



Manuel Alejandro Vázquez Bautista

Proyecto: Geoposicionamiento

Universidad de Sonora

Dpto. Ingeniería Industrial y de Sistemas

Ingeniería en Sistemas de Información

Centro de Servicios de Tecnologías de Información

ÍNDICE

1. Introducción.....	3
2. Administración de usuarios (Log In)	3
2.1. CSTI	4
3. Justificación.....	4
4. Objetivos	5
5. Actividades a realizar.....	5
5.1. Realizar análisis de los requisitos del sistema inteligente de geo- posicionamiento.....	5
5.2. Geo-posicionar objetos dentro de un mapa de Google Maps.....	6
5.3. Diseñar tablas necesarias para la gestión de información de ubicaciones.....	6
5.4. Diseñar la interfaz del sitio web de geo-posicionamiento	6
5.5. Administración de Usuarios y Empresas (LOGIN).....	6
5.6. Desarrollar módulo de búsqueda personalizada (Filtros).....	7
6. Alcances y limitaciones.....	7
7. Fundamento teórico de las herramientas y conocimientos aplicados.....	8
7.1. HTML	8
7.2. PHP	8
7.3. SQL.....	9
7.4. MySQL	9
7.5. WAMP	9
7.6. Php Sessions.....	10
7.7. Session_destroy()	10
7.8. Session_name().....	10
7.9. Sentencia If – Else.....	10
7.10. Formularios HTML	10
7.11. Etiqueta FORM.....	10
8. Procedimientos empleados y actividades desarrolladas.....	12
8.1. Index.....	12
8.2. Login usuario.....	15
8.3. Logout.....	15
8.4. Registrar Usuario	16
8.5. Registro de empresa	18
8.6. Validar login.....	22
8.7. Clase Usuarios.....	24
8.9. Autocompletar.....	26
8.10. Modificar Usuario	28
9. Resultados obtenidos	36
10. Conclusiones y recomendaciones	37
11. Bibliografía.....	39

1. Introducción

En el siguiente documento se hablará sobre las características que conlleva el proyecto de geoposicionamiento, asignado por el dpto. de CSTI de la universidad de sonora, el impacto que tendrá, así como las diferentes problemáticas y como se éstas se fueron solucionando, con énfasis en el aspecto técnico, se hablará de los métodos y herramientas utilizadas para el desarrollo de dicho proyecto.

Según la enciclopedia virtual Wikipedia, tenemos el concepto de geoposicionamiento como, al posicionamiento con el que se define la localización de un objeto espacial (representado mediante punto, vector, área, volumen) en un sistema de coordenadas y datum determinado. Este proceso es utilizado frecuentemente en los Sistemas de Información Geográfica. Este concepto será aplicado a las tecnologías web con el fin de facilitar a los usuarios a encontrar diferentes tipos de negocios o establecimientos que se encuentran en su cercanía para tener consumir productos o servicios de una manera más eficiente y práctica.

2. Administración de usuarios (Log In)

Es importante que cada sitio web o aplicación tenga un control sobre el uso y control de los datos. Esto ayudara a mantener segura la aplicación y poder hacer estadísticas sobre futuras mejoras dependiendo de las opiniones de los usuarios finales. El Log in nos hace apoyo en este aspecto de la aplicación, no solamente controla el flujo de los usuarios de la aplicación, también brinda un uso personalizado a los usuarios para que estos mismos creen un perfil el cual serán visto por las empresas o por los mismos usuarios.

En el ámbito de seguridad informática, login o logon (en español ingresar o entrar) es el proceso mediante el cual se controla el acceso individual a un sistema informático mediante la identificación del usuario utilizando credenciales provistas por el usuario.

Un usuario se puede log in a un sistema para obtener acceso y se puede log out o log off (en español salir o desconectar) cuando no se precisa mantener el acceso. Log out consiste en cerrar el acceso personal a un sistema informático, al cual anteriormente se había realizado el login.

2.1. CSTI

Es un equipo de trabajo situado en el departamento de ingeniería industrial y de sistemas de la Universidad de Sonora. Está integrado por diferentes estudiantes y personal del área de Ing. En sistemas de información, con el fin de apoyar a los diferentes departamentos de la universidad de sonora en las diferentes necesidades de software que tengan éstos y proporcionar experiencia en un entorno laboral a los estudiantes de la carrera de Ing. en Sistemas de Información.

3. Justificación

Se espera contribuir a la mejora de la imagen de las PYMES en la localidad, lugares de la ciudad que no son muy conocidos al público y que pueden ofrecer algún servicio o producto al mercado.

El módulo se realizó con el fin de documentar, controlar y monitorear el flujo de datos en el sistema, en esto nos ayudará el Log In. Tener un control de los usuarios y las empresas que utilizarán el sistema, registrar datos para futuras estadísticas.

Los análisis serán realizados por medio de gráficos estadísticos, en los cuales se medirá y mostrara la cantidad de usuarios que hagan uso del sistema, así como mostrar y registrar la crítica del usuario sobre el sistema y como éste ha contribuido al crecimiento del negocio.

Creo muy necesaria la aplicación de geoposicionamiento para las empresas a nivel local, serán reconocidas, encontradas de manera más eficiente y esto contribuirá al crecimiento de las mismas. Las empresas ahora podrán monitorear

de cierta manera a sus competencias desconocidas, y podrán hacer promociones para los consumidores a traves del espacio asignado en la aplicación.

4. Objetivos

Desarrollo de un sistema de administración de usuarios (Log in) en Php, éste será integrado a la aplicación de geoposicionamiento.

La aplicación deberá integrar una sección de index donde reconocerá a los usuarios que ya han iniciado sesión, un módulo de modificación y registro de usuarios y eliminación de los mismos. Los datos serán almacenados en una base de datos SQL que estarán montadas en un servidor web, esta será utilizada para todas las operaciones que tendrá el Log in.

Realizar módulos de búsquedas y búsqueda avanzada que serán integradas a las diferentes interfaces de la aplicación web de geoposicionamiento que dará funcionalidad a la búsqueda de las diferentes empresas ya registradas en el sistema, esto para la facilidad del usuario a encontrar empresas en su localidad. También se deberán realizar una serie de avisos emergentes para la aprobación del registro de la ubicación del usuario.

Realizar pruebas necesarias para el apoyo del desarrollo de la aplicación de geoposicionamiento.

5. Actividades a realizar

A continuación se hablara de los sub-objetivos o actividades a realizar para el óptimo funcionamiento del sistema de geoposicionamiento.

5.1. Realizar análisis de los requisitos del sistema inteligente de geoposicionamiento

Identificar los requerimientos del sistema de geo-posicionamiento haciendo una lista de actividades que debe realizar la aplicación.

Se elaborará un cronograma con los tiempos que se le dedicará a cada una de las actividades a realizar.

5.2. Geo-posicionar objetos dentro de un mapa de Google Maps

Se harán las pruebas necesarias para posicionar un objeto en Google Maps para el funcionamiento del sistema.

Usando herramientas de JavaScript, HTML y PHP se logrará posicionar algún elemento en determinada ubicación en Google Maps. Esto con el fin de colocar un icono indicando la ubicación de un lugar determinado por el usuario.

5.3. Diseñar tablas necesarias para la gestión de información de ubicaciones

Diseño y desarrollo de la base de datos que se utilizará para administrar la información de las empresas y usuarios que darán uso de la aplicación web.

5.4. Diseñar la interfaz del sitio web de geo-posicionamiento

Desarrollar el sitio web donde se montará la aplicación según las especificaciones previamente definidas. Haciendo uso de HTML, PHP para el desarrollo web y la herramienta Photoshop para el diseño de las diferentes imágenes a utilizar.

5.5. Administración de Usuarios y Empresas (LOGIN)

Desarrollar el módulo dedicado al registro de usuarios y empresas mediante un Login, donde los usuarios deberán llenar un formulario PHP el cual enviará la información a una base de datos donde serán almacenados.

Contendrá módulos de modificación de usuarios y modificación de empresa, en el cual cada usuario o empresa podrá actualizar sus datos personales para mantener a los clientes actualizado sobre los cambios que se han presentado.

5.6. Desarrollar módulo de búsqueda personalizada (Filtros)

El usuario podrá realizar una consulta de algún lugar o negocio, se mostrará en el mapa los diferentes lugares o negocios relacionados con la consulta que realizó el usuario.

El usuario podrá realizar varios filtrados para reducir los resultados de la consulta realizada.

6. Alcances y limitaciones

Como alcances el proyecto puede llegar a mucho, el proyecto con una buena cantidad de personal enfocada a su desarrollo creo que puede llegar a ser un sostén para todas las empresas respecto a publicidad, las empresas tendrán un espacio donde serán reconocidas por los usuarios y podrán llegar a tener más clientes y esto generará capital para las mismas empresas.

Un proyecto de esta magnitud puede llegar a ser un gran apoyo para la economía del estado al brindar el apoyo que muchas empresas no tienen para publicidad.

Sería bueno que este proyecto tuviera seguimiento por otras generaciones de estudiantes de la carrera de ingeniería en sistemas para que pueda crecer y llegar al punto que sea una aplicación de uso estatal como previamente fue mencionado.

Hablando de limitaciones creo que sería el punto en donde se empalmaría con otras aplicaciones, ya hay existencia de otras aplicaciones muy parecidas sin embargo generalmente se encuentran en dispositivos móviles o son de otros países. Creo que como limitación tendríamos las fronteras de otros estados o del país.

Hablando particularmente de mis actividades realizadas las limitaciones se presentaron en la falta de conocimiento de las herramientas utilizadas en el desarrollo de la aplicación, fueron herramientas que jamás había tenido la

oportunidad de utilizar y fue un poco lento el procedimiento de aprendizaje de las mismas. Quizá otra limitación se podría mencionar que fue la falta de comunicación del equipo de trabajo, en algunas ocasiones se presentaba esta problemática e impedía un óptimo seguimiento en los diferentes módulos de la aplicación.

7. Fundamento teórico de las herramientas y conocimientos aplicados

A continuación se hablará de las diferentes herramientas de programación y procedimientos lógicos utilizados para el desarrollo de la aplicación de geoposicionamiento

7.1. HTML

Siglas de **HyperText Markup Language** («lenguaje de marcado de hipertexto»), hace referencia al lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). **HTML** también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir unscript (por ejemplo, JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

7.2. PHP

PHP es un lenguaje de programación de uso general de script del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento **HTML** en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. **PHP** ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. PHP

puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

7.3. SQL

El **lenguaje de consulta estructurado o SQL** (por sus siglas en inglés structured query language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla información de interés de bases de datos, así como hacer cambios en ella.

7.4. MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

7.5. WAMP

El servidor HTTP **Apache** es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual.

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

A continuación se mencionarán algunas de las instrucciones o métodos utilizados para la realización de la aplicación geoposicionamiento.

7.6. Php Sessions

Sessions hace referencia a una serie de instrucciones PHP con los cuales crea una sesión virtual de un usuario con los valores de usuario y contraseña. Es esencial para crear logins en php ya que contribuye en gran medida a la administración de la información al momento de programar.

7.7. Session_destroy()

Destruye todos los datos asociados a la sesión en curso.

7.8. Session_name()

Regresa el valor del nombre de la sesión en curso.

7.9. Sentencia If – Else

Es una sentencia en la cual se realiza una validación de una condición, si ésta se cumple da paso a una serie de instrucciones a realizar, si la condición no se cumple puede realizar otro tipo de instrucciones o bien terminar la sentencia.

7.10. Formularios HTML

Los formularios interactivos permiten a los autores de páginas Web poner elementos interactivos en sus páginas, por ejemplo, cuadros de texto, botones, lista de elementos, cuadros de comentarios, entre otros más.

7.11. Etiqueta FORM

Los formularios están limitados con la etiqueta <FORM> ...instrucciones...</FORM>, ésta etiqueta permite colocar todos los elementos del formulario necesarios, botones, cuadros de texto, etc.

- **METHOD** indica cómo se enviarán los datos de las respuestas o contenido de los formularios. “POST” es el valor que envía los datos al agente de procesamiento almacenándolos en el cuerpo del formulario, en tanto que “GET” envía los datos agregándolos a la dirección URL y separándolos de la dirección con un signo de interrogación
- **ACTION** indica la dirección a la que se enviará la información (un script CGI o dirección de correo electrónico).

La etiqueta *FORM* se comporta como un contenedor que almacena elementos que permiten al usuario seleccionar o introducir datos. Todos los datos se enviarán a la dirección URL indicada en el atributo *ACTION* de la etiqueta *FORM*, por el método indicado en el atributo *METHOD*.

Se puede insertar cualquier elemento HTML en una etiqueta *FORM* (como texto, botones, tablas y enlaces), pero los elementos interactivos son los más interesantes. Estos elementos interactivos son:

- La etiqueta **INPUT**: Todos los botones y casillas de texto
- La etiqueta **TEXTAREA**: una casilla de texto
- La etiqueta **SELECT**: una lista de opciones múltiples

La etiqueta *INPUT* es una etiqueta esencial para los formularios, ya que se usa para crear muchos elementos “interactivos”. La sintaxis de esta etiqueta es la siguiente:

```
<INPUT type="Nombre de campo" value="Valor predeterminado" name="Nombre de elemento">
```

El atributo *name* es esencial, ya que permite al script CGI reconocer qué campo está asociado con un par nombre/valor, lo que significa que el nombre del campo estará seguido de un signo igual (“=”) seguido de un valor que el usuario introdujo, o si el usuario no introdujo ningún valor, por el valor predeterminado de la etiqueta *value*.

El atributo *type* se usa para especificar qué tipo de elemento se representa con la etiqueta *INPUT*. Estos son los valores posibles:

- **Checkbox:** Las casillas de elección pueden adoptar uno de dos estados: checked (seleccionado) o unchecked (no seleccionado). Cuando la casilla es seleccionada, el par nombre/valor se envía al CGI.
- **File:** Un campo que permite al usuario especificar una ruta de archivo que lleva al archivo que se enviará con el formulario. Los tipos de archivo que pueden ser enviados deben especificarse utilizando el atributo ACCEPT de la etiqueta FORM.
- **Image:** Un botón de envío personalizado que aparece cuando se ubica una imagen en la ubicación definida por el atributo SRC.
- **Password:** Una casilla de texto donde los caracteres escritos aparecen como asteriscos para camuflar el texto de entrada.
- **radio:** Un botón que permite al usuario elegir entre varias opciones. Cada uno de estos botones debe tener el mismo atributo **name**. El par nombre/valor del botón radio seleccionado se enviará al CGI. Al aplicar el atributo checked para uno de estos botones se definirá como seleccionado de forma predeterminada.
- **submit:** Un botón de envío para enviar el formulario. El texto en el botón puede definirse usando el atributo **value**.
- **Text:** Una casilla de texto para escribir una línea de texto. El tamaño de la casilla puede definirse usando el atributo **size** y la extensión máxima del texto con el atributo **maxlength**.

8. Procedimientos empleados y actividades desarrolladas

Se mostrará la explicación del código de los diferentes módulos realizados para el Login de la aplicación de geoposicionamiento.

8.1. Index

Para empezar con el index simplemente comenzamos colocando el código general de los elementos de HTML, así como la importación de los diferentes scripts utilizados en el código.

```

<html>
<head>
<meta charset="utf-8" />
    <title></title>
    <link rel="stylesheet" href="reveal.css">
    <script type="text/javascript" src="http://code.jquery.com/jquery-
1.6.min.js"></script>
    <script type="text/javascript" src="jquery.reveal.js">
    </script>
    <style type="text/css">

```

En este bloque de código utilizamos una herramienta de JQuery para desplegar un aviso emergente proporcionándole al usuario información para que permita el geoposicionamiento en su browser.

```

        body { font-family: "HelveticaNeue", "Helvetica-Neue",
"Helvetica", "Arial", sans-serif; }
        .big-link { display: block; margin-top: 100px; text-align: center;
font-size: 70px; color: #06f; }
    </style>
    <script type="text/javascript">
    $(document).ready(function() {
    $('#myModal').reveal();
    });
</script>
</head>
<body>
Bienvenido
<?php
session_start();
?>
<?php

```

En el siguiente segmento de código utilizamos ciertas validaciones para reconocer si el usuario que se ha conectado ha sido una empresa o un usuario de consulta,

para poder proporcionarle las diferentes opciones de modificación de su perfil o Logout.

El reconocimiento se logra accedendo a una herramienta que proporciona Php, la de session, en la cual nos proporciona información sobre la sesión que está activada y poder consultar información en una base de datos o en los registros donde contenga la información del usuario. Validamos la variable class la cual cada uno de los usuarios la tiene con el valor de “usuario” o “empresa” dependiendo del caso.

```
if (isset($_SESSION['k_username'])) {
    $_SESSION['k_class'];
    echo '<b>'. $_SESSION['k_username']. '</b>.';
    echo '<p><a href="logout.php">Logout</a></p>';
    if ($_SESSION['k_class']== 'usuario'){
        echo '<a href="modificar_usuario.php?user='.$_SESSION['k_username'].'">Modificar
Información</a>';
    }else{
        echo '<a href="modificar_empresa.php?empresa='.$_SESSION['k_username'].'">Modificar
Información</a>';
    }
}else{
    echo '<p><a href="login_usuario.php">Login</a></p>
<p><a href="registrar_usuario.php">Registrar un Usuario</a></p>
<p><a href="registrar_empresa.php">Registrar una Empresa</a></p>';
}
?>
```

En este bloque de código se da seguimiento al aviso desplegable accedendo a una clase Modal la cual contiene las características de la animación del anuncio

```
<a href="#" class="big-link" data-reveal-id="myModal" data-animation="fade"></a>

<div id="myModal" class="reveal-modal">
    <h1>Instrucciones para permitir tu geoposicionamiento</h1>
```

```
<p></p>
<a class="close-reveal-modal">&#215;</a>
</div>
</body>
</html>
```

8.2. Login usuario

Se utiliza la función form action para enviar variables a otra página HTML/php donde se validará los datos registrados en el html form con una base de datos donde están los datos de los usuarios registrados para así verificar que los datos sean los correctos y dar acceso a los contenidos de la página.

```
<form action="validar_usuario.php" method="get">
Usuario:<input type="text" name="usuario" size="10" maxlength="10" />
<br />
Password:<input type="password" name="password" size="10" maxlength="10" />
<br />
<input type="submit" value="Ingresar" />
</form>
```

Se hace lo mismo con los datos de la empresa.

```
<form action="validar_empresa.php" method="get">
Usuario:<input type="text" name="usuario" size="10" maxlength="10" />
<br />
Password:<input type="password" name="password" size="10" maxlength="10" />
<br />
<input type="submit" value="Ingresar" />
</form>
```

8.3. Logout

Aquí simplemente borramos todos los datos de la función session y re-direccionamos a la página Index.

```
<?php
```

```

session_start();
session_destroy();
echo 'Ha terminado la session <p><a href="index.php">index</a></p>';
?>
<SCRIPT LANGUAGE="javascript">
location.href = "index.php";
</SCRIPT>

```

8.4. Registrar Usuario

En esta sección solicitamos los datos del usuario a registrar, password, nombre de usuario y correo electrónico.

```

<?php
session_start();
//datos para establecer la conexion con la base de mysql.
mysql_connect('localhost','root','')or die ('Ha fallado la conexión: '.mysql_error());
mysql_select_db('geoposicionamiento')or die ('Error al seleccionar la Base de Datos:
'.mysql_error());
function formRegistro(){
?>

```

En esta sección se crea el HTML FORM donde se solicitara los datos a registrar para posteriormente enviarlos a otra sección del código para su inserción en la base de datos.

```

<form action="registrar_usuario.php" method="GET">
    Usuario (max 10):
    <input type="text" name="username" size="10" maxlength="10" /><br />
    Password (max 10):
    <input type="password" name="password" size="20" maxlength="20" />
    Confirma: <input type="password" name="password2" size="20"          maxlength="20"
/><br />
    E-Mail: <input type="text" name="email" size="20" maxlength="20" /><br />
    <input type="submit" value="Registrar" />
</form>

```

```
<?php
}
```

Verificamos si se han enviado ya las variables necesarias y se hacen un tipo de validaciones para verificar si los datos están completos o si se repite alguno con algún registro en la base de datos.

```
if (isset($_GET["username"])) {
    $username = $_GET["username"];
    $password = $_GET["password"];
    $password2 = $_GET["password2"];
    $email = $_GET["email"];
    // Hay campos en blanco
    if($username==NULL|$password==NULL|$password2==NULL|$email==NULL) {
        echo "un campo está vacío.";
        formRegistro();
    }else{
        // ¿Coinciden las contraseñas?
        if($password!=$password2) {
            echo "Las contraseñas no coinciden";
            formRegistro();
        }else{
```

Comprobamos si el nombre de usuario o la cuenta de correo ya existían con una serie de consultas en SQL.

```
$checkuser = mysql_query("SELECT usuario FROM miembro WHERE usuario='$username'");
$username_exist = mysql_num_rows($checkuser);
$checkemail = mysql_query("SELECT email FROM miembro WHERE email='$email'");
$email_exist = mysql_num_rows($checkemail);
if ($email_exist>0|$username_exist>0) {
    echo "El nombre de usuario o la cuenta de correo estan ya en uso";
    formRegistro();
}else{
```

Si todas las validaciones resultan satisfactorias se da seguimiento a la consulta MySQL para hacer el registro de los datos ingresados.

```
$query = 'INSERT INTO miembro (class, usuario, password, email)VALUES ("usuario",
\''. $username. '\',\''. $password. '\',\''. $email. '\)';
mysql_query($query) or die(mysql_error());
echo 'El usuario ' . $username. ' ha sido registrado de manera satisfactoria.<br />';
echo 'Ahora puede entrar ingresando su usuario y su password <br />';?>
```

Cuando todos los datos son ingresados satisfactoriamente se re-direcciona a otra sección del código para que inicie sesión el nuevo usuario registrado.

```
<FORM ACTION="validar_usuario.php" METHOD="GET">
    Usuario :<INPUT TYPE="text" NAME="usuario" SIZE=10      MAXLENGTH=10><br />
    Password: <INPUT TYPE="password" NAME="password" SIZE=10
    MAXLENGTH=10><br />
<INPUT TYPE="submit" VALUE="Ingresar">
</FORM>
<?php
}}}}else{
    formRegistro();
}
?>
```

8.5. Registro de empresa

Todo el procedimiento de Registrar Usuario se repite, solo que en éste caso se agregaran las variables correspondientes al registro de una empresa, a continuación el código.

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<?php
session_start();
//datos para establecer la conexion con la base de mysql.
mysql_connect('localhost','root','')or die ('Ha fallado la conexión: ' .mysql_error());
```

```
mysql_select_db('geoposicionamiento')or die ('Error al seleccionar la Base de Datos:
'.mysql_error());
function formRegistro(){
?>
```

```
<form action="registrar_empresa.php" method="GET">
```

A diferencia del registro de usuario el registro de empresa llevará una tabla donde mantendrá todos los campos a ingresar en orden.

```
<table>
<tr>
<td>Nombre de la Empresa:</td>
<td><input type="text" name="nombre_empresa" size="20" maxlength="20"/></td></tr><tr>
<td>Contrase&ntildea:</td>
<td><input type="password" name="password" size="20" maxlength="20" /></td></tr>
<tr>
<td>Confirmar contrase&ntildea:</td>
<td><input type="password" name="password2" size="20" maxlength="20" /></td></tr>
<tr>
<td> Direcci&oacuten:</td>
<td><input type="text" name="direccion" size="20" maxlength="20" /></td></tr>
<tr>
<td>Colonia:</td>
<td><input type="text" name="colonia" size="20" maxlength="20" /></td></tr>
<tr>
<td>Tel&eacutefono:</td>
<td><input type="text" name="telefono" size="20" maxlength="20"/></td></tr>
<tr>
<td>C&oacutedigo Postal:</td>
<td><input type="text" name="codigo_postal" size="20" maxlength="20" /></td></tr>
<tr>
<td>Horario de Apertura:</td>
<td><input type="text" name="horario_entrada" size="20" maxlength="20"/> Favor de colocar el
horario de entrada con el siguiente formato HH:MM:SS Ejemplo: 07:00:00</td></tr>
<td>Horario de Cierre:</td>
```

```

<td><input type="text" name="horario_salida" size="20" maxlength="20"/> Favor de colocar el
horario de entrada con el siguiente formato HH:MM:SS Ejemplo: 07:00:00</td></tr>
<tr>
<td>Correo:</td>
<td><input type="text" name="correo" size="20" maxlength="20"/></td></tr>
<tr>
<td>Latitud:</td>
<td><input type="text" name="latitud" size="20" maxlength="20" /></td></tr>
<tr>
<td>Longitud:</td>
<td><input type="text" name="longitud" size="20" maxlength="20"></td></tr>
<tr>
<td>Raiting:</td>
<td><input type="text" name="raiting" size="20" maxlength="20" /></td></tr>
<tr>
<td>Tipo de empresa:</td>
<td><input type="text" name="tipo_empresa" size="20" maxlength="20" /></td>
</tr>
</table>
<input type="submit" value="Registrar" />
</form>
<?php
}

```

Verificamos si se han enviado ya las variables necesarias.

```

if (isset($_GET["nombre_empresa"])) {

    $nombre_empresa = $_GET["nombre_empresa"];
    $password = $_GET["password"];
    $password2 = $_GET["password2"];
    $direccion = $_GET["direccion"];
    $colonia = $_GET["colonia"];
    $telefono = $_GET["telefono"];
    $codigo_postal = $_GET["codigo_postal"];
    $horario_entrada = $_GET["horario_entrada"];
    $horario_salida = $_GET["horario_salida"];
}

```

```

$correo = $_GET["correo"];
$latitud = $_GET["latitud"];
$longitud = $_GET["longitud"];
$raiting = $_GET["raiting"];
$tipo_empresa = $_GET["tipo_empresa"];

// Hay campos en blanco?
if($nombre_empresa==NULL|$password==NULL|$password2==NULL) {
    echo "un campo está vacío.";
    formRegistro();
}else{
    // ¿Coinciden las contraseñas?
    if($password!=$password2) {
        echo "Las contraseñas no coinciden";
        formRegistro();
    }else{
        // Comprobamos si el nombre de usuario o la cuenta de correo ya existían
        $checkuser = mysql_query("SELECT nombre_empresa FROM
EMPRESA WHERE nombre_empresa='$nombre_empresa'");
        $username_exist = mysql_num_rows($checkuser);
        //$checkemail = mysql_query("SELECT email FROM miembro
WHERE email='$email'");
        //$email_exist = mysql_num_rows($checkemail);
        if ($username_exist>0) {
            echo "El nombre de usuario o la cuenta de correo estan
ya en uso";
            formRegistro();
        }else{
            $query = 'INSERT INTO EMPRESA (class,
nombre_empresa, direccion, colonia, telefono,
codigo_postal, horario_entrada, horario_salida, correo,
latitud,
longitud, raiting, tipo_empresa, password)
VALUES ("empresa",
\'.$nombre_empresa.\',\'.$direccion.\',\'.$colonia.\',\'.$telefono.\',\'.$codigo
_postal.\'
\',\'.$horario_entrada.\',\'.$horario_salida.\',\'.$correo.\',\'.$latitud.\',\'.$longitud.\',\'.$raiting
.\',\'.$tipo_empresa.\',\'.$password.\');

```



```
mysql_select_db('geoposicionamiento')or die ('Error al seleccionar la Base de Datos:'.mysql_error());
```

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

```
$usuario= $_GET["usuario"];
$password= $_GET["password"];
```

```
if($usuario != "" && $password != "")
{
    $usuario = strtolower(htmlentities($_GET["usuario"], ENT_QUOTES));
    $password = $_GET["password"];
```

Se hace una consulta con los datos de sesión

```
$result = mysql_query('SELECT class, password, usuario FROM miembro WHERE
usuario=\'".$usuario."\');
if($row = mysql_fetch_array($result)){
    if($row["password"] == $password){
        $_SESSION["k_username"] = $row['usuario'];
        $_SESSION["k_class"] = $row['class'];
        echo 'Has sido logueado correctamente
        '.$_SESSION['k_username'].' <p>';
        echo '<a href="index.php">Index</a></p>';
    }else{
        echo 'Password incorrecto';
        echo '<p><a href="login_usuario.php">Login</a></p>';
    }
}
}else{
```

Si no encuentra los datos según clase sea “usuario” entonces se validara con clase igual a “empresa”.

8.7. Clase Usuarios

A continuación se muestra la clase usuarios que se creó para facilitar la búsqueda por consultas SQL.

```
<?php
class Usuarios
{
```

Se crearon las variables para la conexión al host donde está situada la base de datos.

```
public function __construct() {
    $dbhost = 'localhost';
    $dbuser = 'root';
    $dbpass = '';
    $dbname = 'geoposicionamiento';

    mysql_connect($dbhost, $dbuser, $dbpass);

    mysql_select_db($dbname);
}
```

Se crea una un método el cual tiene la función de hacer una consulta SQL, ésta consiste en buscar los nombres de la empresa según aparezcan los caracteres que se estarán guardando en NombreUsuario, estos valores serán ingresados en un TextBox en otra página. Los valores que genere mediante la consulta serán almacenados en un arreglo.

```
public function buscarUsuario($nombreUsuario){
    $datos = array();
```

```

$sql = "SELECT * FROM EMPRESA
      WHERE nombre_empresa LIKE '$nombreUsuario%";

$resultado = mysql_query($sql);

while ($row = mysql_fetch_array($resultado, MYSQL_ASSOC)){
    $datos[] = array("value" => $row['nombre_empresa'], "nombre" => $row['nombre_empresa'],
"direccion" => $row['direccion']);
    }

    return $datos;
}

```

Se crea otro método el cual hará una búsqueda similar a la anterior pero ahora buscara según la categoría de la empresa.

```

public function buscarUsuarioPorCategoria($nombreUsuario){
    $datos = array();

    $sql = "SELECT * FROM EMPRESA
          WHERE tipo_empresa LIKE '$nombreUsuario%";

    $resultado = mysql_query($sql);

    while ($row = mysql_fetch_array($resultado, MYSQL_ASSOC)){
        $datos[] = array("value" => $row['tipo_empresa'], "nombre" => $row['nombre_empresa'], "direccion"
=> $row['direccion']);
        }

        return $datos;
    }
}

```

8.8. Búsqueda de usuario/empresa (Ajax.php)

En éste segmento de código solo es un puente para poder ejecutar comandos de **Jason**. Este código crea el objeto usuario importando los elementos de clase usuario y ejecuta el método de buscarUsuario utilizando como variable los valores **term**, ésta es una variable que se va obteniendo según sean ingresados valores en un textbox.

```
<?php
include_once 'usuarios.class.php';

$usuario = new Usuarios();

echo json_encode($usuario->buscarUsuario($_GET['term']));
```

Se crea otro objeto para utilizar la búsqueda con el otro método, buscarUsuarioPorCategoria.

```
<?php
include_once 'usuarios.class.php';

$usuario = new Usuarios();

echo json_encode($usuario->buscarUsuarioPorCategoria($_GET['term']));
```

8.9. Autocompletar

Se creó otro Index para realizar pruebas para un buscador que tenga la función de autocompletar, este buscador hace el uso de los elementos de clase usuario, Ajax1 y ajax2, estas páginas hacen el funcionamiento de éste buscador.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<link type="text/css" rel="stylesheet" href="css/jquery-ui-1.8.4.custom.css" />
<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>
```

```
<script type="text/javascript" src="js/jquery-ui-1.8.4.custom.min.js"></script>
<script type="text/javascript">
```

Lo que se agregaría al index principal es este bloque de código, éste llama a una función el cual es autocompletar, el autocompletar dará resultados según la consulta que se hace en **Class usuario**, con el método de buscarusuario.

```
$(function(){
    $('#buscar_usuario').autocomplete({
source : 'ajax.php',
select : function(event, ui){
    $('#resultados').slideUp('slow', function(){
$('#resultados').html(
    '<h2>Detalles de usuario</h2>' +
```

En este segmento se solicita cierta información que se quiere consultar según sea el resultado de la búsqueda.

```
'<strong>Nombre: </strong>' + ui.item.nombre + '<br/>' +
    '<strong>Direccion: </strong>' + ui.item.direccion
    );
    });
    $('#resultados').slideDown('slow');
    }
    });
});
</script>
<title></title>
</head>
<body>
    Busqueda:<br>
<table>
    <tr>
    <td>
```

Aquí se coloca el textbox que estará enviando variables a ajax1 para realizar las consultas en la creación del objeto con la clase usuario.

```
<input type="text" id="buscar_usuario" name="buscar_usuario" value="Nombre de la empresa"
onfocus="this.value=""/></td>
    <td>
        <a href="busqueda_avanzada.php">Busqueda Avanzada</a>
    </td>
</td>
</tr>
</table>
<?php
?>
<div id="resultados">
</div>
</body>
</html>
```

8.10. Modificar Usuario

En esta página agregamos la función de modificar la información del usuario. Para el funcionamiento de esta página, el usuario previamente debió haber iniciado sesión. Una vez que ya tena una sesión iniciada la página tomará las variables de sesión para obtener el nombre de la cuenta del usuario, posteriormente hará la consulta para obtener toda la información del usuario y ser colocada en **Textbox** modificables, una vez ingresada la nueva información se da click en el botón guardar y la información será actualizada mediante una consulta en SQL con la función update.

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<?php
session_start();
//datos para establecer la conexion con la base de mysql.
mysql_connect('localhost','root','')or die ('Ha fallado la conexión: '.mysql_error());
mysql_select_db('geoposicionamiento')or die ('Error al seleccionar la Base de Datos:
'.mysql_error());
```

```
function formRegistro(){
```

```
//Inicializas las variables
```

```
$usuario = "";
```

```
$password = "";
```

```
$password2 = "";
```

```
$email = "";
```

Se obtiene el nombre de usuario.

```
    $user = $_SESSION["k_username"];
```

```
//Suponiendo que ya tienes creada la conexión a la BD paso directamente al query MySQL y su ejecución
```

```
$query = "SELECT * FROM miembro WHERE usuario='$user'";
```

```
$execute = mysql_query($query); /* $conexion es la variable de la conexión a la BD */
```

```
//Compruebo si hay algún resultado
```

```
if($row = mysql_fetch_array($execute)){
```

```
//Guardo los datos de la BD en las variables de php
```

```
    $usuario = $row["usuario"];
```

```
    $password = $row["password"];
```

```
    $email = $row["email"];
```

```
}
```

```
?>
```

Se llenan los **Textbox** según sea la información original.

```
<form action="modificar_empresa.php" method="GET">
```

```
<table>
```

```
<tr>
```

```
<td>Nombre de usuario:</td>
```

```
<td><input type="text" value="<?php echo $usuario; ?>" name="usuario" size="20"
```

```
maxlength="20"/></td></tr><tr>
```

```
<td>Contrase&ntildea:</td>
```

```

<td><input type="text" value="<?php echo $password; ?>"name="password" size="20"
maxlength="20" /></td></tr>
<tr>
<td>Confirmar contrase&ntilde;a:</td>
<td><input type="text" value="<?php echo $password; ?>"name="password2" size="20"
maxlength="20" /></td></tr>
<tr>
<td> Direcci&oacute;n:</td>
<td><input type="text" value="<?php echo $email; ?>" name="email" size="20" maxlength="20"
/></td></tr>
</table>
<input type="submit" value="Guardar" />
</form>
<?php
}

```

// verificamos si se han enviado ya las variables necesarias.

```

if (isset($_GET["usuario"])) {

    $usuario = $_GET["usuario"];
    $password = $_GET["password"];
    $password2 = $_GET["password2"];
    $email = $_GET["email"];

    // Hay campos en blanco
    if($usuario==NULL|$password==NULL|$password2==NULL) {
        echo "un campo está vacío.";
        formRegistro();
    }else{
        // ¿Coinciden las contraseñas?
        if($password!=$password2) {
            echo "Las contraseñas no coinciden";
            formRegistro();
        }else{
            // Comprobamos si el nombre de usuario o la cuenta de correo ya existían
            $checkuser = mysql_query("SELECT nombre_empresa FROM EMPRESA
WHERE nombre_empresa='$nombre_empresa");
$username_exist = mysql_num_rows($checkuser);

```

```

//$checkemail = mysql_query("SELECT email FROM miembro WHERE email='$email");
//$email_exist = mysql_num_rows($checkemail);
if ($username_exist>0) {
    echo "El nombre de usuario o la cuenta de correo estan ya en uso";
    formRegistro();
}else{

```

Se hace la consulta SQL para actualizar los datos del usuario.

```

$query = 'UPDATE `geoposicionamiento`.`empresa` SET `nombre_empresa` =
\'".$nombre_empresa."\` WHERE `empresa`.`nombre_empresa` = \\'".$nombre_empresa2."\` LIMIT
1 ;';
mysql_query($query) or die(mysql_error());
echo 'Los datos de '.$nombre_empresa.' ha sido modificados satisfactoriamente.<br />';
echo 'Ahora puede entrar ingresando su usuario y su password <br />';
?>
<FORM ACTION="validar_usuario.php" METHOD="GET">
Usuario :<INPUT TYPE="text" NAME="usuario" SIZE=10 MAXLENGTH=10><br />
Password: <INPUT TYPE="password" NAME="password" SIZE=10 MAXLENGTH=10><br />
<INPUT TYPE="submit" VALUE="Ingresar">
</FORM>
<?php
}}}}else{
    formRegistro();
}
?>

```

8.11. Modificar empresa

Se crea una página con las mismas funciones de Modificar usuario, pero ahora se hará la consulta para las empresas.

```

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<?php
session_start();
//datos para establecer la conexion con la base de mysql.
mysql_connect('localhost','root','')or die ('Ha fallado la conexión: '.mysql_error());

```

mysql_select_db('geoposicionamiento')or die ('Error al seleccionar la Base de Datos:

'.mysql_error());

function formRegistro(){

//Inicializas las variables

\$nombre_empresa = "";

\$password = "";

\$password2 = "";

\$direccion = "";

\$colonia = "";

\$telefono = "";

\$codigo_postal = "";

\$horario_entrada = "";

\$horario_salida = "";

\$correo = "";

\$latitud = "";

\$longitud = "";

\$raiting = "";

\$tipo_empresa = "";

\$empresa = \$_SESSION["k_username"];

//Suponiendo que ya tienes creada la conexión a la BD paso directamente al query MySQL y su ejecución

\$query = "SELECT * FROM empresa WHERE nombre_empresa='\$empresa'";

\$execute = mysql_query(\$query); /* \$conexion es la variable de la conexión a la BD */

//Compruebo si hay algún resultado

if(\$row = mysql_fetch_array(\$execute)){

//Guardo los datos de la BD en las variables de php

\$nombre_empresa = \$row["nombre_empresa"];

\$password = \$row["password"];

\$direccion = \$row["direccion"];

\$colonia = \$row["colonia"];

\$telefono = \$row["telefono"];

\$codigo_postal = \$row["codigo_postal"];

\$horario_entrada = \$row["horario_entrada"];

\$horario_salida = \$row["horario_salida"];

\$correo = \$row["correo"];

```

$latitud = $row["latitud"];
$longitud = $row["longitud"];
$raiting = $row["raiting"];
$tipo_empresa = $row["tipo_empresa"];
$nombre_empresa2 = $nombre_empresa;
}
?>
<form action="modificar_empresa.php" method="GET">
<table>
<tr>
<td>Nombre de la Empresa:</td>
<td><input type="text" value="<?php echo $nombre_empresa; ?>" name="nombre_empresa"
size="20" maxlength="20"/></td></tr><tr>
<td>Contrase&ntildea:</td>
<td><input type="text" value="<?php echo $password; ?>"name="password" size="20"
maxlength="20" /></td></tr>
<tr>
<td>Confirmar contrase&ntildea:</td>
<td><input type="text" value="<?php echo $password; ?>"name="password2" size="20"
maxlength="20" /></td></tr>
<tr>
<td> Direcci&ocuten:</td>
<td><input type="text" value="<?php echo $direccion; ?>" name="direccion" size="20"
maxlength="20" /></td></tr>
<tr>
<td>Colonia:</td>
<td><input type="text" value="<?php echo $colonia; ?>" name="colonia" size="20" maxlength="20"
/></td></tr>
<tr>
<td>Tel&eacute;fono:</td>
<td><input type="text" value="<?php echo $telefono; ?>" name="telefono" size="20"
maxlength="20"/></td></tr>
<tr>
<td>C&ocutedigo Postal:</td>
<td><input type="text" value="<?php echo $codigo_postal; ?>" name="codigo_postal" size="20"
maxlength="20" /></td></tr>
<tr>
<td>Horario de entrada:</td>

```

```

<td><input type="text" value="<?php echo $horario_entrada; ?>" name="horario_entrada"
size="20" maxlength="20"/>Favor de colocar el horario de entrada con el siguiente formato
HH:MM:SS </td></tr>
<td>Horario de salida:</td>
<td><input type="text" value="<?php echo $horario_salida; ?>" name="horario_salida" size="20"
maxlength="20"/>Favor de colocar el horario de entrada con el siguiente formato HH:MM:SS
</td></tr>
<tr>
<td>Correo:</td>
<td><input type="text" value="<?php echo $correo; ?>" name="correo" size="20"
maxlength="20"/></td></tr>
<tr>
<td>Latitud:</td>
<td><input type="text" value="<?php echo $latitud; ?>" name="latitud" size="20" maxlength="20"
</td></tr>
<tr>
<td>Longitud:</td>
<td><input type="text" value="<?php echo $longitud; ?>" name="longitud" size="20"
maxlength="20"></td></tr>
<tr>
<td>Raiting:</td>
<td><input type="text" value="<?php echo $raiting; ?>" name="raiting" size="20" maxlength="20"
/></td></tr>
<tr>
<td>Tipo de empresa:</td>
<td><input type="text" value="<?php echo $tipo_empresa; ?>" name="tipo_empresa" size="20"
maxlength="20" /></td>
</tr>
</table>
<input type="submit" value="Guardar" />
</form>
<?php
}

```

// verificamos si se han enviado ya las variables necesarias.

```

if (isset($_GET["nombre_empresa"])) {
    $nombre_empresa = $_GET["nombre_empresa"];
    $password = $_GET["password"];
}

```

```

$password2 = $_GET["password2"];
$direccion = $_GET["direccion"];
$colonia = $_GET["colonia"];
$telefono = $_GET["telefono"];
$codigo_postal = $_GET["codigo_postal"];
$horario = $_GET["horario"];
$correo = $_GET["correo"];
$latitud = $_GET["latitud"];
$longitud = $_GET["longitud"];
$raiting = $_GET["raiting"];
$tipo_empresa = $_GET["tipo_empresa"];
$empresa = $_SESSION["k_username"];

// Hay campos en blanco
if($nombre_empresa==NULL|$password==NULL|$password2==NULL) {
    echo "un campo está vacío.";
    formRegistro();
}else{
    // ¿Coinciden las contraseñas?
    if($password!=$password2) {
        echo "Las contraseñas no coinciden";
        formRegistro();
    }else{
        // Comprobamos si el nombre de usuario o la cuenta de correo ya existían
        $checkuser = mysql_query("SELECT nombre_empresa FROM EMPRESA
WHERE nombre_empresa='$nombre_empresa'");
        $username_exist = mysql_num_rows($checkuser);
        //$checkemail = mysql_query("SELECT email FROM miembro WHERE
email='$email'");
        //$email_exist = mysql_num_rows($checkemail);
        if ($username_exist>0) {
            echo "El nombre de usuario o la cuenta de correo estan ya en uso";
            formRegistro();
        }else{
            $query = 'UPDATE `geoposicionamiento`.`empresa` SET
`nombre_empresa` = \''.$nombre_empresa.'\' WHERE `empresa`.`nombre_empresa` =
\''.$empresa.'\' LIMIT 1 ;';
            mysql_query($query) or die(mysql_error());

```

```

        echo 'Los datos de '.$nombre_empresa.' ha sido modificados
satisfactoriamente.<br />';

        echo 'Ahora puede entrar ingresando su usuario y su password <br
/>';

        ?>
        <FORM ACTION="validar_usuario.php" METHOD="GET">
        Usuario :<INPUT TYPE="text" NAME="usuario" SIZE=10
MAXLENGTH=10><br />
        Password: <INPUT TYPE="password" NAME="password"
SIZE=10 MAXLENGTH=10><br />
        <INPUT TYPE="submit" VALUE="Ingresar">
        </FORM>
        <?php}}}}
    }else{
        formRegistro();
    }
    ?>

```

9. Resultados obtenidos

Se puede mencionar como resultados obtenidos el término de los módulos asignados, se logró realizar los módulo para el óptimo funcionamiento de la aplicación de geoposicionamiento.

En el módulo de Registro de usuario se obtuvo una página con formularios HTML la cual es capaz de registrar una serie de campos en los cuales estará almacenada la información de la empresa que se quiere registrar. Se hicieron varias pruebas validando el funcionamiento de la misma tanto de manera local como en plataforma web. Fue el mismo caso con la página de modificación de usuario en las cuales mediante una sesión previamente realizada se llenan los campos de la empresa en campos de texto modificables, la empresa logra cambiar su información satisfactoriamente. También se realizaron las pruebas necesarias local y en plataforma web.

En la búsqueda avanzada se obtuvo como resultado una sección en el portal web en donde el usuario podrá ingresar el nombre de la aplicación dándole la opción de autocompletar la búsqueda para una consulta más rápida y eficiente, esto da la oportunidad al usuario de recordar si bien el nombre de una empresa o buscar alguna otra parecida. Se hicieron varias pruebas para lograr ésta función, sin embargo, mediante ciertas modificaciones se dio un resultado satisfactorio.

La aplicación contará con una administración de usuarios eficiente donde las empresas podrá registrarse para tener un mejor uso de la aplicación. Las empresas ahora pueden realizar cambios en su perfil de usuario, cambiar su dirección u horario de trabajo, etcétera.

Veo la experiencia obtenido como resultado de éste proyecto al desenvolverme en un entorno laboral y al haber formado parte de un equipo de trabajo.

10. Conclusiones y recomendaciones

Como conclusión puedo agregar muchos puntos importantes que se podría destacar, el aspecto personal y el aspecto en resultados para las empresas.

Empezando por el aspecto personal puedo concluir la importancia de éste tipo de proyectos en mi formación académica, la oportunidad de poder formar parte de un equipo integro de trabajo, poder resolver las diferentes situaciones o problemáticas que se presenten en el mismo y aprender sobre ellas. Haciendo énfasis en el aspecto de trabajo en equipo puedo mencionar la importancia de comunicación en el mismo, es crucial que todos los integrantes del equipo estén enterados de cada cambio o movimiento que se haga en la planeación del proyecto o del sistema ya que si uno llegase a no estar comunicado de algo éste podría truncar todo el seguimiento del proyecto por tiempo indefinido. También es muy importante mencionar el rol de una buena documentación y planeación del proyecto, con una buena planeación se pueden definir si no es el total sí una gran

mayoría de las características y aspectos que conllevará el proyecto, así todos los integrantes del equipo sabrán cómo y con qué trabajar.

En aspectos del proyecto se concluye que todo el proyecto se puede mejorar, con un equipo de trabajo más grande y mayor comunicación con los integrantes, creo que el módulo de administración de usuario es algo en lo que se tiene que dedicar muchísimo trabajo ya que es lo que gestiona todo el flujo de usuarios, quienes pueden o no usar la aplicación, que privilegios tienen ciertos usuarios a otros.

En el módulo de registro de usuario se presentaron varios problemas respecto a los campos que iban a ser solicitados, eso se pudo haber evitado con una buena comunicación entre los integrantes de trabajo y una buena definición de los requerimientos del sistema.

En el módulo de búsqueda avanzada se puede mejorar bastante, se pueden agregar varios campos para hacer un filtrado mucho más eficiente, en este trabajo solo se abarcó las secciones de, categoría y nombre, pero se pueden agregar horario de trabajo, producto o servicio entre algunas otras cosas más. También se le pueden agregar algún otro método de búsqueda, como solo buscar por categorías, nuestro buscador de vio limitado a ingresar el nombre como campo obligatorio.

El sistema puede tener muchas mejoras y sin lugar a dudas puede llegar a ser muy útil para los usuarios, puede haber algún espacio para las empresas y que éstas tengan una “página web” a una escala mínima en el portal de geoposicionamiento, un espacio donde pueda haber mucha más información sobre las mismas, no solo la dirección, categoría o el nombre. En mi punto de vista los otros módulos se pueden mejorar de forma mínima.

Otro aspecto que quisiera agregar es en el ámbito del aprendizaje, ahora más que nunca me percaté que las enseñanzas obtenidas a lo largo de la carrera no son

suficientes para tener un desempeño íntegro en un ámbito laboral, hay que estar en continuo aprendizaje, actualizarse de las diferentes tecnologías que se están presentando, todo esto para lograr la superación de las diferentes problemáticas que se van presentando a lo largo de un proyecto.

11. Bibliografía

- <http://es.wikipedia.org/wiki/HTML>
- <http://es.wikipedia.org/wiki/PHP>
- <http://es.wikipedia.org/wiki/SQL>
- http://www.w3schools.com/html/html_forms.asp
- <http://es.kioskea.net/contents/html/htmlform.php3>
- <http://es.wikipedia.org/wiki/AJAX>
- <http://es.wikipedia.org/wiki/MySQL>
- <http://www.w3schools.com/js/default.asp>
- <http://es.wikipedia.org/wiki/JavaScript>