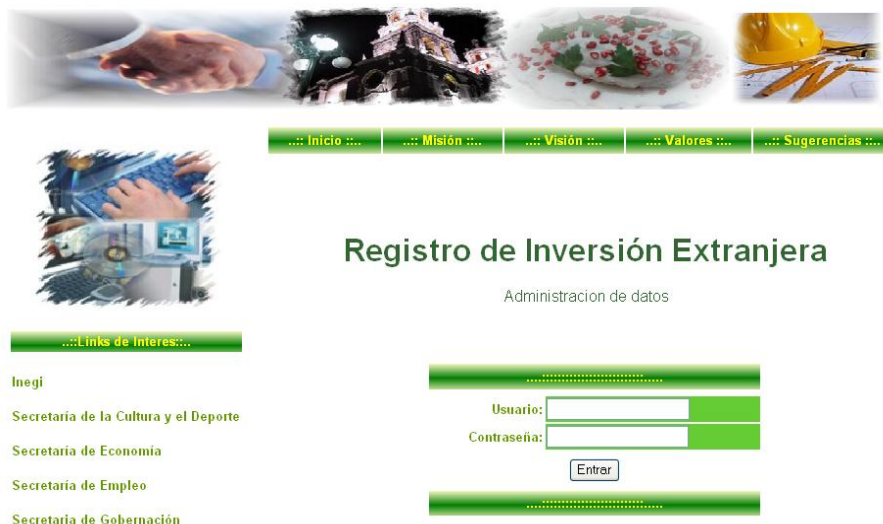


Figura.5.8 Pantalla de Administrador

En la siguiente pantalla el administrador podrá elegir una de las opciones que desee realizar tanto actualización de datos como eliminar los mismos Figura.5.9



Figura.5.9 Menú de administrador

Como se menciona en el Capítulo 3 Análisis de Requerimientos, sólo el administrador es el que podrá registrar a empleados así como modificar y eliminar información de los mismos, y también del manejo de la información de los inversionistas Figura.5.10.

Descripción de las áreas de la pantalla: Un área en el que el administrador, podrá registrar a los empleados que verificaran la información de los inversionistas.

Funcionalidad. El administrador solo llenará los campos requeridos para dar de alta al empleado.



[...: Inicio ...](#) [...: Misión ...](#) [...: Visión ...](#) [...: Fichas ...](#) [...: Sugerencias ...](#)



Registro de Inversión Extranjera

Administración de datos

[...: Links de Interés:...](#)

Inegi

Secretaría de la Cultura y el
Deporte

Secretaría de Economía

Secretaría de Empleo

Secretaría de Gobernación

Nombre del Empleado:

Cargo del Empleado:

Email:

Fig.5.10 Registro de Empleados al sistema.

CAPÍTULO 6.

PRUEBAS Y RESULTADOS.

En este capítulo se realizan pruebas del sistema y su implementación, a través de una PALM y una computadora portátil, mostrando los resultados obtenidos.

6.1 INGRESO AL SISTEMA A TRAVÉS DE UN MÓVIL

Para registrar los datos de una empresa a través de un móvil será necesario que esté dentro del alcance de una red inalámbrica, como se muestra en la Figura. 6.1. Una vez que el móvil se haya conectado a una red inalámbrica, se deberá ingresar la dirección donde se ubica el sistema de “Registro de Inversión Extranjera” como se muestra en la Figura 6.2

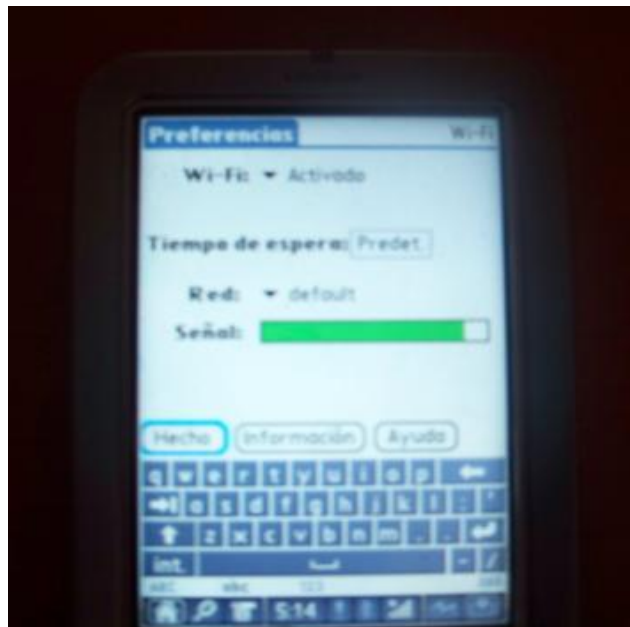


Figura.6.1 Conectando a una Red Inalámbrica disponible

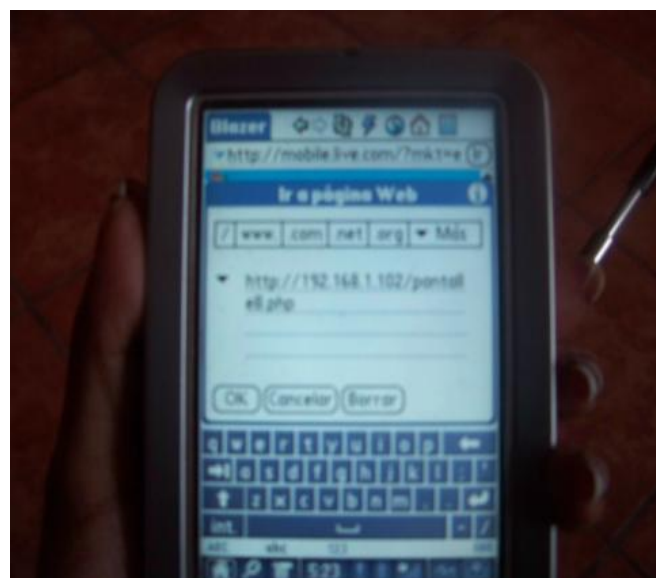


Figura.6.2 Ingreso de la dirección donde se ubica el sistema

6.1.2 REGISTRO DE UNA EMPRESA A TRAVÉS DE UN MÓVIL.

Una vez que se ha ingresado la dirección del sistema, se mostrará la pantalla principal del sistema que verán los inversionistas, para este caso, se ingresarán los datos de una empresa que no se encuentra registrada en la base de datos, por lo que se dará clic en la opción “Registrarse”, Figura 6.3

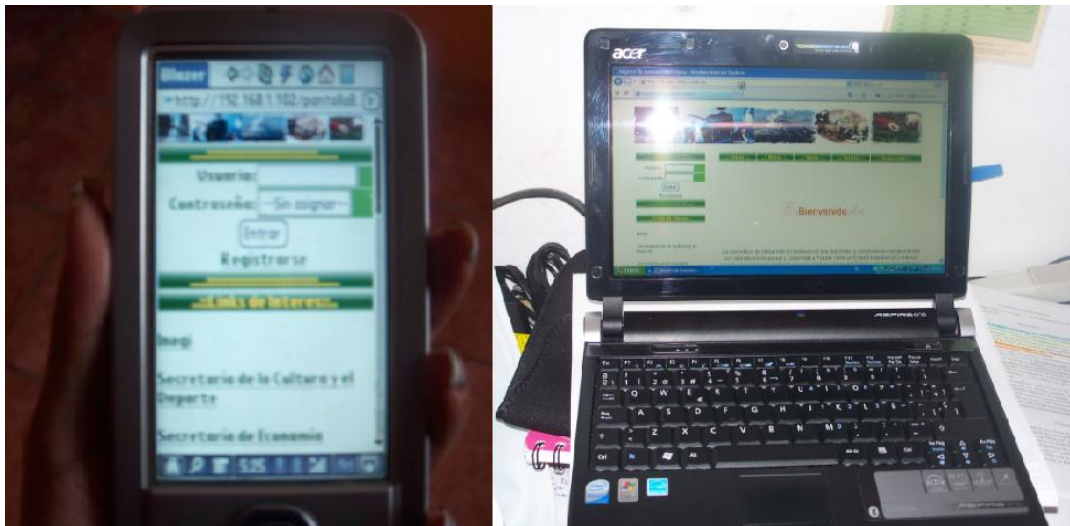


Figura.6.3 Pantalla Principal del sistema

Una vez que se haya dado clic en “Registrarse” se mostrará una pantalla donde se dará inicio al registro de la empresa, en la Figura 6.4.

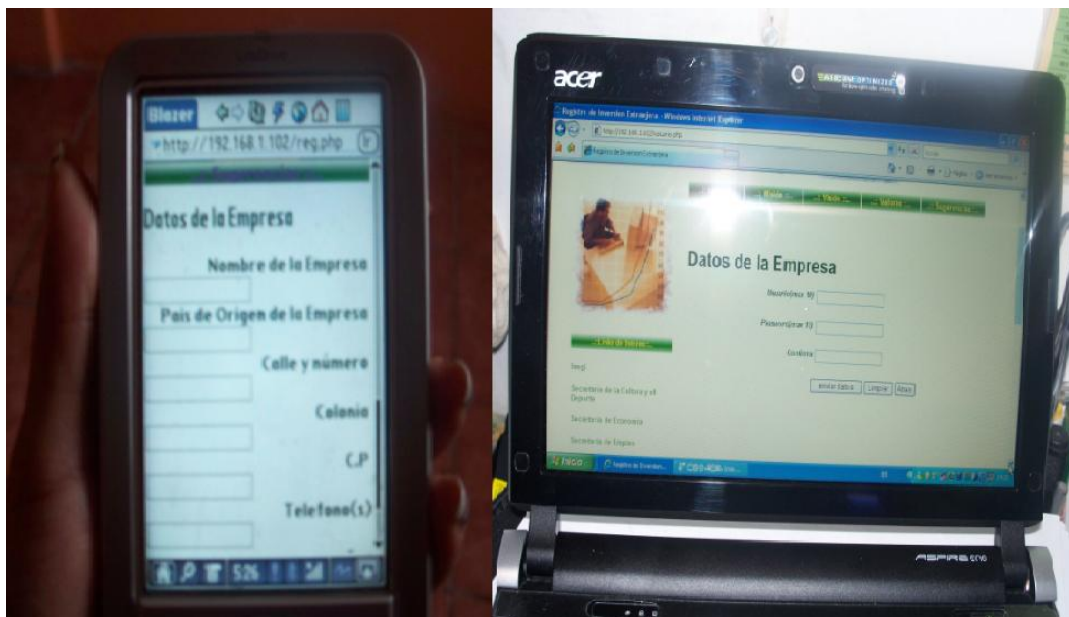


Figura.6.4 Pantalla de registro de un inversionista nuevo.

Cuando el inversionista haya ingresado los datos correspondientes a la información de su empresa, dará clic en el botón “Enviar datos”, donde se mostrará otra pantalla, en la cual se ingresarán los datos del representante su empresa, como se muestra en la Figura. 6.5

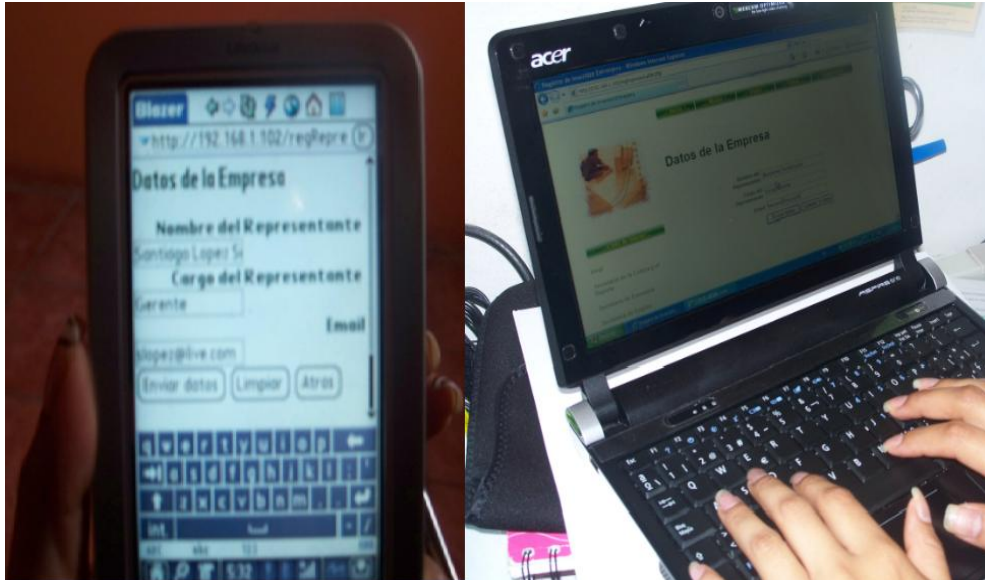


Figura.6.5 Registro de Datos de Representante.

Una vez que el inversionista haya ingresado los datos del representante de su empresa, el inversionista dará clic en el botón “Enviar datos”, donde se deberá ingresar los datos de la inversión, especificando el sector económico al cual pertenece, como se muestra en la Figura 6.6.

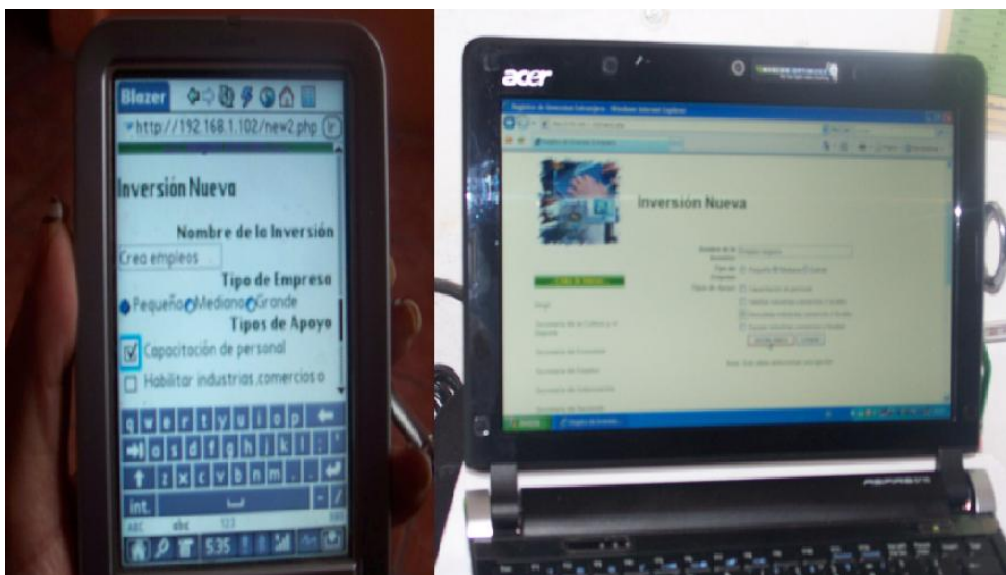


Figura.6.6 Pantalla de Datos de Inversión.

El inversionista dará clic nuevamente en el botón de “Enviar datos”, y procederá a ingresar los datos correspondientes a una Inversión Nueva, como se muestra en la Figura 6.7.

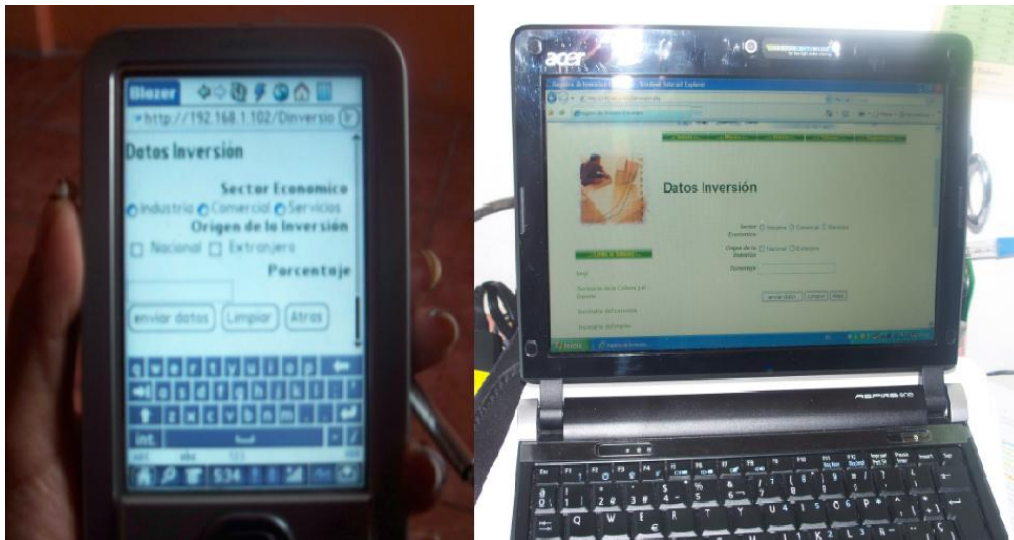


Figura.6.7 Registro de Proyecto.

Registrado los datos del proyecto, se dará clic en el botón de “Enviar datos”, y se mostrará otra pantalla concluyendo el registro del proyecto de inversión a través de un Móvil, como se muestra en la Figura 6.8.

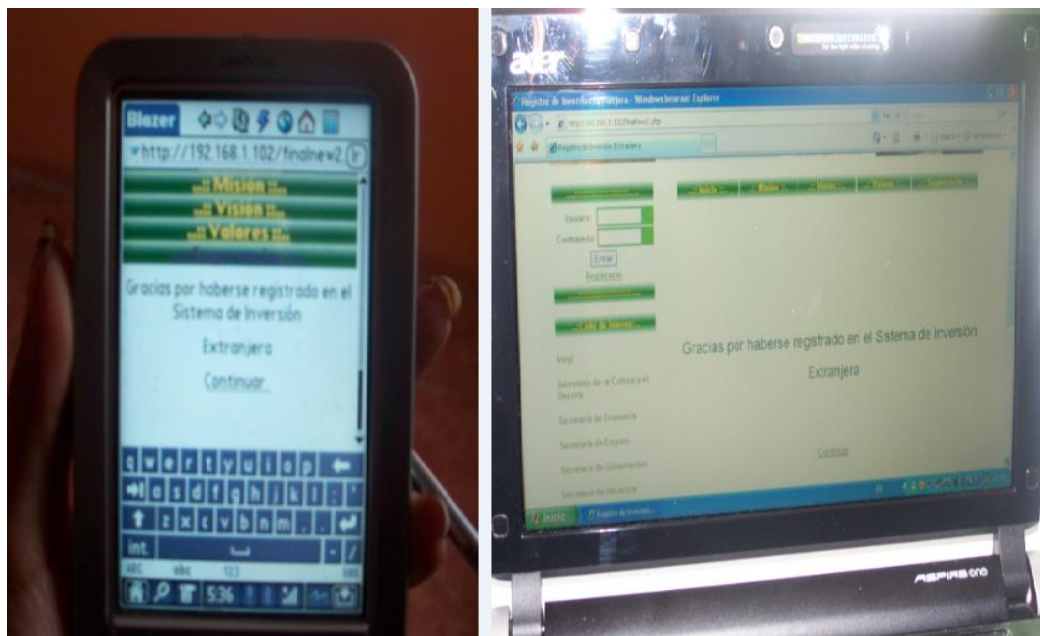


Figura.6.8 Pantalla final del Registro de un proyecto de Inversión.

6.2 RESULTADOS OBTENIDOS DEL REGISTRO DE INVERSIONISTAS TRAVEZ DE UN DISPOSITIVO MÓVIL.

En la actualidad, en el Departamento que se encarga de la Inversión Extranjera, cuenta con computadoras de escritorio, por lo que mostrará como un empleado puede visualizar los datos de un inversionista como se muestra en la Figura. 6.9.



Figura. 6.9 Pantalla principal para un empleado.

Para que un empleado pueda ingresar al sistema y ver los datos de cada inversionista registrado este deberá contar con un usuario y contraseña, una vez ingresados el usuario y contraseña requeridos, se mostrará una pantalla donde se muestra un menú, dando lugar a que el empleado pueda elegir la forma en que podrá visualizar dichos datos, como se muestra en la Figura. 6.10.



Figura.6.10 Menú de visualización de datos.

Se elegirá la primera opción del menú, donde se verán los datos de las empresas registradas, así como los datos de los representantes como se muestra en la Figura 6.11, para ver con más detalle la información que el empleado observa desde su computadora de escritorio ver la Figura 6.12



Figura.6.11 Datos de las Empresas Registradas.

Cabe mencionar, que las empresas que se registraron con la ayuda de una Palm y una computadora portátil, se utilizaron nombres de empresas distintas, se registro el nombre de la empresa Zonet utilizando la Palm, y NWI México S de RL de CV con la computadora portátil como se muestra en la Figura.6.13, esto con la finalidad, de que si el proyecto que se registro fuera aprobado el empleado tendría los datos necesarios para fijar una cita formal donde se expondrían puntos del proyecto de inversión que se registro.

[Imprimir Información](#)
[Regresar a Menú](#)

idempresa	nombreempresa	origenempresa	calle_numero	Colonia	cpempresa	telefono	fax
1	Calsa	Nacional	priv a sur	Mayorazgo	72000	2190561	1234
3	Zapateria Carmen	Nacional	Calle 2 sur 9928	Loma Linda	72000	2285696	12345
36	NWI Mexico S de RL d	Mexico	Colima 410 Int 401	Roma	7800	5553478	3456
16	cosco	Nacional	priv b ote	zapata	72000	2568974	45678
17	burguer	nacional	calle s/n	agua zaul	7200	45689	12569
14	computo	mexicana	101 ote 9930	zapata	7200	2568936	12345
18	Pegamentos S.A	Nacional	2 sur 9928	Jardin	72480	2456958	24569
19	medico's	extranjera	bonita num. 9	chula chula	2	2224531	
35	Zonet	Mexico	Av.Maestros 197	Mexico 68	7800	5551308	13145

Figura. 6.13 Datos de empresas que observa un empleado

Teniendo en cuenta que las empresas que registraron no existían en la base de datos, el proyecto de inversión que se registro, se almaceno como una “inversión nueva” Figura. 6.14.

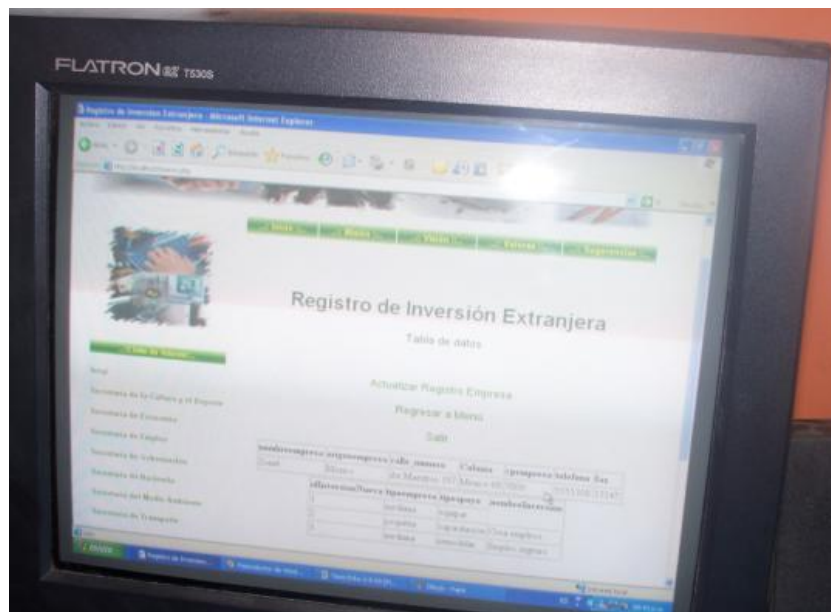


Figura. 6.14 Datos de Inversión Nueva.

Todo esto es con la finalidad a que si el inversionista requiere que se de seguimiento al proyecto que registro al cual se le denomina “Reinversión” o bien registrar un proyecto nuevo, el cual se denomina “inversión nueva”, pueda registrarlos sin la necesidad que registrar los datos de la empresa.

CONCLUSIONES

Con las pruebas realizadas y resultados obtenidos se puede decir que los objetivos se han cubierto, es decir:

- ✓ Se desarrolló un sistema de Registro de Inversión Extranjera donde se facilita al inversionista registrar un proyecto de inversión utilizando tecnología de punta.
- ✓ Se desarrolló una Base de Datos que contiene la información de los datos de la empresa, datos de los representantes de la empresa, datos de acceso del sistema, inversión nueva, reinversión, datos de empleados etc.
- ✓ Se puede acceder a los portales de las secretarías del estado por medio del apartado de ligas de interés.
- ✓ Se provee de información detallada que requiere un empleado cuando esta le sea solicitado, además de proveer al inversionista una manera sencilla de enviar su proyecto.
- ✓ Se configuró un Access Point como punto de acceso al sistema de Registro de Inversión Extranjera, para los dispositivos móviles.
- ✓ Se puede concluir que los objetivos se cumplieron, y se espera que el medio de comunicación para el registro de inversión sea el sistema, y que los inversionistas tengan una herramienta que vaya de acuerdo con la tecnología de punta que está ahora en día.

Perspectivas

Para que el sistema de “Registro de Inversión Extranjera” pueda funcionar correctamente se recomienda contar con:

- ✓ Software: Internet Explorer 5.5 o superior, PHP5, MySQL, Apache
- ✓ Hardware: Pentium u otro, 32 MB de RAM














En un futuro se deja a consideración los siguientes puntos para mejora y aprovechamiento del sistema de “Registro de Inversión Extranjera”:

- ✓ Mejora del formato del sistema de “Registro de Inversión”.
- ✓ Desarrollo y mejora del sistema de “Registro de Inversión Extranjera” en un ambiente Windows.
- ✓ Desarrollo y mejora del sistema de “Registro de Inversión Extranjera” como aplicación propia de un dispositivo móvil.
- ✓ Con la nueva tecnología que hay hoy en día, se espera que el dispositivo móvil se convierta en una herramienta de trabajo primordial.

ANEXOS

SUGERENCIAS GRÁFICAS PARA EL SISTEMA DE “REGISTRO DE INVERSIÓN EXTRANJERA”

1. Colores Oficiales:

Rojo		#AD4032	173 / 64 / 50	 Verde #006233 0 / 98 / 51
Beige		#CFCF89	207 / 207 / 137	
Verde		#94B900	148 / 185 / 0	
Naranja		#EA7500	234 / 117 / 0	
Azul		#2F8EBD	47 / 142 / 189	
Rojo		#D69F98	214 / 159 / 152	 Rojo #DC241F 220 / 36 / 31
Beige		#EDED3	237 / 237 / 211	
Verde		#C9DC7F	201 / 220 / 127	
Naranja		#F4BA7F	244 / 186 / 127	
Azul		#97C6DE	151 / 198 / 222	
				 Gris #767A7D 118 / 122 / 125

1. Colores Hexadecimales y RGB, Portal Único del Gobierno del Estado de Puebla

2. Sugerencias para el desarrollo de formas de Registro:

- Alinear todos los Campos de Texto, y Listas/Menú
- Utilizar Arial como fuente tipográfica
- Utilizar colores oficiales, se pueden utilizar más tenues si tienden hacia el blanco.

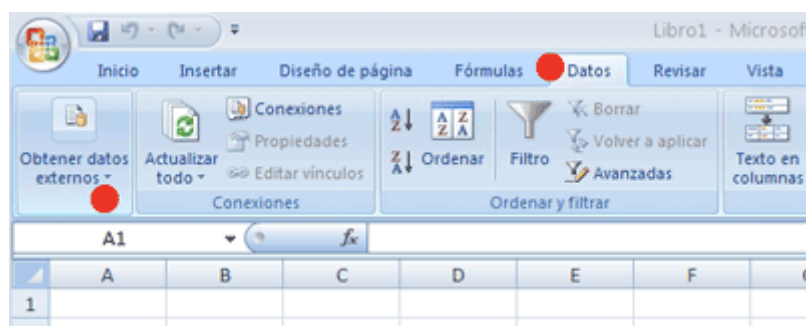
Migrar una Base de Datos MySql a Excel

Herramientas a Utilizar:

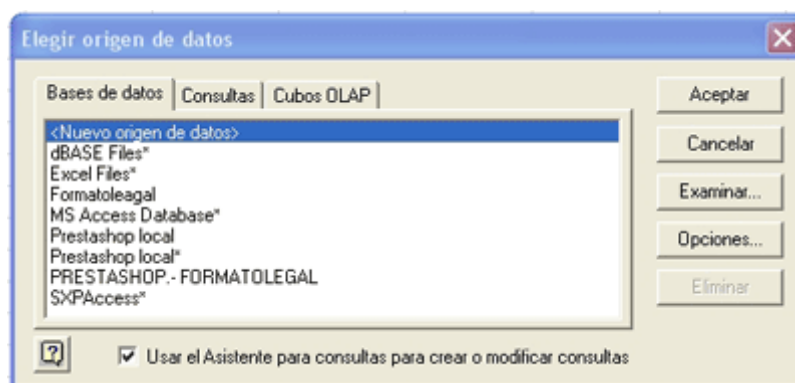
- ✓ Conector ODBC
- ✓ Hoja de Calculo
- ✓ MySql y phpMyAdmin

Lo primero es instalar un driver ODBC al ordenador que será el que conecte a la base de datos y la descargue a Excel para poder tratar los datos y compatibilizarlos.

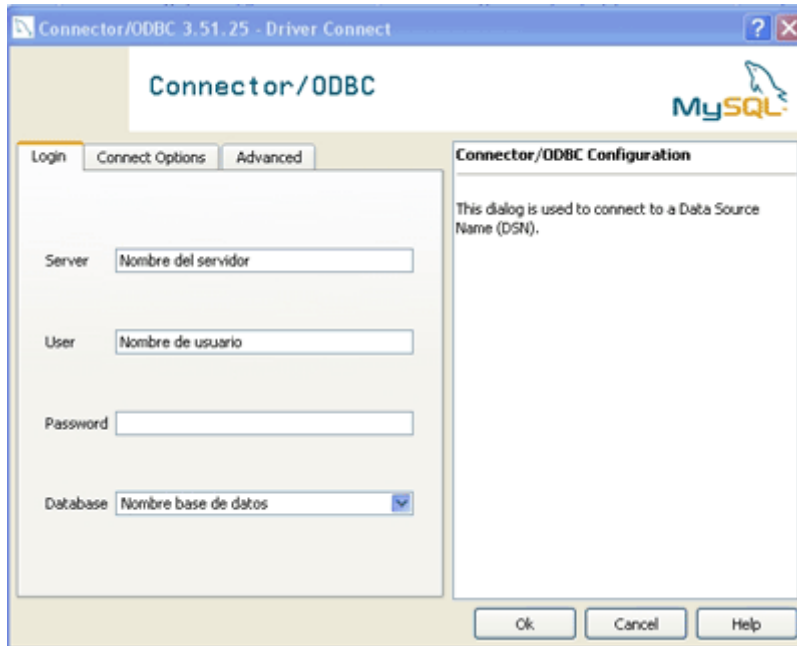
Abrir Excel y seguimos los siguientes pasos:



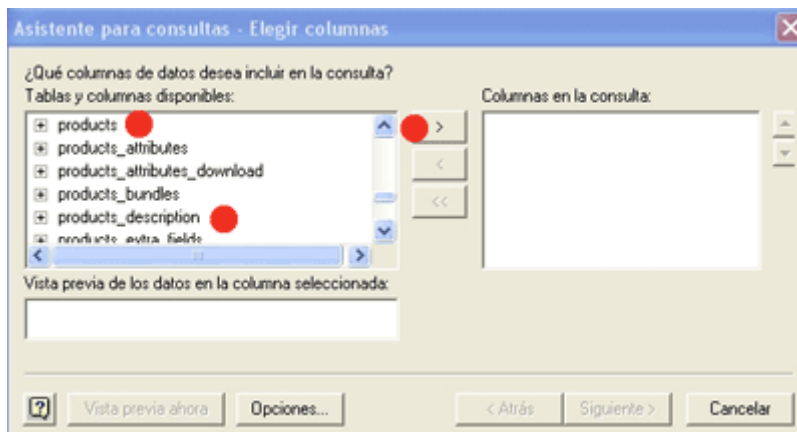
Datos → Obtener datos Externos → De otras fuentes → Desde Microsoft Query (importar datos desde para un formato no instalado utilizando el asistente para consultas de Microsoft y ODBC) nos aparece una pantalla para elegir el origen de los datos y establecer la conexión con el conector ODBC (previamente configurado).



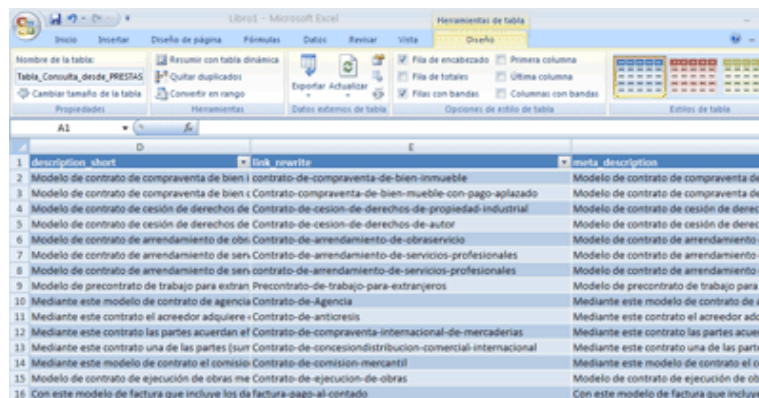
Hacemos doble clic en la conexión que conecta a la base de datos y se muestra otra pantalla donde se pide los datos de acceso.



Seguidamente y con la conexión hecha vemos los nombres de las tablas y sus campos y nos pide que le digamos lo que queremos descargar.



Ahora hacemos lo mismo y descargamos las tablas, si lo tenemos todavía en local hacemos la conexión con el driver ODBC con el nombre del servidor localhost, nombre de usuario y contraseña. Recordar que será la dirección URL (http://) y por tanto no admite ni espacios en blanco ni “ñ” ni otros signos de puntuación; dicho esto cuando tengamos los datos preparados guardamos la hoja de Excel.



BIBLIOGRAFIA

- [1].Análisis y Diseño de Bases de Datos; i.t Hawryszkiewicz. Editorial: Megabyte Noriega Editores.
- [2].Introducción a los Sistemas de Bases de Datos; C.J Date. Editorial: Addison-Wesley Iberoamericana.
- [3].Ingeniería de Software; Ian Sommerville.
- [4].Ingeniería de Software; Richard Fairley. Editorial: Mc Graw Hill.
- [5].Redes Globales de Información tcp/ip Tercera Edición; Douglas E. Comer. Editorial: Hall Hispanoamericana.
- [6].Redes de Computadoras; Andrew S. Tanenbaum. Editorial: Prentice Hall.
- [7].Comunicación de datos, redes de computadores y sistemas abiertos; Fred Halsal. Editorial: Pearson Education.
- [8].MySQL; Paul Dobais. Editorial: New Riders.
- [9].La biblia de MySQL; Ian Ullman. Editorial: Prentice Hall.
- [10]. PHP 4 a través de ejemplos. Abraham Gutiérrez, Ginés Bravo. Editorial: Alfaomega.
- [11]. MySQL Edición Especial, Paul Dubois Editorial: Prentice Hall.
- [12]. La Biblia de JavaScript, Lázaro Issi Camy, Anaya Multimedia.
- [13]. Manual de Referencia JavaScript, Thomas Pwell, Fritz Schneider, Editorial: McGraw Hill.
- [14]. <http://www.desarrolloweb.com/articulos/392.php>
- [15]. <http://www.masadelante.com/faqs/wap>
- [16]. http://mipagina.cantv.net/palmmania/que_es_una_palm.htm
- [17]. <http://www.mitecnologico.com/Main/NivelesAbstraccionBaseDatos>
- [18]. <http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html>