



# **BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA**

---

## **FACULTAD DE CIENCIAS DE LA COMPUTACION**

**PORTAFOLIO DE PRODUCTOS Y  
SERVICIOS DE LA DESIT-BUAP**  
(División de Estudios Superiores de Ingeniería y Tecnología)

**TRABAJO DE TESIS QUE PRESENTA**

**NESTOR AGUSTIN CASTILLO**

**DENTRO DEL PROGRAMA DE DIPLOMADO EN  
COMPUTACION**

**ESPECIALIDAD: BASES DE DATOS**

**PARA OBTENER EL TITULO DE  
LICENCIADO EN CIENCIAS DE LA COMPUTACION**

**ASESOR: PROFESOR JORGE JIMENEZ GONZALEZ**

Puebla, Pue.

2012

*A mis padres porque no fue fácil y porque  
sin ellos hubiera sido imposible gracias  
los amo muchísimo y a mi hermano que  
siempre creyó en mí.*

*A mi esposa e hija que con sus presencias me inspiraron  
a continuar, soportando mis ausencias LAS AMO.*

## INDICE

<b>CAPÍTULO I. INTRODUCCIÓN</b>	<b>4</b>
I.1 Planteamiento del problema	4
I.2 Objetivo general	4
I.2.1 Objetivos específicos	4
I.3 Metodología	4
I.4 Resultados obtenidos	5
<b>CAPÍTULO II. MARCO DE REFERENCIA TEÓRICO</b>	<b>6</b>
II.1 Metodologías de análisis y diseño estructurado	6
II.1.1 Modelado de datos	7
II.1.1.1 Modelo conceptual de la base de datos	7
II.1.1.2 Objetos de datos, atributos y relaciones	8
II.1.1.3 Diagramas entidad-relación	9
II.2.1 Bases de datos relacionales	9
II.1.2 Modelado funcional	10
II.1.2.1 Modelo de flujo de datos	11
II.1.2.2 Modelo de flujo de control	11
II.1.2.3 Diagramas de Flujo de Datos	11
II.1.3 Modelado del comportamiento	13
II.1.3.1 Diagramas de Transición de Estados.	13
II.2. Diseño lógico de una Base de Datos	13
II.2.1 Transformación del esquema conceptual al lógico estándar	14
II.2.1.1 Transformación de entidades.	14
II.2.1.2 Transformación de atributos de entidades	14
II.2.2 Normalización	14
II.3. Lenguajes de Definición y Manipulación de Datos.	16
II.3.1 SQL	17
II.4. Lenguajes de programación para aplicaciones Web Dinámicas.	17
II.4.1 HTML	17
II.4.2 PHP	18
II.5 Servidores Web	18
<b>CAPÍTULO III. ESPECIFICACIÓN DE LOS REQUISITOS DEL SISTEMA</b>	<b>20</b>
III.1 Descripción General del sistema	20
III.2 Objetivos	20
III.3 Alcance	21
III.4 Casos de Uso	21
III.4.1 Actores	22
III.4.2 Documentación de los casos de uso	23

III.4.3 Modelo de casos de uso	26
III.5 Requerimientos Funcionales	27
III.6 Requerimientos No Funcionales	27
<b>CAPÍTULO IV. ANÁLISIS Y DISEÑO DEL SISTEMA</b>	<b>29</b>
IV.1 Análisis del sistema	29
IV.1.1 Modelo conceptual de la base de datos del sistema	29
IV.1.1.1 Entidades, Atributos y Relaciones	30
IV.1.1.2 Diagrama entidad-relación del sistema	34
IV.1.2 Modelo funcional	35
IV.1.3 Diagramas de flujo de datos	36
IV.4 Diseño lógico	40
IV.4.1 Esquema Relacional de la Base de Datos	40
IV.3.2 Normalización	41
IV.5 Diseño de la Interfaz de Usuario	42
<b>CAPÍTULO V. DISEÑO FÍSICO E IMPLEMENTACIÓN</b>	<b>44</b>
V.1. Instalación y configuración de los Servidores Web y de Bases de Datos	44
V.2. Creación de tablas utilizando el lenguaje SQL	47
V.3. Estructura final de la base de datos	49
V.4. Consultas	52
V.5. CSS	55
V.6. Pantallas del sistema	59
V.7. Reportes	62
<b>CONCLUSIONES</b>	<b>64</b>
<b>BIBLIOGRAFÍA</b>	<b>65</b>
<b>LINKOGRAFÍA</b>	<b>66</b>

# CAPITULO I. INTRODUCCIÓN

## I.1 Planteamiento del problema.

En la División de Estudios Superiores de Ingeniería y Tecnología (DESIT) de la Benemérita Universidad Autónoma de Puebla se está elaborando un Plan estratégico de Integración Social (PEIS), entre las acciones a desarrollar se tiene el establecimiento de un Catálogo o Portafolio de Productos y Servicios que integre todas las capacidades con las que se cuenta para resolver problemáticas productivas o tecnológicas con el objetivo de propiciar su comercialización.

## I.2 Objetivo general

En este proyecto se pretende realizar un Prototipo de Sistema de Gestión del Portafolio de Productos y Servicios para el registro y catalogación de los diversos productos y servicios que los miembros de la División de Estudios Superiores de Ingeniería y Tecnología (DESIT-BUAP) pueden ofrecer a la sociedad, con el objeto de potenciar su comercialización.

### I.2.1 Objetivos específicos

- Gestión de los datos referentes a los productos y servicios que se pueden ofrecer en la DESIT.
- Gestión de los datos de los universitarios, colaboradores externos y profesores que proveerán los productos y servicios del Portafolio.
- Gestión de los datos de contacto de los interesados en los productos y servicios del Portafolio.
- Presentación del listado de productos y servicios del Portafolio de acuerdo a diferentes criterios de búsqueda.

## I.3 Metodología.

La Metodología que se seguirá será el Análisis y Diseño Estructurado y se implementará el prototipo en un DBMS Relacional.

1. Planteamiento del Problema.
2. Especificación de Requisitos.
3. Modelo Conceptual de la Base de Datos.
4. Modelo Funcional del Sistema.

5. Modelo de Comportamiento del Sistema.
6. Diseño de la Base de Datos.
7. Diseño de la Arquitectura del Sistema.
8. Diseño de la Interfaz.
9. Diseño Procedimental
10. Codificación.

#### **I.4 Resultados obtenidos.**

Dentro de los resultados obtenidos se logró consolidar cada uno de los requerimientos del sistema antes mencionados en los objetivos específicos mediante una aplicación web que puede ser visualizada desde cualquier navegador de internet, ya que el sistema logra administrar los datos de los proyectos que se realizan en la DESIT pudiendo el administrador del sistema así como también el usuario con el rol de proveedor ingresar, eliminar o actualizar los datos de los productos y/o servicios que se desean comercializar, así mismo muestra un listado con los proyectos ya almacenados en el sistema y que pueden ser consultados por los diferentes tipos de criterios que usa el sistema, también se logró administración de los datos personales y de contacto, de los diferentes tipos de usuario administrador, proveedor y cliente, por lo que se logró el objetivo que pretendía el portafolio de productos y servicios.

# CAPÍTULO II.

## MARCO DE REFERENCIA TEÓRICO

### II.1 Metodologías de análisis y diseño estructurado

El análisis estructurado<sup>1</sup>, como todos los demás métodos de análisis de requisitos, es una actividad de construcción de modelos. Mediante una notación que es única de este método, se crean modelos que reflejan el flujo y el contenido de la información, se parte el sistema funcionalmente y según los distintos comportamientos, se establece la esencia de lo que se debe construir.

Algunos de los problemas comunes que los desarrolladores encuentran en la construcción de software de cierta complejidad son los siguientes:

- El dominio de aplicación no es conocido.
- La comunicación con el usuario.
- La comunicación con el grupo de desarrollo.
- La carencia de buena documentación.

Por esta razón, es necesario seguir una serie de pasos, los cuales están constituidos por diferentes etapas:

- **Especificación de requerimientos:** Se realizan entrevistas con el usuario identificando los requerimientos y necesidades del usuario.
- **Análisis:** Modela los requerimientos del usuario.
- **Diseño:** Se modela la solución del sistema, teniendo en cuenta el ambiente de implementación a utilizar, por ejemplo, si el sistema es centralizado o distribuido, la base de datos a utilizar, lenguaje de programación, performance deseada, etc.
- **Implementación:** Dado el lenguaje de programación elegido se implementa el sistema.
- **Testeo:** En esta etapa se verifica y valida el sistema teniendo en cuenta algunos criterios determinados por el grupo correspondiente.

---

<sup>1</sup>(2011, 02). Analisis Estructurado. BuenasTareas.com. Recuperado 02, 2011, de <http://www.buenastareas.com/ensayos/Analisis-Estructurado/1625190.html>

## II.1.1 Modelado de datos

Por lo general, un modelo de datos permite describir las estructuras de una base de datos (el tipo de datos y la forma en que se relacionan), las restricciones de integridad (las condiciones que los datos deben cumplir para reflejar correctamente la realidad deseada), las operaciones de manipulación de los datos (agregado, borrado, modificación y recuperación de los datos de la base). El modelo de datos en un enfoque más amplio, permite describir los elementos que intervienen en una realidad o en un problema dado y la forma en que se relacionan dichos elementos entre sí.

Por lo general, un modelo de datos presenta dos sub-lenguajes:

- Lenguaje de Definición de Datos o DDL<sup>2</sup> (Data Definition Language), cuya función es describir, de una forma abstracta, las estructuras de datos y las restricciones de integridad.
- Lenguaje de Manipulación de Datos o DML<sup>3</sup> (Data Manipulation Language), que se orienta a describir las operaciones de manipulación de los datos, la parte del DML enfocada a la recuperación de datos, se la suele conocer como Lenguaje de Consulta o QL (Query Language).

La clasificación de los modelos de datos se realiza de acuerdo al nivel de abstracción. Los modelos de datos conceptuales son aquellos que describen las estructuras de datos y restricciones de integridad. Se utilizan durante la etapa de análisis de un problema dado y están orientados a representar los elementos que intervienen y sus relaciones. Los modelos de datos lógicos se centran en las operaciones y se implementan en algún manejador de base de datos, se puede mencionar a los modelos de datos físicos, que son estructuras de datos a bajo nivel implementadas dentro del propio manejador.

### II.1.1.1 Modelo conceptual de la base de datos

Es una descripción de alto nivel de la estructura de la base de datos, independientemente del SGBD que se vaya a utilizar para manipularla. Un modelo conceptual es un lenguaje que se utiliza para describir esquemas conceptuales. El objetivo del diseño conceptual es describir el contenido de información de la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar esta información. Existen distintos tipos de modelos conceptuales:

#### Basados en registros:

---

<sup>2</sup> Pablo, Reyes , (1997) Introducción al SQL de InterBase: DDL y DML, Grupo: DanySoft, Recuperado Abril de 2002 de [blearning.itmina.edu.mx/dep/.../tablas\\_integridad.pdf](http://blearning.itmina.edu.mx/dep/.../tablas_integridad.pdf)

<sup>3</sup> Pablo, Reyes , (1997) Introducción al SQL de InterBase: DDL y DML, Grupo: DanySoft, Recuperado Abril de 2002 de [blearning.itmina.edu.mx/dep/.../tablas\\_integridad.pdf](http://blearning.itmina.edu.mx/dep/.../tablas_integridad.pdf)

**Jerárquico:** datos en registros, relacionados con apuntadores y organizados como colecciones de árboles

**Redes:** datos en registros relacionados por apuntadores y organizados en gráficas arbitrarias.

**Relacional:** datos en tablas relacionados por el contenido de ciertas columnas.

### **Basados en objetos:**

**Orientado a objetos:** datos como instancias de objetos (incluyendo sus métodos).

**Entidad-relación:** datos organizados en conjuntos interrelacionados de objetos (entidades) con atributos asociados.

### **II.1.1.2 Objetos de datos, atributos y relaciones <sup>4</sup>**

#### **Entidades.**

Cualquier tipo de objeto o concepto sobre el que se recoge información: cosa, persona, concepto abstracto o suceso. Por ejemplo: coches, casas, empleados, clientes, empresas, oficios, diseños de productos, conciertos, excursiones, etc. Las entidades se representan gráficamente mediante rectángulos y su nombre aparece en el interior. Un nombre de entidad sólo puede aparecer una vez en el esquema conceptual.

Hay dos tipos de entidades: fuertes y débiles. Una entidad débil es una entidad cuya existencia depende de la existencia de otra entidad.

#### **Relación.**

Es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función. Las relaciones se representan gráficamente mediante rombos y su nombre aparece en el interior. Una relación recursiva es una relación donde la misma entidad participa más de una vez en la relación con distintos papeles. El nombre de estos papeles es importante para determinar la función de cada participación. La Cardinalidad con la que una entidad participa en una relación específica el número mínimo y el número máximo de correspondencias en las que puede tomar parte cada ocurrencia de dicha entidad

---

<sup>4</sup> Fundamentos de Bases de Datos, Abraham, Silberschatz, Sudarshan, Mc Graw Hill, 2006, Capítulo 6

## Atributo

Es una característica de interés o un hecho sobre una entidad o sobre una relación. Los atributos representan las propiedades básicas de las entidades y de las relaciones. Toda la información extensiva es portada por los atributos. Gráficamente, se representan mediante bolitas que cuelgan de las entidades o relaciones a las que pertenecen. Cada atributo tiene un conjunto de valores asociados denominado dominio. El dominio define todos los valores posibles que puede tomar un atributo. Puede haber varios atributos definidos sobre un mismo dominio.

En el diseño se pueden considerar tres categorías de atributos

- **Simple o compuestos:** ya sea que el atributo sea un todo o bien este compuesto
- **Con valores simples o multivaluados:** en base a si consisten de un solo valor o un conjunto de valores.
- **Derivados:** que se pueden calcular en base a otros atributos

### II.1.1.3 Diagramas entidad-relación <sup>5</sup>

Se denomina así debido a que precisamente permite representar relaciones entre entidades (objetivo del modelado de datos). El modelo debe estar compuesto por:

- Entidades
- Atributos
- Relaciones
- Cardinalidad
- Llaves

### II.2.1 Bases de datos relacionales

Es un conjunto de información relacionada entre sí, referente a un tema o propósito en particular. Las bases de datos relacionales están constituidas por una o más tablas que contienen la información ordenada de una forma organizada. Cumplen las siguientes leyes básicas:

- Generalmente, contendrán muchas tablas.
- Una tabla sólo contiene un número fijo de campos.
- El nombre de los campos de una tabla es distinto.

---

<sup>5</sup> Fundamentos de Bases de Datos, Abraham, Silberschatz, Sudarshan, Mc Graw Hill, 2006, Capitulo 6

- Cada registro de la tabla es único.
- El orden de los registros y de los campos no está determinados.
- Para cada campo existe un conjunto de valores posible.

## II.1.2 Modelado funcional <sup>6</sup>

Es un bosquejo que representa un conjunto real con cierto grado de precisión y en la forma más completa posible, pero sin pretender aportar una réplica de lo que existe en la realidad.

Los modelos son muy útiles para describir, explicar o comprender mejor la realidad, cuando es imposible trabajar directamente en la realidad en sí. Un buen modelo funcional toma en cuenta todos los factores esenciales e ignora por completo los detalles superfinos.

Por eso, es de suma importancia disponer de un propósito muy claro y preciso antes de comenzar a elaborar el modelo. Los requisitos primordiales para construir cualquier modelo son:

- Un propósito claramente definido.
- Identificar las consideraciones esenciales.
- Desechar consideraciones superfluas.
- El modelo debe representar la realidad en forma simplificada.

El modelo funcional especifica lo que sucede y cuándo sucede, y el modelo de objetos sobre qué entidades sucede, ya que define el significado de:

- Las operaciones y restricciones del Modelo de Objetos.
- Las acciones del Modelo Dinámico.
- Sólo expresa qué valores de salida se derivan de qué valores de entrada.
- Consta de múltiples diagramas de flujos de datos que muestran el flujo de valores desde las entradas externas, pasando por las operaciones y almacenes internos, hasta las salidas externas.

Las ventajas de un modelo gráfico son:

- Todos los rasgos esenciales están expuestos. La estructura y el contenido del modelo son percibidos con claridad y precisión. El propósito del modelo se percibe netamente.
- Es una representación física que no se altera fácilmente. El modelo no requiere sea memorizado y puede replicarse fácilmente cuando es

---

<sup>6</sup> Juan J. Alcaraz Espín, Tecnologías de red de transporte de operadora  
[http://ait.upct.es/~jjalcaraz/teaching/tema\\_1\\_modelo\\_funcional.pdf](http://ait.upct.es/~jjalcaraz/teaching/tema_1_modelo_funcional.pdf)

necesario utilizarlo. Es una manera muy fácil y rápida de transferir a otras personas -a través del tiempo y del espacio - ideas y conceptos.

- La representación elaborada no puede ser modificada; el concepto representado tiene que ser constante. Si las condiciones cambian quizás será preciso construir un nuevo modelo, pero eso no invalida el modelo original.

### **II.1.2.1 Modelo de flujo de datos**

Es un modelo lógico-gráfico, que ayuda a representar el funcionamiento de un sistema, este permite incorporar opciones para el depurado de algoritmos, facilitando la localización de errores de ejecución y lógicos más habituales.

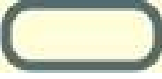





### **II.1.2.2 Modelo de flujo de control**

Este modelo muestra el orden en el que se ejecutan las instrucciones de un programa, siendo las propias instrucciones las que determinan o controla dicho flujo. En un programa, a menos que el flujo de control se vea modificado por una instrucción de control, las instrucciones siempre se ejecutan secuencialmente, una detrás de otra, en orden de aparición, de izquierda a derecha y de arriba abajo, que es el flujo natural de un programa.

### **II.1.2.3 Diagramas de Flujo de Datos**

Diagrama de Flujo consiste en representar gráficamente hechos, situaciones, movimientos o relaciones de todo tipo, por medio de símbolos. Son una de las cuatro herramientas del análisis estructurado. Es una herramienta gráfica que se emplea para describir y analizar el movimiento de los datos a través de un sistema, ya sea este manual o automatizado, incluyendo procesos, lugares para almacenar datos y retrasos en el sistema. Los DFD, como se les conoce popularmente son la herramienta más importante y la base sobre la cual se desarrollan otros componentes. La transformación de datos de entrada en salida por medio de procesos puede describirse en forma lógica e independiente de los componentes físicos asociados con el sistema. Un diagrama de flujo es una representación gráfica de un algoritmo o de una parte del mismo. Los diagramas de flujo ayudan en la comprensión de la operación de las estructuras de control. La ventaja de utilizar un algoritmo es que se lo puede construir independiente mente de un lenguaje de programación, pues al momento de llevarlo a código se lo puede hacer en cualquier lenguaje. Dichos diagramas se construyen utilizando ciertos símbolos de uso especial como son rectángulos, diamantes, óvalos, y pequeños círculos, estos símbolos están conectados entre sí por flechas, conocidas como líneas de flujo.

En la figura 2.1 se detallarán los símbolos que se utilizan en los diagramas de flujo.

Nombre	Símbolo	Función
<b>Terminal</b>		Representa el inicio y fin de un programa. También puede representar una parada o interrupción programada que sea necesaria realizar en un programa.
<b>Entrada / salida</b>		Cualquier tipo de introducción de datos en la memoria desde los periféricos o registro de información procesada en un periférico.
<b>Proceso</b>		Cualquier tipo de operación que pueda originar cambio de valor, formato o posición de la información almacenada en memoria, operaciones aritméticas, de transformaciones, etc.
<b>Decisión</b>		Indica operaciones lógicas o de comparación entre datos (normalmente dos) y en función del resultado de la misma determina (normalmente si y no) cuál de los distintos caminos alternativos del programa se debe seguir.
<b>Conector Misma Página</b>		Sirve para enlazar dos partes cualesquiera de un diagrama a través de un conector en la salida y otro conector en la entrada. Se refiere a la conexión en la misma página del diagrama
<b>Indicador de dirección o línea de flujo</b>		Indica el sentido de la ejecución de las operaciones


<b>Salida</b>		Se utiliza en ocasiones en lugar del símbolo de salida. El dibujo representa un pedazo de hoja. Es usado para mostrar datos.
---------------	---	--

Figura 2.1. Describe la acción de cada uno de los símbolos que se utilizan en los diagramas de flujo.

### II.1.3 Modelado del comportamiento

El software puede responder a acontecimientos del mundo exterior que hacen que el sistema cambie de un estado a otro, el estímulo-respuesta es la base del modelo de comportamiento. Un estado es un modo de comportamiento tal como calculando, imprimiendo, esperando, haciendo cola, etc. que cambia solo cuando ocurre un acontecimiento: un movimiento de ratón, un sistema externo, etc. El modelo de comportamiento es una representación de los estados del software y los acontecimientos que causan que cambie de estado.

#### II.1.3.1 Diagramas de Transición de Estados.

El diagrama de transición de estados DTE representa el comportamiento de un sistema que muestra los estados y los sucesos que hacen que el sistema cambie de estado. Un estado es un modo observable de comportamiento, por ejemplo monitoreando, comprobando, calculando, etc. Cada estado representa un modo de comportamiento y el DTE indica cómo se mueve el sistema de un estado a otro.

## II.2. Diseño lógico de una Base de Datos <sup>7</sup>

En el diseño lógico, llamado también diseño conceptual o esquema, deben describirse todos los elementos que forman la Base de Datos. Más aún, el diseño lógico debe condicionar al diseño físico. Algunos sistemas comerciales llaman diseño lógico a una definición global que debe hacerse y diseño físico a la forma en que la Base de Datos es almacenada dentro de la computadora. El diseño lógico de sistemas se refiere lo que hará el nuevo sistema, mientras que el diseño físico de sistemas es la forma en la que se lograrán las tareas del sistema, lo que incluye la manera de combinar sus componentes y las funciones que realizará cada uno de estos; en otras palabras, es la expresión conceptual de lo que hará el sistema para resolver los problemas identificados en el análisis previo. El diseño lógico, incluye planear el propósito de cada elemento del sistema, sin hacer consideraciones de hardware y software.

---

<sup>7</sup> Johnson, James Lee.

Data base: Models, languages, design. Español, Bases de datos : modelos, lenguajes, diseño / James L. Johnson ; traducción Eduardo Ramírez Grycuk, Jorge Humberto Romo Muñoz Oxford University Press, 2000.

## II.2.1 Transformación del esquema conceptual al lógico estándar

En el diseño lógico se deben coordinar exigencias casi siempre encontradas, como son eliminar redundancias, conseguir la máxima simplicidad para así obtener una estructura lógica adecuada que establezca el equilibrio entre las exigencias de los usuarios y la eficiencia.

Las tres reglas básicas para convertir un esquema en el modelo E/R al relacional son las siguientes:

- 1) Todo tipo de entidad se convierte en una relación.
- 2) Todo tipo de interrelación N: M se transforma en una relación.
- 3) Para todo tipo de interrelación 1: N

### II.2.1 .1 Transformación de entidades.

"Cada tipo de entidad se convierte en una relación". Esto es, el modelo lógico estándar posee el objeto RELACION o TABLA mediante el cual representamos las entidades. La tabla se llamará igual que el tipo de entidad de donde proviene

### II.2.1.2 Transformación de atributos de entidades

Cada atributo de una entidad se transforma en una columna de la relación a la que ha dado lugar la entidad

Atributos identificadores.

Atributos identificadores alternativos.

Atributos no identificadores.

1) Interrelaciones N:M. se transforma en una relación que tendrá como clave primaria la concatenación de los AIP de los tipos de entidad que asocia.

2) Interrelaciones 1:N. Existen dos soluciones para la transformación de una interrelación 1:N:

3) Interrelaciones 1: 1 es un caso particular de una N:M o, también, de una 1:N, por lo que no hay regla fija para la transformación de este tipo de interrelación al modelo relacional estándar.

## II.2.2 Normalización

La normalización es el proceso de organizar los datos de una base de datos. Se incluye la creación de tablas y el establecimiento de relaciones entre ellas según reglas diseñadas tanto para proteger los datos como para hacer que la base de datos sea más flexible al eliminar la redundancia y las dependencias incoherentes.

Los datos redundantes desperdician el espacio de disco y crean problemas de mantenimiento. Si hay que cambiar datos que existen en más de un lugar, se deben cambiar de la misma forma exactamente en todas sus ubicaciones. Como ejemplo aplicado al proyecto en turno un cambio en la dirección de un cliente es mucho más fácil de implementar si los datos sólo se almacenan en la tabla CONTACTOS y no en algún otro lugar de la base de datos. Hay algunas reglas en la normalización de una base de datos. Cada regla se denomina una "forma normal". Si se cumple la primera regla, se dice que la base de datos está en la "primera forma normal". Si se cumplen las tres primeras reglas, la base de datos se considera que está en la "tercera forma normal". Aunque son posibles otros niveles de normalización, la tercera forma normal se considera el máximo nivel necesario para la mayor parte de las aplicaciones. Al igual que con otras muchas reglas y especificaciones formales, en los escenarios reales no siempre se cumplen los estándares de forma perfecta.

### **Primera forma normal**

- Eliminar los grupos repetidos de las tablas individuales.
- Crear una tabla independiente para cada conjunto de datos relacionados.
- Identificar cada conjunto de datos relacionados con una clave principal.

No usar varios campos en una sola tabla para almacenar datos similares.

### **Segunda forma normal**

- Crear tablas independientes para los conjuntos de valores que se apliquen a varios registros.
- Relacionar estas tablas con una clave externa.

Los registros no deben depender de nada que no sea una clave principal de una tabla, una clave compuesta si es necesario.

### **Tercera forma normal**

- Eliminar los campos que no dependan de la clave.

Los valores de un registro que no sean parte de la clave de ese registro no pertenecen a la tabla. Existe una excepción cumplir la tercera forma normal, aunque en teoría es deseable, no siempre es práctico. En teoría, la normalización merece el trabajo que supone. Sin embargo, muchas tablas pequeñas pueden degradar el rendimiento o superar la capacidad de memoria o de archivos abiertos.

Puede ser más factible aplicar la tercera forma normal sólo a los datos que cambian con frecuencia. Si quedan algunos campos dependientes, hay que diseñar

la aplicación para que pida al usuario que compruebe todos los campos relacionados cuando cambie alguno.

### **Otras formas de normalización**

La cuarta forma normal, también llamada Forma normal de Boyce Codd (BCNF, Boyce Codd Normal Form), y la quinta forma normal existen, pero rara vez se consideran en un diseño real. Si no se aplican estas reglas, el diseño de la base de datos puede ser menos perfecto, pero no debería afectar a la funcionalidad.

### **II.3. Lenguajes de Definición y Manipulación de Datos.**

Un lenguaje de Manipulación de Datos (Data Manipulation Lenguaje (DML) es un lenguaje proporcionado por el sistema de gestión de bases de datos que permite a los usuarios de la misma llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado. El lenguaje de manipulación de datos más popular hoy en día es SQL, usado para recuperar y manipular datos en una base de datos relacional. Otros ejemplos de DML son los usados por bases de datos IMS/DL1, CODASYL u otras.

Se clasifican en dos grandes grupos:

- Lenguajes de consulta procedimentales.
  - Son los lenguajes en el que los usuarios solicitan información de la base de datos, el usuario da las instrucciones al sistema para que realice una secuencia de operaciones en la base de datos para calcular el resultado deseado.
- Lenguajes de consulta no procedimentales
  - El usuario describe la información deseada sin dar un procedimiento específico para obtener dicha información.

El lenguaje de Consulta Estructurado (Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos, de una forma sencilla.

### II.3.1 SQL <sup>8</sup>

Es lenguaje de consulta estructurado o SQL por sus siglas en inglés Structured Query Language, se lo conoce como un lenguaje declarativo de acceso a bases de datos relacionales, permite especificar diversos tipos de operaciones en estas. Una de sus características principales es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar -de una forma sencilla- información de interés de una base de datos, así como también hacer cambios sobre ella. Las aplicaciones en red son cada día más numerosas y versátiles. En muchos casos, el esquema básico de operación es una serie de scripts que rigen el comportamiento de una base de datos.

Debido a la diversidad de lenguajes y de bases de datos existentes, la manera de comunicar entre unos y otras sería realmente complicada a gestionar de no ser por la existencia de estándares que nos permiten el realizar las operaciones básicas de una forma universal. Es de eso de lo que trata el Structured Query Language que no es más que un lenguaje estándar de comunicación con bases de datos. Hablamos por tanto de un lenguaje normalizado que nos permite trabajar con cualquier tipo de lenguaje (ASP o PHP) en combinación con cualquier tipo de base de datos (MS Access, SQL Server, MySQL, etc.).

El hecho de que sea estándar no quiere decir que sea idéntico para cada base de datos. En efecto, determinadas bases de datos implementan funciones específicas que no tienen necesariamente que funcionar en otras.

### II.4. Lenguajes de programación para aplicaciones Web Dinámicas.

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas.

#### II.4.1 HTML <sup>9</sup>

HTML es el lenguaje con el que se definen las páginas web., básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web, es un lenguaje estático para el desarrollo de sitios web HyperText Markup Language, sus características:

- Sencillo que permite describir hipertexto.
- Texto presentado de forma estructurada y agradable.
- No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web.
- Archivos pequeños.

---

<sup>8</sup> Gilfillan, Ian. La biblia de MySQL / Ian Gilfillan. Madrid : Anaya Multimedia, 2003.

<sup>9</sup> [www.aulaclic.es/html/t\\_1\\_1.htm](http://www.aulaclic.es/html/t_1_1.htm)

- Despliegue rápido.
- Lenguaje de fácil aprendizaje.
- Lo admiten todos los exploradores.

## II.4.2 PHP <sup>10</sup>

PHP un lenguaje de programación utilizado para la creación de sitio web. PHP es un acrónimo recursivo PHP Hypertext Pre-processor, es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP, sus características son:

- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

## II.5 Servidores Web

En Internet, un servidor es una computadora remota que provee los datos solicitados por parte de los navegadores de otras computadoras.

En redes locales se entiende como el software que configura una pc como servidor para facilitar el acceso a la red y sus recursos.

Los Servidores almacenan información en forma de páginas web y a través del protocolo HTTP lo entregan a petición de los clientes (navegadores web) en formato HTML.

Un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse a la computadora físicamente hablando en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

---

<sup>10</sup> Welling, Luke, 1972-

Desarrollo Web con PHP y MySQL / Luke Welling, Laura Thomson.  
Madrid : Anaya Multimedia, 2003.

Este uso dual puede llevar a confusión. Por ejemplo, en el caso de un servidor web, este término podría referirse a la máquina que almacena y maneja los sitios web, y en este sentido es utilizada por las compañías que ofrecen hospedaje. Alternativamente, el servidor web podría referirse al software, como el servidor de http de Apache, que funciona en la máquina y maneja la entrega de los componentes de las páginas web como respuesta a peticiones de los navegadores de los clientes.

Los archivos para cada sitio de Internet se almacenan y se ejecutan en el servidor. Hay muchos servidores en Internet y muchos tipos de servidores, pero comparten la función común de proporcionar el acceso a los archivos y servicios.

Un servidor sirve información a los ordenadores que se conecten a él. Cuando los usuarios se conectan a un servidor pueden acceder a programas, archivos y otra información del servidor.

En la web, un servidor web es un ordenador que usa el protocolo http para enviar páginas web al ordenador de un usuario cuando el usuario las solicita.

Los servidores web, servidores de correo y servidores de bases de datos son a lo que tiene acceso la mayoría de la gente al usar Internet.

Algunos servidores manejan solamente correo o solamente archivos, mientras que otros hacen más de un trabajo, ya que un mismo ordenador puede tener diferentes programas de servidor funcionando al mismo tiempo. Los servidores se conectan a la red mediante una interfaz que puede ser una red verdadera o mediante conexión vía línea telefónica o digital.

# **CAPÍTULO III.**

## **ESPECIFICACIÓN DE LOS REQUISITOS DEL SISTEMA**

### **III.1 Descripción General del sistema**

En la División de Estudios Superiores de Ingeniería y Tecnología (DESIT) de la Benemérita Universidad Autónoma de Puebla se está elaborando un Plan estratégico de Integración Social (PEIS), entre las acciones a desarrollar se tiene el establecimiento de un Catálogo o Portafolio de Productos y Servicios que integre todas las capacidades con las que se cuenta para resolver problemáticas productivas o tecnológicas con el objetivo de propiciar su comercialización, se utilizó la metodología de desarrollo de Análisis y Diseño Estructurado y se implementará el prototipo en un DBMS Relacional, ya que es el más conveniente para este caso como se menciona en el capítulo II.

### **III.2 Objetivos**

En este proyecto se pretende realizar un Portafolio de Productos y Servicios para el registro y catalogación de los diversos productos y servicios que los miembros de la División de Estudios Superiores de Ingeniería y Tecnología (DESIT-BUAP) pueden ofrecer a la sociedad, con el fin de potenciar su comercialización.

El sistema debe contar con las siguientes características específicas:

- Proporcionar información acerca de los productos y servicios que los integrantes de la DESIT va realizando u obteniendo a lo largo de sus investigaciones.
- Proporcionar información acerca de los proveedores de los proyectos con el fin de poder vender estos a clientes y potenciar su comercialización.
- Proporcionar información sobre los posibles clientes que pueden adquirir los servicios y productos.

### **III.3 Alcance**

El Sistema abarca las siguientes funciones:

- Gestión de los datos referentes a los productos y servicios que se pueden ofrecer en la DESIT.
- Gestión de los datos de los universitarios, colaboradores externos y profesores que proveerán los productos y servicios del Portafolio.
- Gestión de los datos de contacto de los interesados en los productos y servicios del Portafolio.
- Presentación del listado de productos y servicios del Portafolio de acuerdo a diferentes criterios de búsqueda.

El sistema deberá operar bajo las siguientes restricciones:

- Será una aplicación Web, es decir, se accederá desde una máquina cliente mediante un browser y estará residente en un servidor web remoto.
- La base de datos se implementará en un Manejador de Bases de Datos Relacional.

### III.4 Casos de Uso

En la tabla 3.1 se presenta una lista de los casos de usos a desarrollar con su descripción general:

Num.	Caso de Uso	Descripción
1	Registro de Proveedor.	Ingresa la información del proveedor.
2	Registro del Contacto.	Ingresa la información del contacto.
3	Registro de producto y/o servicio	Ingresa la información del producto y/o servicio
4	Acceso de usuarios.	Controla el acceso de los usuarios al sistema, verificando nombre de usuario y password, una vez autorizado el acceso el proveedor, podrá consultar y/o registrar un producto y/o servicio, en el

		caso del contacto solo podrá consultar los servicios y/o productos del portafolio.
5	Consulta de producto y/o servicio	Presentación de los servicios del portafolio mediante una página web y de acuerdo al criterio solicitado por el usuario.
6	Registro del administrador.	Permite realizar el registro de administrador otorgando un nombre y contraseña, el cual el administrador de la base de datos le otorga permisos.
7	Validar administrador	Verificar que el nombre de usuario y password ingresado por el administrador sea el correcto una vez autorizado el acceso, podrá gestionar los datos de los proveedores, contactos y servicios y/o productos.

Tabla 3.1 Describe los casos de uso que presenta el sistema.

### III.4.1 Actores.

En esta sección se describen con más detalle las acciones que podrá realizar cada uno de los actores y de acuerdo a su rol en sistema.

#### Usuario PROVEEDOR

Podría gestionar la siguiente información de:

- Datos referentes a los productos y/o servicios que él ya ha ingresado al sistema.
- Ingresar nuevos producto y/o servicio.
- Modificar sus datos personales.
- Consultar los productos y/o servicios ya registrados en el sistema

#### Usuario CLIENTE

- Podría gestionar la información referente a sus datos personales.

- Realizar consultas sobre los productos y servicios registrados en el sistema.
- Contactar a los proveedores de los proyectos.

### **Usuario ADMINISTRADOR**

Podrá gestionar la información de:

- Productos y/o Servicios
- Clientes
- Proveedores
- Cuentas de los usuarios (password y nombres de usuario)
- Categorías
- Sectores económicos
- Áreas funcionales
- Reportes del sistema

### **III.4.2 Documentación de los casos de uso**

En esta sección se explica paso a paso cada uno de los casos de uso ya mencionados en el apartado **III.4 Casos de Uso**.

#### **III.4.2.1 Caso de uso registro de proveedor.**

1. El usuario PROVEEDOR ingresa al sistema mediante un explorador web, y dará clic en el botón **REGISTRARSE**.
2. Se mostrará un pantalla con 2 links para este caso el usuario dará clic en link **PROVEEDOR SERVICIO**.
3. Se muestra una ventana solicitando que ingrese los siguientes datos: CURP, No. Plaza, Matricula, Nombre,Unidad Académica, Profesión, Mail, Celular, Tipo (A = si es alumno o P= profesor de la universidad), Usuario, Password.
4. Una vez ya ingresados los datos antes mencionados dará clic en el botón **INSERTAR**, si el almacenamiento de los datos es correcto, se será enviado a una página donde ingresara sus datos para acceder al sistema.

4.1 En caso de que la conexión con el servidor o a internet se perdiera, se mostrara una pantalla indicando que de clic en el botón **INTENTAR NUEVAMENTE** (este botón lo regresara al paso 2 y un texto que indique **INTENTAR MAS TARDE**.

5. Para ingresar al sistema ver descripción caso de uso **Acceso de usuarios**.

#### III.4.2.2 Caso de uso registro de cliente.

1. El usuario **CLIENTE** ingresa al sistema mediante un explorador web, y dará clic en el botón **REGISTRARSE**.
2. Se mostrará un pantalla con 2 links para este caso el usuario dará clic en link **CLIENTE**.
3. Se muestra una ventana solicitando ingrese los siguientes datos: CURP, RFC, Nombre, Dirección, Teléfono, Celular, Mail, Municipio, Estado Usuario, Password.
4. Una vez ya ingresados los datos antes mencionados dará clic en el botón **INSERTAR**, si el almacenamiento de los datos es correcto, se será enviado a una página donde ingresara sus datos para acceder al sistema.
  - 4.1 En caso de que la conexión con el servidor o a internet se perdiera, se mostrara una pantalla indicando que de clic en el botón **INTENTAR NUEVAMENTE** (este botón lo regresara al paso 2 y un texto que indique **INTENTAR MAS TARDE**.

5. Para ingresar al sistema ver descripción caso de uso **Acceso de usuarios**

#### III.4.2.3 Caso de uso acceso de usuarios.

1. El usuario ingresa al sistema mediante un explorador web, y dará clic en el botón **LOGIN**.
2. Se mostrará una pantalla con tres links **PROVEEDOR, CONTACTO, ADMINISTRADOR**, el usuario dará clic en el botón que corresponda a su perfil.
3. Se mostrara una pantalla donde el usuario ingresara su nombre de usuario y Password.
  - 3.1 Si el usuario no está registrado tendrá que dar clic en el boto **REGISTRARSE** para poder ingresar al sistema (para más información ver casos de uso Registro de proveedor, Registro de cliente).
  - 3.2 Si la contraseña o nombre de usuario fueran incorrectos se mandara una notificación indicando que la contraseña y/o nombre de usuario no coinciden y se solicitara sean nuevamente escritos.
  - 3.3 Si el usuario no recordara su Password y/o nombre de usuario dará clic en el link **RECUPERAR DATOS** y se le enviara un e-mail con su contraseña y el nombre de usuario.

4. Una vez verificado el usuario, se mostrara la página donde el usuario podrá realizar la consulta de los servicios y/o productos esto solo para el caso exclusivo del Usuario: cliente, para los usuarios Proveedor podrán además de consultar gestionar sus productos y/o servicios, para los usuarios Administradores, podrán gestionar todos los datos del sistema.

#### III.4.2.4 Caso de uso registro de administrador

1. El usuario administrador ingresa al sistema mediante un explorador web, y dará clic en el botón **REGISTRARSE**.
2. Una vez que el administrador ingresa al sistema (ver caso de uso acceso de usuarios) dará clic en el botón **INSERTAR**
3. Se muestra una ventana solicitando ingrese los siguientes datos: Nombre, Usuario, Password.
4. Una vez ya ingresados los datos antes mencionados dará clic en el botón **INSERTAR**, y será enviado a la página principal del sistema.
  - 4.1 En caso de que la conexión con el servidor o a internet se perdiera, se mostrara una pantalla indicando que de clic en el botón **INTENTAR NUEVAMENTE** (este botón lo regresara al paso 2 y un texto que indique **INTENTAR MAS TARDE**.
5. Si el almacenamiento de los datos es correcto, se mostrara la página para ingresar al sistema usando su usuario y password.
6. Para ingresar al sistema ver descripción caso de uso **Acceso de usuarios**.

#### III.4.2.5 Caso de uso consulta de productos y/o.

1. Una vez accedido al sistema, se mostrará una página con dos botones y dará clic en el botón **CONSULTAR**.
2. Se desplegara una página donde el usuario podrá consultar el producto y/o servicio por: área funcional, sector económico y/o tipo.
  - 2.1 Si la consulta **NO** arrojará datos se mostrara un mensaje indicando que no hay elementos que coincidan con su búsqueda y se solicitara se realice otro intento con nuevos parámetros de búsqueda.
3. Si la consulta fuera exitosa se mostrará una lista con los resultados y de acuerdo a los parámetros introducidos por el usuario.

#### III.4.2.6 Caso de uso ingreso de producto y/o servicio usuario proveedor

1. Una vez que haya accedido al sistema, el usuario (proveedor) el sistema desplegara índice, y el usuario dará clic en el botón **INSERTAR**.

2. Se mostrará una página donde el usuario introducirá los datos del producto o servicio: Nombre proyecto, Descripción, Tipo, Categoría, Área, Sector. Una vez ya ingresados los datos antes mencionados dará clic en el botón **GUARDAR**.
  - 2.1 En caso de que la conexión con el servidor o a internet se perdiera, se mostrará una pantalla indicando que de clic en el botón **INTENTAR NUEVAMENTE** (este botón lo regresara al paso 2) y un texto que indique **INTENTAR MAS TARDE**.
  - 2.2 Si el registro ya existiera se le informara al usuario y se mostrará el producto que ya estaba registrado. (Esto se verificara mediante el nombre del producto).
3. Si el almacenamiento de los datos es correcto, se mostrara una página donde se mostrara el servicio o producto recién ingresado.

### III.4.3 MODELO DE CASOS DE USO

En la Figura 3.1 se muestra de manera gráfica las acciones que puede realizar cada usuario y de acuerdo a su rol en el sistema.

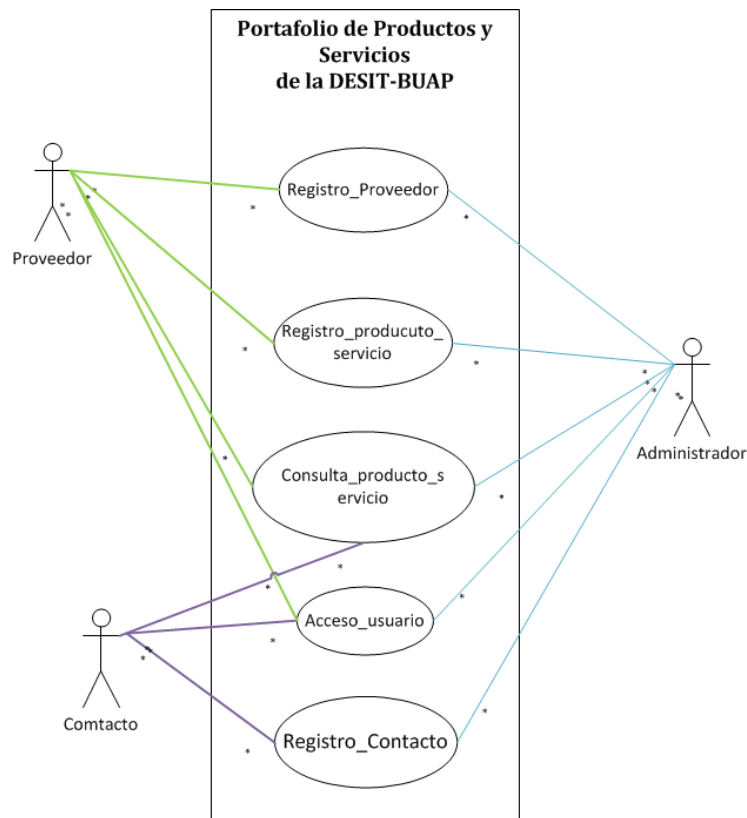


Figura 3.1 Modelo de casos de uso del sistema.

### III.5 Requerimientos Funcionales

Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar

en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer, para este caso en particular el sistema cuenta con ciertos requerimientos sobre los datos personales de los usuarios y que estarán de acuerdo a su rol en el sistema.

#### **Gestión de los proveedores de los Productos y Servicios.**

- El Sistema permitirá un alumno, profesor, colaborador externo, darse de alta registrando sus datos personales básicos (CURP, RFC, No.Plaza, Matrícula, Nombre\_proveedor, Unidad académica, Grado\_estudios, Profesion, Mail y Tipo). El campo tipo indicara el tipo de rol usuario, este puede ser P= Profesor y A= alumno.

#### **Gestión de Productos y Servicios.**

- El Sistema permitirá a un proveedor, dar de alta y baja un servicio o producto, siempre y cuando el proveedor ya este registrado en el sistema.

#### **Gestión de los datos del cliente interesado en los productos y servicios del Portafolio de la DESIT-BUAP.**

- El Sistema permitirá a un contacto, darse de alta registrando sus datos personales básicos (RFC, Password, Nombre Contacto, Dirección, Teléfono fijo, Celular, E-mail, E-mail\_2, Municipio, Estado, Cliente).

#### **Despliegue de la información referente a Productos y Servicios.**

- El sistema mostrará a detalle la información de los servicios y/o productos de acuerdo al criterio solicitado por el cliente, siempre y cuando el cliente ya este registrado en el sistema.

### **III.6 Requerimientos No Funcionales**

Los requerimientos no funcionales no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema, como se explica a continuación:

### **III.6.1 Interfaz con usuario.**

La aplicación será accedida por medio de un navegador Web. El sistema debe ser fácil de operar y no requerir un entrenamiento especializado. Sólo se requiere que el usuario sepa usar un navegador Web.

### **III.6.2 Interfaces con otro software o hardware.**

#### **III.6.2.1 Confiabilidad.**

El sistema deberá realizar las consultas requeridas y la actualización de la base de datos sin errores. En caso de presentarse algún problema en alguna transacción deberá de recuperar el estado inmediato anterior a la transacción.

#### **III.6.2.2 Portabilidad.**

El sistema deberá ser portable a diferentes plataformas como Linux, Unix, Windows X.X Server, Windows 2000, XP, Vista y 7.

#### **III.6.2.3 Interoperatividad.**

El sistema deberá poder ser accedido desde cualquier máquina cliente con un navegador web y sistemas operativos Linux, Unix o Windows.

### **III.6.3 Restricciones de diseño y construcción**

El sistema tendrá una arquitectura cliente-servidor, operará en un servidor Web Apache 2.x, con un manejador de bases de datos MySQL 5.x y será desarrollado en HTML y PHP 4.x o 5.x.

# CAPITULO IV. ANÁLISIS Y DISEÑO DEL SISTEMA

## IV.1 Análisis del sistema

En este proyecto se pretende realizar un Portafolio de Productos y Servicios para el registro y catalogación de los diversos productos y servicios que los miembros de la División de Estudios Superiores de Ingeniería y Tecnología (DESIT-BUAP) pueden ofrecer a la sociedad, con el fin de potenciar su comercialización.

### IV.1.1 Modelo conceptual de la base de datos del sistema

El objetivo principal es construir un modelo conceptual (ver Figura 4.1) de los datos de acuerdo con los requisitos previamente analizados y que estén enteramente independientes de los detalles de la implementación y para que sirva de base en las demás etapas del diseño.

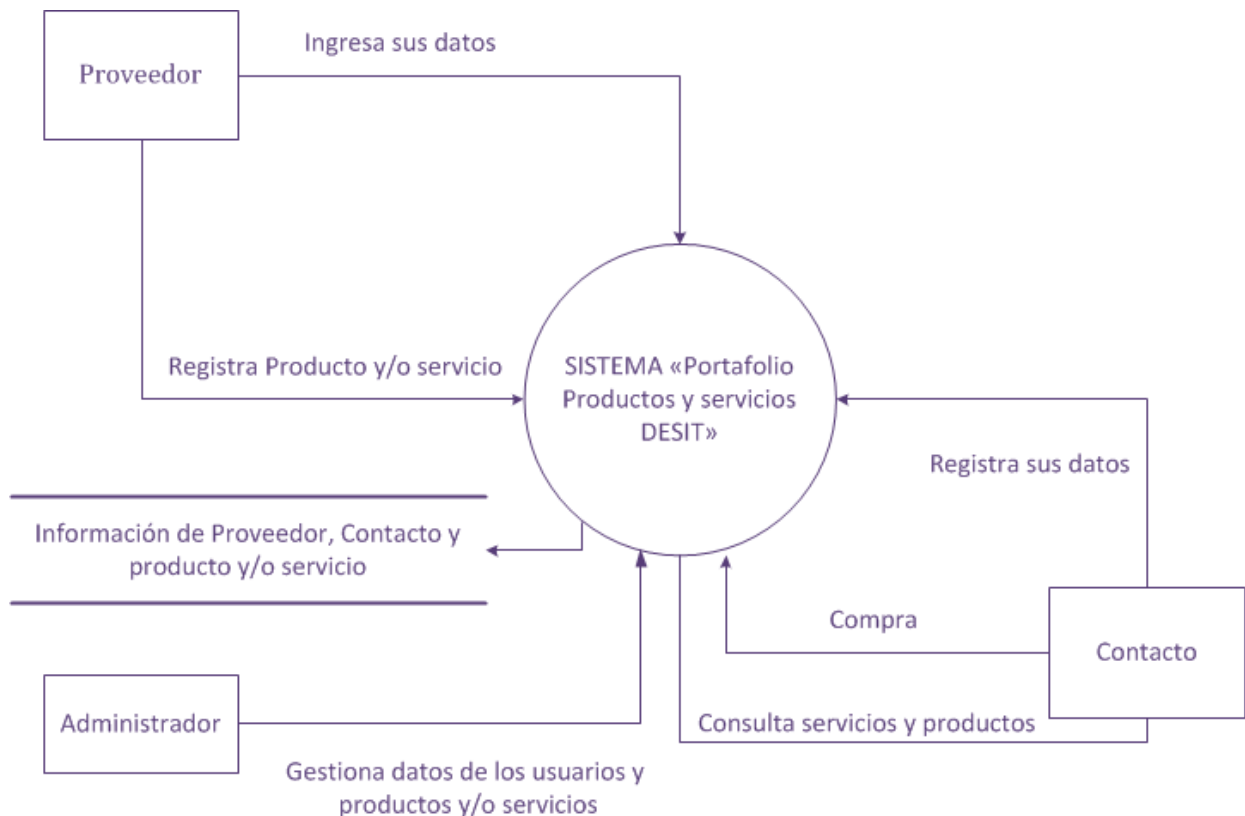


Figura 4.1 Modelo conceptual del sistema.

#### **IV.1.1.1 Entidades, atributos y Relaciones**

El sistema cuenta con varias entidades, quienes a su vez cuentan con varios atributos necesarios para el sistema, y que están relacionados entre sí para poder tener una buena gestión de los datos almacenados en la base de datos.

##### **IV.1.1.1.1 Entidades, atributos**

###### **ENTIDAD PROVEEDOR**

Esta entidad presenta los siguientes atributos:

- CURP
- no\_plaza
- no\_matricula
- nombre\_proveedor
- unidd\_academica
- profesion
- mail
- movil
- Tipo (A= ALUMNO P = PROFESOR)

Como se observa cada uno de los atributos indica el dato que almacenará, el identificador de cada proveedor será su CURP ya que este elemento es único e irrepitable, el campo Tipo esta predefinido y tiene solo dos opciones A Y P (A= ALUMNO P = PROFESOR) el usuario solo podrá seleccionar una de las dos.

Se tomó la decisión de estos datos ya que la gestión de los proveedores será más sencilla además de que esto facilitara una alta y baja de los proveedores en la base de datos.

###### **ENTIDAD CONTACTO**

Esta entidad presenta los siguientes atributos:

- RFC
- Nombre\_cliente
- Dirección
- Teléfono fijo
- Celular
- E-mail
- Municipio
- Estado
- Cliente

Como se observa cada uno de los atributos indica el dato que almacenará, el identificador de cada proveedor será su RFC ya que este elemento es único e irrepetible, el campo Cliente será modificado por el administrador cuando el contacto haya comprado un proyecto y entonces se convertirá ya en cliente. Se tomó la decisión de estos datos ya que la gestión de los contactos será más sencilla además de que esto facilitara una alta y baja de los contactos en la base de datos.

## **ENTIDAD SERVICIOS\_PRODUCTOS**

Esta entidad presenta los siguientes atributos:

- Id\_sp
- Nombre\_sp
- Descripción
- Tipo (servicio o producto)
- Categoría
- Área
- Sector
- Rol

El atributo Id\_sp lo genera el sistema de manera automática, este será el identificador del proyecto ya que será único e irrepetible, para los demás campos su nombre indica la información que será almacenada en ellos, el atributo Tipo esta predefinido y tiene solo dos opciones servicio o producto, el usuario solo podrá seleccionar una de las dos, el campo rol sirve para identificar si el proveedor del servicio es responsable del proyecto o solo tiene una participación en el.

Estos datos se ocuparon con el objetivo de tener una gestión de los datos referentes a los productos y/o servicios que se pueden ofrecer en la DESIT, para su Alta y en determinado momento una baja en la base de datos.

## **ENTIDAD CATEGORÍA**

Esta entidad presenta los siguientes atributos:

- ID\_categoria
- Descripcion\_categoria
- Nivel\_categoria
- Categoria\_superior

El atributo Id\_categoria lo genera el sistema de manera automática, este será el identificador de la categoría ya que será único e irrepetible, para el campo descripción\_categoria indica la información que será almacenada en él, los atributos Nivel\_categoria y Categoria\_superior serán dados por el sistema y es

una forma de controlar las categorías y que el administrador pueda tener un mejor control de los proyectos almacenados en el sistema, ya que una categoría puede tener varios niveles por ejemplo Informática tiene como subcategoría desarrollo de software.

Estos datos se ocuparon con el objetivo de tener una gestión de las categorías a las que pertenecen los proyectos que se pueden ofrecer en la DESIT, para su Alta y en determinado momento una baja en la base de datos.

### **ENTIDAD SECTOR\_ECONOMICO**

Esta entidad presenta los siguientes atributos:

Código  
Descripcion\_economico

El atributo Código lo genera el sistema de manera automática, este será el identificador del sector económico ya que será único e irrepitable, para el campo descripción\_economico a indica la información que será almacenada en él.

Estos datos se ocuparon con el objetivo de tener una gestión de los sectores económicos a los que pertenecen los proyectos que se pueden ofrecer en la DESIT, para su Alta y en determinado momento una baja en la base de datos.

### **ENTIDAD AREA\_FUNCIONAL**

Esta entidad presenta los siguientes atributos:

Id\_area  
Descripcion\_area

El atributo Id\_area lo genera el sistema de manera automática, este será el identificador del área funcional ya que será único e irrepitable, para el campo descripción\_area a indica la información que será almacenada en él.

Estos datos se ocuparon con el objetivo de tener una gestión sencilla de las áreas funcionales a las que pertenecen los proyectos que se pueden ofrecer en la DESIT, para su Alta y en determinado momento una baja en la base de datos.

En el siguiente párrafo se explica del porqué de los datos de las entidades SERVICIOS PRODUCTOS, CATEGORÍA, SECTOR ECONOMICO, AREA FUNCIONAL.

La presentación del listado de productos y servicios del Portafolio de acuerdo a diferentes criterios de búsqueda y página web va de acuerdo al criterio solicitado por el cliente, estos se clasifican en categorías y subcategorías y pueden ser usados o aplicarse en un conjunto de áreas funcionales (ejemplo ventas, mercadotecnia, almacén, etc.) y/o en determinados sectores económicos (actividades o conjuntos de actividades económicas), se ha hecho clásica la

clasificación del conjunto de actividades económicas de un país en tres grandes grupos o sectores: sector primario (minería, agricultura, ganadería, pesca y silvicultura), sector secundario (industrias manufactureras o transformadoras) y sector terciario (comercio y servicios en general), a los cuales se añade más recientemente el denominado sector cuaternario, de contenido no bien definido todavía, en el cual se suelen incluir las actividades económicas relacionadas con el ocio y las nuevas de servicios de sofisticada tecnología. A continuación se comentan los datos que tendrán las tablas para poder catalogar los servicios y/o productos.

### **ENTIDAD USUARIO**

Esta entidad presenta los siguientes atributos:

Login  
Password

Cada atributo indica el tipo de información que almacenara esto con fin de tener gestión de los datos para el acceso al sistema de los interesados y proveedores ya registrados en el sistema.

### **ENTIDAD ADMIN**

Esta entidad presenta los siguientes atributos:

usuario  
password  
nombreadm  
mail\_admin

Cada atributo indica el tipo de información que almacenara esto con fin de tener gestión de los datos de cada posible administrado que tendrá acceso al sistema.

#### **IV.1.1.1.2 Relaciones**

Estas son las relaciones con las que cuenta la base de datos.

Proveedor → **Provee un** → Servicio/producto

Contacto → **se interesa en un** → Servicio/producto

Servicio/producto → **pertenece a una** → categoría

Servicio/producto → **sirve para un** → sector económico

Servicio/producto → es requerido en un → área funcional

Proveedor → le corresponde un → usuario

Contacto → le corresponde un → usuario

Administrador → le corresponde un → usuario

#### IV.1.1.2 Diagrama entidad-relación del sistema

En el diagrama (figura 4.2) se representa de manera gráfica y global las relaciones que hay entre cada una de las entidades mencionadas con anterioridad.

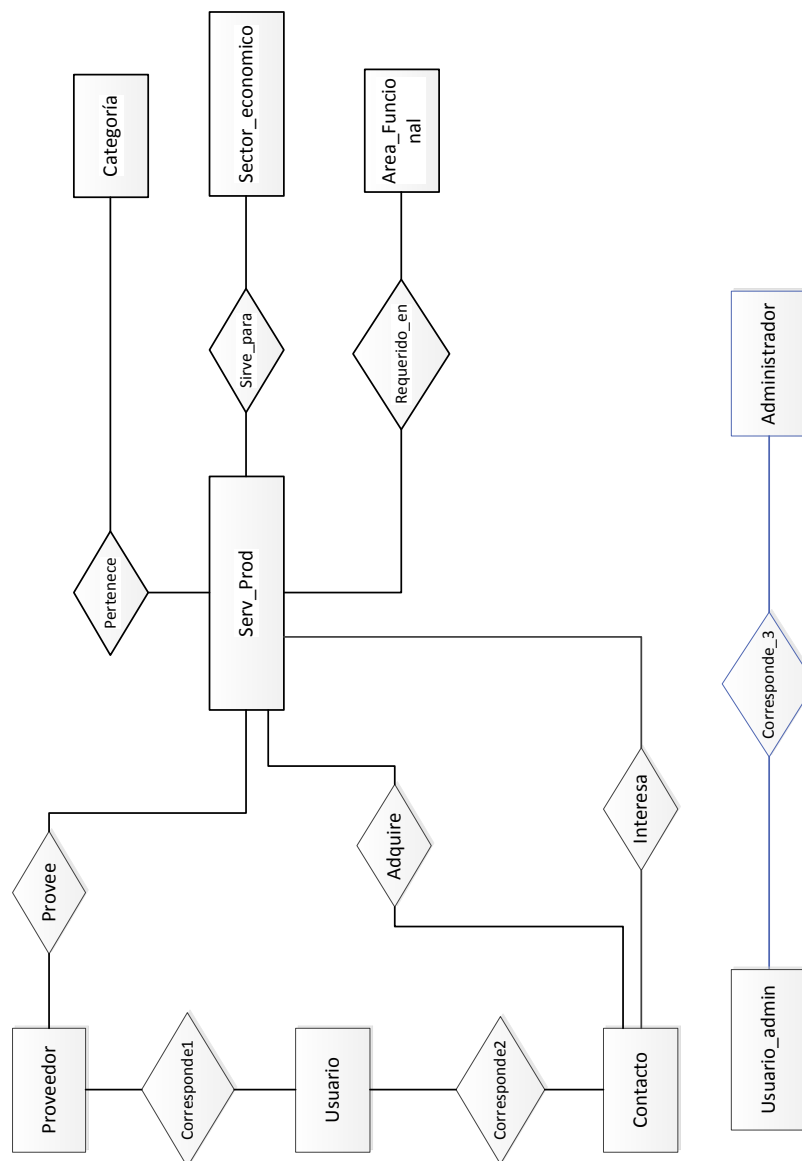


Figura 4.2 Diagrama entidad relación del sistema.

## IV.1.2 Modelo funcional y diagramas de flujo de datos (DFD's)

### IV.1.2.1 Modelo funcional

EL modelo funcional (Figura 4.3) describe los comportamientos y operaciones de los objetos, además muestra la dependencia de datos en el sistema, de la misma manera describe la computación dentro del sistema, así como los valores de salida que se derivan de los valores de entrada, sin importar el orden en que son computados

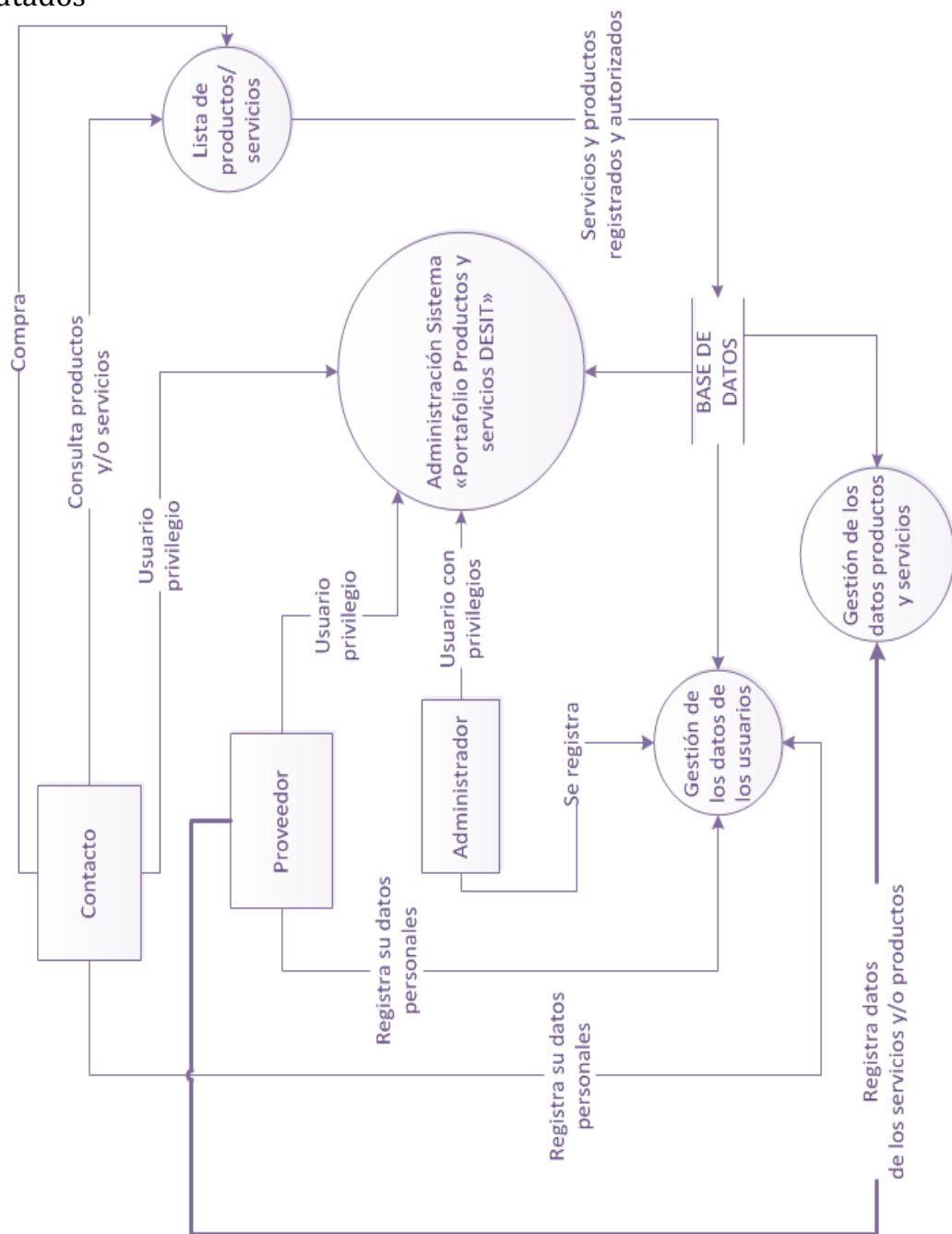


Figura 4.3 Modelo funcional del sistema.

#### IV.1.2.2 Diagramas de flujo de datos (DFD's)

El Diagrama de flujo muestra en qué forma se procesan los datos, entre las principales funciones del sistema.

##### Diagrama de flujo del caso de uso registro de proveedor.

En el diagrama (Figura 4.4) se muestra paso a paso lo que tiene que hacer el usuario con el rol de proveedor para ingresar sus datos personales y de esta manera quedar registrado en el sistema.

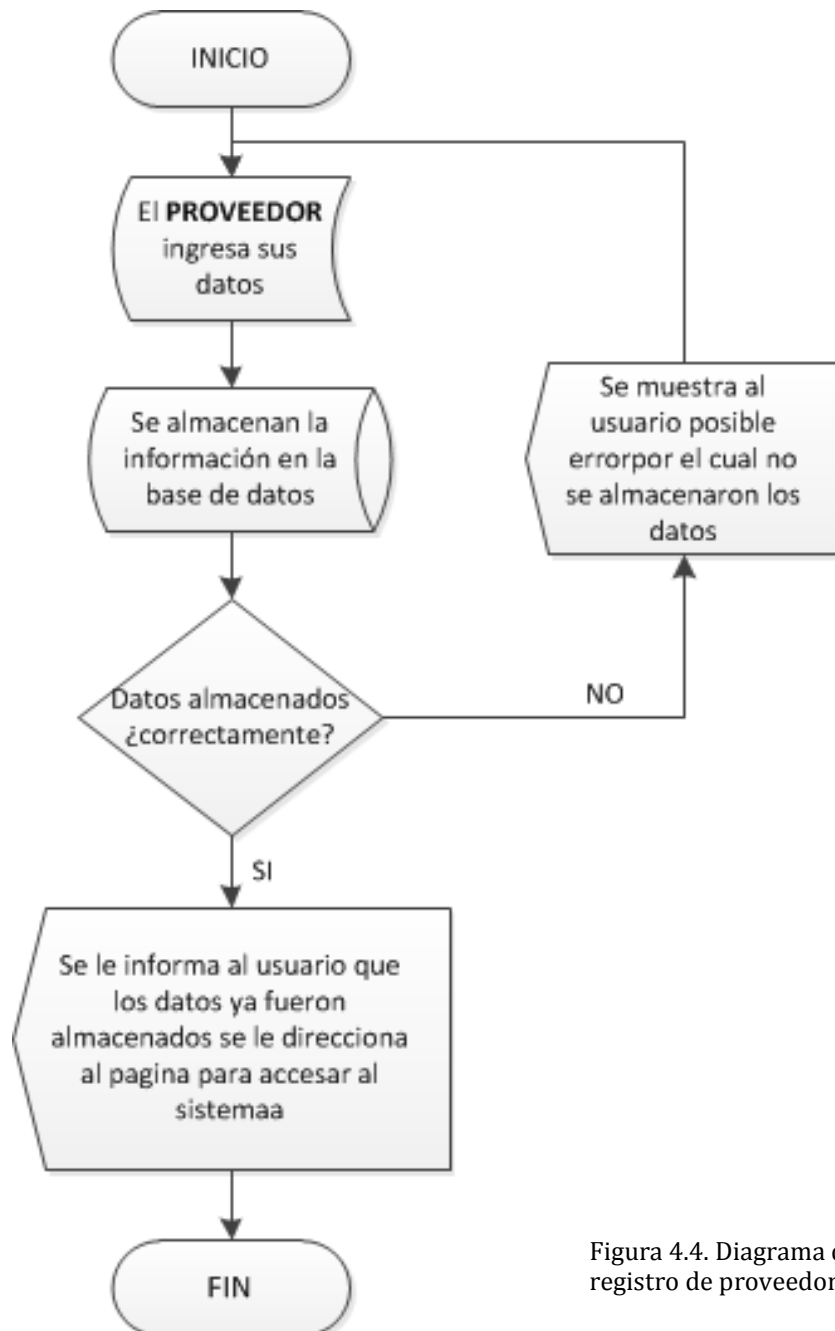


Figura 4.4. Diagrama de flujo para el caso de uso registro de proveedor.

## Diagrama de flujo del caso de uso registro de contacto.

En el diagrama (Figura 4.5) se muestra paso a paso lo que tiene que hacer el usuario con el rol de contacto para ingresar sus datos personales y de esta manera quedar registrado en el sistema.

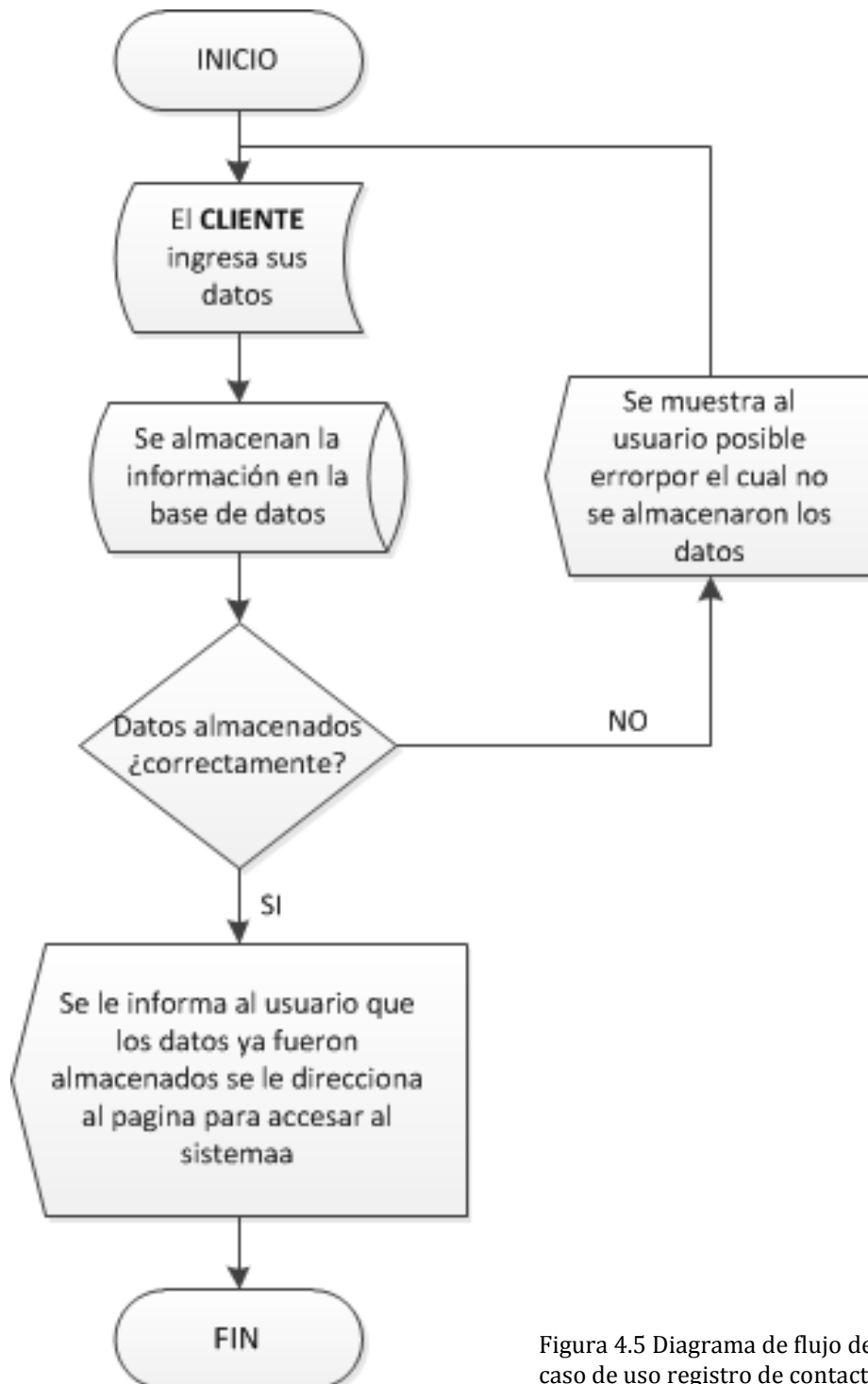


Figura 4.5 Diagrama de flujo del caso de uso registro de contacto.

## Diagrama de flujo del caso de uso acceso usuario.

En el diagrama (Figura 4.6) que es genérico ya que funcionara para los tres tipos de usuarios (administrador, contacto y proveedor) que tiene el sistema, se muestra paso a paso lo que tiene que hacer el usuario para ingresar al sistema.

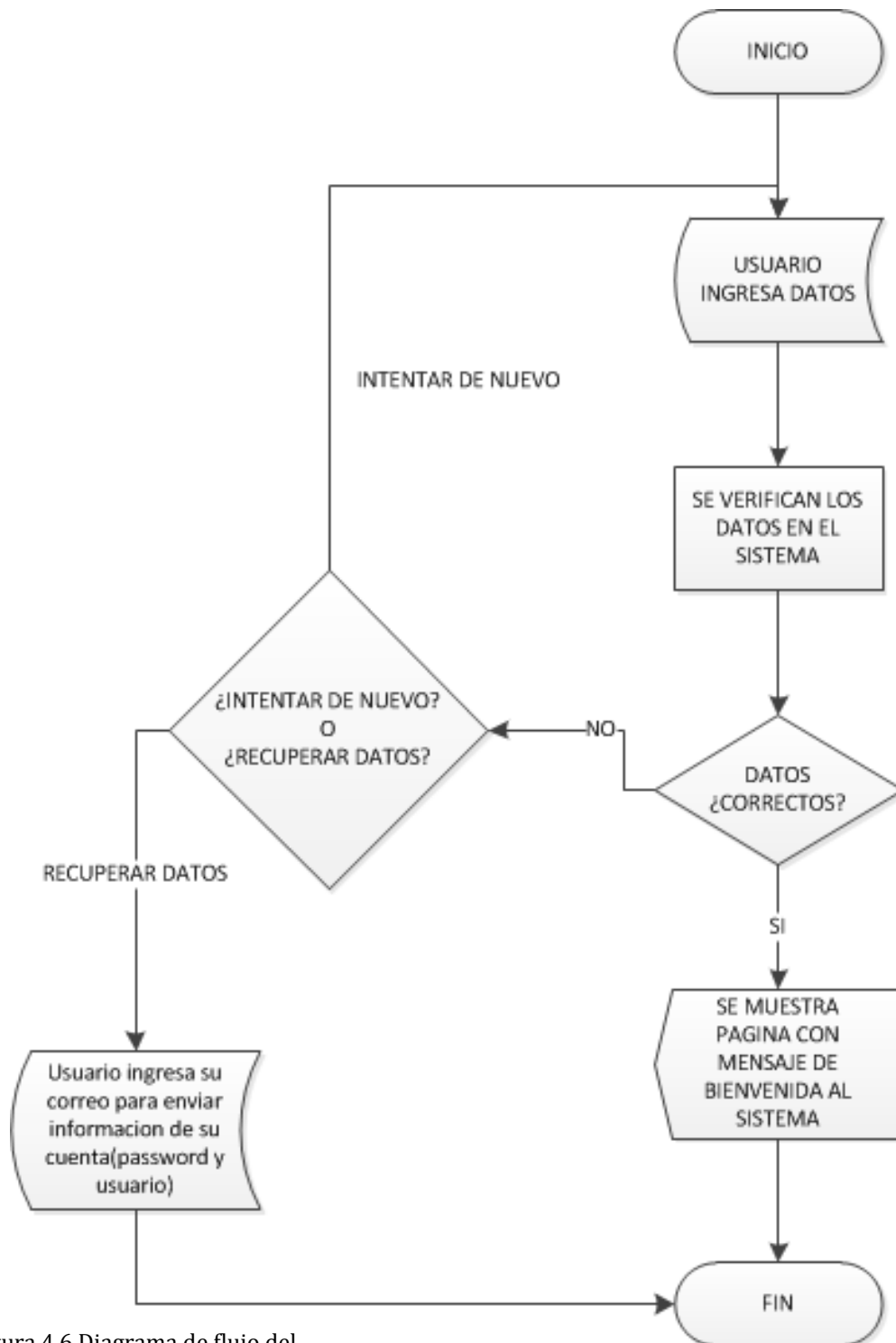


Figura 4.6 Diagrama de flujo del caso de uso acceso usuario.

## Diagrama de flujo del caso de uso registro de producto y/o servicio.

En el diagrama (Figura 4.7) se muestra paso a paso lo que tiene que hacer el usuario con el rol de proveedor para dar de alta un proyecto en el sistema.

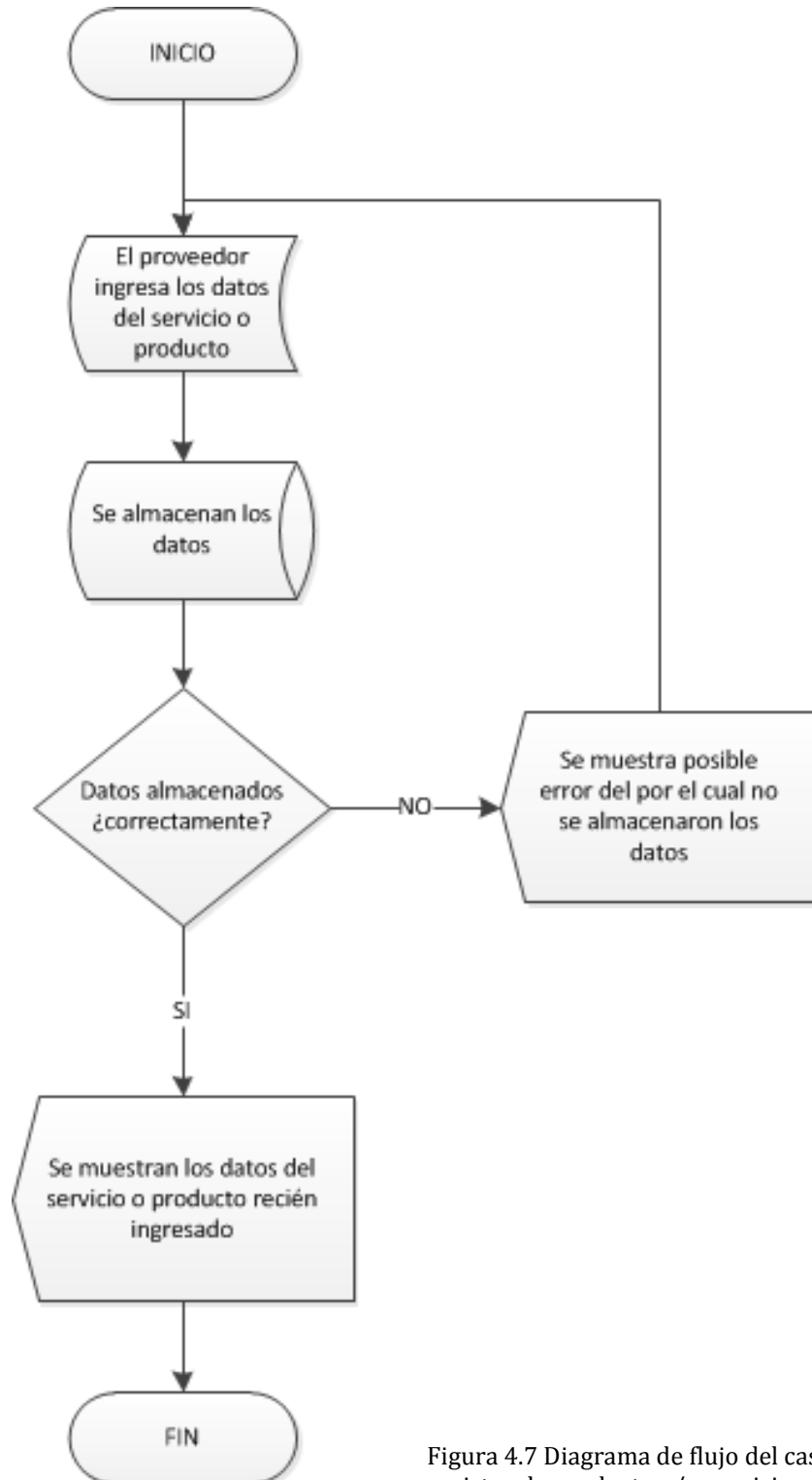


Figura 4.7 Diagrama de flujo del caso de uso registro de producto y/o servicio.

### IV.3 Diseño lógico.

El diseño lógico parte del diseño conceptual obtenido en la fase anterior, se llega un diseño lógico. Transformamos las entidades y relaciones obtenidas del modelo anterior en tablas. Para ello usamos la normalización tema explicado en el siguiente apartado **IV.3.2 Normalización**.

#### IV.3.1 Esquema Relacional de la Base de Datos.

En la Figura 4.8 se muestra el diagrama lógico obtenido del programa Toad Data Modeler 3.6, donde se pueden apreciar las entidades con sus atributos y las relaciones entre ellos ya explicadas en el apartado **IV.1.1.1.2 Relaciones**.

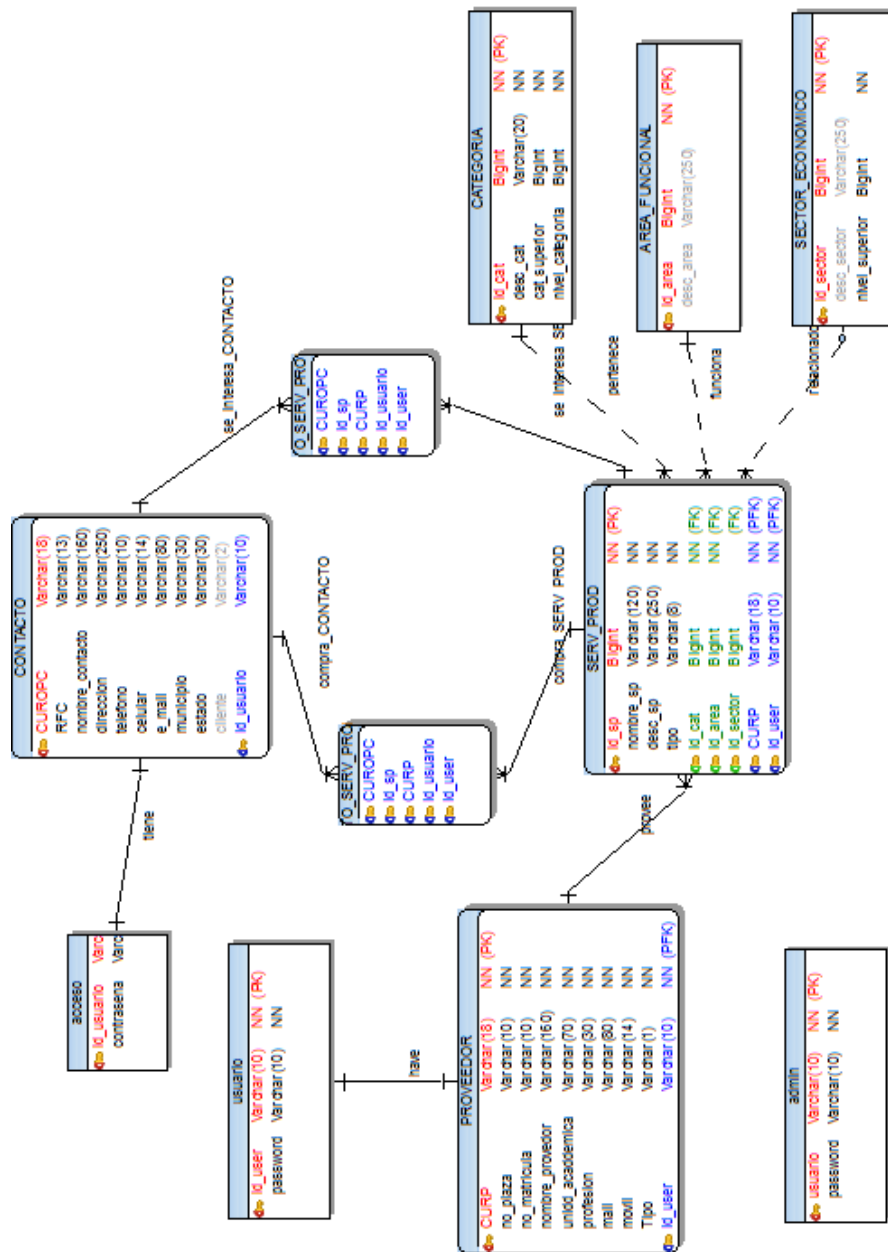


Figura 4.8 Diagrama relacional de la base de datos.

### IV.3.2 Normalización.

A continuación se describe la dependencia funcional de las tablas servicio y producto, categoría, área funcional y sector económico.

SERVICIO Y PRODUCTO → id, nombre, descripción, tipo

CATEGORÍA → id, descripción, categoría superior

AREA FUNCIONAL → id, descripción

SECTOR ECONÓMICO → id, descripción

Para cumplir con las 3 formas normales se busca tener atributos atómicos, en la tabla servicio-producto (Tabla 4.1) se puede identificar que los datos nombre, descripción y tipo dependen totalmente de la clave principal id, pues esta será quien identifique a un proyecto.

SERVICIO/PRODUCTO			
id	nombre	descripción	tipo
4	RESERVACIONES PLUS PLANES&HOTELS	PROGRAMA WEB PARA RESERVAR BOLETOS DE AVION Y HOTELES	Servicio
5	ABONO MINAL B	Es un complejo de elementos menores activos de reposición y de lenta liberación para cubrir las pérdidas naturales e inevitables de los suelos debido a su explotación.	Producto

Tabla. 4.1 Se muestra la dependencia que existe sobre el campo id.

De la misma forma se normalizó la tabla Categoría (Tabla 4.2) ya que descripción y categoría superior dependen exclusivamente del id\_cat, sucede lo mismo con las tablas área funcional (Tabla 4.3) y sector económico (Tabla 4.4).

CATEGORIA		
id_cat	descripción	categoría superior
13	TURISMO	1
11	AGRICULTURA Y ALIMENTACIÓN	1

Tabla. 4.2 Se muestra la dependencia que existe sobre el campo id\_cat.

AREA FUNCIONAL	
id_area	descripción
15	AGENCIAS DE VIAJE
11	AGROQUÍMICA

Tabla. 4.3 Se muestra la dependencia que existe sobre el campo id\_area.

SECTOR ECONOMICO	
id_sector	descripción
1	TERCIARIO
9	PRIMARIO

Tabla. 4.4 Se muestra la dependencia que existe sobre el campo id\_sector.

Dónde en cada entidad se tiene un elemento atómico y único e irrepetible fácil de ubicar con su clave principal.

Al consultar ya en el sistema se mostrara la información más completa (Figura 4.9), como se presenta a continuación.

NOMBRE PROYECTO	DESCRIPCION	TIPO	PROVEEDOR	MAIL	CATEGORIA	AREA	SECTOR
RESERVACIONES PLUS PLANES&HOTELS	PROGRAMA WEB PARA RESERVAR BOLETOS DE AVION Y HOTELES	Servicio	Proveedor1	proveedor1@mail.com	TURISMO	AGENCIAS DE VIAJE	TERCIARIO
ABONO MINAL B	Es un complejo de elementos menores activos de reposición y de lenta liberación para cubrir las pérdidas naturales e inevitables de los suelos debido a su explotación.	Producto	Proveedor2	proveedor2@mail.com	AGRICULTURA Y ALIMENTACIÓN	AGROQUÍMICA	PRIMARIO

Figura. 4.9 Muestra cómo se presenta la información en sistema.

#### IV.4 Diseño de la Interfaz de Usuario

El diseño de interfaces de usuario es una tarea que ha adquirido relevancia en el desarrollo de un sistema. La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso. Los principios que se presentan son de utilidad para creación de interfaces funcionales y de fácil operación. El uso de Internet, ha favorecido la implementación de interfaces invisibles. Esto significa que el usuario siempre ve una página específica, pero nunca puede conocer la totalidad del espacio de páginas de Internet. La navegación en las aplicaciones debe ser reducida a la mínima expresión. El usuario debe sentir que se mantiene en un único lugar y que el que va variando es su trabajo. Esto no solamente elimina la necesidad de mantener mapas u otras ayudas de navegación, sino que además brindan al usuario una sensación de autonomía, por lo que para el diseño de la interfaz se realizó utilizando el software Dreamweaver 8, PHP 5 y HTML para la codificación de sistema, para poder acceder al sistema se necesita un explorador web, a continuación se muestra la pantalla principal del sistema así como los botones con los que se cuenta.

La pantalla principal (Figura 4.10) del sistema presentan los siguientes botones:

**INICIO** → Envía directamente al menú principal del sistema.

**CONSULTAR** → Muestra los servicios y productos en el sistema

**LOGIN** → Se introduce usuario y password para acceder al sistema y de acuerdo al tipo de acceso que tenga (proveedor, cliente y administrador).  
**REGISTRARSE** → Se registran los usuarios.



Figura 4.10 Pantalla principal del sistema.

# CAPÍTULO V.

## DISEÑO FÍSICO E IMPLEMENTACIÓN

En este capítulo se documentará la Instalación y configuración de los Servidores Web y de Bases de Datos, la creación de tablas utilizando el lenguaje SQL, se mostrará la estructura final de la base de datos, como se realizaron las consultas, pantallas del sistema y los reportes del mismo.

### V.1. Instalación y configuración de los Servidores Web y de Bases de Datos.

En este punto se describe la instalación de wamperver 2.2 que incluye Apache, PHP y MySQL.

El instalador es bastante sencillo de seguir (Figuras 5.1 a 5.7), se anexan algunas capturas de pantalla.

Figura 5.1  
Primera pantalla  
del instalador.

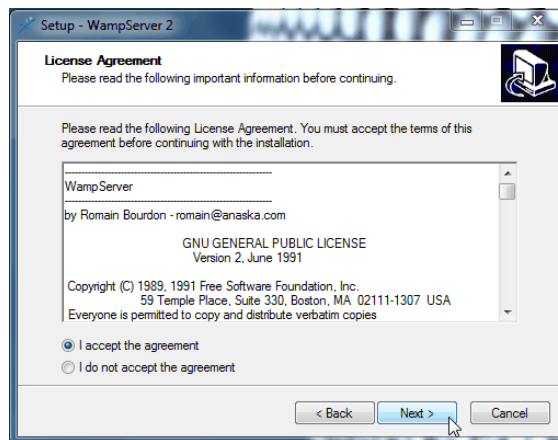
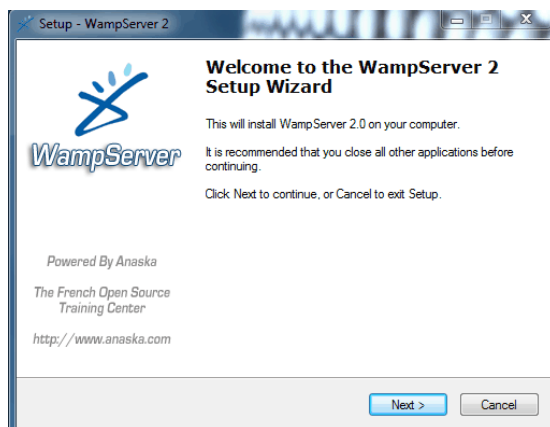


Figura 5.2 Segunda  
pantalla del instalador.

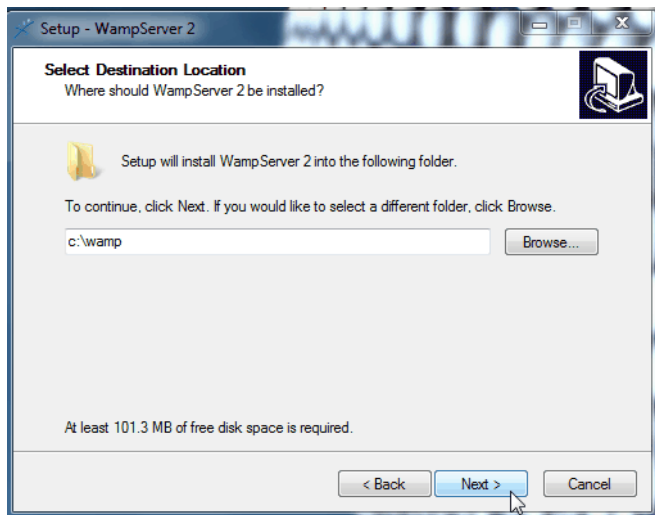


Figura 5.3 Tercera pantalla del instalador.

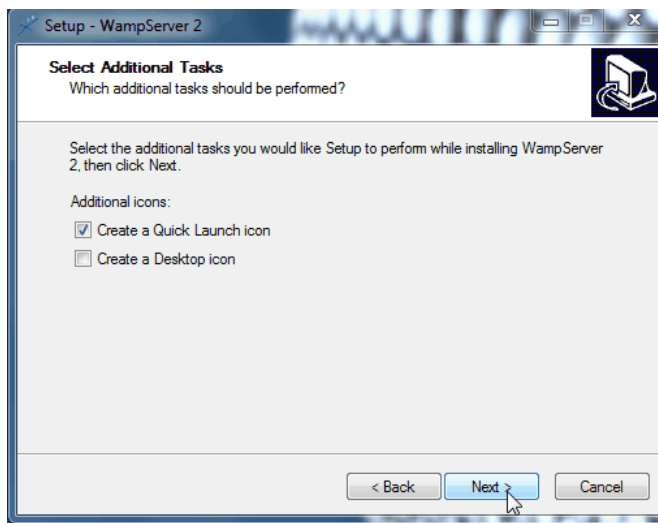


Figura 5.4 Cuarta pantalla del instalador.

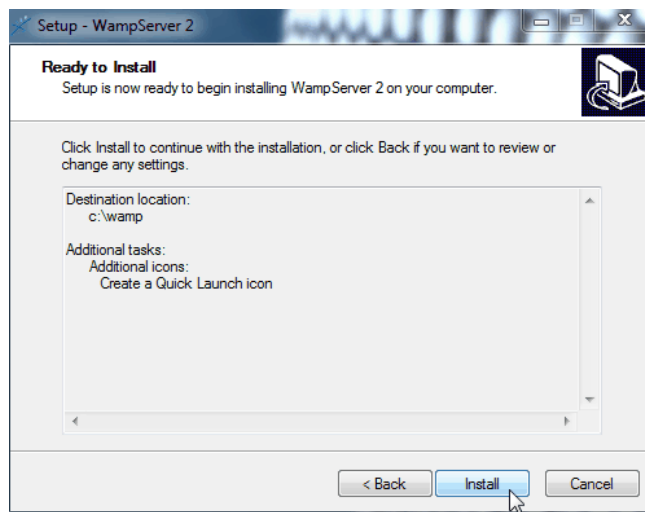


Figura 5.5 Quinta pantalla del instalador.

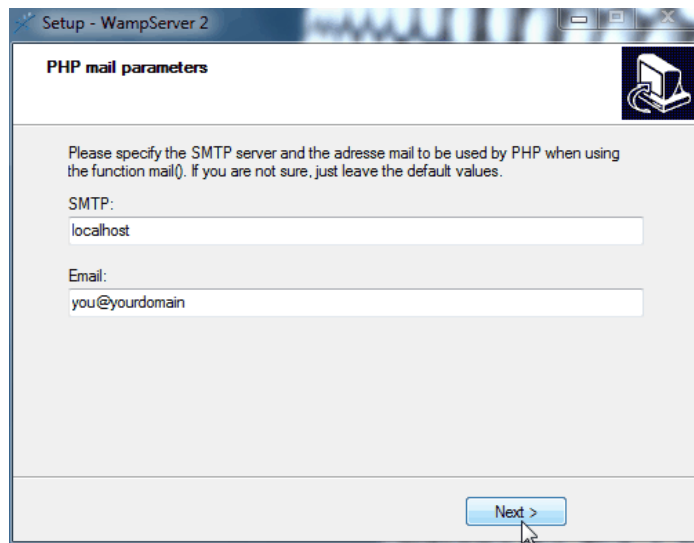


Figura 5.6 Sexta pantalla del instalador.

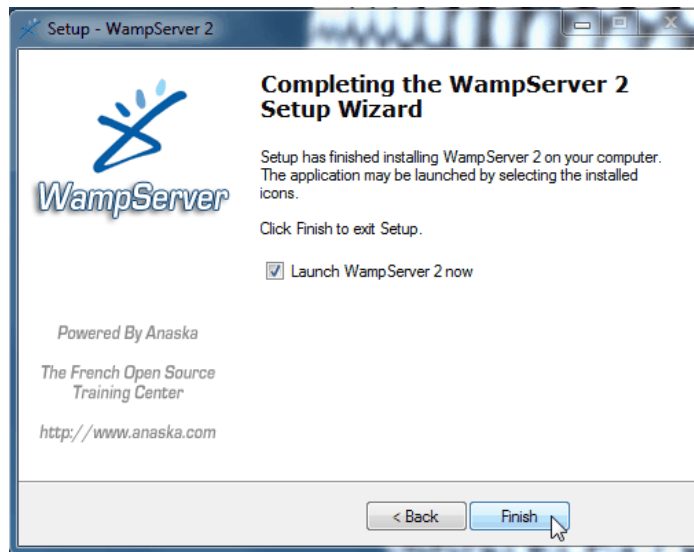


Figura 5.7 Séptima pantalla del instalador.

Una vez instalado podrán ver el icono de WampServer en la Barra de Tareas (Figura 5.8). Si se le da clic al icono el menú de opciones aparecerá una listas con las opciones de wampserver 2.2 (figura 5.9).

Figura 5.8 Icono en barra de tareas de Windows.

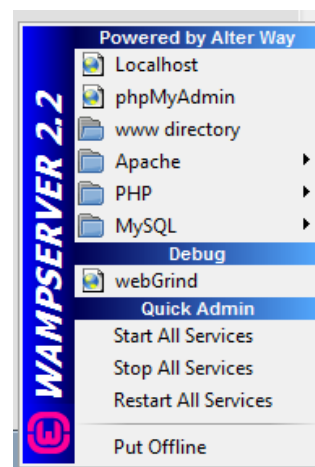


Figura 5.9 Menú de opciones de wampserver 2.2

Al escoger "Localhost" en el menú se abre el navegador defecto con la dirección: <http://localhost> mostrando la página `index.php` (Figura 5.10) creada por el programa. En ella se desglosan la configuración de PHP así como también las herramientas disponibles.



Figura 5.10 Pantalla de la página localhost de wampserver 2.2

Lo próximo será crear una base de datos y un usuario MySQL con el cual podamos conectar con Dreamweaver. Para eso usamos phpMyAdmin. (phpMyAdmin es una herramienta para trabajar con MySQL hecha en PHP, es por eso que se trabaja con un navegador web (Para el sistema se usó la consola de MySQL). Se puede acceder phpMyAdmin usando el enlace que está en la imagen anterior o dando clic al icono de WampServer y escoger phpMyAdmin.

## V.2. Creación de tablas utilizando el lenguaje SQL

Existen tres sentencias que se emplean en SQL para crear o modificar una base de datos. Estas sentencias son:

- CREATE, que define y crea un objeto en la base de datos.
- DROP, que elimina un objeto existente en la base de datos.
- ALTER, que modifica la definición de un objeto de la base de datos.

La estructura más importante de una base de datos relacional es la tabla, y se ocupa la sentencia CREATE TABLE. Las columnas aparecen en una lista separada por comas e insertada entre paréntesis. El orden de las definiciones de las columnas determina el orden de izquierda a derecha de las columnas en la tabla.

Cada definición específica:

- El nombre de la columna, que se utiliza para referirse a la columna en las sentencias SQL. Cada columna de la tabla debe tener un nombre único, pero los nombres pueden ser iguales a los de las columnas de otras tablas.
- El tipo de datos de la columna, que identifica la clase de datos que la columna almacena.
- Si la columna no puede contener datos nulos, la cláusula NOT NULL impide que aparezcan valores NULL en la columna, en caso contrario se permiten los valores nulos en la columna.
- Un valor por omisión es opcional para la columna. Se utiliza este valor cuando una sentencia INSERT aplicada a la tabla no especifica un valor para la columna.

Ejemplo de cómo se creó un par de tablas del sistema:

```
CREATE TABLE PROVEEDOR
( CURP Varchar(18) NOT NULL,
  no_plaza Varchar(10) NOT NULL,
  no_matricula Varchar(10) NOT NULL,
  nombre_proveedor Varchar(160) NOT NULL,
  unidd_acaddemica Varchar(70) NOT NULL,
  profesion Varchar(30) NOT NULL,
  mail Varchar(80) NOT NULL,
  movil Varchar(14) NOT NULL,
  Tipo Varchar(1) NOT NULL,
  id_user Varchar(10) NOT NULL);
ALTER TABLE PROVEEDOR ADD PRIMARY KEY (CURP,id_user);
```

Estructura física de la tabla PROVEEDOR (tabla 5.1):

CURP	no_plaza	no_matricula	nombre_proveedor	unidd_acaddemica	profesion	mail	movil	Tipo	id_user
------	----------	--------------	------------------	------------------	-----------	------	-------	------	---------

Tabla 5.1 Muestra la estructura final de la tabla Proveedor

```
CREATE TABLE CONTACTO
( CUROPC Varchar(18) NOT NULL,
  RFC Varchar(13) NOT NULL,
  nombre_contacto Varchar(160) NOT NULL,
  direccion Varchar(250) NOT NULL,
  telefono Varchar(10) NOT NULL,
  celular Varchar(14) NOT NULL,
  e_mail Varchar(80) NOT NULL,
```

```

municipio Varchar(30) NOT NULL,
estado Varchar(30) NOT NULL,
cliente Varchar(2),
id_usuario Varchar(10) NOT NULL);
ALTER TABLE CONTACTO ADD PRIMARY KEY (CUROPC,id_usuario)

```

Estructura física de la tabla CONTACTO (Tabla 5.2):

CUROPC	RFC	nombre_contacto	direccion	telefono	celular	e_mail	municipio	estado	id_usuario
--------	-----	-----------------	-----------	----------	---------	--------	-----------	--------	------------

Tabla 5.2 Muestra la estructura final de la tabla Contacto

```

CREATE TABLE SERV_PROD
(id_sp Bigint NOT NULL,
nombre_sp Varchar(120) NOT NULL,
desc_sp Varchar(250) NOT NULL,
tipo Varchar(8) NOT NULL,
id_cat Bigint NOT NULL,
id_area Bigint NOT NULL,
id_sector Bigint,
CURP Varchar(18) NOT NULL,
id_user Varchar(10) NOT NULL );
ALTER TABLE SERV_PROD ADD PRIMARY KEY (id_sp,CURP,id_user);

```

Estructura física de la tabla SER\_PROD (productos y servicios, Tabla 5.3):

id_sp	nombre_sp	desc_sp	tipo	id_cat	id_area	id_sector	CURP	id_user	rol
-------	-----------	---------	------	--------	---------	-----------	------	---------	-----

Tabla 5.3 Muestra la estructura final de la tabla Ser\_Prod.

### V.3. Estructura final de la base de datos

#### Estructura de la tabla acceso.

Se ocupó el tipo varchar ya que es una cadena de longitud variada, para este caso la dimensión será de 10 caracteres como máximo y con esto se podrán almacenar el login del usuario y la contraseña del usuario (Tabla 5.4).

Columna	Tipo	Nulo	Predeterminado
<i>id_usuario</i>	varchar(10)	No	
<b>contrasena</b>	varchar(10)	No	

Tabla 5.4 Muestra la estructura final de la tabla acceso.

### Estructura de tabla para la tabla admin.

Para esta tabla se ocupó el tipo varchar una cadena de longitud variada para todos los elemento de la tabla y con esto se podrán almacenar el login, contraseña, nombre y mail del administrador (Tabla 5.5).

Columna	Tipo	Nulo	Predeterminado
<i>usuario</i>	varchar(10)	No	
<b>password</b>	varchar(10)	No	
<b>nombreadm</b>	varchar(160)	No	
<b>mail_adm</b>	varchar(120)	No	

Tabla 5.5 Muestra la estructura final de la tabla admin.

### Estructura de tabla para la tabla area\_funcional.

En esta tabla se ocupó el tipo bigint para almacenar el id de del are funcional y su manejo sea más sencillo a la hora de buscar un elemento de esta tabla (Tabla 5.6).

Columna	Tipo	Nulo	Predeterminado
<i>id_area</i>	bigint(20)	No	
<b>desc_area</b>	varchar(250)	Sí	NULL

Tabla 5.6 Muestra la estructura final de la tabla area\_funcional.

### Estructura de tabla para la tabla categoría

En esta tabla se ocupó el tipo bigint para almacenar el id de la categoría, su categoría superior y su nivel de categoría, y de esta forma su manejo sea más sencillo a la hora de buscar un elemento (Tabla 5.7).

Columna	Tipo	Nulo	Predeterminado
<i>id_cat</i>	bigint(20)	No	
<b>desc_cat</b>	varchar(150)	No	
<b>cat_superior</b>	bigint(20)	No	
<b>nivel_categoria</b>	bigint(20)	No	

Tabla 5.7 Muestra la estructura final de la tabla categoria.

### Estructura de tabla para la tabla contacto

Para esta tabla se ocupó el tipo varchar una cadena de longitud variada para todos los elemento de la tabla y con esto se podrán almacenar el de una manera más precisa la información del usuario contacto (Tabla 5.8).

Columna	Tipo	Nulo	Predeterminado
<i>CUROPC</i>	varchar(18)	No	
<b>RFC</b>	varchar(13)	No	
<b>nombre_contacto</b>	varchar(160)	No	
<b>direccion</b>	varchar(250)	No	
<b>telefono</b>	varchar(10)	No	
<b>celular</b>	varchar(14)	No	
<b>e_mail</b>	varchar(80)	No	
<b>municipio</b>	varchar(30)	No	
<b>estado</b>	varchar(30)	No	
<b>cliente</b>	varchar(2)	Sí	NULL
<i>id_usuario</i>	varchar(10)	No	

Tabla 5.8 Muestra la estructura final de la tabla contacto.

### Estructura de tabla para la tabla proveedor.

En esta tabla se ocupó el tipo varchar una cadena de longitud variada para todos sus elementos, con esto se podrán almacenar el de una manera más precisa la información del usuario proveedor (Tabla 5.9).

Columna	Tipo	Nulo	Predeterminado
<i>CURP</i>	varchar(18)	No	
<b>no_plaza</b>	varchar(10)	No	
<b>no_matricula</b>	varchar(10)	No	
<b>nombre_proveedor</b>	varchar(160)	No	
<b>unidd_academica</b>	varchar(70)	No	
<b>profesion</b>	varchar(30)	No	
<b>mail</b>	varchar(80)	No	
<b>movil</b>	varchar(14)	No	
<b>Tipo</b>	varchar(1)	No	
<i>id_user</i>	varchar(10)	No	

Tabla 5.9 Muestra la estructura final de la tabla proveedor.

### Estructura de tabla para la tabla sector\_economico.

En esta estructura se ocupó el tipo bigint para almacenar el id del sector económico, de esta forma su manejo sea más sencillo a la hora de buscar un elemento (Tabla 5.10).

Columna	Tipo	Nulo	Predeterminado
<i>id_sector</i>	bigint(20)	No	
<b>desc_sector</b>	varchar(250)	Sí	NULL
<b>nivel_superior</b>	bigint(20)	No	

Tabla 5.10 Muestra la estructura final de la tabla sector\_economico.

### Estructura de tabla para la tabla serv\_prod.

Para esta tabla se ocupó el tipo varchar una cadena de longitud variada y el bigint para los elemento de la tabla de esta manera se podrán almacenar de una manera más precisa la información del servicio y/o producto (Tabla 5.11).

Columna	Tipo	Nulo	Predeterminado
<i>id_sp</i>	bigint(20)	No	
<b>nombre_sp</b>	varchar(120)	No	
<b>desc_sp</b>	varchar(250)	No	
<b>tipo</b>	varchar(8)	No	
<b>id_cat</b>	bigint(20)	No	
<b>id_area</b>	bigint(20)	No	
<b>id_sector</b>	bigint(20)	Sí	NULL
<i>CURP</i>	varchar(18)	No	
<i>id_user</i>	varchar(10)	No	
<b>rol</b>	varchar(12)	No	

Tabla 5.12 Muestra la estructura final de la tabla serv\_prod.

### Estructura de tabla para la tabla usuario.

Se ocupó el mismo tipo de datos que se usó para la tabla administrador (Tabla 5.13).

Columna	Tipo	Nulo	Predeterminado
<i>id_user</i>	varchar(10)	No	
<b>password</b>	varchar(10)	No	

Tabla 5.13 Muestra la estructura final de la tabla usuario.

## V.4. Consultas

En este apartado se muestra como se realizaron algunas consultas de sistema, para la localización de los proyectos previamente almacenados en sistema, dependiendo del criterio de búsqueda que el usuario solicite.

### **Consulta para localizar los proyectos por el campo tipo (servicio y producto).**

La consulta se formuló de la siguiente forma, buscar un proyecto por su tipo y en esta consulta deberían aparecer datos como descripción, categoría, área funcional sector y por ultimo una opción para contactar al proveedor del proyecto, el primer dato que se toma es el campo tipo para así obtener todos los proyectos que cumplen con el criterio de búsqueda (servicio o tipo) del usuario, el siguiente dato que se toman de la tabla es el CURP para de esta manera obtener el nombre del proveedor y su correo electrónico para poder contactarlo pero todo esto de la tabla proveedor, de la misma forma para obtener la descripción del sector económico, categoría, y área funcional ya que en la estructura de la tabla serv\_prod solo aparecen los id's que son números y no servirían para una presentación descriptiva del proyecto. De la misma forma se realiza las demás búsquedas de los proyectos, lo único que cambia es el campo detonante ya que si se solicita encontrar un proyecto por su categoría, área funcional, sector económico, nombre del proyecto o nombre del proveedor, será el primer elemento de búsqueda y así de manera consecutiva como se explica anteriormente.

```
SELECT * FROM serv_prod, proveedor,categoria,area_funcional,sector_economico
        WHERE serv_prod.tipo='$tipo'
        and
        serv_prod.CURP = proveedor.CURP
        and
        serv_prod.id_cat=categoria.id_cat
        and
        serv_prod.id_area = area_funcional.id_area
        and
        serv_prod.id_sector=sector_economico.id_sector
```

### **Consulta para localizar los proyectos por el campo Categoría.**

En esta consulta el dato detonador es **cat**, y de esta manera se obtienen los proyectos que pertenecen a la categoría que el usuario solicito.

```
SELECT * FROM serv_prod, proveedor,categoria,area_funcional,sector_economico
        WHERE categoria.desc_cat='$cat'
        and
        serv_prod.CURP = proveedor.CURP
        and
        serv_prod.id_cat=categoria.id_cat
        AND
        serv_prod.id_area = area_funcional.id_area
        and
```

```
serv_prod.id_sector=sector_economico.id_sector
```

### **Consulta para localizar los proyectos por el campo Área Funcional.**

En esta consulta el dato detonador es **área**, y de esta manera se obtienen los proyectos que funcionan en el are que el usuario solicito.

```
SELECT * FROM serv_prod, proveedor,categoria,area_funcional,sector_economico
      WHERE area_funcional.desc_area='$area'
      and
      serv_prod.CURP = proveedor.CURP
      and
      serv_prod.id_cat=categoria.id_cat
      AND
      serv_prod.id_area = area_funcional.id_area
      and
      serv_prod.id_sector=sector_economico.id_sector
```

### **Consulta para localizar los proyectos por el campo Sector económico.**

En esta consulta el dato detonador es **sec**, y de esta manera se obtienen los proyectos que se relacionan al sector económico que el usuario solicito.

```
SELECT * FROM serv_prod, proveedor,categoria,area_funcional,sector_economico
      WHERE sector_economico.desc_sector='$sec'
      and
      serv_prod.CURP = proveedor.CURP
      and
      serv_prod.id_cat=categoria.id_cat
      AND
      serv_prod.id_area = area_funcional.id_area
      and
      serv_prod.id_sector=sector_economico.id_sector
```

### **Consulta para localizar los proyectos por el campo Nombre del proyecto.**

En esta consulta el dato detonador es **nom**, y de esta manera se obtienen el proyecto con ese nombre en específico y que el usuario solicito.

```
SELECT * FROM serv_prod, proveedor,categoria,area_funcional,sector_economico
      WHERE serv_prod.nombre_sp='$nom'
      and
      serv_prod.CURP = proveedor.CURP
      and
```

```
serv_prod.id_cat=categoria.id_cat
AND
serv_prod.id_area = area_funcional.id_area
and
serv_prod.id_sector=sector_economico.id_sector
```

### **Consulta para localizar los proyectos por el campo Nombre del proveedor.**

En esta consulta el dato detonador es **pro**, y de esta manera se obtienen los proyectos que provee un determinado proveedor y que el usuario solicito.

```
SELECT * FROM serv_prod, proveedor,categoria,area_funcional,sector_economico
WHERE proveedor.nombre_proveedor='$pro'
and
serv_prod.CURP = proveedor.CURP
and
serv_prod.id_cat=categoria.id_cat
AND
serv_prod.id_area = area_funcional.id_area
and
serv_prod.id_sector=sector_economico.id_sector
```

### **Consulta todos los proyectos registrados en el sistema.**

En esta consulta no hay un dato detonador ya que se regresarán todos los proyectos dados de alta en el sistema.

```
SELECT * FROM serv_prod, proveedor,categoria,area_funcional,sector_economico
WHERE
serv_prod.CURP = proveedor.CURP
and
serv_prod.id_cat=categoria.id_cat
AND
serv_prod.id_area = area_funcional.id_area
and
serv_prod.id_sector=sector_economico.id_sector
```

## **V.5. CSS**

Las CSS's es una tecnología que permite crear páginas web de una manera más exacta. Gracias a las CSS es mucho más fácil ver los resultados finales de la página, pudiendo hacer muchas cosas que no se podía hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores... etc.

Las CSS son las siglas de Cascading Style Sheets, en español Hojas de estilo en Cascada.

Se muestra parte del código del style. css que se utilizó para crear el sistema. Esta parte código se utilizó para el cuerpo de la página.

```
body {  
    margin-top: 0px;  
    margin-bottom: 30px;  
}
```

Este container se ocupa para el título de cada una de las páginas (Figura 5.1).

```
#container {  
    width: 750px;  
    margin: 0 auto;  
    font-family: Verdana, Arial, Helvetica, sans-serif;  
    font-size: 11px;  
    line-height: 1.6em;  
    color: #666666;  
    background-color: #FFFFFF;  
}
```



PORTAFOLIO DE  
PRODUCTOS Y SERVICIOS  
DESIT BUAP

Figura 5.1 Estilo de los encabezados que presentan todas las páginas del sistema

Este contenedor se ocupó para dar formato al contenido de las páginas del sistema (Figura 5.2).

```
#container2 {  
    width: 746px;  
    margin: 0 auto;  
    font-family: Verdana, Arial, Helvetica, sans-serif;  
    font-size: 11px;  
    line-height: 1.6em;  
    color: #666666;  
    background-color: #FFFFFF;  
    border-bottom: 2px solid #2763A5; }  
}
```

**Objetivo del portafolio**

En la División de Estudios Superiores de Ingeniería y Tecnología (DESIT) de la Benemérita Universidad Autónoma de Puebla se está elaborando un Plan estratégico de Integración Social (PEIS), entre las acciones a desarrollar se tiene el establecimiento de un Catálogo o Portafolio de Productos y Servicios que integre todas las capacidades con las que se cuenta para resolver problemáticas productivas o tecnológicas con el objetivo de propiciar su comercialización.

Figura 5.2 Estilo del cuerpo de las páginas del sistema.

```

h1 {
  font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
  font-weight: normal;
  font-size: 32px;
  color: #6DA6E2;
  margin-bottom: 30px;
  background-color: #FFFFFF;
}

```

Esta sección del código se ocupó para los headers y los menús de las pantallas del sistema.

```

#header {
  width: 750px;
  border: 0px;
  padding-bottom: 10px;
  padding-top: 10px;
  clear: both;
}

#header p {
  margin-top: -20px;
  margin-left: 30px;
  color: #999999;
}

```

Los tabs (figura 5.3) se ocuparon para el formato de los botones y el efecto que tienen al posicionar el mouse sobre ellos, el efecto consiste en oscurecer un poco el color azul del botón, y para el color de la línea que se encuentra debajo de los botones se ocupa una imagen `tableft10.gif` que está alojada en una carpeta adjunta a los archivos del sistema.

```

#tabs10 {
  float:left;
  width:750px;
  font-size:93%;
  padding-bottom: 1px;
  border-bottom:4px solid #2763A5;
  line-height:normal;
}

```

```

#tabs10 ul {

```

```

        margin:0;
        padding:10px 10px 0 50px;
        list-style:none;
    }
    #tabs10 li {
        display:inline;
        margin:0;
        padding:0;
    }

    #tabs10 a {
        float:left;
        background:url('imagenes/tableft10.gif') no-repeat left top;
        margin:0;
        text-decoration:none; padding-left:4px; padding-right:0; padding-top:0;
padding-bottom:0
    }

    #tabs10 a span {
        float:left;
        display:block;
        background:url('imagenes/tabright10.gif') no-repeat right top;
        color:#FFF; padding-left:6px; padding-right:15px; padding-top:5px; padding-
bottom:4px
    }

```



Figura 5.3 Estilo de los botones del sistema

```

#tabs10 a span {
    float:none;
}

#tabs10 a:hover {
    background-position:0% -42px;
}

#tabs10 a:hover span {
    background-position:100% -42px;
}

```

## V.6. Pantallas del sistema

Página principal del sistema o índice del sistema (Figura 5.4).



Figura 5.4 Pantalla principal del sistema.

Presenta cuatro botones:

- **INICIO:** Manda siempre al índice del sistema y aparece en todas las ventas sistema.
- **CONSULTAR:** Muestra una lista de todos los proyectos registrados en el sistema.
- **LOGIN:** Permite a usuarios ya registrados acceder al sistema.
- **REGISTRARSE:** Permite a usuarios no registrados tener cuenta de acceso al sistema.

Índice del perfil ADMINISTRADOR del sistema (Figura 5.5).



Figura 5.5 Index del usuario con rol de administrador.

Presenta seis botones:

- **INICIO:** Manda siempre al índice del usuario ADMINISTRADOR y aparece en todas las vistas del sistema.
- **CONSULTAR:** El administrador podrá realizar las siguientes acciones:
  - Consultar servicio o producto
  - Consultar clientes
  - Consultar proveedores
  - Consultar administradores
  - Consultar interés en productos
  - Consultar productos o servicios vendidos
- **INSERTAR:** Ingresa datos de nuevos administradores.
- **ELIMINAR:** El administrador podrá borrar los siguientes datos:
  - Servicio o producto
  - Cliente
  - Proveedor
  - Administrador
- **ACTUALIZAR:** El usuario podrá actualizar los siguientes datos
  - Datos de clientes
  - Datos proveedores
  - Datos de los administradores
- **SALIR:** Este botón sale del perfil ADMINISTRADOR y manda al índice principal del sistema.

Índice del perfil PROVEEDOR del sistema (Figura 5.6).

**INDEX PROVEEDOR**  
FCC BUAP Bienvenido nestor

**INICIO** **CONSULTAR** **ALTA** **ELIMINAR** **ACTUALIZAR DATOS** **EDITAR SP** **SALIR**

**Como proveedor de servicios**

Podra gestionar la siguiente informacion de:

- Datos referentes a los productos y servicios que tu has ingresado.
- Modificar tus datos personales.
- Consultar los productos y servicios registrados en el sistema

**Ligas de interes**

[Google](#)

[BUAP](#)

[FCC](#)

Valid XHTML 1.0 | Copyright © Your Name Here | Design by super\_j\_man

Figura 5.6 Índice del usuario con rol de proveedor.

Presenta siete botones:

- INICIO: Manda siempre al índice del usuario PROVEEDOR y aparece en todas las ventas sistema.
- CONSULTAR: El usuario podrá consultar solo los proyectos que el haya ingresado al sistema.
- ALTA: El proveedor ingresara, los datos el proyecto.
- ELIMINAR: El usuario podrá eliminar solo los proyectos que el haya ingresado al sistema.
- ACTULIZAR DATOS: EL proveedor podrá actualizar sus datos personales.
- EDITAR SP: EL usuario podrá actualizar los datos de sus productos y/o servicios, previamente almacenados en el sistema.
- SALIR: Este botón sale del perfil PROVEEDOR y manda al índice principal del sistema.

Índice del perfil CLIENTE del sistema (Figura 5.7).



Figura 5.7 Índice del usuario con rol de cliente o contacto.

Presenta siete botones:

- INICIO: Manda siempre al índice del usuario CLIENTE y aparece en todas las ventas sistema.
- CONSULTAR: El cliente podrá consultar los productos y servicios almacenados en el sistema de acuerdo a los siguientes criterios:

- TIPO
  - CATEGORIA
  - AREA FUNCIONAL
  - SECTOR ECONOMICO
  - NOMBRE PROYECTO
  - PROVEEDOR
  - MOSTRAR TODOS PRODUCTOS Y SERVICIOS
- ACTUALIZAR DATOS: EL cliente podrá actualizar sus datos personales.
  - PROYECTOS DE INTERES: El usuario podrá ver los proyectos en los que se ha interesado y ha contactado a los proveedores de esos proyectos.
  - SALIR: Este botón sale del perfil CLIENTE y manda al índice principal del sistema.

## V.7. Reportes

El sistema genera tres reportes esenciales para el portafolio interés en proyectos, responsables y participantes de proyectos.

### Interés en proyectos

Aquí se muestra el interés que ha existido sobre los proyectos existentes en el portafolio, por parte de los usuarios con el perfil contacto, se muestran los siguientes datos (Figura 5.8):

<span>INICIO</span> <span>CONSULTAR</span> <span>INSERTAR</span> <span>ELIMINAR</span> <span>ACTUALIZAR</span> <span>SALIR</span>								
NOMBRE CONTACTO	MAIL CONTACTO	NOMBRE PROVEEDOR	MAIL PROVEEDOR	NOMBRE PROYECTO	DESCRIPCION PROYECTO	CATEGORIA	AREA	TIPO

Figura 5.8 Encabezado del reporte que genera el sistema al buscar los proyectos que han generado un interés de posible compra.

### Responsables de proyectos

Muestra la información de quien es responsables de que producto, esto se realiza ya que a la hora de ingresar un proyecto se solicita que se especifique al usuario proveedor indique cuál es su rol en el proyecto pues puede existir la posibilidad de que sea responsable de uno o más proyectos y que suceda lo mismo si es que solo participara en algunos proyectos (Figura 5.9).

<span>INICIO</span> <span>CONSULTAR</span> <span>INSERTAR</span> <span>ELIMINAR</span> <span>ACTUALIZAR</span> <span>SALIR</span>				
NOMBRE RESPONSABLE	MAIL	PROYECTO	DESCRIPCION	TIPO

Figura 5.9 Encabezado del reporte responsables de proyectos.

## Participantes de proyectos

Muestra la información de quien es participante de que producto, esto se realiza ya que a la hora de ingresar un proyecto se solicita que se especifique al usuario proveedor indique cuál es su rol en el proyecto pues puede existir la posibilidad de que sea participante de uno o más proyectos y que suceda lo mismo si es que fuera responsable de algunos proyectos (Figura 5.10).

INICIO CONSULTAR INSERTAR ELIMINAR ACTUALIZAR SALIR				
NOMBRE PARTICIPANTE	MAIL	PROYECTO	DESCRIPCION	TIPO

Figura 5.10 Encabezado del reporte participantes de proyectos.

# CONCLUSIONES

Al terminar este proyecto se cumplió en su totalidad con el propósito general que consistía en realizar un Prototipo de Sistema de Gestión del Portafolio de Productos y Servicios para el registro y catalogación de los diversos productos y servicios que los miembros de la DESIT-BUAP, así como también de los propósitos particulares la gestión de los datos referentes a los productos y servicios que se pueden ofrecer, gestión de los datos de los universitarios, colaboradores y profesores que proveerán los productos y servicios del Portafolio, gestión de los datos de contacto de los interesados en los proyectos y presentación del listado de productos y servicios del Portafolio de acuerdo a diferentes criterios de búsqueda. Todo esto se realizó siguiendo la Metodología de análisis y diseño estructurado y se implementará el prototipo en un DBMS Relacional.

De los resultados obtenidos, se logró consolidar cada uno de los requerimientos del sistema mediante una aplicación web que puede ser visualizada desde cualquier navegador de internet, ya que el sistema logra administrar los datos de los proyectos que se realizan en la DESIT pudiendo el administrador del sistema así como también el usuario con el rol de proveedor ingresar, eliminar o actualizar los datos de los productos y/o servicios que se desean comercializar, así mismo muestra un listado con los proyectos ya almacenados en el sistema y que pueden ser consultados por los diferentes tipos de criterios que usa el sistema, también se logró administración de los datos personales y de contacto, de los diferentes tipos de usuario administrador, proveedor y cliente, por lo que se logró objetivo que pretendía el portafolio de productos y servicios.

# BIBLIOGRAFÍA.

Johnson, James Lee.

Data base: Models, languages, design. Español

Bases de datos : modelos, lenguajes, diseño / James L. Johnson ; traducción Eduardo Ramírez Grycuk, Jorge Humberto Romo Muñoz

Oxford University Press, 2000.

Hansen, Gary W. (Gary William), 1943-

Database management and design. Español

Diseño y administración de bases de datos / Gary W. Hansen, James V. Hansen ; traducción Miguel Katrib Mora ; coordinación de la traducción y revisión técnica Luis Joyanes Aguilar ; revisión técnica Sebastián Dormido Bencomo.

Madrid : Prentice Hall, 1998, c1997.

Silberschaltz, Abraham.

Fundamentos de bases de datos / Abraham Silberschaltz, Henry F. Korth, S. Sudarshan ; traducción Fernando Sáenz Pérez.

Madrid : McGraw-Hill, 1998.

Stucky, Matthew.

MySQL : building user interfaces / Matthew Stucky.

Indianapolis, Indiana. : New Riders, 2001.

Gilfillan, Ian.

La biblia de MySQL / Ian Gilfillan.

Madrid : Anaya Multimedia, 2003.

Minera, Francisco José.

PHP y MySQL / Francisco José Minera.

Buenos Aires : MP Ediciones, 2005.

Desarrollo web con PHP, Apache y MySQL / Michael Glass ... [et al.].

Madrid : Anaya Multimedia, c2004.

Welling, Luke, 1972-

Desarrollo Web con PHP y MySQL / Luke Welling, Laura Thomson.

Madrid : Anaya Multimedia, 2003.

Murphy, Christopher.

HTML and CSS Web standards solutions [electronic resource] : a Web standardistas' approach / Christopher Murphy, Nicklas Persson.

Berkeley, CA : Friends of ED ; New York : Distributed to the trade by Springer-Verlag, c2009.

# LINKOGRAFIA.

[www.desarrolloweb.com](http://www.desarrolloweb.com)  
[www.aulaclie.es/html/t\\_1\\_1.htm](http://www.aulaclie.es/html/t_1_1.htm)  
[html.conclase.net/tutorial/html/](http://html.conclase.net/tutorial/html/)  
[www.w3.org/Style/Examples/011/firstcss.es.html](http://www.w3.org/Style/Examples/011/firstcss.es.html)  
[www.desarrolloweb.com/php/](http://www.desarrolloweb.com/php/)  
[www.original-design.es/tutosphp.php](http://www.original-design.es/tutosphp.php)  
[www.phpya.com.ar/](http://www.phpya.com.ar/)  
[www.programacion.com/.../tutorial\\_de\\_php\\_y\\_mysql\\_7](http://www.programacion.com/.../tutorial_de_php_y_mysql_7)  
[www.gamarod.com.ar/recursos/tutoriales/php/](http://www.gamarod.com.ar/recursos/tutoriales/php/)  
[www.pablin.com.ar/computer/cursos/phpmysql/index.htm](http://www.pablin.com.ar/computer/cursos/phpmysql/index.htm)  
[mit.ocw.universia.net/curso11208/11/.../lecture5-2.html](http://mit.ocw.universia.net/curso11208/11/.../lecture5-2.html) Compartir  
<http://www.icomputo.com/content/estructura-de-las-bases-de-datos-relacionales-modelo-relacional>  
<http://www.desarrolloweb.com/articulos/estructura-modelo-relacional.html>  
<http://www.inf.udec.cl/~mvaras/estprog/cap3.html>  
[lifedrivedevelopment.wordpress.com/.../normalizacion](http://lifedrivedevelopment.wordpress.com/.../normalizacion)  
[www.slideshare.net/.../normalizacion-base-de-datos](http://www.slideshare.net/.../normalizacion-base-de-datos)