

UNIVERSIDAD DE SONORA

DIVISIÓN DE INGENIERÍA

Departamento de Ingeniería Industrial

Sistema para administrar la adquisición de materiales eléctricos para el Departamento de Programación y Confiabilidad de Bienes de la Comisión Federal de Electricidad, zona Hermosillo.

Reporte de Prácticas Profesionales

PRESENTA:

RUBÉN ANDRÉS BARRAZA GARCÍA

INGENIERO EN SISTEMAS DE INFORMACIÓN

Director: Dr. Guzmán Gerardo Alfonso Sánchez Schmitz

Hermosillo, Sonora. Marzo 2017.

Índice

1. INTRODUCCIÓN.....	1
1.1 BREVE EXPLICACIÓN DEL PROYECTO O ACTIVIDAD.....	2
1.2 OBJETIVO.....	3
2. DESCRIPCIÓN DEL CONTEXTO.....	5
2.1 EQUIPAMIENTO E INSTALACIONES.....	7
2.2 REGLAS DE OPERACIÓN DE LA UNIDAD RECEPTORA.....	9
2.3 ENTORNO DONDE SE UBICA LA UNIDAD RECEPTORA.....	10
3. FUNDAMENTO TEÓRICO DE LAS HERRAMIENTAS Y CONOCIMIENTOS APLICADOS.....	12
3.1 CICLO DE VIDA DEL DESARROLLO DE SOFTWARE (DESARROLLO EN CASCADA).....	12
3.2 HTML Y BOOTSTRAP.....	15
3.2 PHP.....	16
3.3 BASES DE DATOS MySQL.....	17
3.4 JAVASCRIPT.....	18
3.5 JQUERY.....	19
3.6 AJAX.....	20
3.7 CODEIGNITER.....	22
3.8 MODELO VISTA CONTROLADOR.....	22
4. DESCRIPCIÓN DETALLADA DE LAS ACTIVIDADES REALIZADAS.....	24
4.1 PRIMERA ETAPA.....	25
4.2 SEGUNDA ETAPA.....	28
4.3 TERCERA ETAPA.....	31
5. ANÁLISIS DE LA EXPERIENCIA ADQUIRIDA.....	33
5.1 ANÁLISIS GENERAL DEL PROYECTO.....	33
5.2 ANÁLISIS DE LOS OBJETIVOS DE LAS PRÁCTICAS.....	34
5.3 ANÁLISIS DE LAS ACTIVIDADES REALIZADAS.....	34
5.4 ANÁLISIS DE LA METODOLOGÍA UTILIZADA.....	35
6. CONCLUSIONES Y RECOMENDACIONES.....	37
7. REFERENCIAS BIBLIOGRÁFICAS Y VIRTUALES.....	39

Índice de figuras

Figura 2. 1 Ejemplos de proveedores de materiales eléctricos	6
Figura 2. 2 Vista exterior de las Oficinas Divisionales.....	7
Figura 2. 3 Vista superior de las Oficinas Divisionales.....	8
Figura 2. 4 Instalaciones de la UTEC.....	10
Figura 3. 1 Modelo de desarrollo en cascada	13
Figura 3. 2 Tecnologías agrupadas bajo el concepto de AJAX.....	21
Figura 3. 3 Colaboración entre los componentes de un MVC	23
Figura 4. 1 Cronograma de actividades	24
Figura 4. 2 Cronograma con fechas estimadas de inicio a fin.....	25
Figura 4. 3 Plantilla usada en el diseño de la aplicación	28
Figura 4. 4 Muestra del código para crear un proveedor.....	29
Figura 4. 5 Función que usa AJAX para guardar un arreglo en la base de datos	30
Figura 4. 6 Ejemplo de cómo funciona la vista del listado de Proveedores.....	32

1. INTRODUCCIÓN

En la reglamentación de la Universidad de Sonora se tiene contemplado que en todos los planes de estudio se incluyan actividades de vinculación con el sector social o productivo con el propósito de contemplar la formación de los estudiantes a través de la aplicación de los conocimientos obtenidos en las diversas materias de los programas de licenciatura. Así, el programa de Ingeniería en Sistemas de Información, del Departamento de Ingeniería Industrial, incluye en su plan de estudios las prácticas profesionales con un valor de 20 créditos, equivalentes a 340 horas.

Para cumplir con este requisito pedí de manera formal a la empresa del sector público Comisión Federal de Electricidad, ser aceptado para realizar mis prácticas profesionales. Primeramente, me comuniqué con Ivette Salazar; quien está a cargo de encontrar a practicantes de diferentes carreras para cubrir los diferentes departamentos de la CFE. De ahí fui referido con el Ing. Carlos Emilio Acosta, jefe del Departamento de Programación y Confianza de Bienes. Este departamento se encarga de comprar materiales para obras e instalaciones eléctricas usando recursos públicos, para hacer esta labor deben investigar precios con diferentes proveedores, elaborar solicitudes de compra de materiales donde se especifiquen cantidades, precios, etc. Una vez elaborados todos estos oficios, se mandan a otro departamento que los examina y deben ser aprobados antes de realizar la compra.

La idea de este proyecto es digitalizar el proceso de investigación de mercado, creando una aplicación web con una base de datos que contenga toda la información que debe ser tratada. Esta aplicación también será capaz de elaborar los documentos necesarios en PDF para que se envíen rápidamente al departamento correspondiente. Este proyecto surge con la necesidad de agilizar los procesos antes mencionados, pues en la actualidad toda la información manejada se encuentra en hojas de cálculo de Microsoft Excel y los documentos se elaboran manualmente en Microsoft Word.

Además del Ing. Carlos Emilio Acosta, conté con el apoyo del Lic. en Informática, José Urías, quien me explicó con más detalle los objetivos del proyecto y la programación. El

proyecto y las funciones se desarrollaron entre los meses de febrero y marzo de 2017. Durante este periodo de tiempo se estuvieron aplicando diversos conocimientos adquiridos en clase, así como también ciertos conocimientos desarrollados de manera autodidacta.

En el presente documento se encuentran todas las actividades realizadas, metodologías, consultas, etc. que se realizaron para poder cumplir con los objetivos que se acordaron en la institución receptora.

1.1 BREVE EXPLICACIÓN DEL PROYECTO O ACTIVIDAD

El Departamento de Programación y Confiabilidad de Bienes tiene la tarea de hacer investigaciones de mercado para la compra de diferentes materiales para obras e instalaciones eléctricas que se realizan en la ciudad, carreteras y resto del Estado. Algunos ejemplos de estos materiales son postes de concreto reforzados, cableados, antifaunas, etc. (Figura 1.1) Para este proceso se siguen los siguientes pasos:

- Se manda un oficio petición a los diferentes proveedores de materiales eléctricos que existen en la región, indicando los materiales que se desean adquirir y las cantidades necesarias.
- Los proveedores responden con una propuesta, en la que indican el precio unitario de cada material, cantidades disponibles, tamaños y cualquier otra característica que se deba especificar.
- El Departamento de Programación y Confiabilidad de Bienes analiza las propuestas de los proveedores, elige las mejores opciones y las manda al departamento que se encarga de destinar los recursos públicos a la compra de bienes.
- El Departamento de Programación y Confiabilidad de Bienes recibe una respuesta con las órdenes que se han sido aprobadas para su compra.
- Ya con la autorización aprobada, se adquieren los materiales necesarios con el proveedor o los proveedores seleccionados.



Figura 1. 1 Postes PCR, anti faunas y cables

El proyecto con el nombre “Sistema para administrar la adquisición de materiales eléctricos para el Departamento de Programación y Confiabilidad de Bienes” consiste en una aplicación web en la cual se puede consultar la información de todo el catálogo de proveedores, los materiales y servicios que ofrecen. Los usuarios pueden seleccionar de diferentes proveedores los materiales que se necesitan, indicar cantidades, seleccionar los contactos dentro de las empresas proveedoras y así elaborar el oficio petición de una manera más rápida y en formato PDF.

1.2 OBJETIVO

El objetivo principal de este proyecto es agilizar el proceso con el que se consulta la información de los proveedores, los materiales y la elaboración de los oficios petición

que se envían a los proveedores. El proceso actual consume mucho tiempo, pues se debe estar buscando en diferentes archivos de Excel los catálogos de los proveedores y los contactos que hay en cada una de esas empresas. Otros objetivos son los siguientes:

- Creación de un sistema web para la consulta de información y elaboración de documentos oficiales.
- Crear una base de datos con toda la información que se vaya a manejar: proveedores, contactos, empleados de CFE, documentos creados, catálogo de materiales, etc.
- Mejoramiento del proceso actual para realizar las investigaciones de mercado.
- Eliminación del proceso actual y anticuado de realizar esta tarea.
- Creación de plantillas para los oficios de petición que se mandan a proveedores.
- Capacidad para capturar nuevos datos en la base de dato: agregar o eliminar nuevos proveedores, contactos, empleados, materiales, etc.
- Generación automática de los oficios de petición para su impresión y almacenaje.
- Realizar consultas a la base de datos y ver la información de una manera limpia y organizada.

2. DESCRIPCIÓN DEL CONTEXTO

Para la Comisión Federal de Electricidad es de suma importancia tener un excelente control de los materiales que adquieren y de los proveedores con los que tratan los precios de estos materiales, especialmente considerando que usan recursos públicos para adquirirlos y proporcionar un servicio eficaz y de alta calidad a todos los ciudadanos de este país que cuentan con servicio eléctrico. Para ello se cuenta con un control de revisiones de investigaciones de mercado, peticiones de materiales, un catálogo con los proveedores y los materiales que ofrecen, y un presupuesto que deben seguir para adquirir estos bienes y servicios. El área responsable de tratar este tipo de actividades es el Departamento de Programación y Confiabilidad de Bienes.

Para iniciar el proceso de adquisición de nuevos materiales, el Departamento de Programación y Confiabilidad de Bienes determina qué bienes materiales o servicios son lo que necesita adquirir la Comisión Federal de Electricidad para realizar sus obras e instalaciones eléctricas. Este departamento elabora un oficio petición donde se indica los materiales que se necesitan y las cantidades necesarias.

Los proveedores son empresas que se encargan de fabricar o distribuir materiales eléctricos. La CFE tiene uno o varios contactos en cada una de estas empresas. Algunos ejemplos proveedores que trabajan son IUSA y 3M (Figura 2.1). Cada proveedor tiene su propio inventario de materiales disponibles, sus precios, tipos de embalaje. Las diferentes categorías de materiales se dividen en grupos denominados *Familias*; Postes reforzados de concreto (PCR), antifaunas y cables son ejemplos de familias. Es importante conocer qué familias manejan los proveedores para poderlos encontrar más fácil y empezar con el proceso de adquisición más rápido.



Figura 2. 1 Ejemplos de proveedores de materiales eléctricos

Una vez que se elabora el oficio petición indicando las familias de materiales o servicios que se necesitan, el documento se envía a los proveedores que fueron seleccionados. En contacto en la empresa proveedora recibe el documento, lo analiza y consulta en su inventario para saber si podrá suplir con la solicitud de la Comisión Federal de Electricidad. En caso de que se pueda cumplir con la petición, ya sea total o parcialmente, el contacto proveedor responde de vuelta e indica qué materiales puede surtir, el precio unitario, el tipo de embalaje y las cantidades con las que puede cumplir.

Ya teniendo respuesta de los proveedores, el Departamento de Programación y Confiabilidad de Bienes analiza las propuestas de los proveedores. Es importante tener en consideración el precio unitario de los productos que ofrecen, si son productos propios o distribuidos, la calidad de los materiales, duración, resistencia y cualquier otro factor que se deba tomar en cuenta al momento de estar trabajar con estos materiales. Queda en manos de este departamento elegir las mejores opciones y enviar una decisión al Departamento de Compras, quien está a cargo de conseguir los recursos y autorizar la compra de los bienes.

2.1 EQUIPAMIENTO E INSTALACIONES

El Departamento de Programación y Confiabilidad de Bienes de la CFE se encuentra ubicado en la dirección: Benito Juárez y San Luis Potosí C.P. 83000 Col. Centro, Hermosillo, Sonora. (Figura 2.2)



Figura 2. 2 Vista exterior de las Oficinas Divisionales

Esta zona está conformada por los edificios que corresponden a las oficinas divisionales de Distribución Noroeste. (Figura 2.3)

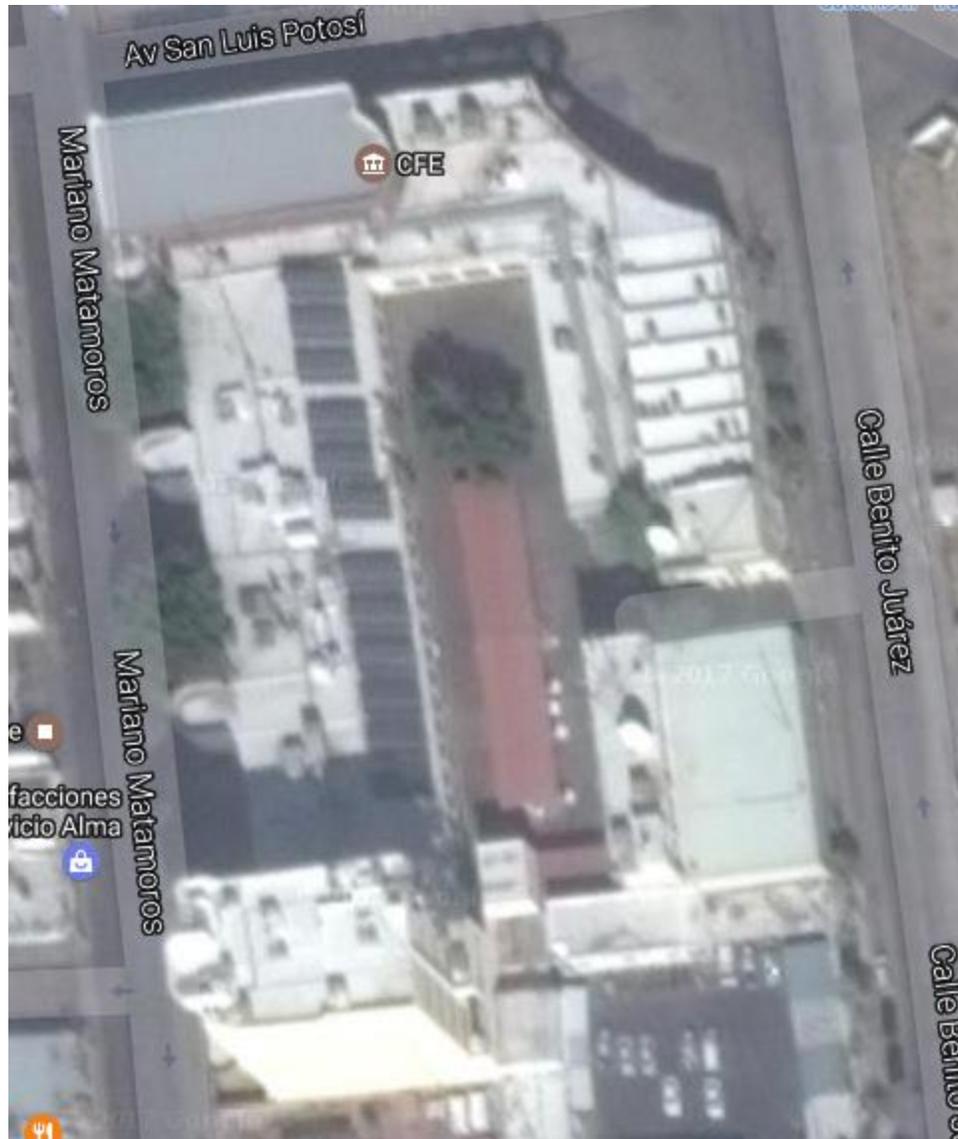


Figura 2. 3 Vista superior de las Oficinas Divisionales

Por otra parte, se dispone de equipo cómputo de distinto tipo, así como herramientas de captura especiales (lectores de códigos de barras, magnéticos, etc.) que se utilizan para operaciones internas. El equipo utilizado en el Departamento de Programación y Confiabilidad de Bienes es básicamente computadoras de escritorio y laptops, impresoras y multifuncionales con que se realizan actividades diarias necesarias. Cuentan con mobiliario de oficina y una flotilla de vehículos destinados para transporte de personal y materiales.

2.2 REGLAS DE OPERACIÓN DE LA UNIDAD RECEPTORA

La Comisión Federal de Electricidad cuenta con una convocatoria permanente para estudiantes de nivel medio superior y superior que deseen presentar tanto Servicio Social como Prácticas Profesionales. Para poder prestar cualquiera de las dos opciones es necesario presentarse en las instalaciones de CFE Oficinas Divisionales con la dirección antes mencionada para conocer los requisitos y el tipo de practicantes que se estén solicitando en ese momento. Los requisitos son: identificación oficial, comprobante de domicilio, constancia que acredite la pertenencia a una institución educativa superior o media superior, así como un comprobante de afiliación al Seguro Social o algún otro seguro médico.

Una vez que se presenta la debida documentación ante Ivette Salazar, la persona encargada de Prácticas Profesionales en la UTEC , se procede a un breve tiempo de espera, tras el cual se es contactado para informar cuál de los proyectos disponibles pudiera ser del interés del solicitante según su área de estudio.

Ya con un proyecto elegido, se fijan los horarios de entrada y salida según se hayan acordado con el tutor de las prácticas de CFE y el alumno. Se asigna un área de trabajo en la cual puede contar con mobiliario y equipo de cómputo según sea el carácter del proyecto a desarrollar. También se le asigna al practicante uno o varios miembros del personal ante los cuales se presentarán las actividades realizadas, avances de las tareas asignadas o las dudas que puedan surgir.

Independientemente de las normas y requisitos administrativos, se sigue un código y reglas de seguridad sumamente rigurosas que buscan asegurar la integridad tanto del personal como de los practicantes y personas que se encuentren dentro y fuera de las instalaciones, esto se debe a que se está en constante presencia de maquinaria pesada, vehículos y diferentes materiales que pudieran causar algún daño o lesión.

2.3 ENTORNO DONDE SE UBICA LA UNIDAD RECEPTORA

La unidad receptora es una dependencia Federal con sede en la colonia Centro en Hermosillo, Sonora. Está ubicada en un sector destinado para este tipo de complejos, ya que dentro de la misma colonia se encuentran oficinas de otras dependencias del gobierno estatal y municipal. Las instalaciones constan de un área de oficinas para el departamento administrativo y dirección, vestidores, baños, sala común, cubículos, salón de eventos y la UTEC (Universidad Tecnológica de la CFE) (Figura 2.4) donde se les otorga capacitación a los empleados para aprender nuevas tecnologías, nuevos temas o procedimientos. El personal del Departamento de Programación y Confiabilidad de Bienes está conformado por su director, encargados de diversas áreas, ingenieros y técnicos, secretarías y personal de apoyo.



Figura 2. 4 Instalaciones de la UTEC

Hablando del entorno se puede decir que tiene una ubicación estratégica en cuanto a distancia de las demás agencias a su cargo y de las instalaciones con las que se cuentan en todo Hermosillo como, por ejemplo, los Centros de Atención al Cliente, los

CFEMáticos, Subestaciones, Almacenes, etc. sin dejar de mencionar que las áreas habitacionales están lo suficientemente alejadas como para no molestar o causar inconvenientes por el constante entrar y salir de camiones o la carga y descarga de materiales.

3. FUNDAMENTO TEÓRICO DE LAS HERRAMIENTAS Y CONOCIMIENTOS APLICADOS

A continuación, se presenta la información que respalda el uso de las tecnologías y metodologías que fueron utilizadas durante el desarrollo del proyecto, con el fin de plantear una base de dónde partir para el sustento de las afirmaciones y conclusiones que se hagan.

Cabe destacar que, al tratarse de un proyecto sin fines de lucro, se utilizó en su totalidad: software, frameworks, plantillas de diseño y otras herramientas completamente gratuitas o de código abierto.

3.1 CICLO DE VIDA DEL DESARROLLO DE SOFTWARE (DESARROLLO EN CASCADA)

Para el desarrollo del proyecto se optó por la utilización de una metodología conocida como “Modelo de Desarrollo en Cascada”. Es una de las más utilizadas, debido a que sigue una secuencia y un orden que permite llevar el proyecto por etapas bien definidas y permite la evaluación de avances y corrección de errores en cada una de las etapas.

El desarrollo en cascada consta de varias etapas o fases, siguen una secuencia bien definida, con metas claras que contribuyen al alcance de los objetivos. (Figura 3.1)

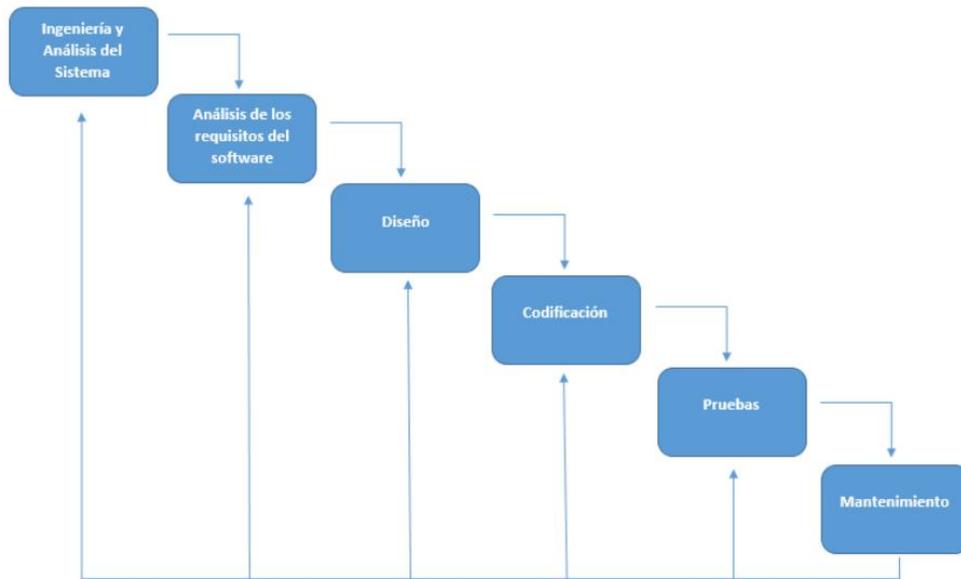


Figura 3. 1 Modelo de desarrollo en cascada

1. **Ingeniería y Análisis del Sistema:** Debido a que el software es siempre parte de un sistema mayor, el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.

Para ello es fundamental conocer a la perfección el funcionamiento del proceso que se está buscando mejorar o automatizar, quienes son los responsables de llevarlo a cabo y que posiblemente serán los usuarios finales, sin dejar de lado las limitaciones que se puedan tener, tales como tiempo, recursos, equipo, etc.

2. **Análisis de los requisitos del software:** el proceso de recopilación de los requisitos se centra e intensifica especialmente en el software. El ingeniero de software debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas. Esto se logra mediante distintas técnicas como los cuestionarios y aplicación de entrevistas con los responsables del desarrollo de los procesos a mejorar, el modelado de escenarios, o la observación de las actividades directamente por parte del desarrollador del software.

- 3. Diseño:** el diseño del software se enfoca en cuatro atributos distintos del programa; la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación. Este es uno de los puntos claves del proceso, ya que es cuando los datos y la información recabada durante las etapas previas se traducen en instrucciones operacionales que se deberán de seguir en adelante. Se tomará en cuenta la plataforma en donde se implemente el software y los requerimientos extras o especiales que puedan surgir por parte de los usuarios finales.
- 4. Codificación:** el diseño debe traducirse en una forma legible para la máquina. Si el diseño se realiza de una manera detallada, la codificación puede realizarse mecánicamente. En esta etapa entran en juego las habilidades que el o los programadores puedan tener, así como la destreza para la utilización del lenguaje anfitrión, obviamente también de la capacidad para comprender el grado de abstracción con que se haya traducido el proceso real a lo digital.
- 5. Prueba:** una vez que se ha generado el código comienza la prueba del programa. La prueba se centra en la lógica interna del software y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren. Las pruebas del código pueden presentar errores de múltiple naturaleza, por lo general se agrupan en tres tipos:

 - I. Léxico:** tiene que ver con la utilización en sí de uno u otro carácter, según el lenguaje utilizado, una palabra puede o no utilizarse, la lengua en que se programe también entra en este apartado (por lo general todos los lenguajes de programación utilizan léxico en inglés).
 - II. Sintaxis:** puede que el léxico utilizado sea el correcto pero el orden en que se estructuran los comandos, palabras, signos, operadores, etc. no sea el correcto, la sintaxis es una serie de reglas que permite al compilador traducir

el alto nivel en que se programa al bajo nivel (binario) en que la computadora interpreta.

III. Semántica: en ocasiones el léxico utilizado es el correcto, la sintaxis no ha presentado error alguno pero el resultado obtenido simplemente no es el esperado. Es donde la semántica entra en juego y ayuda a que lo codificado siga una lógica para obtener resultados lógicos.

6. Mantenimiento: el software sufrirá cambios después de que se entrega al cliente. Los cambios ocurrirán debidos a que se haya encontrado errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos) o a que el cliente requiera ampliaciones funcionales o del rendimiento. Y esta es una de las etapas en que el software adquiere su “forma” final, precisamente los pequeños o grandes cambios que aquí se realicen marcarán la forma en que el cliente vea el producto final y en ocasiones dicta si el software es o no utilizado en realidad.

3.2 HTML Y BOOTSTRAP

HTML son las siglas de Lenguaje de Marcas de HyperTexto y se utiliza en la elaboración de páginas web. Es un estándar que define una estructura básica y un código para la definición de contenido de un sitio o página web.

Mediante la utilización de etiquetas se puede modelar un entorno que los navegadores pueden interpretar para que el usuario visualice lo que se desea mostrar. Es un lenguaje que permite hacer referencia a elementos externos al código en sí, como imágenes, clips, audio, videos, etc. pero estos elementos no se encuentran el código HTML sino sólo se muestran mediante una referencia, lo que hace que este tipo de páginas web sean menos pesadas y rápidas de cargar.

La gran ventaja de la utilización de HTML es que permite combinar su estructura con otros lenguajes o sub-lenguajes (como php, css, javascript, etc.) para crear entornos interactivos y funcionales que permitan al usuario almacenar o consultar archivos o contenido almacenado en el servidor, descargar o subir su propia información, etc. Las

posibilidades son ilimitadas si se tiene el conocimiento necesario y la lógica para combinar las etiquetas de hipertexto con alguno de los otros lenguajes.

Existen varias versiones, la más actual es HTML5 y conserva la estructura básica de sus predecesores, pero incorpora nuevos elementos que aprovechan mejor las cualidades multimedia que hoy en día se tienen al alcance de la mano. También ofrece mayor rendimiento en dispositivos móviles y menos tiempo de desarrollo, entre otras muchas ventajas.

Bootstrap es un proyecto que surgió por el interés de varias personas alrededor del mundo (Ingenieros, programadores, diseñadores, estudiantes, etc.) que se dieron la tarea de crear un framework o plantilla que permitiera el Desarrollo Web Responsivo de una forma rápida y con muy buena estética.

Este Framework consta de un conjunto de paquete que debe de ser descargados desde el sitio oficial, consta de archivos .CSS, .JS, .MAP, .TTF, .WOFF, etc. así como imágenes e iconos, los cuales permiten la maquetación de un sitio web en algunos minutos mediante HTML y el llamado a clases en las etiquetas de los distintos componentes como botones, inputs, tablas, etc. La principal ventaja es que con una sola vez que diseñes el sitio web bastará para que pueda ser visualizado en múltiples dispositivos sin importar la resolución de pantalla o distribución de los elementos en la página, estos se ajustarán según sea necesario.

3.2 PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. En lugar de usar muchos comandos para mostrar HTML (como en C o en Perl), las páginas de PHP contienen HTML con código incrustado que hace "algo". El código de PHP está encerrado entre las etiquetas especiales de comienzo y final `<?php` y `?>` que permiten entrar y salir del "modo PHP".

Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá

el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga. Aunque el desarrollo de PHP está centrado en la programación de scripts del lado del servidor, se puede utilizar para muchas otras cosas.

Básicamente, PHP se encarga de ejecutar HyperTexto que se encuentra incrustado en las etiquetas de HTML, o por separado, en el lado del servidor. Esto quiere decir que el cliente no verá este código pues son sólo instrucciones visibles para el servidor, quien sólo devuelve un resultado visible para el cliente.

3.3 BASES DE DATOS MySQL

Es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también la marca. Inicialmente, MySQL carecía de algunos elementos esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de esto, atrajo a los desarrolladores de páginas web con contenido dinámico por su simplicidad, de tal manera que los elementos faltantes fueron complementados por la vía de las aplicaciones que la utilizan. Poco a poco estos elementos faltantes están siendo incorporados tanto por desarrolladores internos como por desarrolladores de software libre.

En las últimas versiones se pueden destacar las siguientes características principales:

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas: puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.

- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas.

Ventajas de MySQL

- Velocidad al realizar las operaciones, lo que lo hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de sistemas operativos.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Conectividad y seguridad.

Desventajas de MySQL

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo como otros programas (Microsoft Access).

3.4 JAVASCRIPT

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (*client-side*), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

Desde el 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1, una versión de javascript. Los navegadores más antiguos soportan por lo menos ECMAScript 3. La sexta edición se liberó en julio del 2015.

JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX. JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

3.5 JQUERY

jQuery es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. jQuery es la biblioteca de JavaScript más utilizada.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

3.6 AJAX

El término AJAX se presentó por primera vez en el artículo "Ajax: A New Approach to Web Applications" publicado por Jesse James Garrett el 18 de febrero de 2005. Hasta ese momento, no existía un término normalizado que hiciera referencia a un nuevo tipo de aplicación web que estaba apareciendo. En realidad, el término AJAX es un acrónimo de Asynchronous JavaScript + XML, que se puede traducir como "JavaScript asíncrono + XML". El artículo define AJAX de la siguiente forma: *Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes.* Las tecnologías que forman AJAX son:

- XHTML y CSS para crear una presentación basada en estándares.
- DOM para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON para el intercambio y la manipulación de información.
- XMLHttpRequest para el intercambio asíncrono de información.
- Javascript para unir todas las demás tecnologías

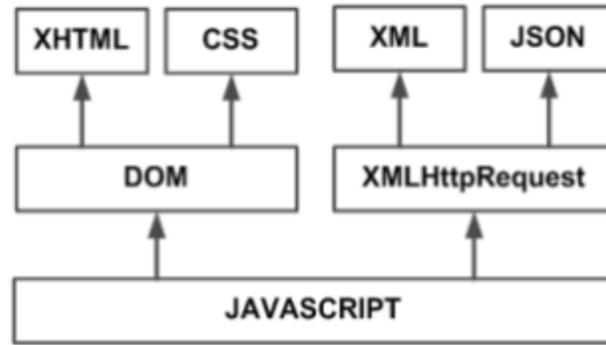


Figura 3. 2 Tecnologías agrupadas bajo el concepto de AJAX

En las aplicaciones web tradicionales, las acciones del usuario en la página (pinchar en un botón, seleccionar un valor de una lista, etc.) desencadenan llamadas al servidor. Una vez procesada la petición del usuario, el servidor devuelve una nueva página HTML al navegador del usuario.

AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano.

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

Aparte de la significativa mejora en la interacción del usuario con el sistema web, se obtiene una disminución de consumo de datos cuando se accede mediante dispositivos móviles, ya que sólo se recarga una sección de la página y no su totalidad, al mismo tiempo que esto incrementa la velocidad de respuesta del servidores mejorando la eficiencia.

3.7 CODEIGNITER

CodeIgniter es un framework para aplicaciones web de código abierto para crear sitios web dinámicos con PHP. “Su objetivo es permitir que los desarrolladores puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero, brindando un conjunto de bibliotecas para tareas comunes, así como una interfaz simple y una estructura lógica para acceder a esas bibliotecas”.

También hay que destacar que CodeIgniter es más rápido que muchos otros entornos. Incluso en una discusión sobre entornos de desarrollo con PHP, Rasmus Lerdorf, el creador de PHP, expresó que le gustaba CodeIgniter “porque es rápido, ligero y parece poco un entorno”.

Kohana ha creado una rama a partir de CodeIgniter 1.5.4 en 2007, como resultado del disgusto de la comunidad por la falta de corrección de errores y el deseo de incorporar otras características demandadas por los usuarios.

3.8 MODELO VISTA CONTROLADOR

Modelo–vista–controlador (MVC) es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador; es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

De manera genérica, los componentes de MVC se podrían definir como sigue:

- El Modelo: Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio).

Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.

- El Controlador: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.
- La Vista: Presenta el 'modelo' (información y *lógica de negocio*) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.

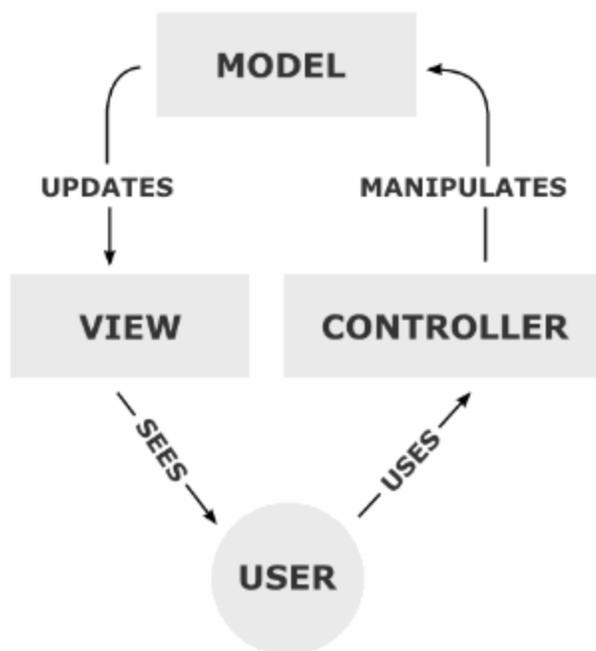


Figura 3. 3 Colaboración entre los componentes de un MVC

4. DESCRIPCIÓN DETALLADA DE LAS ACTIVIDADES REALIZADAS

Cuando se aprobó la solicitud de prácticas profesionales se tuvo una plática acerca del proyecto con el Ing. Carlos Emilio Acosta y el Lic. José Urías. Este último ya tenía empezado un análisis sobre el proyecto, pues tenía pensado desarrollarlo él mismo, pero la falta de tiempo y la carga laboral de su puesto no lo dejaron avanzar. José me explicó la necesidad de crear este sistema web para agilizar el proceso de adquisición de bienes y servicios, que contaría con una base de datos y generación de reportes en PDF.

Ya habiendo establecido las actividades que tendría que realizar, el horario y los días que estaría asistiendo a la CFE, elaboramos un cronograma indicando la duración de las tareas señaladas. Algunos tiempos tuvieron que ser modificados para poder adaptarse a las necesidades y requerimientos que fueron surgiendo durante la marcha; afortunadamente, no hubo ningún cambio considerablemente grande y el proyecto pudo ser terminado en tiempo y forma. El periodo de duración de estas prácticas profesionales abarca desde el 7 de febrero de 2017 hasta el 29 de marzo de 2017, trabajando con un horario de 8 horas diarias de lunes a sábado. Las actividades principales se dividieron en tres etapas de desarrollo con una duración de, aproximadamente, tres semanas cada una.

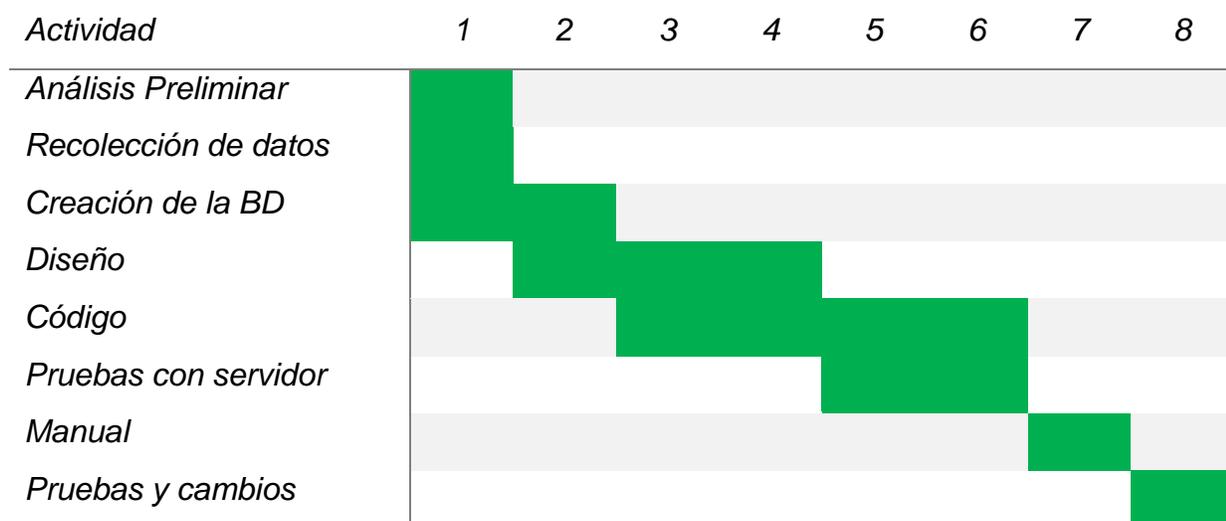


Figura 4. 1 Cronograma de actividades

<i>Actividad</i>	<i>Fecha de Inicio</i>	<i>Duración (en semanas)</i>	<i>Fecha de terminación</i>
<i>Análisis preliminar</i>	07/02/17	1	11/02/17
<i>Recolección de datos</i>	07/02/17	1	11/02/17
<i>Creación de la BD</i>	07/02/17	2	18/02/17
<i>Diseño</i>	13/02/17	3	04/03/17
<i>Código</i>	20/02/17	4	18/03/17
<i>Pruebas con servidor</i>	06/03/17	2	18/03/17
<i>Manual</i>	20/03/17	1	25/03/17
<i>Pruebas y cambios</i>	27/03/17	1	29/03/17

Figura 4. 2 Cronograma con fechas estimadas de inicio a fin

Las actividades mencionadas abarcan el ciclo de vida de desarrollo de software, donde se puede encontrar todo lo necesario para la creación de un producto de calidad y se enfoca en todos los aspectos que se deben tomar en cuenta desde el comienzo, como el análisis y acercamiento con el cliente, la implementación, retroalimentación y finalizando con el mantenimiento del producto para así cerrar el círculo.

4.1 PRIMERA ETAPA

La primera actividad que se realizó fue el acercamiento con los encargados del Departamento de Programación y Confiabilidad de Bienes, ya que ellos eran quienes estaban solicitando practicantes para desarrollar una solución a la situación actual que se presentaba en ese entonces de manera cotidiana. Se sostuvieron una serie de reuniones en las cuales se buscó comprender a profundidad cómo se estaba trabajando en ese momento y hacia dónde querían dirigir sus esfuerzos en el corto plazo. De estas pláticas y reuniones se fueron visualizando puntos clave que posteriormente serían utilizados en el desarrollo del proyecto.

Los puntos clave que antes mencionados fueron analizados desde la perspectiva de un Ingeniero de Software, es decir, se buscaba establecer una metodología que permitiera el establecer parámetros para determinar si lo que se propondría era lo indicado, si lo

que se hacía estaba cumpliendo con su propósito, si lo entendido había sido correcto, si se necesita hacer cambios en las notas, etc.

Como ya se mencionó anteriormente, el proceso de análisis del proyecto ya había sido comenzado por el Lic. en Informática José Urías, quien ya tenía una estructura para la base de datos con la que contaría la aplicación web. Estos fueron algunos puntos que describen con qué recursos se trabaja:

1. Actualmente el proceso de recolección de datos, almacenamiento, procesamiento y presentación de la información consume demasiado tiempo y requiere de la participación de varios empleados.
2. Actualmente se utilizan varias herramientas de ofimática como Word, Excel, PowerPoint, archivos PDF, imágenes, etc. para llevar a cabo el proceso de adquisición y cotización de bienes y servicios, esto se hace en varias computadoras con distintos empleados lo que afecta la integridad de información lleva a la pérdida de la misma o la omisión de detalles importantes en los reportes.
3. El proceso actual necesita realizarse por personal que haya sido capacitado previamente y conozca los aspectos a revisar en cada uno de los oficios y cotizaciones que se elaboran.

Estos puntos fueron los de mayor importancia y relevancia que se derivaron de las juntas de trabajo. Gracias a la detección de estos puntos clave se confirmó la propuesta final que se realizaría para resolver la problemática encontrada, fue el desarrollo de un Sistema Web para la Adquisición y Cotización de Bienes, este proyecto fue aceptado y se procedió a llevarlo a cabo.

Con el análisis hecho por José Urías, ya contábamos con el diseño de una base de datos (BD) que permite almacenar, organizar y consultar los datos e información necesarios para los trámites que se deben hacer en el departamento.

Para este paso decidimos usar software libre, como WampServer versión 2.2 que consta de un manejador de base de datos MySQL versión 5.6.17, un Servidor local

Apache versión 2.2.22 y un administrador de PHP versión 5.4.3. La BD se diseñó en base a los requerimientos detectados y la información que cada uno de las consultas que se necesitan hacer y los documentos que se necesitan crear con esa información.

La base de datos consta de un total de 19 tablas en las que se almacenan desde los empleados de CFE, proveedores y su categoría, familias, tipos de embalaje, los diferentes oficios que se elaboran, etc.

La BD obtenida cuenta con distintas tablas, algunas son catálogos fijos a los cuales sólo se hace referencia, mientras que otras tablas son de relación y guardan los ID de otras tablas para evitar redundancias innecesarias. Se utilizaron llaves únicas para evitar el registro de elementos repetidos. El propósito de diseñar una BD como esta es el de asegurar la integridad de datos y brindar rápido acceso a la información cuando se le necesite.

La última actividad importante que se realizó en esta etapa fue la de selección de una plantilla o tema en Bootstrap. El diseño responsivo era indispensable para lograr una buena interacción cliente-sistema y una interface amigable e intuitiva, esto se puede lograr utilizando un Framework como Bootstrap. Tomando en cuenta que el tiempo del que se disponía para la realización del proyecto era limitado se optó por la utilización de una plantilla gratuita que contara con diseño responsivo y pudiera ser adaptada a las necesidades que se tuvieran.

Se eligió el tema AdminLTE 2 versión 2.3 de Almsaeed Studio7, este tema es gratuito y cumplía con todos los requerimientos previstos para el Desarrollo del Sistema de revisiones. Una vez seleccionado se procedió al estudio de su documentación para conocer el funcionamiento interno y cómo es que se hacen los cambios necesarios para adecuarlo a las necesidades específicas. Este tema trabaja con CSS y JavaScript los cuales le otorgan la mayoría de sus funcionalidades y le da el aspecto estilizado y atractivo, aunado al diseño responsivo antes mencionado.

El primer paso del rediseño del tema fue el cambiar el lenguaje de todos los componentes y archivos del inglés al español, la tarea aquí se complica un poco ya que debes de hacer recorrido a lo largo de todos los módulos que se necesitarán del tema y ver cuándo es que se hace llamado a un archivo externo para ir a ese archivo y buscar

entre miles de líneas de código lo que se debe de modificar, para hacer esto tardé un poco, ya que debí comprender la lógica que se utiliza y los recorridos que se hacen al momento de usar JavaScript. Lo bueno de esto es que al utilizar HTML y este ser un lenguaje de etiquetas que hacen referencia a archivos en lugar de contenerlos en cada una de las páginas, cuando encontraba y modificaba uno de los ficheros automáticamente se cambiaba en el resto de las páginas. Finalmente se creó una plantilla propia que tomaba todas las ventajas del tema, pero adaptado al lenguaje y necesidades particulares. A continuación, una muestra de la plantilla obtenida. (Figura 3.6)

The screenshot displays a web application interface for managing suppliers. On the left is a dark sidebar with a user profile (John Doe) and a menu under 'Investigación de Mercado' and 'Catálogos'. The main content area is titled 'Proveedores' and features a search bar, a filter dropdown set to 'Todos', and a table with the following data:

Razón Social	Contacto	Teléfono Fijo	Teléfono Móvil	Correo Electrónico	Tipo	
00001	00001	1	1	00001	B	
00002	00002	2	2	00002	B	
00003	00003	3	3	00003	B	
00004	00004	4	4	00004	B	
00005	00005	5	5	00005	B	
00006	00006	6	6	00006	B	

Figura 4. 3 Plantilla usada en el diseño de la aplicación

Con la obtención de esta plantilla se concluyó la primera etapa del proyecto y fue presentada para recibir el visto bueno y retroalimentación de parte de los usuarios finales, los cuales quedaron muy satisfechos con el diseño elegido y la plataforma utilizada para su despliegue.

4.2 SEGUNDA ETAPA

Una vez que se contaba con la estructura del diseño bien definida, se procedió a dale funcionalidad, para ello se comenzó por cambiar las extensiones de las páginas que componen los distintos módulos de .html a .php y se ubicaron en una carpeta especial

para que pudieran ser accedidos desde el localhost del Servidor Local (WampServer) instalado en la computadora en donde se estaba desarrollando el proyecto. La combinación de la estructura de diseño en HTML con “toques” de código en PHP brindan un sin fin de posibilidades y les dan funcionalidad a las páginas. Se logra pasar de módulos estáticos a módulos completamente dinámicos. Este paso fue fundamental para implementar lo realizado en la siguiente actividad que era la conexión con la base de datos y creación del CRUD (operaciones básicas de Crear, Leer, Actualizar y Borrar) con una combinación de HTML, PHP, JavaScript y MySQL. A continuación, se muestra un ejemplo del código de creado creado es el siguiente (Figura 3.7).

```

add.php  Proveedor.php x  controllerComboBoxes.php  index.php  Proveedormodel.php  config.php
36      * Adding a new proveedor
37      */
38      function add()
39      {
40          $this->load->library('form_validation');
41
42          $this->form_validation->set_rules('clave','Clave','max_length[15]|required');
43          $this->form_validation->set_rules('rfc','Rfc','max_length[15]|required');
44          $this->form_validation->set_rules('razonSocial','RazonSocial','max_length[150]|required');
45          $this->form_validation->set_rules('direccion','Direccion','max_length[150]|required');
46          $this->form_validation->set_rules('codigoPostal','CodigoPostal','exact_length[5]|required');
47          $this->form_validation->set_rules('idMunicipio','IdMunicipio','required');
48          $this->form_validation->set_rules('nombre1','Nombre1','max_length[100]|required');
49          $this->form_validation->set_rules('direccion1','Direccion1','max_length[150]|required');
50          $this->form_validation->set_rules('codigoPostal1','CodigoPostal1','exact_length[5]|required');
51          $this->form_validation->set_rules('idMunicipio1','IdMunicipio1','required');
52          $this->form_validation->set_rules('telefonoFijo1','TelefonoFijo1','max_length[11]|required');
53          $this->form_validation->set_rules('telefonoMovil1','TelefonoMovil1','max_length[11]|required');
54          $this->form_validation->set_rules('correoElectronico1','CorreoElectronico1','max_length[100]|required');
55          $this->form_validation->set_rules('extension1','Extension1','max_length[11]|required');
56          $this->form_validation->set_rules('nombre2','Nombre2','max_length[100]');
57          $this->form_validation->set_rules('direccion2','Direccion2','max_length[150]');
58          $this->form_validation->set_rules('idMunicipio2','IdMunicipio2');
59          $this->form_validation->set_rules('codigoPostal2','CodigoPostal2','exact_length[5]');
60          $this->form_validation->set_rules('telefonoFijo2','TelefonoFijo2','max_length[11]');
61          $this->form_validation->set_rules('telefonoMovil2','TelefonoMovil2','max_length[11]');
62          $this->form_validation->set_rules('correoElectronico2','CorreoElectronico2','max_length[100]');
63          $this->form_validation->set_rules('extension2','Extension2','max_length[11]');
64          $this->form_validation->set_rules('nombre3','Nombre3','max_length[100]');
65          $this->form_validation->set_rules('direccion3','Direccion3','max_length[150]');
66          $this->form_validation->set_rules('idMunicipio3','IdMunicipio3');
67          $this->form_validation->set_rules('codigoPostal3','CodigoPostal3','exact_length[5]');
68          $this->form_validation->set_rules('telefonoFijo3','TelefonoFijo3','max_length[11]');
69          $this->form_validation->set_rules('telefonoMovil3','TelefonoMovil3','max_length[11]');
70          $this->form_validation->set_rules('correoElectronico3','CorreoElectronico3','max_length[100]');
71          $this->form_validation->set_rules('extension3','Extension3','max_length[11]');
72          $this->form_validation->set_rules('estatus','Estatus','required');

```

Figura 4. 4 Muestra del código para crear un proveedor

En el anterior ejemplo se muestra cómo es que se hacen los CREATE en la BD en la tabla “proveedor”. Así como este, existen otros archivos que se encargan de los READ, UPDATE y DELETE de los distintos módulos del Sistema. De ahí se obtuvieron

métodos que con el simple hecho de mandarle parámetros distintos permitían funcionar para varios propósitos. Alguno de los métodos creados fueron el de crearRelación(), que se encarga de crear una relación entre el proveedor y las familias de materiales que maneja; la función agarra ambos valores y los inserta en la tabla “relaciónproveedorfamilia”.

Una vez que los métodos para la Base de Datos fueron codificados se procedió con las pruebas de integridad de datos y validación de los parámetros enviados, esto a través de recorridos y pruebas individuales que consistían en hacer inserciones y llamadas a la base de datos desde el Sistema y corroborar los resultados obtenidos directamente con la consola de MySQL.

Se crearon Scripts que en conjunto con los métodos del CRUD de la BD permiten al usuario hacer las llamadas e interacciones con la BD a través de AJAX. Por ejemplo, al agregar un nuevo usuario se muestra un mensaje de Éxito o Error seguido de la recarga de sólo la tabla donde se muestran a los usuarios, las demás partes del módulo quedan estáticas. Otro de los ejemplos de la implementación de AJAX ocurre cuando se utilizan los distintos Combos de Selección (Select en HTML) los cuales se cargan dinámicamente según los elementos seleccionados previamente. (Figura 3.8)

```
$("#botonGuardar").click(function(){
    var seleccion = $("#listaSeleccion li");
    var familias_seleccion = [];

    seleccion.each(function() {
        familias_seleccion.push($(this).text().replace(/Quitar/, ''));
    });
    $.ajax({
        url: '<?php echo base_url();?>index.php/Proveedor/crearRelacion',
        method: 'POST',
        data: {familias_seleccion: familias_seleccion}
    });
});
```

Figura 4. 5 Función que usa AJAX para guardar un arreglo en la base de datos

La detección y corrección de errores en este punto fue fundamental, cada módulo que se iba avanzando o terminando era probado varias veces hasta asegurar su correcto

funcionamiento. En caso de que un error era detectado se procedía a hacer un recorrido desde el inicio de la secuencia hasta llegar al punto del error y terminar con el fin de la secuencia. Esto permitió que la detección de errores fuera un tanto rápida al igual que la corrección de los mismos. No obstante, cuando los errores se encontraban en los archivos .js era un poco más complicado de resolver pues independientemente de tener que revisar el código en el lenguaje anfitrión (HTML y PHP) se tenía que ir al archivo en cuestión y realizar una segunda búsqueda, saber si los parámetros llegaban y si eran los correctos, ver que era lo que se devolvía una vez procesados los parámetros y demás cuestiones por el estilo que representaron un pequeño reto a lo largo del desarrollo.

4.3 TERCERA ETAPA

Una vez que las sesiones fueron probadas y se aseguró su correcto funcionamiento se procedió a realizar una presentación de avances ante el Jefe del Departamento. Esto con el fin de obtener retroalimentación pues ya estaba por concluir el tiempo que se había calculado para el proyecto. Se presentó una sección de los módulos que ya era completamente funcional y se realizaron pruebas tanto de mi parte como por los usuarios finales del sistema, detectando faltas ortográficas que fueron corregidas en el mismo momento. En general la respuesta de los usuarios y el entusiasmo por finalmente contar con esta nueva herramienta fue muy positiva y las reacciones de empatía y agradecimiento no se hicieron esperar.

Razón Social	Contacto	Teléfono Fijo	Teléfono Móvil	Correo Electrónico	Tipo	
00001	00001	1	1	00001	B	 
00002	00002	2	2	00002	B	 
00003	00003	3	3	00003	B	 
00004	00004	4	4	00004	B	 
00005	00005	5	5	00005	B	 
00006	00006	6	6	00006	B	 

Figura 4. 6 Ejemplo de cómo funciona la vista del listado de Proveedores

Una vez concluida la presentación de avances, siguió la parte de corrección y ajustes en la Base de Datos debido a que se necesitó reajustar las longitudes de algunos de los campos de varias tablas. También se hicieron algunos otros ajustes visuales y la etapa de codificación se dio por concluida. Rápidamente me di a la tarea de crear el manual de usuario que se debería de entregar en conjunto con el código del sistema y su base de datos correspondiente. En este manual se describe a detalle cada uno de los módulos con que el Sistema de revisión de Edificios cuenta, así como los posibles errores o advertencias que los usuarios pueden recibir del mismo.

5. ANÁLISIS DE LA EXPERIENCIA ADQUIRIDA

Es necesario invertir el tiempo y dedicación indicada en cada uno de los procedimientos para poder concluir con un trabajo de calidad, no es solamente poner código y trabajar de vez en cuando, por algo todo lleva un orden el cual tienes que seguir al pie de la letra y siempre darle la importancia adecuada. Ser un programador no es únicamente conocer los distintos tipos de lenguajes y tener el conocimiento de las herramientas informáticas y software. Un Ingeniero en Sistemas de Información también lleva mucho el trato con el cliente y la comunicación es un tema básico, la investigación y el autoaprendizaje son cosas que van de la mano pues con ellas creces tanto como profesionista y persona.

También me dio gusto adentrarme más al campo de la programación, que es un área que me ha llamado la atención y no practicamos mucho en la carrera. Trabajé con herramientas que no conocía mucho como PHP, Javascript y AJAX; que todavía son tecnologías ampliamente utilizadas para este tipo de aplicaciones.

Uno de los desafíos más grande que tuve en este proyecto fue familiarizarme con la arquitectura MVC, pues Codeigniter está pensado para funcionar bajo esa arquitectura. Las primeras semanas tenía muchas dudas sobre cómo hacer las consultas a la base de datos, cómo usar las variables, etc. pero al final me fui familiarizando y me alegra haber aprendido algo nuevo.

El hecho de que este proyecto se inclinara hacia este tipo de tecnologías fue de gran ayuda para complementar conocimientos básicos con aspectos más específicos y que en conjunto hacen que se puedan lograr grandes cosas.

5.1 ANÁLISIS GENERAL DEL PROYECTO

En las primeras visitas que se tuvo con el Jefe del Departamento de Programación y Confiabilidad de Bienes de CFE Oficinas Divisionales, se hablaba de fomentar el uso de herramientas web para facilitar el trabajo de revisiones y así poder reducir tiempos y hacer el trabajo mucho más preciso, también se habló del control que habría con el sistema, por el hecho de quienes fuesen los que pudieran tener acceso.

La idea estaba planteada desde hace años, pero nunca había podido hacerse realidad por la falta de tiempo por parte del personal que podía hacer el desarrollo. Una vez concordada la responsabilidad de la elaboración del sistema web, se acordó la cooperación por parte del Departamento de Programación y Confiabilidad de Bienes para la obtención de información sobre los procesos involucrados.

5.2 ANÁLISIS DE LOS OBJETIVOS DE LAS PRÁCTICAS

Los objetivos siempre estuvieron muy presentes, siendo el objetivo en general modificar la forma de trabajo de una manera muy radical, todo esto mediante el uso de las tecnologías en este caso, mediante un sistema web. Se trabajó el almacenaje de información, procesamiento, orden, consulta e impresión de los reportes necesarios para el obtener los bienes materiales y servicios que la empresa necesita.

Otro objetivo era tener rápidamente a la mano toda la información de las empresas proveedores y los contactos que se tienen en cada una de ellas, toda esta información se puede consultar rápidamente en la aplicación y además, se pueden aplicar filtros para encontrar la información más fácil y rápido.

5.3 ANÁLISIS DE LAS ACTIVIDADES REALIZADAS

Comenzando con la captura de información, fue un proceso bastante largo pues era demasiada información la que debía ser capturada. Esta llevaba una secuencia y orden pues si no se seguía habría conflictos en el momento o en un futuro.

La carga de información en la Base de Datos es donde mayor cuidado se tenía que tener ya que las tuplas de las distintas tablas estaban referenciadas unas a otras de tal manera que el simple cambio en una de ellas desencadenaría problemas en las demás y la integridad de datos es una de las partes que más se cuidaron en este proyecto, no sólo por ser lo correcto sino que se necesitaba para que el sistema web de verdad

marcara una diferencia en la forma en que se está trabajando en el Departamento de Programación y Confiabilidad de Bienes.

Si bien las actividades fueron programadas en un cronograma, no siempre se pudieron realizar en ese orden o en el tiempo deseado, a veces fue antes pero también hubo ocasiones en que se tuvo que replantear completamente la estrategia que se estaba siguiendo y rediseñar las actividades programadas como consecuencia. No quiere decir que siempre se estuvieron cambiando, más bien debemos de mostrarnos abiertos a nuevas formas de trabajar para ser flexibles y adaptarnos a los inconvenientes que se pudieran presentar.

Debo de destacar que estas actividades, todas y cada una de ellas, no sólo se realizaban y terminaban en el tiempo y fechas laborales que las prácticas profesionales muestran, en ocasiones las ideas para resolver algún problema que una de las actividades planteaba llegaba en un momento inesperado, tal vez en la noche, un domingo, etc. el punto es que no puedes dejar por un lado el hecho de que se necesita verdadero compromiso cuando se está realizando un trabajo el cual sabes que importa y puede marcar una diferencia.

5.4 ANÁLISIS DE LA METODOLOGÍA UTILIZADA

Aplicar los conocimientos adquiridos durante la carrera fue un aspecto de suma importancia durante la realización de las prácticas profesionales, tanto de manera teórica como práctica, esto sentó las bases para poder ofrecer un trabajo de calidad y que cumpliera con los objetivos y expectativas que fueron planteadas desde un principio.

Desde un principio fueron puestas en práctica estrategias y mecanismos que, si bien se nos enseñan en la carrera, es muy importante el ser autodidactas para contar con información actualizada y poder utilizar las tecnologías y herramientas más actuales a nuestra disposición. Un ejemplo de estas estrategias, y uno de las más básicas como desarrollador de software, es el simple hecho de sentarse con los involucrados, las personas que realizan el trabajo que se quiere mejorar mediante la implementación de

una tecnología, los usuarios finales que conocen a profundidad los procesos y que son quienes en un momento dado utilizarán lo que se llegue a desarrollar, y una vez reunidos comenzar una de tantas pláticas que serán necesarias.

Hablando del aspecto técnico que involucra tanto los distintos lenguajes de programación utilizados, frameworks, software de desarrollo, etc. el esfuerzo se enfocó en el desarrollo rápido, tomando en cuenta el poco tiempo disponible. Se optó por una metodología de desarrollo de software llamada Modelo en Cascada, la cual permite utilizar un principio básico de la programación que es “Divide y Vencerás”, esto se refiere a que las actividades no se deben de ver como una gran y compleja red de problemas a solucionar, sino descomponer esa compleja red en módulos menos complejos y estos a su vez en tareas menos complejas y así sucesivamente hasta que se logran crear pequeñas actividades e ir solucionándolas para que al final todas trabajen y funcionen juntas para solucionar el problema inicial. El modelo en Cascada plantea 6 etapas por lo general antes descrito), las cuales contuvieron una serie de actividades que se realizaron una a la vez, desarrollarlas, probarlas, hacer que trabajaran en conjunto hasta que el proceso de desarrollo concluyó.

6. CONCLUSIONES Y RECOMENDACIONES

El proyecto se finalizó en tiempo y forma el código del Sistema Web y la Base de Datos necesaria para su funcionamiento, con algunos cambios que se hicieron a última hora debido a que el Departamento de Programación y Confiabilidad de Bienes optó por un sistema local por el momento debido a que cualquier gasto debe de ser presupuestado y en este caso los gastos involucrados son el costo de dominio y host (pago anual y mensual, respectivamente) que un sistema web acarrea consigo.

Se logró equipar una computadora de escritorio y una portátil con el Sistema de Revisión de Edificios, las cuales servirán como piloto para que sea implementado dentro del departamento. En un futuro cercano, se planea subir el sistema a un servidor privado o uno propio de CFE, según se llegue a un acuerdo con el área de informática, para que sea operativo al 100%.

Uno de los factores que más obstáculos representó fue el hecho de la comunicación para obtención de información específica. Se llegaron a presentar retrasos en el cronograma debido a que los archivos o información solicitados eran proporcionados por una persona en particular y no se podía contactar o estaba fuera de las instalaciones. Esto claro no pasó de una simple plática para exponer lo que sucedía y se corrigió el problema.

Otro punto relevante durante la realización de este proyecto es que en ocasiones tachamos de irrelevante o repetitivo, en realidad es de mucha ayuda, al igual que la experiencia previa que se pudiera tener. No se debe de menospreciar ninguna de las actividades o experiencias que tendremos a lo largo de nuestra vida como estudiantes o como profesionistas ya que en conjunto la suma de todas ellas se refleja en la calidad del trabajo que es presentado por el Ingeniero en Sistemas de Información en este caso.

Está claro que el proceso de Prácticas Profesionales es una gran experiencia como alumno y como persona, pues nos podemos dar cuenta de muchas de la vida laboral ya real aun siendo alumnos, te das cuenta también de muchas cosas que en la escuela o en ningún lado las aprendes más que en la práctica.

Ahora bien, según lo observado durante este proceso de casi 2 meses en que se tuvo la oportunidad de conocer cómo es que se trabaja en el Departamento de Programación y Confiabilidad de Bienes, se presentan adelante algunas recomendaciones:

- Buscar la manera de que el sistema web se aloje en un servidor para que pueda prestar sus servicios a cualquier otra agencia dependientes de la Zona Hermosillo en el menor tiempo posible.
- Implementar una aplicación móvil para consultar la información necesaria desde un celular o cualquier otro dispositivo móvil.
- El hecho de que se estén acercando a las soluciones que las Tecnologías de la Información (TI) les ofrecen para mejorar sus procesos es un buen comienzo. Recomendaría que se acerque a las instituciones educativas como la Universidad de Sonora donde existen instancias que los pueden ayudar en sus intentos de utilizar las TI y así llegar a acuerdos que beneficien a ambas partes.
- Los proyectos que se tiene pensados comenzar deben de llevarse a la realidad ya que son basten buenos y no requieren de grandes inversiones o desembolsos de dinero, en cambio, representarían grandes beneficios para sus Departamentos y cambiarían la forma en que se realizan los trabajos.
- La capacitación del personal en cualquier nueva tecnología siempre es necesaria y no debe de verse como un gasto, es siempre una inversión que toda empresa o institución importante debe de hacer, ya que le permite tener personal listo para enfrentar los retos que se presentarán sin duda alguna en el corto plazo, es mejor estar preparados y no esperar a que los problemas surjan para arreglarlos.

7. REFERENCIAS BIBLIOGRÁFICAS Y VIRTUALES

Rosenberg, D.; Scott, K. (2005). Ingeniería del software. Cataluña, España: Editorial UOC. (Página 20 en adelante).

Ble, Carlos. (2010). Desarrollo en Cascada, de LibrosWeb Sitio web: http://librosweb.es/libro/tdd/capitulo_1/modelo_en_cascada.html

Pavón, J. (2013). Bootstrap 3.0, de Universidad Complutense Madrid Sitio web: <https://www.fdi.ucm.es/profesor/jpavon/web/26-Bootstrap.pdf>

Marino, J. (2009). Manual de AJAX: Las Entrañas de AJAX. Sitio web: <http://www.uco.es/~lr1maalm/manualdeajax.pdf>

Manuales de PHP: <http://php.net/manual/es/>

Manuales de Administrador de Base de Datos: <http://dev.mysql.com/doc/refman/5.7/en/connecting-disconnecting.html>

Manuales de AJAX: <http://api.jquery.com/jquery.ajax/>