



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Creación de un sistema de  
inscripciones a prácticas de  
campo**

**TESIS**

Que para obtener el título de  
**Ingeniero en Computación**

**P R E S E N T A**

Alan Gabriel Barrios Muñoz

**DIRECTORA DE TESIS**

M. en I. Tanya Itzel Arteaga Ricci



Ciudad Universitaria, Cd. Mx., 2021

## **Sistema de Inscripciones a Prácticas de Campo del Departamento de Sistemas, Planeación y Transporte SIPRAC**

Introducción	4
Capítulo 1 Marco Teórico	7
1.1 <i>WEB 2.0</i>	7
1.2 Ventajas de los sistemas basados en <i>Web</i> .	10
1.2.1 Ventajas y Compatibilidad	10
1.2.1.1 HTML5	11
1.2.1.2 CSS3	14
1.2.1.3 PHP	18
1.3 <i>Frameworks</i>	19
1.3.1 <i>Codeigniter</i>	19
1.3.2 <i>Materialize</i>	21
1.3.3 <i>Jquery</i>	24
Capítulo 2. Metodologías de desarrollo	25
2.1 CMMI-DEV	25
2.1.1 <i>Specific Goal - Establecer estimaciones (Establish Estimates)</i>	26
2.1.1.1 <i>Specific Practice - Definición de las fases del ciclo de vida del proyecto</i>	26
2.1.1.2 <i>Specific Practice - Estimar el alcance del proyecto (Estimate the Scope of the Project)</i>	27
2.1.1.2.1 <i>Project Charter</i>	27
2.1.1.2.2 <i>Descripción del alcance del producto (Product Scope Description)</i>	28
2.1.1.3 <i>Specific Practice - Establecer estimaciones de productos de trabajo y atributos de tareas (Establish Estimates of Work Product and Task Attributes)</i>	56
2.1.1.3.1 <i>Estimación del proyecto</i>	56
2.1.1.3.2 <i>Specific Practice - Estimar el esfuerzo y el costo (Estimate Effort and Cost)</i>	73
2.1.2 <i>Specific Goal - Desarrollar un plan de proyecto (Develop a Project Plan)</i>	75
2.1.2.1 <i>Specific Practice - Establish the Budget and Schedule &amp; Specific Practice - Establish the Project Plan</i>	76
2.1.2.2 <i>Specific Practice - Identificar Riesgos del Proyecto (Identify Project Risk)</i>	85
2.1.2.3 <i>Specific Practice - Planificar la gestión de datos (Data Management Plan)</i>	85

2.1.2.4 Specific Practice - Plan de Recursos del Proyecto (Plan the Project's Resources)	86
2.1.2.5 Specific Practice - Plan de conocimientos y habilidades necesarios (Plan Needed Knowledge and Skills)	87
2.1.2.6 Specific Practice - Planificar la participación de las partes interesadas (Plan Stakeholder Involvement)	87
2.1.2.7 Specific Practice - Comprometerse con el plan (Obtain Commitment to the Plan)	87
Capítulo 3. Desarrollo de la base de datos	89
3.1 Bases de datos	89
3.2 Diagrama Entidad Relación	93
3.3 SQL	101
3.3.1 Lenguaje de manipulación de datos	102
3.3.1.1 Insertar los datos en una tabla	102
3.3.1.2 Consultar los datos ingresados	103
3.3.1.3 Modificar datos	103
3.3.2 <i>Data Definition Language (DDL)</i>	104
3.3.2.1 Crear una base de datos en MySQL	104
Figura 3.16 Creación de las tablas en la base de datos siprac2	106
3.3.3 Control de concurrencia en la base de datos MySQL	106
Capítulo 4. Implementación	109
4.1 <i>Front</i>	109
4.2 <i>Backend</i>	113
4.4 Seguridad del sistema	117
4.4.1 SQL Injection	117
4.4.2 Certificados de seguridad	119
4.4.2 Seguridad en las contraseñas	122
Capítulo 5. Experiencia de usuario y pruebas del sistema	123
Capítulo 6. Validación del proyecto y conclusiones	143

# Introducción

El Departamento de Sistema, Planeación y Transporte de la División de Ingenierías Civil y Geomática de la Facultad de Ingeniería de la UNAM, tiene como una de sus funciones la administración de prácticas de campo de las asignaturas Sistemas de Transporte e Integración de Proyectos. La realización de estas prácticas de campo propicia un mejor aprendizaje en los alumnos por lo que es necesario efectuar un proceso sistematizado de inscripción y gestión de prácticas de campo. Normalmente el proceso a seguir es:

- 1 El alumno debe asistir personalmente al departamento para inscribir la práctica o prácticas
- 2 El departamento le muestra en una hoja de papel la lista de prácticas y fechas elegibles.
- 3 El alumno se registra en una de ellas y se pide ir a pagar a cajas de la Facultad.
- 4 El alumno regresa al departamento entrega el ticket y se le asegura el lugar firmando en la hoja.

Los 4 pasos anteriores solamente abarcan la inscripción a la práctica, falta el trámite administrativo interno que se tiene que hacer para llevar a cabo la visita, posteriormente otra serie de pasos que conllevan tiempo y esfuerzo a los colaboradores del departamento y que representa, además, bastante tiempo en la entrega de resultados.

Es por esto que el Jefe del Departamento solicitó se desarrolle un sistema de acceso WEB para gestionar todo el proceso de Prácticas de Campo.

La presente tesis abarca justamente ese desarrollo que se solicitó, y que abarca en los siguientes temas:

## Capítulo 1.- Marco Teórico

En este capítulo se abordarán las herramientas para el desarrollo del sistema, además se expondrá el diseño responsivo implementado para que los alumnos y el administrador vean la información en la mayoría de los dispositivos de manera adecuada ya sea un celular, una tableta o una computadora. También los lenguajes de programación utilizados junto con los *frameworks* que facilitaron el desarrollo de todo el sistema web.

## Capítulo 2. Metodología de desarrollo

En este capítulo se explicará cómo las buenas prácticas, que pueden ser dictadas por *The Capability Maturity Model Integration for Development* (CMMI-DEV), logran mejorar el proceso que se llevaba a cabo en el Departamento para la creación de sistemas web, éste

es un modelo que se enfoca en productos y servicios para empresas, que ayude a mejorar el rendimiento del proceso.

### **Capítulo 3. Desarrollo de la base de datos**

En este capítulo lo que se muestra es cómo utilizando todas las herramientas adquiridas durante la formación académica, se pudo implementar una base de datos en MySQL para almacenar principalmente datos de los alumnos, así como de las prácticas o las inscripciones de los alumnos a prácticas, además, cómo se utiliza SQL para consultarlos.

### **Capítulo 4. Implementación**

En este capítulo se mostrará el patrón de desarrollo Modelo-Vista-Controlador para desarrollar el sistema con el *framework CodeIgniter*, que implica dividir el código para tener un orden con la finalidad de encontrar fácilmente los archivos y además de elementos de seguridad para el sistema.

### **Capítulo 5. Experiencia de usuario y pruebas del sistema**

En este capítulo se muestra el correcto funcionamiento del sistema web para su manipulación, en el cual, podrán hacer las inscripciones, la carga de prácticas, generación e impresión de listas, toma de asistencia, generación e impresión de constancias de asistencia y estadísticas, entre otras.

También se explicará la forma en que se llevaron a cabo pruebas para el correcto funcionamiento del sitio web con Postman.

### **Capítulo 6. Validación del proyecto y conclusiones**

En este último capítulo se tiene los resultados de la implementación del proyecto junto con la aceptación de este por el Departamento. También se muestran los resultados durante un ejercicio práctico con los datos de alumnos de Ingeniería Civil al ser utilizado y alimentado con las prácticas del semestre corriente, además de la gran experiencia adquirida al desarrollar todo el sistema desde cero.

## **Objetivo general**

El principal objetivo de este proyecto es contar con una herramienta que facilite la gestión académica-administrativa que implica el tener prácticas de campo en una de las licenciaturas más demandadas de la Facultad de Ingeniería, coordinar y entregar resultados de una manera práctica e inmediata.

## **Objetivos Particulares**

Los objetivos particulares son:

- 1 Implementar conocimientos de programación para llevar a cabo la lógica de negocio.
- 2 Implementar el diseño de bases de datos para brindar información
- 3 Diseñar las vistas del sistema web para ser visualizadas correctamente en la mayoría de los dispositivos, implementando *responsive designer*.

# Capítulo 1 Marco Teórico

Todo sistema computacional está desarrollado con uno o varios lenguajes de programación y *frameworks* que faciliten su desarrollo, estos sistemas pueden estar albergados en *clusters* para recibir una cantidad grande de usuarios o bien en un solo servidor cuando el número de usuarios sea bajo; también estos *frameworks* permiten desarrollar sistemas que se puedan mostrar en cualquier dispositivo. Con esto dicho, en este apartado se mostrarán los lenguajes de programación y *frameworks* que hicieron posible el desarrollo del sistema SIPRAC a la solicitud del Departamento.

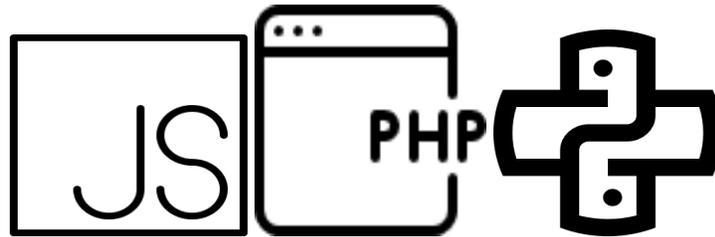
## 1.1 WEB 2.0

Con la llegada del internet fue posible obtener información visitando lo que se llama páginas web. Estas páginas solamente el usuario podía leer la información más no interactuar con ellas, es decir, las páginas eran estáticas, la información mostrada nunca cambiaba, pero con la llegada de *web 2.0* se tienen páginas dinámicas, la información cambia dependiendo de los datos que se ingresan o se modifican, con servidores donde reside la lógica de negocio, que es la forma en cómo se manipulan los datos y se toman decisiones de acuerdo a las acciones que tenga el usuario en la página *web*.



**Figura 1.1 Web dinámica**

Todo esto fue posible gracias a lo que se llama “*backend*”, que en conjunto son instrucciones en algún lenguaje de programación orientado a web que realizan ciertas acciones estudiadas previamente, es decir toda la programación está en el servidor y cuando hay una solicitud de un usuario para obtener su información, el código que reside en el servidor la recupera y se la regresa en forma de página web o *JSON*. Algunos de los lenguajes de programación que logran esto son *PHP*, *Javascript*, *Python*. Lo que facilita más el desarrollo del *backend* son los *frameworks*, por mencionar unos: *Django*, *Flask*, *CodeIgniter* o *NodeJS*.



**Figura 1.2 Lenguajes de programación**

El marco general de cómo funciona es el siguiente:

El usuario solicita una página *web* al servidor (ya sea *Apache2*, *NodeJS* o *NGINX*), éste regresa la página web con imágenes, texto y la forma o diseño en que se va a mostrar la página al usuario, después el usuario interactúa con la página web y lo que puede suceder es que se le solicite un usuario o contraseña de manera que se autentifique la persona y así mostrar contenido exclusivo para usuarios registrados. Un ejemplo común, sería el del portal principal de *Netflix*, en este caso hay dos opciones de respuesta del servidor al usuario:

- 1 La información ingresada es correcta por lo tanto el servidor mostrará una página en donde seleccionará alguna de las películas en *Netflix*.
- 2 El servidor muestre un mensaje indicando que las credenciales ingresadas son incorrectas.

Ahora se tienen sistemas que pueden recibir peticiones de usuarios en pocos momentos, pero si se tiene más de un sistema, los cuales reciben muchas peticiones a la vez, lo más seguro es que se necesite un *datacenter* como los que seguramente cuenta **Google** cuyo objetivo es aceptar todas esas peticiones y responder al instante por cada uno de los sistemas que se tienen involucrados. Por ejemplo, se puede tener el sistema de inscripciones a prácticas de campo, otro sistema podría ser de inventarios u otro sería un sistema hecho con Moodle el cual sirve para que los estudiantes suban tareas, descarguen archivos, entre otros, y cada uno tendría su carga de trabajo. Todo esto es posible gracias a que un *datacenter* es un conjunto de sistemas distribuidos que realiza computación en paralelo para diversos usuarios, algunos trabajadores de *Google* definen sus *datacenters* como:

*“The brain of the internet”*

*“The engine of the internet”*<sup>1</sup>

En un *Google datacenter* lo que se tiene es una cantidad enorme de computadoras conocidos como servidores, todos estos son acomodados en un espacio destinado y confinado coloquialmente llamado *SITE*, como en la siguiente imagen:

---

<sup>1</sup> <https://www.youtube.com/watch?v=XZmGGAhHqa0>



**Figura 1.3 Racks, quadraracks o armarios cerrados**

En el caso de *Google* son diseñados específicamente por ingenieros de *Google*, pero no solo eso es necesario para tener un *datacenter*, también se necesita seguridad lógica por ejemplo en la información y seguridad física como por ejemplo en el acceso al *datacenter*.

En el caso de la seguridad física o perimetral

- 1 Primero se tiene que identificar como trabajador de *Google* e indicar que se trabaja en el *datacenter* en específico al cual se quiere acceder, esto se logra mostrando la identificación de trabajador al policía de seguridad.
- 2 Al entrar al edificio otro oficial de seguridad pedirá que se pase por el detector de metales para saber si no se tiene algún objeto que ponga en peligro la seguridad de las personas en el edificio.
- 3 Para acceder realmente al cuarto donde están los servidores se necesita pasar por un detector de huellas dactilares y de retina para confirmar que en realidad se trata de la persona autorizada.

Una característica importante es el control de la temperatura en un *datacenter*, ya que el calor generado por los servidores al estar en funcionamiento elevan su temperatura ocasionando a estos bajar su rendimiento. En *Google* se concentran en la parte superior de los *racks* unos *pipes* o tuberías que tienen el mismo diseño y material que los enfriadores de computadora. Estos *pipes* contienen agua la cual enfrían los servidores y al subir de temperatura se envía afuera del *datacenter* para enfriarla y una vez a temperatura baja regresan de nuevo a enfriar los servidores.



**Figura 1.4 Ventilador para un procesador con sus pipes**

Todo esto es muy caro y no para todas las empresas es viable crear su propio *datacenter*, por lo que se tiene en la industria la nube o “*cloud*”, la nube son servicios que nos ofrecen préstamo de *hardware* que se encuentra en *datacenters*.

Estos servicios los puede proporcionar *Amazon*, *Microsoft* y muchas más empresas, en su mayoría lo que se debe indicar es la cantidad de *RAM* que se necesita, cantidad de memoria de almacenamiento, cantidad de núcleos que se requieren en el procesador y el sistema operativo que tendrá el sistema para que funcione. Una de las ventajas de la nube es que a las empresas se les da la flexibilidad de poder aumentar servidores o disminuirlos dependiendo de la carga de trabajo que se tenga en ese momento, para así gastar lo que en realidad se necesita, a esto se le llama Elasticidad que es una parte muy importante para tener sistemas en la nube y no tener problemas con ellos.

En el caso de la DICyG se tiene un *SITE* con *racks* que contienen servidores, en uno de ellos es donde está el sistema SIPRAC para que reciba las peticiones de los alumnos o del administrador en cualquier hora del día y también tiene su ventilación para enviar el calor generado por los servidores.

## **1.2 Ventajas de los sistemas basados en *Web*.**

### **1.2.1 Ventajas y Compatibilidad**

Para el desarrollo del sistema SIPRAC se eligió un sitio web, ya que la mayoría de sistemas operativos tiene navegador para visualizarlos. En cambio si se elegía otra plataforma, por ejemplo Android esta aplicación solo iba a llegar a celulares con ese Sistema operativo y además no se podría visualizar en una computadora. Entonces se hubiera desarrollado el mismo sistema para diferentes plataformas y esto hubiera reflejado un aumento en costos y tiempo en dichos desarrollos.

En un sitio web la visualización viene establecido por *HTML5* y *CSS3* que lee el navegador para mostrar los elementos que componen al sitio, esto gracias a su compatibilidad con los navegadores ya sea que estén en un celular, una computadora u otro dispositivo. Su funcionamiento viene establecido por protocolos como *HTTPS* o *HTTP* que permiten la

comunicación entre el cliente y el servidor para que así el servidor conteste con un sitio web o JSON en el caso del desarrollo de SIPRAC, algunos de los navegadores que están en el mercado son:

**Tabla 1.1 Navegadores web**

Firefox	
Chrome	
Opera	
Microsoft Edge	
Safari	
Tor	

La mayoría de ellos están disponibles en computadoras, tablets, celulares, por lo que permite mostrar el sistema a todas partes a lo que se le llama un sistema multiplataforma, es decir que no importa el dispositivo en que se visualice vas a ver la misma información, ya sea celular que tiene los sistemas operativos IOS o Android y para computadoras Chrome OS, Windows, Mac OSX, etc. Por lo que nos permite ver la misma información. Todos estos navegadores tienen algo en común que leen *HTML5*, *CSS3* y *Javascript* por lo que podremos utilizarlos para que funcione de la misma manera y en cualquier dispositivo el sistema.

### **1.2.1.1 HTML5**

*HTML* significa Lenguaje de Marcado para Hipertexto (*Hypertext Markup Language* en inglés) el cual nos ayuda en ingresar texto, imágenes, videos, dibujar, geolocalizar, etc. Que permite con sus muchas características visualizar nuestra página web en varios navegadores.



**Figura 1.5 HTML5**

En cada navegador se muestra la misma información, para ello se hace uso de las etiquetas (tags) de *HTML5* para especificarle a los navegadores de que se trata cada contenido, algunas de las etiquetas utilizadas en esta página son:

**Tabla 1.2 Ejemplos de etiquetas HTML5**

Nombre de la etiqueta	Uso
<code>&lt;!Doctype&gt;</code>	Esta etiqueta sirve para especificar que la versión del <i>HTML</i> es la cinco, anteriormente se especificaba con una etiqueta muy larga por ejemplo <code>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"&gt;</code> que es para especificar la versión 4 de <i>HTML</i> , y con ello el navegador sabrá cómo interpretar cada etiqueta correctamente.
<code>&lt;html&gt;..&lt;/html&gt;</code>	Esta etiqueta sirve para especificar que vamos a ingresar las etiquetas necesarias para que nuestra página se despliegue.
<code>&lt;header&gt;...&lt;/header&gt;</code>	Después de esta etiqueta se especifica el título de la página, los archivos <i>CSS3</i> que se utilizarán, al igual que los archivos <i>Javascripts</i> .
<code>&lt;body&gt;...&lt;/body&gt;</code>	Las etiquetas que prosiguen entre las etiquetas <i>body</i> es para especificar el

	contenido que visualizará el usuario en pantalla del navegador.
<code>&lt;link rel="stylesheet" href="normalize.css"&gt;</code>	Esta etiqueta se utiliza para especificar los archivos CSS3 que se utilizarán en la página web.
<code>&lt;title&gt;...&lt;/title&gt;</code>	Entre estas etiquetas se especifica el nombre de la página el cual se mostrará en la pestaña del navegador.
<code>&lt;ul&gt;...&lt;/ul&gt;</code>	Sirve para especificar una lista.
<code>&lt;a data-toggle="tab" href="#edo2"&gt;&lt;strong&gt;Inscripción&lt;/strong&gt;&lt;/a&gt;</code>	Esta etiqueta es a lo que se refiere <i>hypertext</i> es decir que al darle clic al texto mostrará otra página web.
<code>&lt;div&gt;...&lt;/div&gt;</code>	Se utilizo para dividir el cuerpo de la página que se encuentra entre la cabecera y el pie de la página, en el caso de la página principal se utilizó, para dividir la tabla donde se muestran las prácticas del campo con el mensaje de bienvenida.
<code>&lt;table&gt;...&lt;/table&gt;</code>	Se utiliza para crear tablas en la página web.

Y así hay una gran cantidad de etiquetas para mostrar información en nuestra página web, lo que hace el navegador es leer todas esas etiquetas las cuales les da un diseño específico a cada uno dependiendo del navegador, conforme se va especificando cada etiqueta en el archivo es como se va mostrando la información en la página web.

En las páginas *web* es importante saber manejar *HTML5*, ya que con él podemos especificar nuestro contenido que vamos a mostrarle al cliente. En el proyecto fue muy importante, porque sin él no podríamos agregar nuestros archivos *CSS3*, *Javascript*, las tablas de las prácticas de campo, los menús, al igual que la información del usuario como es: su nombre, las prácticas a las que está inscrita, los lugares disponibles en cada una, al igual que la asistencia de cada una de sus prácticas inscritas.

### 1.2.1.2 CSS3

CSS3 es una de las principales herramientas para desarrollar páginas web ya que nos permite realizar el diseño de cada una de las etiquetas de *HTML5*, CSS3 significa Hojas de Estilo en Cascada (*Cascading Style Sheets*) las cuales nos permiten la elección de color, tamaño del texto, el fondo de la página web, etc. Para así tener el mismo diseño en cualquier navegador, por lo que es muy importante, todo eso lo realiza gracias a las clases e ids que son atributos de las etiquetas en *HTML5* que nos ayudan a identificar a los elementos de una página web, cada clase o id se especifica con un nombre, ese nombre especificado en el archivo *HTML5* se utiliza en el archivo *CSS3* para darle el estilo deseado, por ejemplo:

**Tabla 1.3 Ejemplos para identificar clases e ids en CSS3**

Clases	Ids
<pre>.margeAbajo{   margin: 0; }</pre>	<pre>#practicaPL{   font-size: 17.5px; }</pre>



**Figura 1.6 CSS3**

Como se muestra en la tabla 1.3, las clases se identifican con un punto seguido del nombre asignado y entre llaves se especifican las propiedades de cada etiqueta y para aplicar el diseño deseado a un *id* se especifica con un símbolo de gato seguido del nombre asignado junto con sus propiedades para obtener el diseño deseado, pero entonces se estarán preguntando ¿Cuál es la diferencia entre una clase y un *id*?, la respuesta es que una clase puede ser aplicada a diferentes etiquetas del *HTML5*, en cambio un *id* en si nada más debe haber una etiqueta con ese nombre que se le asignó al *id* por lo que al haber uno o más etiquetas con el mismo *id* solamente al primero se le aplicará el diseño y a los demás ya no. Una de las ventajas del CSS3 es que a pesar de que tengan las etiquetas de *HTML5*

un diseño por default en cada navegador, con el CSS3 se puede estandarizar el diseño mostrándonos así el mismo color, tamaño de texto, etc. En cada navegador en que se visualice la página web.

En el diseño también se incluye lo que es la estructura de la página, por ejemplo que se tenga la parte superior el *header*, en el centro el mensaje de bienvenida, al igual que la tabla que permite visualizar las prácticas de campo Figura 1.8, pero hay veces que necesitamos mostrar un menú en la página de una forma tal que permita dividir el centro en dos columnas donde en la primera columna se visualice un menú vertical y a la derecha la información pertinente a cada uno de los *links* del menú como se muestra en la figura 1.9.

Sistema de Inscripción a Prácticas de Campo

- [Iniciar sesión](#)
- [Crear cuenta](#)

**ACCESO**

Favor de introducir tu usuario y tu contraseña para acceder al SIPRAC

Usuario

Contraseña

[Recuperar contraseña](#) [Cerrar](#)

**Inscripción**

**Alumno**

Es aquella persona que se encuentra de intercambio en la FI y proviene de otra universidad Nacional o internacional. También se considera Alumno a aquella persona que agotó el plazo máximo para estar inscrito, establecido en el Artículo 22 del R.G.I., sin haber concluido sus créditos; o bien, Ha estado inscrito dos veces en forma ordinaria en una misma asignatura sin haberla acreditado (Art. 33 R.G.I).

[Estudiante](#)

Es aquella persona que está formalmente inscrita en una asignatura de forma regular.

[Cerrar](#)

**Alumno**

Favor de venir a la DICyG con el coordinador de las prácticas para realizar tu inscripción.

Edificio "R" DICyG Segundo Piso Cubículo R-219

[Cerrar](#)

**Estudiante**

Proporciona tu número de cuenta, nombre completo en mayúsculas con sus respectivos acentos. Recuerda que este sistema sólo contiene información pertinente a las carreras de Ingeniería Civil e Ingeniería Geomática.

**1**  Número de cuenta

**2**  Nombre(s)

Figura 1.7 Página de bienvenida sin CSS3

Sistema de inscripción a Prácticas de Campo

Bienvenido al SPRAC. Como parte complementaria de las asignaturas teóricas, el departamento de Ingeniería en Sistemas, Planeación y Transporte de la DQSP pone a tu disposición el catálogo de prácticas de campo correspondientes al semestre 2021-2. Te invitamos a que revises y te inscribas a las prácticas que te corresponden. Sólo necesitas registrarte.

Prácticas correspondientes al semestre 2021-2

#	FECHA DE INSCRIPCIÓN	FECHA PARA FIRMAR	FECHA (A/BALDA)	FECHA (E/SALIDA)	GRUPO	PROFESOR	VISTA A	LUGAR DE SALIDA	LUGAR DE LLEGADA	CUPO	Categoría
1	2020-12-05	2020-12-08	2020-02-15 09:45:00	2020-02-15 16:00:00	2	M. FRANCISCO GRANADOS VILLALBETTE	PUESTO CENTRAL DE CONTROL DEL METRO	CALLE DE DELICIAS N° 67 METRO SALTO DEL AGUA	CALLE DE DELICIAS N° 67 METRO SALTO DEL AGUA	30	S
3	2020-12-05	2020-12-05	2020-02-02 09:45:00	2020-02-02 16:00:00	5	M. I. MAURO TOSAH HUERTA	TREN SUBURBANO	ESTACION BUENAVISTA	ESTACION BUENAVISTA	30	S
4	2020-12-05	2020-12-05	2020-02-01 07:00:00	2020-02-01 16:00:00	1	ING. GUILLERMO ESPINEL CADAVEÑA	AEROPUERTO INTERNACIONAL DE CUERNAVACA, MORELOS	EE 10 SUR Y CERRO DEL AGUA (METRO COPOLCO)	EE 10 SUR Y CERRO DEL AGUA (METRO COPOLCO)	30	S
5	2020-12-05	2020-12-05	2020-02-14 09:45:00	2020-02-14 16:00:00	4	ING. GUILLERMO ESPINEL CADAVEÑA	PUESTO CENTRAL DE CONTROL DEL METRO	CALLE DE DELICIAS N° 67 METRO SALTO DEL AGUA	CALLE DE DELICIAS N° 67 METRO SALTO DEL AGUA	30	NO
8	2020-12-17	2020-12-20	2020-12-02 07:00:00	2020-12-02 16:00:00	3	DR. ALEJANDRO LUGO VARGAS	AEROPUERTO INTERNACIONAL DE PUEBLA, PUE.	EE 10 SUR Y CERRO DEL AGUA (METRO COPOLCO)	EE 10 SUR Y CERRO DEL AGUA (METRO COPOLCO)	28	NO
10	2020-12-05	2020-12-05	2020-02-28 07:00:00	2020-02-28 16:00:00	1	ING. GUILLERMO ESPINEL CADAVEÑA	AEROPUERTO INTERNACIONAL DE CUERNAVACA, MORELOS	EE 10 SUR Y CERRO DEL AGUA (METRO COPOLCO)	EE 10 SUR Y CERRO DEL AGUA (METRO COPOLCO)	30	S
12	2020-12-05	2020-12-05	2020-04-25 12:00:00	2020-04-27 21:00:00	VARIOS	VARIOS	PUESTO Y TRABAJOS DE AMPLIACIÓN DEL PUESTO DE VESPAQUIZ, AEROPUERTO INTERNACIONAL DE VESPAQUIZ	EE 10 SUR Y CERRO DEL AGUA (METRO COPOLCO)	EE 10 SUR Y CERRO DEL AGUA (METRO COPOLCO)	40	S
13	2020-12-05	2020-12-05	2020-05-09 07:00:00	2020-05-09 16:00:00	4	ING. GUILLERMO ESPINEL CADAVEÑA	AEROPUERTO INTERNACIONAL DE CUERNAVACA, MORELOS	EE 10 SUR Y CERRO DEL AGUA (METRO COPOLCO)	EE 10 SUR Y CERRO DEL AGUA (METRO COPOLCO)	30	S
14	2021-01-08	2021-01-07	2021-01-11 09:00:00	2021-01-11 21:00:00	VARIOS	VARIOS	INSTITUTO MEXICANO DEL TRANSPORTE, TERMINAL INTERNACIONAL LOGÍSTICA DE HEDDING Y AEROPUERTO INTERNACIONAL DE CUERNAVACA	EE 10 SUR Y CERRO DEL AGUA (METRO COPOLCO)	EE 10 SUR Y CERRO DEL AGUA (METRO COPOLCO)	27	S
15	2020-12-05	2020-12-05	2020-02-17 09:45:00	2020-02-17 16:00:00	1	ING. GUILLERMO ESPINEL CADAVEÑA	TREN SUBURBANO	ESTACION BUENAVISTA	ESTACION BUENAVISTA	30	S

INTEGRACION DE PROYECTOS

Universidad Nacional Autónoma de México, Facultad de Ingeniería  
 Dirección de Ingeniería Civil y Geométrica © 2017  
 Ciudad Universitaria, México - CDMX, Av. Universidad 3300.  
 Copyright © 2017. Todos los derechos reservados.

Redes sociales

Figura 1.8 Página de bienvenida con CSS3

Sistema de inscripción a Prácticas de Campo

ING. ERNESTO BENEVIDADO SANCHEZ

Validación

- Constancias de Asistencia
- Reglamento
- Listas
- Inscripción a materias o materias
- Ver materias inscritas del alumno
- Passar lista alumnos
- Generar lista de asistencia
- Generar Estadísticas
- Estación de prácticas
- Agregar o eliminar prácticas
- Agregar o eliminar alumnos

Agregar alumno

Numero de cuenta

Nombre del alumno Apellido paterno del alumno Apellido materno del alumno

Calle Número Colonia

Delegación/Municipio Estado Federativa Código postal

Correo Número de afiliación a su seguro social Fecha de nacimiento del alumno

Nombre(s) del beneficiario Apellido paterno del beneficiario Apellido materno del beneficiario

Teléfono del beneficiario Sexo Escoge el tipo de estudiante

ENVIAR GUARDAR

Inscribir alumno a una materia

Numero de cuenta Materia Grupo

ENVIAR GUARDAR

Figura 1.9 Menú vertical

Gracias a HTML5 y CSS3 se visualiza en varios navegadores de manera correcta:

## Firefox

The screenshot shows the Firefox browser window displaying the 'Sistema de Inscripción a Prácticas de Campo' website. The page features a dark blue header with the title and navigation links 'Iniciar sesión' and 'Crear cuenta'. Below the header is a large banner image of a coastal city. A text box below the image contains a welcome message. A dark blue bar below the banner lists 'Prácticas correspondientes al semestre 2021-2' with two menu items: 'SISTEMAS DE TRANSPORTE' and 'INTEGRACION DE PROYECTOS'. The footer contains university information and social media icons.

Sistema de Inscripción a Prácticas de Campo Iniciar sesión   Crear cuenta



Bienvenido al SIPRAC Como parte complementaria de las asignaturas teóricas, el departamento de Ingeniería en Sistemas, Planeación y Transporte de la DICyG pone a tu disposición el catálogo de prácticas de campo correspondientes al semestre 2021-2. Te invitamos a que revises y te inscribas a las prácticas que te correspondan. Sólo necesitas registrarte.

**Prácticas correspondientes al semestre 2021-2**

- SISTEMAS DE TRANSPORTE
- INTEGRACION DE PROYECTOS

Universidad Nacional Autónoma de México, Facultad de Ingeniería Redes sociales  
División de Ingenierías Civil y Geomática © 2021     
Ciudad Universitaria, México CDMX. Av. Universidad 3000.

Figura 1.10 Página de bienvenida en *Firefox*

## Opera

The screenshot shows the Opera browser window displaying the same 'Sistema de Inscripción a Prácticas de Campo' website as in the Firefox version. The layout and content are identical, including the header, banner image, welcome text, practice list, and footer.

Sistema de Inscripción a Prácticas de Campo Iniciar sesión   Crear cuenta



Bienvenido al SIPRAC Como parte complementaria de las asignaturas teóricas, el departamento de Ingeniería en Sistemas, Planeación y Transporte de la DICyG pone a tu disposición el catálogo de prácticas de campo correspondientes al semestre 2021-2. Te invitamos a que revises y te inscribas a las prácticas que te correspondan. Sólo necesitas registrarte.

**Prácticas correspondientes al semestre 2021-2**

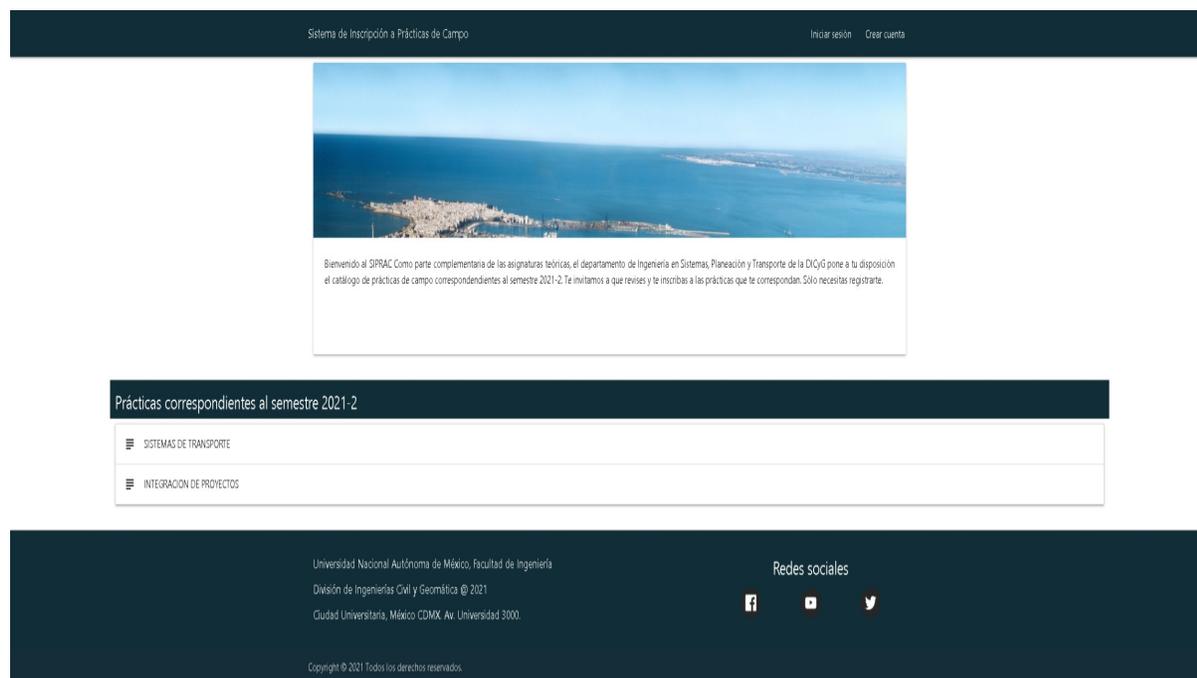
- SISTEMAS DE TRANSPORTE
- INTEGRACION DE PROYECTOS

Universidad Nacional Autónoma de México, Facultad de Ingeniería Redes sociales  
División de Ingenierías Civil y Geomática © 2021     
Ciudad Universitaria, México CDMX. Av. Universidad 3000.

Copyright © 2021 Todos los derechos reservados.

Figura 1.11 Página de bienvenida en *Opera*

## Microsoft Edge



**Figura 1.12** Página de bienvenida en *Microsoft Edge*

### 1.2.1.3 *PHP*

Otra tecnología que se emplea en el desarrollo de páginas web es el lenguaje de programación orientado a objetos *PHP* el cual le da a la página la funcionalidad de mostrar contenido dinámico, se refiere a que dependiendo de la petición del usuario la página mostrará diferentes datos dependiendo de la información que el usuario envió, con este lenguaje de programación todo lo que solicite el usuario al servidor de *PHP* responderá con una página web o *JSON*.



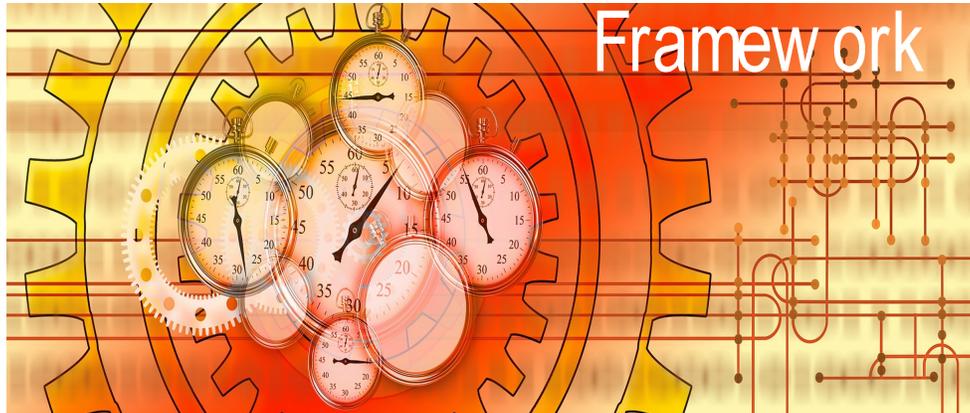
**Figura 1.13** Imagen del lenguaje de programación *PHP*

Principalmente se utiliza en las páginas web para hacer sistemas en que involucre información específica para cada usuario cuando éste ingrese con su usuario y contraseña, en nuestro caso se utiliza para que el alumno al registrarse se pueda inscribir

en prácticas de campo, no todas las prácticas que están en el sistema se le mostrará al alumno; esto dependerá de las materias a las cuales está inscrito, también *PHP* se utiliza mucho con la base de datos *MYSQL* que en este sistema funciona para recuperar la información pertinente al usuario, en *SIPRAC* se utiliza para mostrar las prácticas a las cuales está inscrito el usuario, para las inscripciones a las mismas, para visualizar las prácticas ya inscritas entre otras funciones. Su utilización en este sistema es porque la infraestructura de la *DICyG* está hecha en servidores *Apache* el cual maneja *PHP* y también porque todos los sistemas están hechos con *PHP*.

### 1.3 Frameworks

Es una tecnología que permite al desarrollador de sistemas, implementar rápidamente una metodología para desarrollar ágilmente su proyecto, es lo que se llama un esqueleto en el cual te apoyas para concentrarte en lo que en realidad importa el cómo quieres que quede tu proyecto y no cómo lo vas a desarrollar desde cero, no tendrás que pensar cómo se administraran los archivos, ni cómo se realizan desde cero ciertas acciones como recuperar información del servidor o cómo mostrar mensajes al usuario para hacerle saber que se equivocó al proporcionar sus datos.



**Figura 1.14 Frameworks**

#### 1.3.1 Codeigniter

En el desarrollo web existen muchos *frameworks* que ayudan hacer sitios web uno de los principales que se utiliza es *CodeIgniter* el cual es un *framework* enfocado en el desarrollo web con *PHP*.

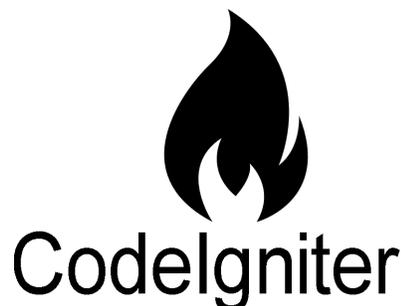
Se escogió *CodeIgniter* ya que es fácil de aprender y porque en la *DICyG* se va a empezar a actualizar o crear los sistemas con él, además tiene varias ventajas por enlistar algunas:

- una de ellas es la posibilidad de conectar fácilmente *PHP* con varias bases de datos como *PostgreSQL*, *MYSQL*, *SQLite*, *ORACLE* entre otros, la base que se utilizo en SIPRAC es *MYSQL*.
- También agiliza lo que es el abrir la conexión a la base de datos para solicitar datos o para guardarlos.
- La metodología que se utiliza es Modelo Vista Controlador la cual nos permite tener administrados correctamente los documentos para encontrarlos rápidamente y para que la programación se divida en diferentes sectores uno es para la administración de los datos, otra es para la lógica del sistema y por último para tener la estructura de como visualizará el usuario la información en los navegadores.
- Asimismo provee lo que se llama rutas amigables, estas rutas le permiten al usuario solicitar una parte del sistema al hacer clic en un *link* o ingresando la *URL* en un navegador. Una *URL* con rutas amigables sería

<http://siprac.com/administrador>

En cambio, sino se utilizaran, las rutas tendrían que recurrir al archivo *index.php* de *CodeIgniter* para poder redirigir la solicitud al archivo correcto, un ejemplo de ello es la siguiente *URL* <http://siprac.com/index.php/adiminstrador>,

- Por último, *CodeIgniter* ya tiene implementada el enfoque orientado a objetos en donde ya se tienen clases programadas por los desarrolladores del *framework*. Estas clases se heredan con el fin de utilizar los métodos que contienen, también ayudan fácilmente en la validación de formularios, ya que tiene sus propias expresiones regulares para validar cada campo que llenó el usuario.



**Figura 1.15 CodeIgniter**

Las carpetas más importantes de este *framework* son:

1. La carpeta "*controllers*" ya que ésta guarda los controladores que vas creando, es decir el que lleva la lógica de cómo funcionan las páginas web que creas.
2. La carpeta "*models*", ésta lleva los archivos que utilizan los controladores para hacer las consultas a la base de datos.

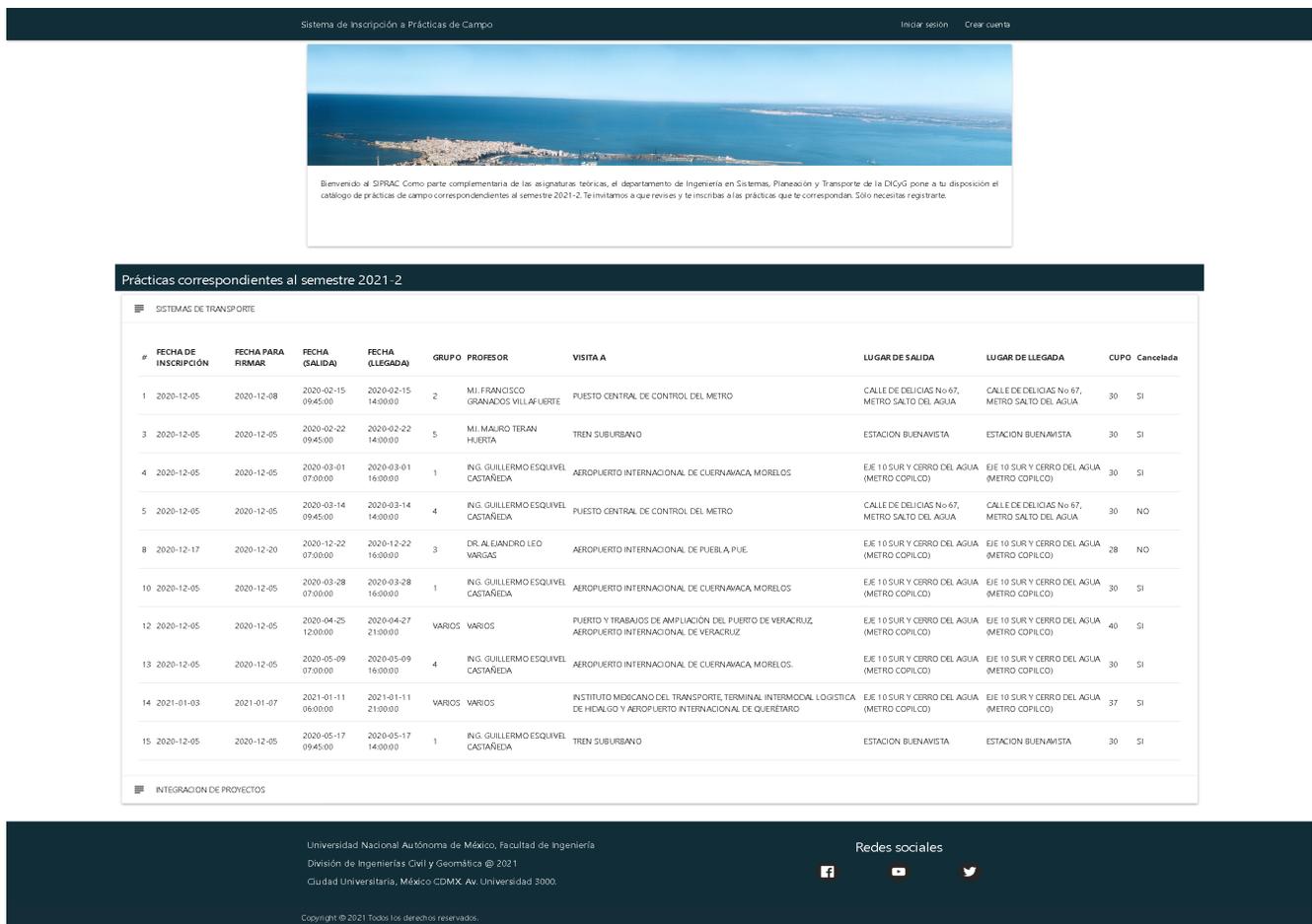
3. La carpeta “views”, ésta lleva los archivos que envían los controladores al regresar una página web (*HTML5, CSS3, Javascript*).

Por último, uno de los archivos importantes es el llamado “index.php” que utiliza apache2 para mostrar correctamente el sistema.

### 1.3.2 Materialize

*Materialize* es un *framework* que nos permite aplicar el diseño deseado de nuestra página y hacerlo *responsive*, es decir que el contenido se acomoda de diferente manera de acuerdo con el tamaño de la pantalla en el dispositivo que se visualiza, otra ventaja es que ya tiene predefinido varios diseños en los botones, *headers*, *modals* entre otras cosas. Además, que permite acomodar fácilmente cada elemento en la página web y por si fuera poco *Materialize* es sencillo de agregar a cualquier proyecto web.

Las siguientes imágenes fueron sacadas desde *Firefox* simulando diferentes dispositivos para mostrar el *responsive design* con *Materialize*:



**Figura 1.16** Página de bienvenida en un monitor para pc marca **SCEPTRE** 2560x1080

☰ Sistema de Inscripción a Prácticas de Campo



Bienvenido al SIPRAC. Como parte complementaria de las asignaturas teóricas, el departamento de Ingeniería en Sistemas, Planeación y Transporte de la DICyG pone a tu disposición el catálogo de prácticas de campo correspondientes al semestre 2021-2. Te invitamos a que revises y te inscribas a las prácticas que te correspondan. Solo necesitas registrarte.

**Prácticas correspondientes al semestre 2021-2**

☰ SISTEMAS DE TRANSPORTE

#	FECHA DE INSCRIPCIÓN	FECHA PARA FIRMAR	FECHA (SALIDA)	FECHA (LEGADA)	GRUPO	PROFESOR	VISITA A
1	2020-12-05	2020-12-08	2020-02-15 09:45:00	2020-02-15 14:00:00	2	M.I. FRANCISCO GRANADOS VILLAFUERTE	PUERTO CENTRAL CONTRA METRO
3	2020-12-05	2020-12-05	2020-02-22 09:45:00	2020-02-22 14:00:00	5	M.I. MAURO TERAN HUERTA	TREN SUBURB
4	2020-12-05	2020-12-05	2020-03-01 07:00:00	2020-03-01 16:00:00	1	ING. GUILLERMO ESQUIVEL CASTAÑEDA	AEROPU. INTERNA/ DE CUERNA MORELO
5	2020-12-05	2020-12-05	2020-03-14 09:45:00	2020-03-14 14:00:00	4	ING. GUILLERMO ESQUIVEL CASTAÑEDA	PUERTO CENTRAL CONTRA METRO
8	2020-12-17	2020-12-20	2020-12-22 07:00:00	2020-12-22 16:00:00	3	DR. ALEJANDRO LEO VARGAS	AEROPU. INTERNA/ DE PUEB
10	2020-12-05	2020-12-05	2020-03-28 07:00:00	2020-03-28 16:00:00	1	ING. GUILLERMO ESQUIVEL CASTAÑEDA	AEROPU. INTERNA/ DE CUERNA MORELO
12	2020-12-05	2020-12-05	2020-04-25 12:00:00	2020-04-27 21:00:00	VIARIOS	VIARIOS	PUERTO TRABAJOS AMPLIA DEL PUE VERACRU. AEROPU. INTERNA/ DE VERACRU.
13	2020-12-05	2020-12-05	2020-05-09 07:00:00	2020-05-09 16:00:00	4	ING. GUILLERMO ESQUIVEL CASTAÑEDA	AEROPU. INTERNA/ DE CUERNA MORELO
14	2021-01-03	2021-01-07	2021-01-11 06:00:00	2021-01-11 21:00:00	VIARIOS	VIARIOS	INSTITUTO MEXICANO DE TRANSPORTE TERMINALES AEROPU. HIDALGO AEROPU. INTERNA/ DE QUERETARO
15	2020-12-05	2020-12-05	2020-05-17 09:45:00	2020-05-17 14:00:00	1	ING. GUILLERMO ESQUIVEL CASTAÑEDA	TREN SUBURB

☰ INTEGRACION DE PROYECTOS

Universidad Nacional Autónoma de México, Facultad de Ingeniería  
 División de Ingenierías Civil y Geomática © 2021  
 Ciudad Universitaria, México CDMX. Av. Universidad 3000.

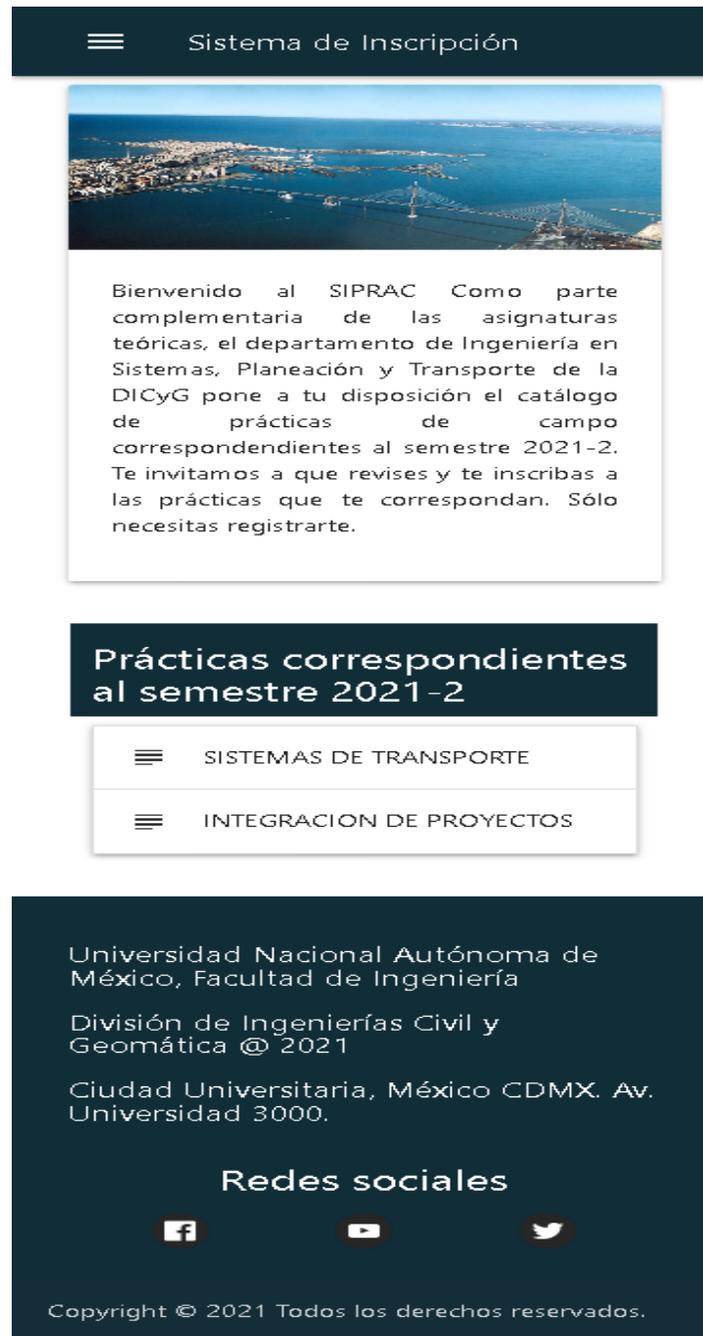
Redes sociales





Copyright © 2021 Todos los derechos reservados.

**Figura 1.17** Página de bienvenida en *Firefox* simulando una pantalla *iPad*, pero en este caso adquiriendo toda la página sin respetar el *viewport*.



**Figura 1.18** Página de bienvenida en *Firefox* simulando la pantalla de un celular *Galaxy S9* sin respetar el *viewport*.

Con esto se demostró que dependiendo de la pantalla en el dispositivo los elementos de la página web se acomodan de diferente manera para visualizarse correctamente. Para el diseño de las páginas web fue fácil su implementación ya que *Materialize* tiene documentación fácil de entender y su implementación en comparación con una implementación de CSS3 puro lleva menos tiempo.

### 1.3.3 *Jquery*

*Javascript* es un lenguaje de programación interpretado por el navegador, esto es posible gracias a un motor que es distinto para cada uno de los navegadores que interpreta línea por línea a *Javascript*, en caso de *Google Chrome* su motor se llama V8 el cual está hecho con el lenguaje de programación C++, pero la mayoría de las veces el utilizar *Javascript* se vuelve un poco cansado por ello se eligió *Jquery* que facilitar la mayoría del trabajo.

*Jquery* es un *framework* desarrollado con *Javascript* que facilita la manipulación de los elementos de la página web como cambiar texto, agregar texto o imágenes, eliminar texto, agregar estilo y además la selección de los elementos por medio de clases o ids lo hace mucho más sencillo, pero la función principal por la cual se eligió fue para enviar solicitudes de rutas específicas del sitio web para mostrar o verificar información pertinente como por ejemplo verificar las credenciales del alumno para saber si existe en la base de datos en caso de que sean correctos se crea la sesión con los datos pertinentes del alumno y se redirecciona a la página del alumno en caso contrario se muestra un mensaje indicando que el usuario o contraseña son incorrectos, todo esto gracias a las rutas que regresan *JSON* y no una página web como tal.

En una página web el usuario interactúa con los elementos del sitio al dar clic en un botón para inscribirse a una práctica de campo, esa acción ejecuta código de *Jquery* que manda la información pertinente al servidor con el fin de almacenar la inscripción en la base de datos, su respuesta hace que se muestre un modal que le informa que su inscripción fue realizada con éxito.

Gracias a todas estas tecnologías fue posible el desarrollo del sistema SIPRAC, su uso fue por la infraestructura que tiene la DICyG y la facilidad para utilizarlas.

## Capítulo 2. Metodologías de desarrollo

Para el desarrollo de un sistema se necesita un orden, saber cuáles son los requerimientos, se necesita planeación, pero lo más importante comunicación y saber correctamente lo que el cliente necesita de acuerdo con los requerimientos, todo esto y más conforma al proceso de desarrollo del sitio web. Este proceso que se tiene para su desarrollo se tiene que evaluar y mejorar, por ello se va a utilizar CMMI-DEV.

### 2.1 CMMI-DEV

Para la creación del sitio web (SIPRAC) se llevó a cabo una serie de pasos que fueron tomados del CMMI-DEV<sup>2</sup> (*Capability Maturity Model Integration for Development / Integración de los Modelos de Madurez de Capacidades*) que te permite saber qué pasos hay que seguir para mejorar e institucionalizar (aplicar el proceso de planeación del proyecto, ya mejorado, en cada uno de los desarrollos que se tengan en la DICyG), existen veintiséis áreas de proceso, pero solo se seleccionó planeación del proyecto (Project Planning), al final el proceso mejorado será implementado en el desarrollo del sistema SIPRAC, pero hay que tener en cuenta que este documento no te dice el cómo hacer la mejora, sino qué tienes que hacer. Cada uno de las áreas de proceso tienen componentes comunes, las cuales se muestran en la siguiente lista:

1. Declaración de propósito: Describe el proposito del área de proceso.
2. Notas introductorias: Describen el concepto principal que cubre el área de proceso.
3. Áreas de proceso relacionadas: Se recomienda más no es obligatorio implementar otras áreas de proceso antes de implementar la que se desea.
4. Objetivos específicos: Son objetivos que ayudarán a mejorar el proceso, en este caso el proceso de planeación del proyecto.
5. Practicas específicas: Son aquellas que describen lo que hay que hacer para lograr los objetivos específicos.
6. Objetivos genericos: Se utilizan en todas las áreas de proceso para estandarizar el proceso mejorado.

Cada área de proceso, objetivos generales y prácticas generales están definidos en el documento de CMMI-DEV. Para su implementación hay dos niveles, el primero llamado “*Capability levels*” y el segundo “*Maturity levels*”. El primer modelo nos permite escoger que proceso mejorar de los veintiséis que existen, el otro nivel te proporciona la opción de escoger un conjunto de áreas de proceso relacionadas con los cuales mejorar los procesos existentes en la empresa.

---

<sup>2</sup> Carnegie Mellon University. (2010). CMMI for Development, Version 1.3. United States of America: Software Engineering Institute.

El área de proceso a implementar en este proyecto es “*Project Planning*” por lo que se hará uso de “*Capability levels*”, pero solo se llegará hasta el nivel “*Performed*”. Para llevar a cabo la mejora en el proceso de planeación del proyecto SIPRAC con el área de proceso, lo primero a hacer es utilizar el primer objetivo general “*GG1 Achieve Specific Goals*” el cual nos indica que se deben cumplir los objetivos específicos en este caso del área de proceso “*Project Planning*”, por ello se comenzará con la definición de las fases del ciclo de vida del proyecto, que es una de las prácticas específicas del objetivo específico llamado “*Establish Estimates*” o establecer estimaciones, ya que considero que debería ser la primera práctica en realizarse porque lleva correlación con las demás prácticas de los objetivos.

### **2.1.1 Specific Goal - Establecer estimaciones (*Establish Estimates*)**

En este apartado se genera documentación del proyecto que incluyen los requerimientos, la planeación de creación de cada una de las especificaciones y así el tiempo estimado y requerido para finalizar el sistema.

#### **2.1.1.1 Specific Practice - Definición de las fases del ciclo de vida del proyecto**

En el proyecto se utilizó el modelo llamado modelo de desarrollo de software en espiral que en esencia tiene cuatro fases: planeación, análisis de riesgos, implementación y validación. Estas fases van en ese orden, por lo que, no se pueden saltar planeación a implementación, la explicación de cada una es la siguiente:

- Planeación: Es la parte del modelo en donde se especifica qué se va a desarrollar en el ciclo. Para esta etapa se realizaban juntas con el jefe del departamento y se especificaban las partes del sitio web a desarrollar, por ejemplo:
  - En primera instancia se desarrolló la página de bienvenida, por lo que se estableció lo que iba a tener, el *header*, el mensaje de bienvenida, cómo se iban a presentar las prácticas de campo y el *footer*.
- Análisis de riesgos: En esta segunda parte se busca todo evento que pudiese afectar el alcance del ciclo. En el desarrollo del sitio web se aclaró que el tiempo de entrega puede ser mucho mayor, ya que no me encargaba solo de los sistemas, sino también apoyaba en soporte técnico, el cual, nunca se tenía con certeza el número de veces que se detenía el desarrollo del sitio web por atenderlos. En esta fase también se hacen los prototipos de las parte(s) del sitio web que se acordaron, por lo que en el primer ciclo se realizó la página de bienvenida, pero sin ningún funcionamiento, es decir, los formularios no funcionaban, las prácticas que se mostraban no se recuperaban de la base de datos, sino que, simplemente se ponían manualmente en la página. Una vez que se establecieron los riesgos y se aceptó el prototipo se sigue con la siguiente fase.

- Implementación: En esta fase se genera código para hacer que la página funcione correctamente, como por ejemplo que se recuperen las prácticas de la base de datos y pasarlas a la vista, que los formularios ya envíen los datos.
- Validación: Aquí se muestra la parte acordada y se verifica que los riesgos se eliminaron, en el desarrollo de este sistema no se pudo mitigar el riesgo principal que es el soporte técnico, lo que se realizó fue que la persona a cargo de los sistemas solo daba soporte técnico sino había nadie más. Además, se tenía una junta para mostrar lo acordado.

Todas estas fases se repiten una y otra vez hasta terminar con el sitio web.

### **2.1.1.2 Specific Practice - Estimar el alcance del proyecto (*Estimate the Scope of the Project*)**

En esta práctica se solicita crear un *Work Breakdown Structure*<sup>3</sup> (*WBS*), por lo que se redactaron dos documentos principales llamados “*Product charter*” y “*Product description*” los cuales establecen los requerimientos del sistema, junto con sus entregables que conllevan a la creación del diagrama *WBS (Work Breakdown Structure)* que es una estructura jerárquica en donde el nivel más alto es el sistema completo y conforme se va bajando de nivel se va visualizando los entregables (*WBS packages*) que establecen la construcción del sistema conforme se vayan completando.

#### **2.1.1.2.1 Project Charter**

##### **Descripción del proyecto (*Project Overview*)**

Este proyecto se desarrollará para la administración de las prácticas de campo, principalmente para dos materias en específico: Sistemas de transporte e Integración de Proyectos. Su desarrollo será un sitio web para ser visualizado en diferentes dispositivos y estará alojado en los servidores del edificio de la DICyG.

##### **Sección I. Propósito del proyecto (*Project Purpose*)**

Este sitio web apoyará a los ingenieros a cargo de la administración de las prácticas, haciendo posible la transición de su administración manual a digital, minimizando el trabajo a realizar y ofreciendo funcionalidades que proporcione el control y administración de los datos almacenados, además de proporcionar una herramienta a los alumnos para poder inscribirse a cualquier práctica que ellos deseen.

---

<sup>3</sup> Eric S. Norman, Shelly A. Brotherton & Robert T. Fried. (2008). *Work Breakdown Structures: Foundation for project management excellence*. Hoboken, New Jersey & Canada: John Wiley & Sons, Inc..

## **Sección II. Alcance del proyecto (*Project Scope*)**

El desarrollo de este sitio web involucra la infraestructura de la DICyG, por lo que se utilizará el siguiente *software* para su alojamiento, desarrollo del *backend* y *frontend*:

- *Apache 2*
- *PHP 5.6*
- *HTML 5*
- *CSS 3*
- *Codelgniter*
- *Excel framework*
- *PDF framework*

El sitio web involucra:

- La página de bienvenida
- La página del alumno al iniciar sesión
- La página de administración de las prácticas de campo.

## **Sección III. Objetivos del proyecto (*Project Objectives*)**

El progreso del sitio web será de la siguiente manera:

1. Diseño de cada página del sitio web completa y aprobada.
2. Diseño de la base de datos completa.
3. Controladores en Codelgniter completa.
4. Uso de JQuery para el envío de la información al backend y solicitud de información completa.
5. Mostrar cómo se usa el sistema.

## **Administración de cambios (*Change Management*)**

Los cambios posteriores al diseño de las páginas aprobadas se harán a través de un formato en el cual se establezca ¿quién quiere que se realice el cambio? y ¿qué quiere cambiar? con este formato se muestra por qué se alargó el desarrollo del sistema y el impacto en el tiempo de entrega directamente a la encargada de la Unidad de Cómputo.

### **2.1.1.2.2 Descripción del alcance del producto (*Product Scope Description*)**

Los usuarios a los cuales va dirigido el desarrollo del sitio web SIPRAC es para los estudiantes de la carrera de ingeniería civil de la Facultad de Ingeniería perteneciente a la Universidad Nacional Autónoma de México, ubicada en Cto. Escolar 04360, C.U., Coyoacán, 04510 Ciudad de México, CDMX. La mayoría (si no que todos) los estudiantes tienen un celular para comunicarse y para navegar en internet, pero se puede dar el caso de que no tengan datos para navegar o conexión de internet vía *WIFI*, entonces podrían

solicitar una computadora en el edificio M o en el edificio principal. Con esta información sabemos que debemos abarcar una variedad de tamaños de pantalla, entonces el sitio web será responsivo, para acomodar los elementos que se muestran de manera dinámica en las diferentes pantallas.

Este sitio web facilitará al estudiante la inscripción a las prácticas de campo, al acceder a este sitio web desde cualquier parte del país con el fin de que puedan cumplir con un requerimiento más en sus materias.

También se tiene cómo usuario al ingeniero encargado de las prácticas de campo, el cual por comodidad accederá desde su computadora personal y se le facilitará la administración de las prácticas de campo con funcionalidades como imprimir la lista de asistencias de cada práctica, inscribir a los alumnos que no estén inscritos en alguna de las materias ya mencionadas, pasar lista de asistencia en cada práctica o imprimir las constancias de asistencia, entre otras funciones.

Actualmente la administración manual de las prácticas de campo se realiza con las siguientes actividades:

1. Los alumnos interesados deben visitar al ingeniero encargado en su cubículo en el edificio de la DICyG para inscribirse en las prácticas de campo con el fin de cumplir con un requerimiento más en sus materias, el ingeniero va llenando un formato en excel para posteriormente imprimirlo.
2. Luego el ingeniero vuelve a solicitar la presencia de los alumnos que solicitaron su inscripción a alguna de las prácticas, para firmar el formato mencionado y pagar la cuota del seguro que es de \$1 peso.
3. El proceso termina con la asistencia de los alumnos a la práctica de campo.

La creación del sitio web permitirá a los administradores reducir su tiempo en las siguientes actividades:

- La creación de los documentos por práctica para que los alumnos firmen. Este documento será generado por el sistema y ya no por el ingeniero.
- Tener que solicitar la presencia de los alumnos solamente dos veces para su firma, para así, asegurar que el alumno quiere ir a la práctica y no desperdiciar un lugar, además para firmar.
- La impresión de constancias para los alumnos.
- El pasar la lista de asistencia por práctica.
- Estadísticas
- Validar la asistencia de los alumnos
- Generar comprobantes de asistencias
- Generar listas de asistencias

El siguiente diagrama de caso de uso servirá para explicar el sistema SIPRAC y el comportamiento que tendrán los diferentes actores con él:

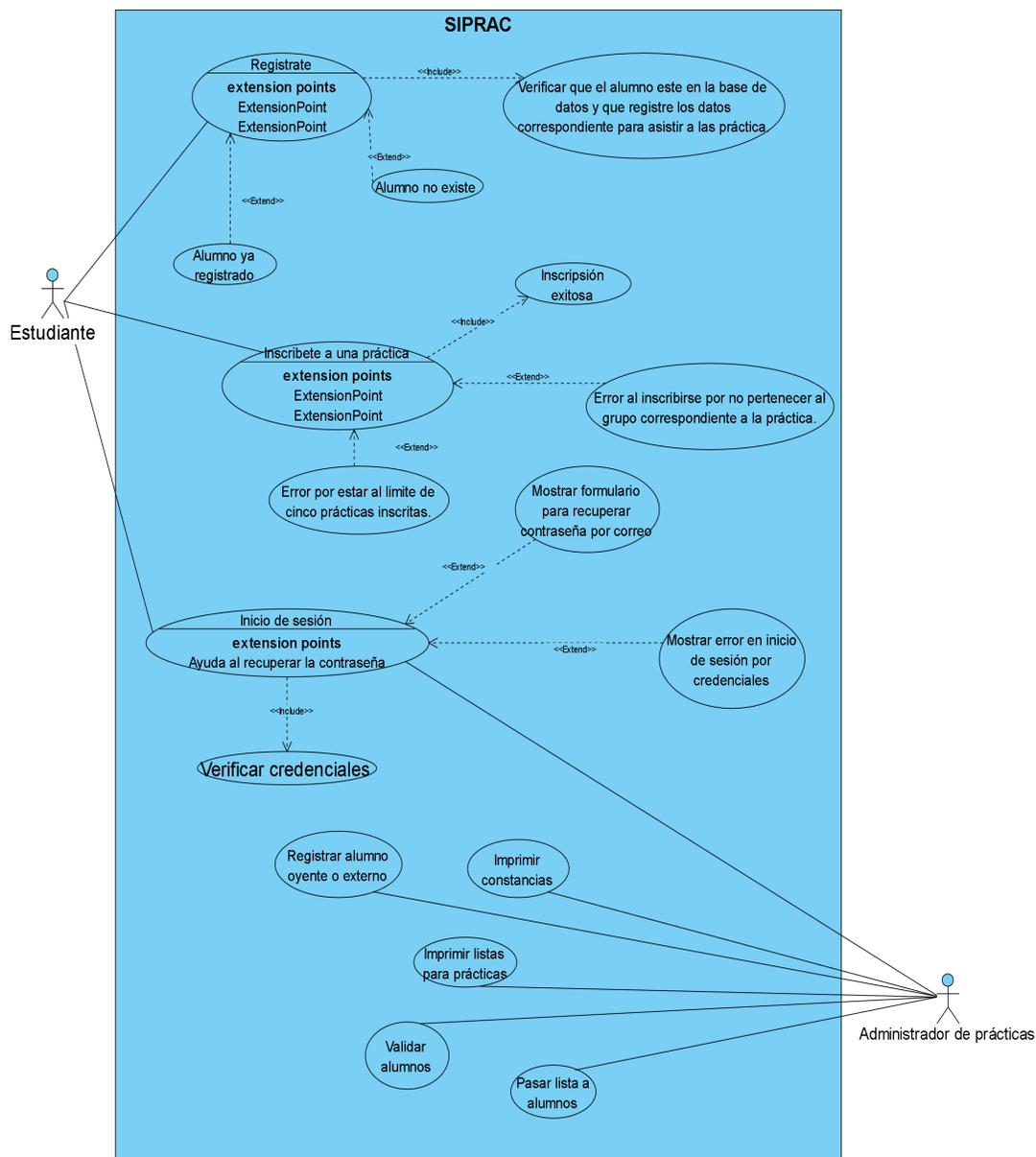
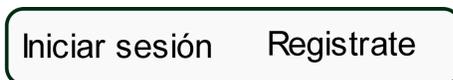


Figura 2.1 Diagrama de caso de uso

La página de bienvenida proporcionará las siguientes funcionalidades y características:

**Un Header**



**Figura 2.2 Header de la página principal**

El usuario del sitio web al dar clic en la opción iniciar sesión deberá abrir un modal que contenga un formulario con los siguientes requerimientos:

<b>Iniciar sesión</b>
<ul style="list-style-type: none"> <li>• Usuario</li> <li>• Contraseña</li> </ul>
Agregar una opción para recuperar la contraseña con el número de cuenta.

**Tabla 2.1 Requerimientos para el formulario de inicio de sesión**

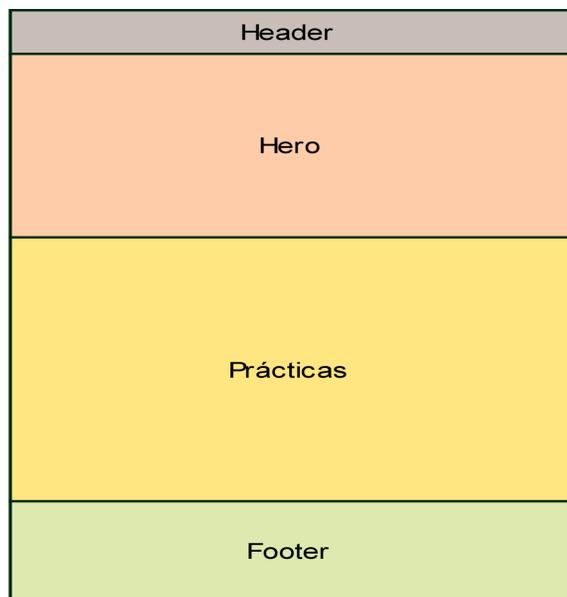
Si el usuario elige la opción regístrate se mostrará un modal con un formulario con los siguientes requerimientos:

<b>Regístrate</b>
<ul style="list-style-type: none"> <li>• Número de cuenta</li> <li>• Nombre</li> <li>• Apellido Paterno</li> <li>• Apellido Materno</li> </ul>
El alumno al ingresar los datos, se verificarán contra la base de datos, en caso de existir, mostrar un formulario más para ingresar los datos solicitados por la universidad, para asistir a las prácticas de campo.
Los datos que se deben solicitar son: <ul style="list-style-type: none"> <li>• Calle del alumno</li> <li>• Número</li> <li>• Colonia</li> <li>• Alcaldía/Municipio</li> <li>• Entidad Federativa</li> <li>• Código Postal</li> <li>• Número de afiliación a su seguro social</li> <li>• Correo del alumno</li> <li>• Fecha de nacimiento</li> <li>• Nombre completo del beneficiario para su seguro de vida</li> <li>• teléfono del beneficiario</li> </ul>
Al finalizar de enviar los datos, se le enviará al alumno su contraseña por correo.

**Tabla 2.2 Requerimientos para las páginas de inicio de sesión y Registro del alumno**

También se tendrá un **Hero** donde con un mensaje se les dé la bienvenida a los alumnos, debajo de este elemento se mostrarán las prácticas de campo por materia y hasta abajo de toda la página un *footer*.

Por lo tanto, la estructura completa de la página de bienvenida se muestra en la siguiente imagen:



**Figura 2.3 Estructura de la página de bienvenida**

El diseño final no deberá cumplir estrictamente la estructura de la imagen de arriba, pero sí debe tener los elementos mostrados.

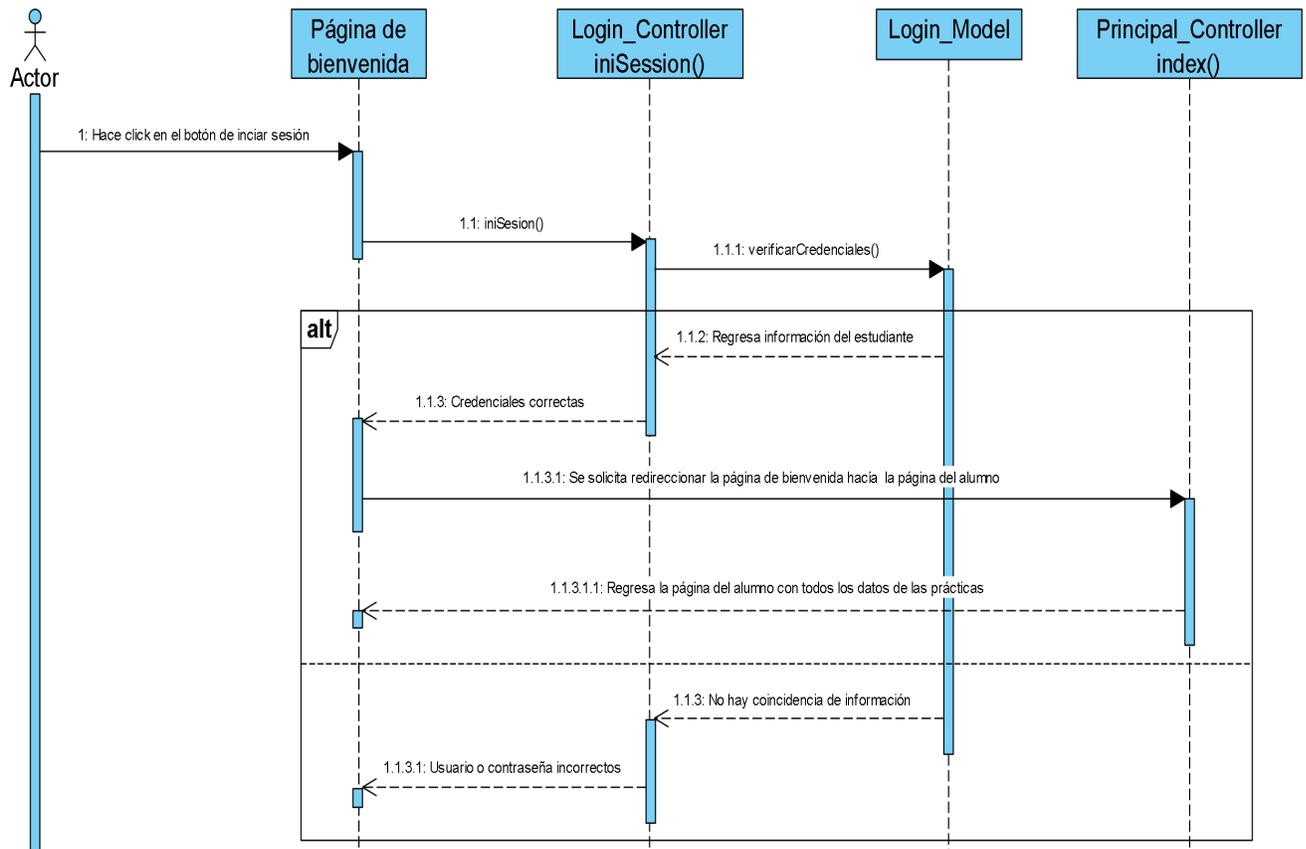
Los datos de las prácticas de campo se proporcionaron en un excel con el siguiente formato:

	A	B	C	D	E	F	G	H	I	J	J	L	M
10	No.	FECHA	ASIGNATURA	GRUPO	PROFESOR	VISITA A	SALIDA		LLEGADA		CUPO	CLAVE	
11							Hora	Lugar	Hora	Lugar			
12	<b>Febrero</b>												
13	1	15	SISTEMAS DE TRANSPORTE	2	M.I. FRANCISCO GRANADOS VILLARJERTE	PUESTO CENTRAL DE CONTROL DEL METRO	9:45	CALLE DE DELICIAS No 67, METRO SALTO DEL AGUA	14:00	CALLE DE DELICIAS No 67, METRO SALTO DEL AGUA	30	781	
14	1	15	SISTEMAS DE TRANSPORTE	2	M.I. FRANCISCO GRANADOS VILLARJERTE	PUESTO CENTRAL DE CONTROL DEL METRO	9:45	CALLE DE DELICIAS No 67, METRO SALTO DEL AGUA	14:00	CALLE DE DELICIAS No 67, METRO SALTO DEL AGUA	30	781	
15	<b>Marzo</b>												
15	1	15	SISTEMAS DE TRANSPORTE	2	M.I. FRANCISCO GRANADOS VILLARJERTE	PUESTO CENTRAL DE CONTROL DEL METRO	9:45	CALLE DE DELICIAS No 67, METRO SALTO DEL AGUA	14:00	CALLE DE DELICIAS No 67, METRO SALTO DEL AGUA	30	781	
16	1	15	SISTEMAS DE TRANSPORTE	2	M.I. FRANCISCO GRANADOS VILLARJERTE	PUESTO CENTRAL DE CONTROL DEL METRO	9:45	CALLE DE DELICIAS No 67, METRO SALTO DEL AGUA	14:00	CALLE DE DELICIAS No 67, METRO SALTO DEL AGUA	30	781	

**Figura 2.4 Contenido del archivo excel con la información de las prácticas de campo**

En la Figura 2.4 se muestra cómo el ingeniero tiene la información en el archivo excel y esta imagen se puede utilizar para empezar a diseñar la parte correspondiente en la página de bienvenida.

En los diagramas de secuencia de la Figura 2.5 y 2.6 permiten visualizar la comunicación del *frontend* y *backend* cuando un usuario realiza alguna acción que requiera enviar o solicitar información en la página de bienvenida:



**Figura 2.5 Diagrama secuencial de inicio de sesión**

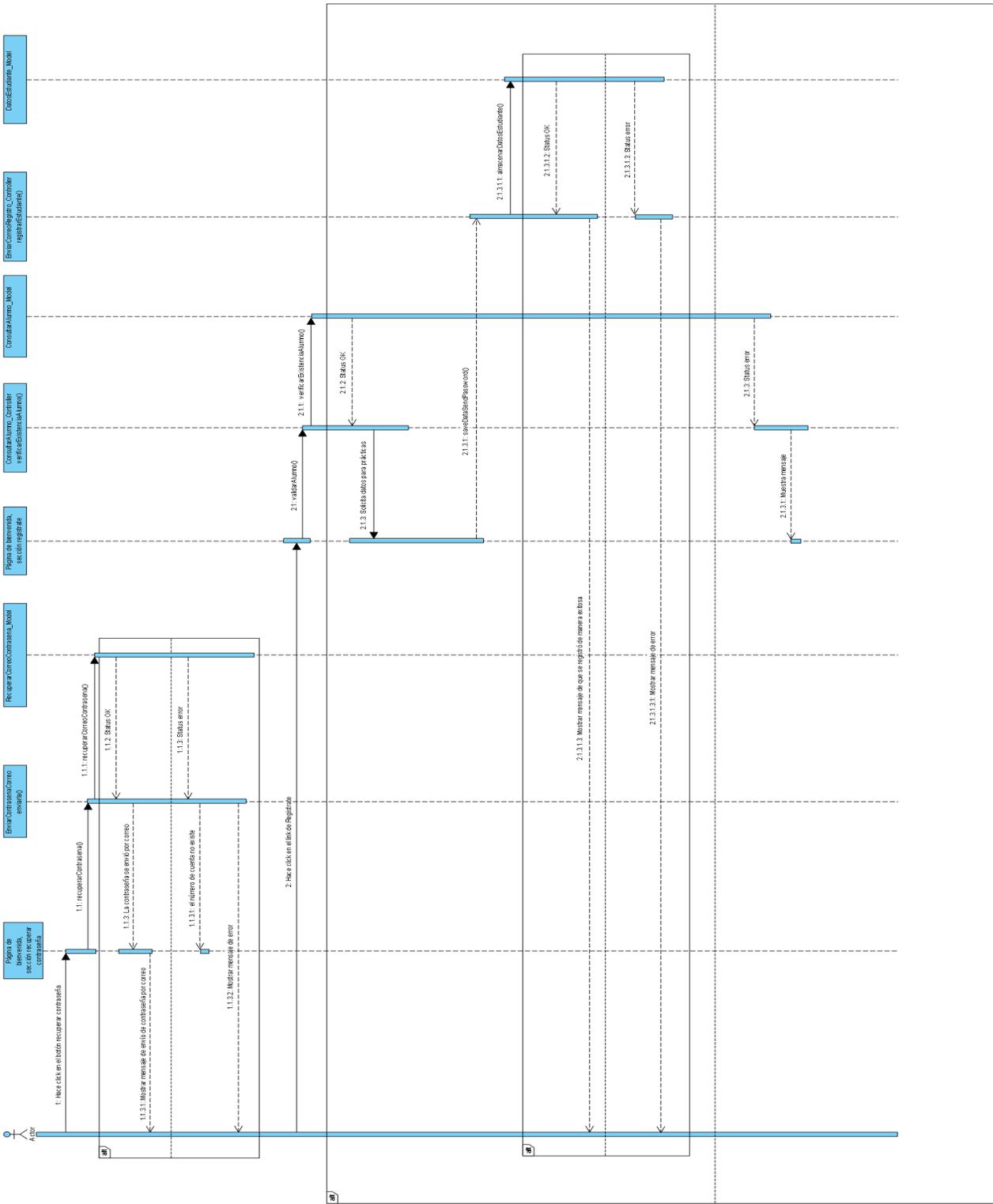
### Escenario principal

En este caso el usuario hará clic en el botón y esta acción activará una función *Javascript* con el nombre “iniSesion” que con *Jquery* enviará el usuario y contraseña a la función del controlador llamada “iniSession”, posteriormente verificará que las credenciales sean correctas utilizando la función “verificarCredenciales” del Modelo “*Login\_Model*” y tendrá éxito al iniciar sesión por lo que la información recuperada en el punto 1.1.2 servirá para crear la sesión en *PHP*, los datos que se manejará en la sesión son: Nombre del estudiante, número de cuenta. Por último, se redireccionará a la página del estudiante.

### Ruta alterna

En este caso las credenciales son incorrectas, por lo que, se mostrará en pantalla un mensaje con el siguiente texto “Usuario o contraseña incorrectos”.

Con el diagrama de arriba se muestra el nombre de los controladores a crearse junto con las funciones (de *Javascript* y *PHP*) que se deben activar para solicitar o almacenar datos a partir de una acción del usuario en la página web. Los nombres de las funciones a usar de algún modelo se mostrarán en la flecha del mensaje que va del controlador al modelo.



**Figura 2.6 Diagrama secuencial de registro del estudiante y de recuperación de contraseña.**

### **Escenario principal del punto uno**

La acción del usuario es dar clic en el botón “Recuperar contraseña”, lo cual mostrará otro modal para que ingrese su número de cuenta y al dar click en el botón enviar se activará una función Javascript con el nombre “recuperarContrasena” para posteriormente hacer un *POST* request con *Jquery*, este request activará la función “enviarla” en el controlador “EnviarContrasenaCorreo” y simplemente se recuperará la contraseña y el correo a partir del número de cuenta del estudiante, finalmente se envía la contraseña al estudiante.

### **Ruta alterna del punto uno**

En caso de que el número de cuenta no se encuentre, solicitar al alumno registrarse en el sistema.

### **Escenario principal del punto dos**

El usuario hará clic en el botón “Regístrate”, lo cual mostrará un formulario en un modal para ingresar su número de cuenta y nombre completo al enviar la información se accionará una función de Javascript llamada “validarAlumno” que solicitará un *POST* request a la función “verificarExistenciaAlumno” en el controlador “ConsultarAlumno\_Controller”, posteriormente se hará una consulta con su número de cuenta como filtro, el cual recuperará su nombre completo para mostrarse en un nuevo formulario para solicitar información pertinente a la asistencia de las prácticas de campo, finalmente se almacenarán los datos y se enviará el correo con la contraseña generada.

### **Ruta alterna del punto dos**

En caso de que los datos no se encuentren se le indicará al alumno solicitar una aclaración con el ingeniero a cargo de la administración de las prácticas de campo.

### **Página del alumno**

La siguiente es la página del alumno al iniciar sesión, la cual debe tener las siguientes funcionalidades y características:

Se reutilizará el mismo diseño de la página de bienvenida, pero eliminando el Hero, agregando varias secciones e incorporando un botón a la información de las prácticas para que el alumno se pueda inscribir.

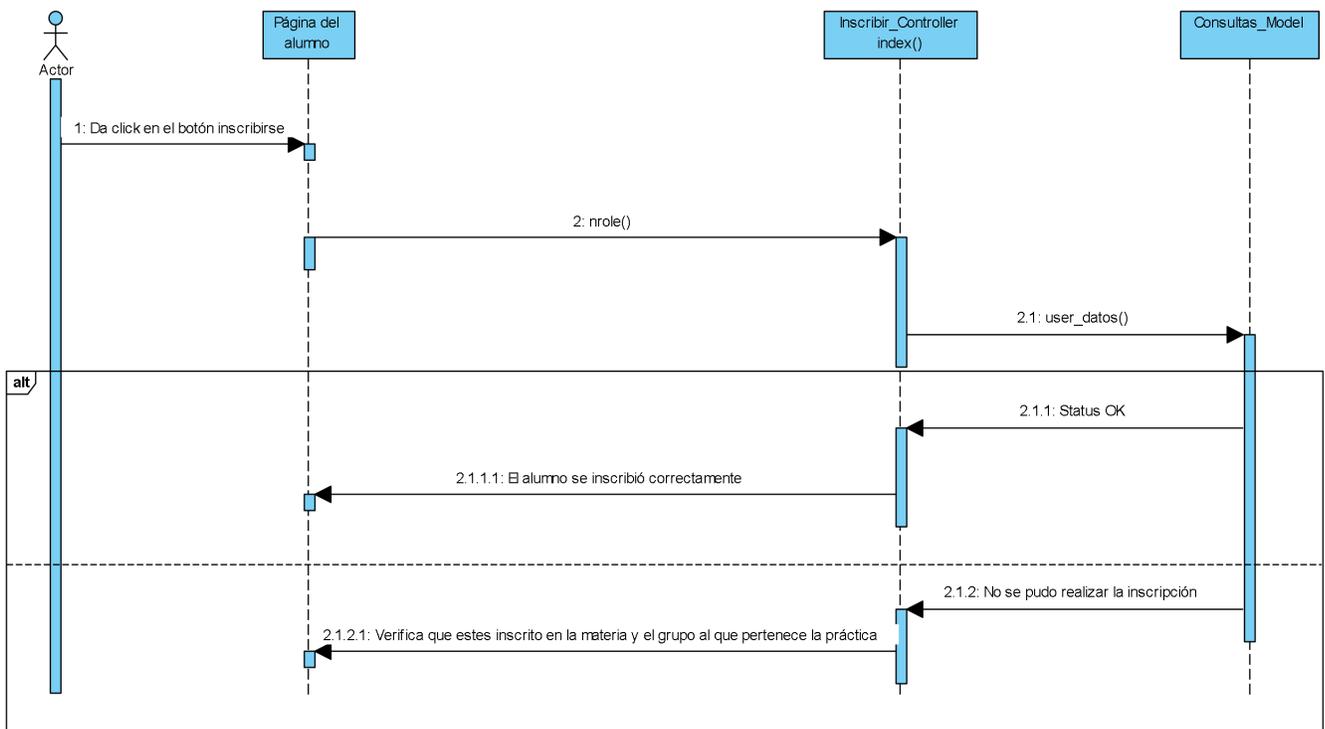
Las secciones que se deben visualizar son:

- INICIO
- INSCRIPCIÓN
- MIS PRÁCTICAS

- REGLAMENTO
- NOTAS

En “INICIO” se describirá las diferentes secciones.

En “INSCRIPCIÓN” se mostrarán las prácticas de campo por materia para que el alumno se pueda inscribir con un botón. El alumno solamente podrá inscribirse a la práctica si y solo si pertenece a la materia correspondiente y además que pertenezca al grupo indicado. Como una regla de negocio se solicitó que solamente puedan inscribirse un máximo de cinco prácticas en total.



**Figura 2.7 Diagrama secuencial para que el alumno se inscriba a una práctica de campo**

### Escenario principal

El alumno dará clic en el botón inscribirse, esto hará que se envíe un *POST request* al servidor y en el caso de que esté inscrito en la materia y grupo que corresponda a la información de la práctica su inscripción se realizará con éxito.

## **Ruta alterna**

En el caso de que el alumno no pertenezca a la materia o no esté inscrito en el grupo adecuado, entonces se le mostrará el siguiente mensaje “Verifica que estés inscrito en la materia y el grupo al que pertenece la práctica”.

En “MIS PRÁCTICAS” se deberá mostrar por materia las prácticas a las que se inscribió el alumno, mostrando si él/ella, ya fue a validar su inscripción.

En “REGLAMENTO” se deberá mostrar el *pdf* que el alumno deberá leer. Su *URL* es el siguiente:

<http://www.administracion.ingenieria.unam.mx/assets/reglamento-general-de-practicas-escolares2.pdf>

En “NOTAS” se deberá mostrar información extra para el alumno, cómo la obligación que tiene para ir a validar su inscripción y a dejar el peso para su seguro.

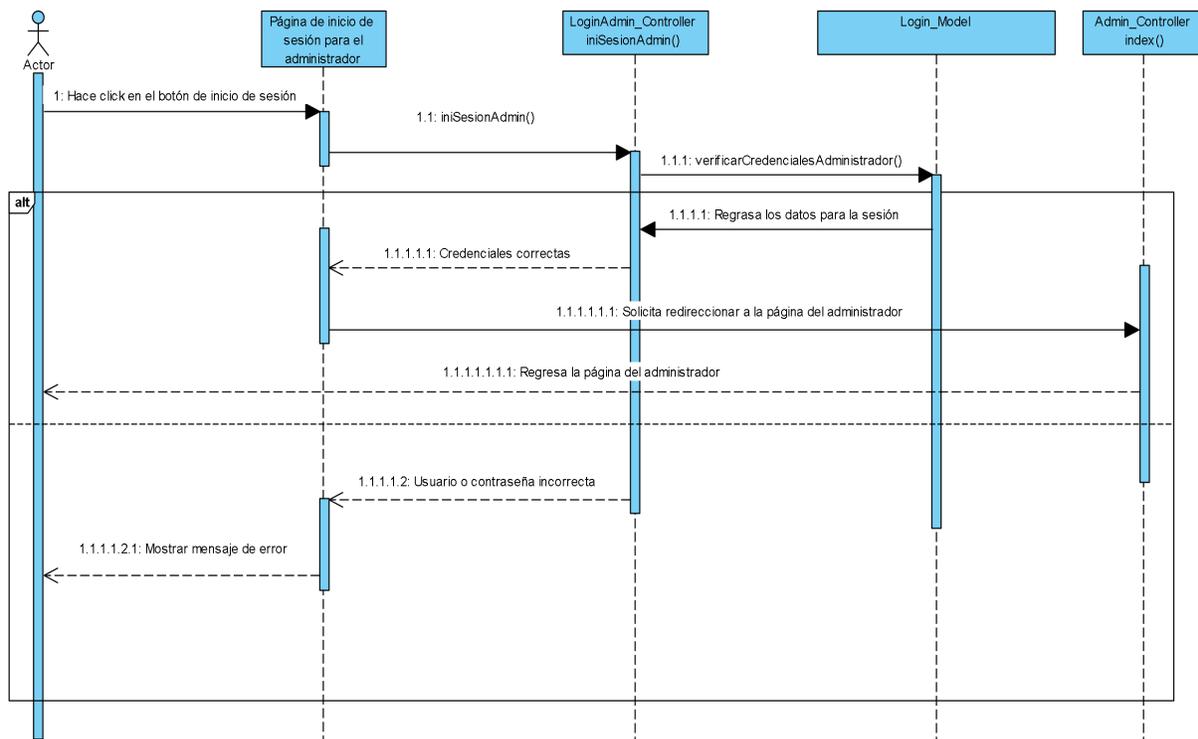
La recuperación de los datos de las secciones para mostrar las prácticas a inscribirse o inscritas, se recuperarán con los datos de la sesión.

## **Página del administrador**

Por último, se tendrá la página del administrador, en donde se deberá desarrollar las siguientes funcionalidades:

- Validación de las prácticas inscritas por los alumnos
- Constancia de Asistencia
- Imprimir las listas para que los alumnos firmen
- Inscripción a oyentes o externos
- Pasar lista a los alumnos por prácticas
- Generar lista de asistencia por grupo y materia
- Generar estadísticas de asistencia por prácticas
- Editar prácticas
- Agregar o eliminar prácticas
- Agregar o eliminar alumnos

El administrador tendrá otra página de inicio de sesión, y su diagrama de secuencia es la figura 2.8:



**Figura 2.8** Diagrama secuencial de inicio de sesión para el administrador

### Escenario principal

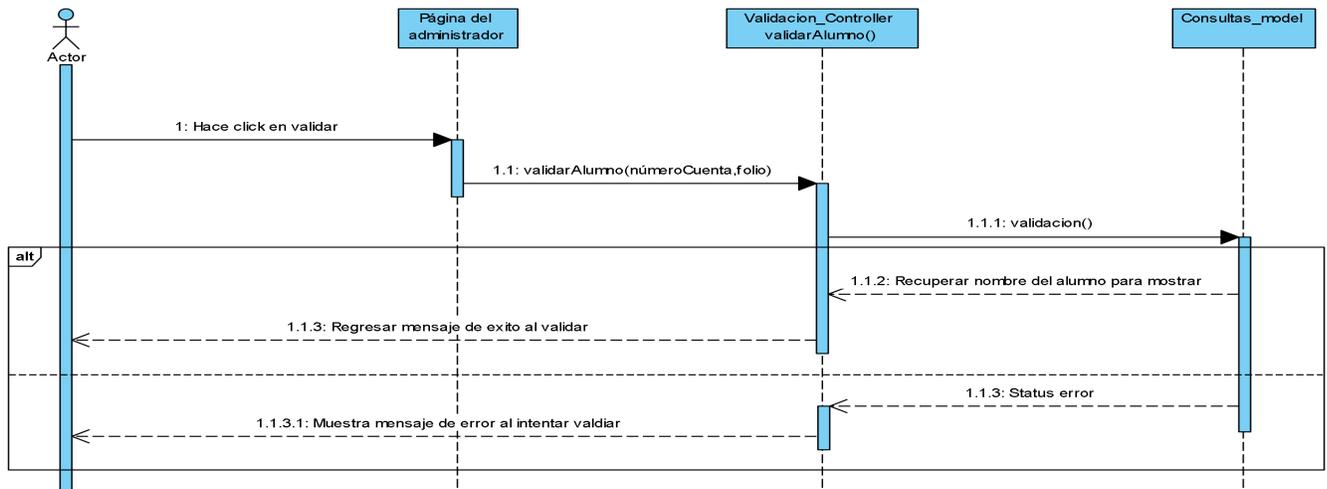
El usuario ingresa en el formulario sus credenciales, posteriormente hace clic en el botón de inicio de sesión, lo cual activará la función de *Javascript* “iniSesionAdmin”, en esta función se utilizará *Jquery* para mandar un *POST request* a la función “iniSesionAdmin” en el controlador “LoginAdmin\_Controller” y finalmente se verificarán contra la base de datos, posteriormente se solicitará con *Javascript* un redireccionamiento a la página del administrador.

### Ruta alterna

En este caso cómo las credenciales son incorrectas se le mostrará al administrador un mensaje que indique “Usuario o contraseña incorrecta”.

### Validación de las prácticas inscritas por los alumnos

En la validación se mostrará una lista de las inscripciones a las prácticas de campo por materia, mostrando: el número de cuenta, Nombre del alumno, Apellidos del alumno, folio de la práctica y un botón para hacer la validación. La siguiente imagen muestra el diagrama de secuencia para la validación de las prácticas:



**Figura 2.9 Diagrama secuencial para validar la inscripción de los alumnos**

### Escenario principal

El alumno se presentará con el ingeniero, se le solicitará su número de cuenta, posteriormente se buscará las prácticas a validar y se finalizará dando clic en cada botón correspondiente a las prácticas inscritas y en cada clic se activará una función en *Javascript* llamada “validarAlumno”, en esta función se solicitará un *POST request* a la función de *PHP* “validarAlumno” en el controlador “Validacion\_Controller”, finalmente se recupera el nombre del alumno de la base de datos para mostrarle al ingeniero el siguiente mensaje “Se validó correctamente la inscripción de Alan Gabriel Barrios Muñoz a la práctica”.

### Ruta alterna

En caso de un error mostrar mensaje de error “Favor de intentarlo más tarde, verifique su conexión a internet esté bien o contacte con el desarrollador. Gracias”.

### Constancia de Asistencia

Las constancias de las prácticas se generarán con la siguiente información:

FACULTAD DE INGENIERÍA  
DIVISIÓN DE INGENIERÍAS CIVIL Y GEOMÁTICA  
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS,  
PLANEACIÓN Y TRANSPORTE

Asunto: CONSTANCIA DE ASISTENCIA A PRÁCTICAS

A QUIEN CORRESPONDA  
PRESENTO

Por medio del presente, se hace constar la asistencia a las prácticas que se indican:

Nombre del alumno (a): MARISOL ORTIZ SÁNCHEZ

Número de cuenta: 620191108

Práctica a la que asistió:  
AEROPUERTO INTERNACIONAL DE PUEBLA, PUE.

Fecha de la práctica:  
2020-03-22

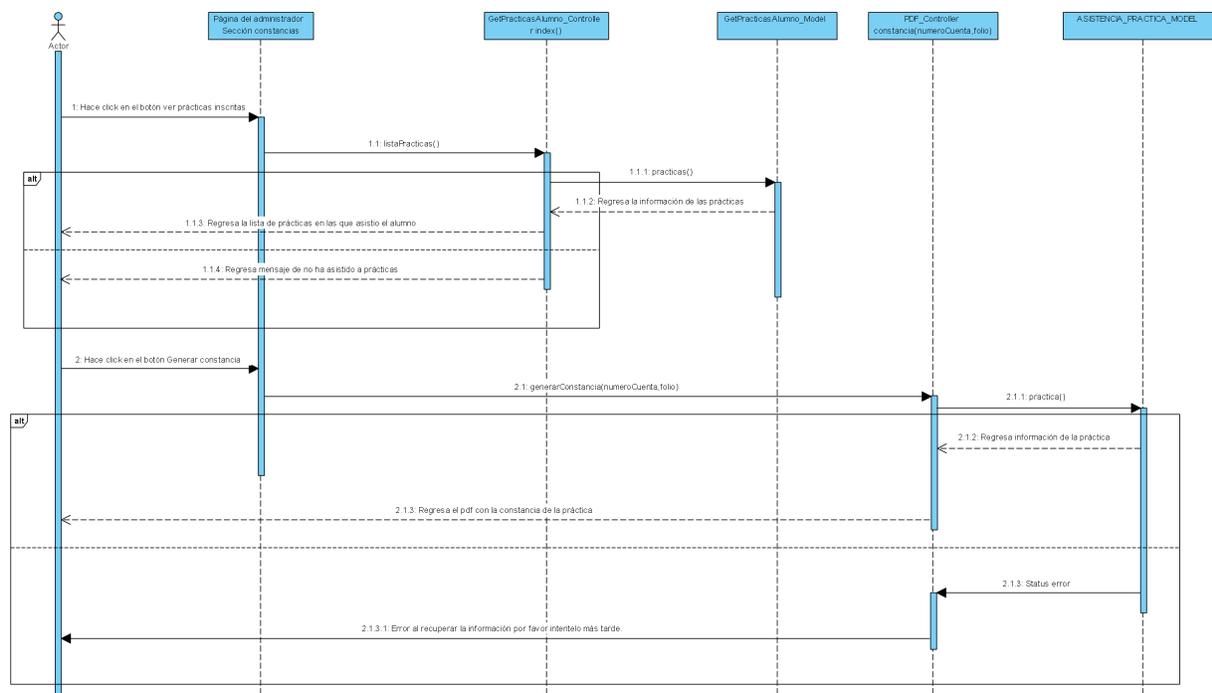
Se extiende la presente constancia a petición del interesado para los fines académicos que considere convenientes.

Atentamente.  
"POR MI RAZA HABLARÁ EL ESPÍRITU"  
Ciudad Universitaria Cd. Mx., a 06 de diciembre del 2020  
EL JEFE DEL DEPARTAMENTO

Ing. Ernesto René Mendoza Sánchez

**Figura 2.10 Formato para generar las constancias de asistencia a las prácticas de campo**

Su diagrama de secuencia es la siguiente:



**Figura 2.11 Diagrama secuencial para generar las constancias de asistencia**

### Escenario principal

El administrador ingresará el número de cuenta del alumno y lo enviará al servidor en respuesta se le mostrarán todas las prácticas de campo a las que se inscribió, validó y asistió el alumno, posteriormente solicitará generar una o todas las constancias de las prácticas del alumno y en respuesta recibirá un *PDF* por cada práctica.

### Ruta alterna

El alumno no asistió a ninguna práctica o a la hora de generar su constancia no tiene conexión a internet.

Como se muestra en la figura 2.11 se obtendrá primero la lista de las prácticas a las cuales el alumno sí asistió y posteriormente se seleccionará la práctica a generar su constancia.

### Imprimir las listas para que los alumnos firmen

El archivo proporcionado para generar la lista de firmas de los alumnos por práctica se muestra en la siguiente imagen:



### **Escenario principal**

En esta opción se le mostrará la lista de las prácticas en un *select tag* en el cual al seleccionar una y dar en el botón generar se le empezará a descargar el archivo excel con los datos de los alumnos inscritos en la práctica seleccionada.

### **Ruta alterna**

En caso de que no se encuentre a ningún alumno se generará el archivo con el mensaje "No hay alumnos".

### **Inscripción a alumnos oyentes o personas externas**

La inscripción a alumnos oyentes o personas externas será permitida, por lo que el formulario para su inscripción contendrá lo mismo que un alumno normal, es decir se le asignará un grupo de la materia que esté relacionada con la práctica a inscribirse, además se solicitará un número de cuenta, su nombre, correo, entre demás información y posteriormente podrá ingresar cómo un alumno normal al sistema para inscribirse. Su diagrama de secuencia es la siguiente:

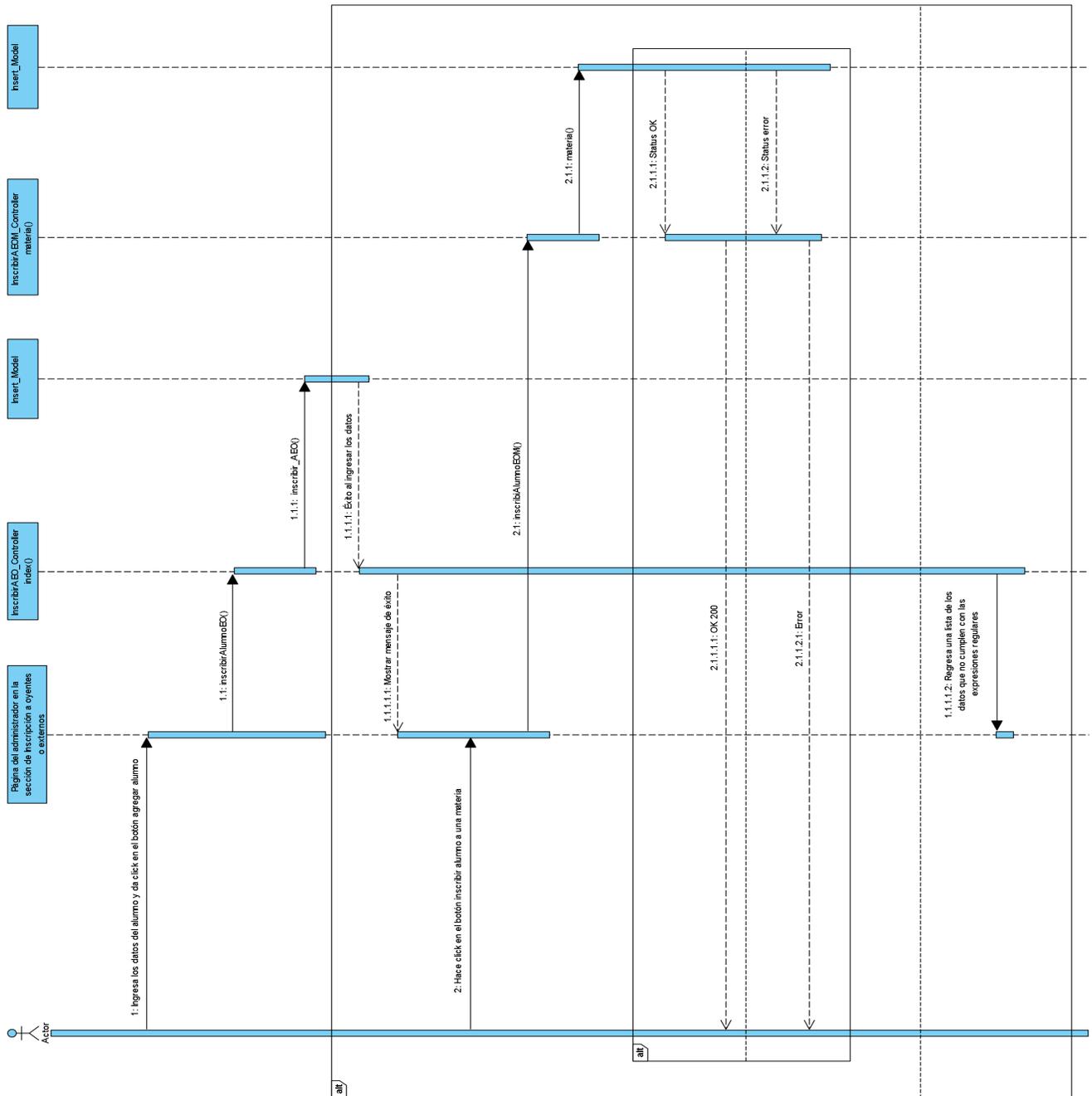


Figura 2.14 Diagrama secuencial para inscribir un alumno externo u oyente

## Escenario principal

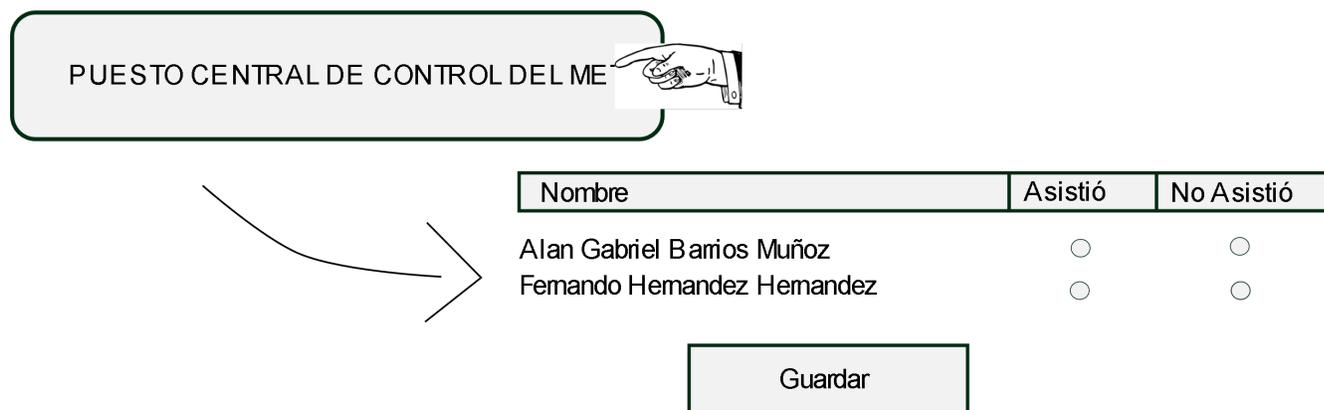
El administrador solicitará al alumno oyente o externo su información la cual será enviada al servidor para ser almacenada, posteriormente se inscribirá al alumno a la materia y grupo correspondiente a las prácticas.

## Ruta alterna

En caso de no tener buena conexión a internet se le mostrará al administrador que lo verifique o que se ponga en contacto con el desarrollador del sistema o en el caso de que los datos ingresados no sean del tipo correspondiente cómo el número de cuenta que debe ser solo números, entonces se regresará una lista de los datos que no pasaron la validación en el backend.

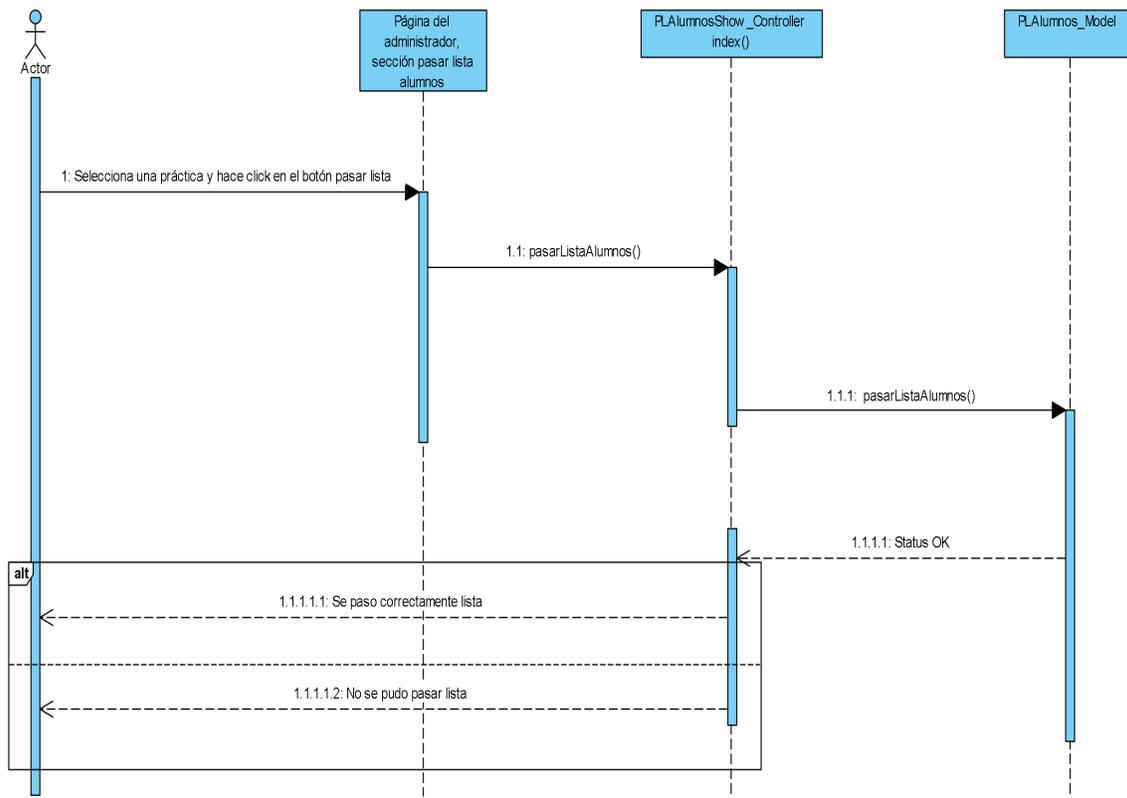
## Pasar lista

En la opción de pasar lista se deberá seleccionar la práctica y posteriormente mostrar los alumnos inscritos, por lo que al lado del nombre tendrá dos opciones asistió o no asistió, en la siguiente imagen se muestra el resultado deseado:



**Figura 2.15 Pasar lista**

El diagrama de secuencia que le corresponde es el siguiente:



**Figura 2.16 Diagrama secuencial para pasar lista**

### Escenario principal

El administrador seleccionará una de las prácticas a través de un select tag al enviar la solicitud al servidor se le mostrará una lista de los alumnos en donde podrá seleccionar una de las dos opciones asistió o no asistió y posteriormente enviar la lista.

### Ruta alterna

En caso de enviar la solicitud en obtener la lista de alumno o al pasar la lista y tener un error, entonces se mostrará el mensaje “Verifique su conexión a internet o comuníquese con la Unidad de Computó”

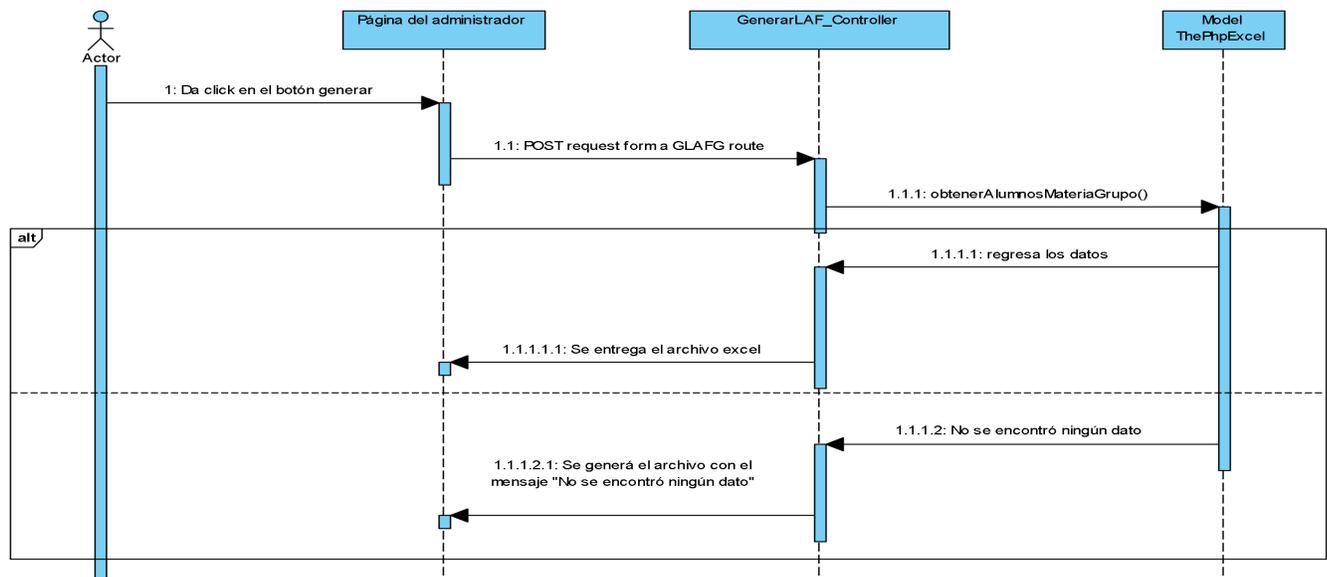
### Generar lista de asistencia por grupo y materia

En este apartado se le permitirá al administrador generar una lista de asistencia al seleccionar la materia y grupo deseado, en esta lista se mostrarán los números de cuenta, nombres completos de los estudiantes, grupo, nombre de la materia, nombre del profesor y nombres completos de las prácticas, pero será una lista de prácticas las que aparecerá y en cada una de ellas un tache que representa la asistencia del alumno a la práctica, por

ejemplo si un estudiante asistió a la práctica “PUESTO CENTRAL DE CONTROL DEL METRO” y a la práctica “LABORATORIO DE COMISION FEDERAL DE ELECTRICIDAD EN CUERNAVACA”, pero otro estudiante asistió a la práctica “TREN SUBURBANO”, entonces la lista de prácticas estará conformada por tres elementos y si algún elemento se repite, solo se mostrará una sola vez, la siguiente imagen muestra el formato del excel que se desea:

		PUESTO CENTRAL DE CONTROL DEL METRO	LABORATORIO DE COMISION FEDERAL DE ELECTRICIDAD EN CUERNAVACA	TREN SUBURBANO
324433520	Alan Gabriel Barrios Muñoz	X	X	X
342255115	Fernando Pérez Hernandez	X	X	X

**Figura 2.17 Formato del archivo excel a generar para ver la asistencia a las prácticas de campo**



**Figura 2.18 Diagrama secuencial para generar el archivo de asistencia a las prácticas de campo por materia y grupo**

### Escenario principal

El administrador seleccionará una práctica y dará clic en el botón generar, el servidor obtendrá la solicitud y recuperará la información con la cual se creará el archivo excel, al administrador se le mostrará en la parte inferior de la mayoría de los navegadores una descarga con el archivo.

## Ruta alterna

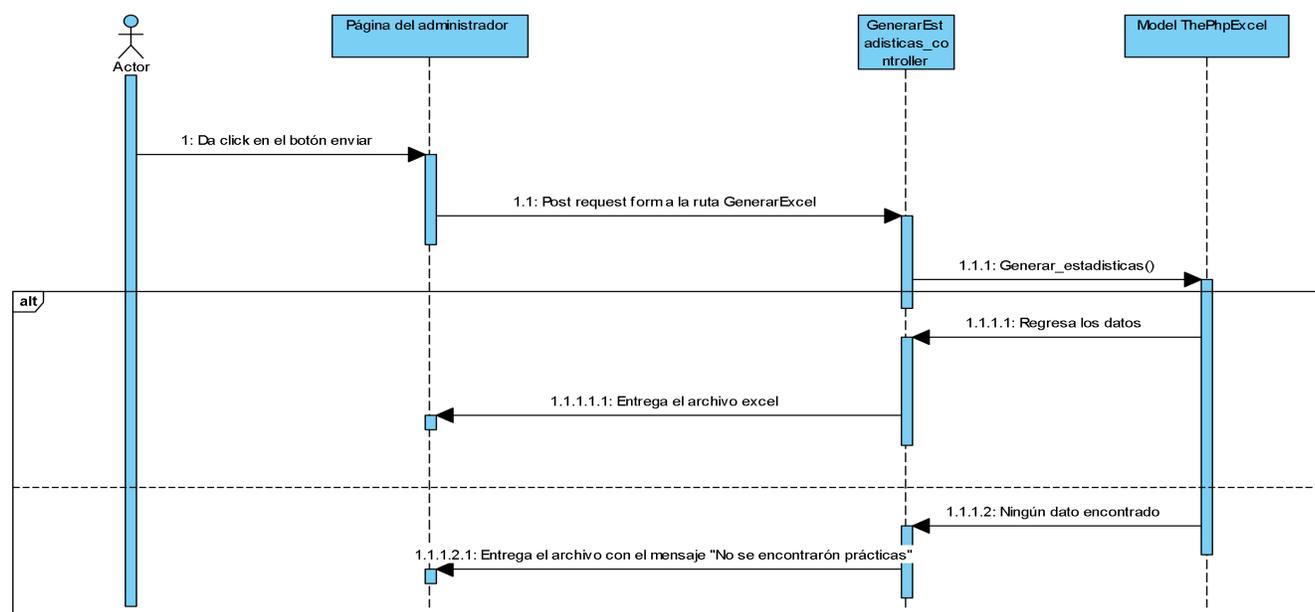
En caso contrario de que no se encuentre ningún dato se generará un archivo con el mensaje No se encontró ningún dato.

## Generar estadísticas de asistencia por práctica

El administrador tendrá la posibilidad de seleccionar alguna de las prácticas para generar las inscripciones contra los estudiantes que sí asistieron, mostrando al igual el folio y el nombre de la práctica, la siguiente imagen muestra la representación del archivo excel:

		Inscritos	Asistieron
7	PUESTO CENTRAL DE CONTROL DEL METRO	20	15
8	LABORATORIO DE COMISION FEDERAL DE ELECTRICIDAD EN CUERNAVACA	15	9

**Figura 2.19** Formato del archivo excel a generar de las listas de estadísticas de asistencia contra inscritos de las prácticas de campo.



**Figura 2.20** Diagrama secuencial para generar archivo excel con la lista de estadísticas de asistencia contra inscritos de las prácticas de campo.

## Escenario principal

El administrador enviará la solicitud de generar el archivo de estadísticas por lo que se le mostrará en el navegador una descarga con los datos solamente de las prácticas en las que ya se pasó lista.

## Ruta alterna

En caso contrario de que no se encontrase ningún dato se generará el archivo, pero con el mensaje “No se encontraron prácticas” es decir que en el momento de generar las estadísticas hasta ese momento todavía no se ha pasado lista en ninguna de las prácticas.

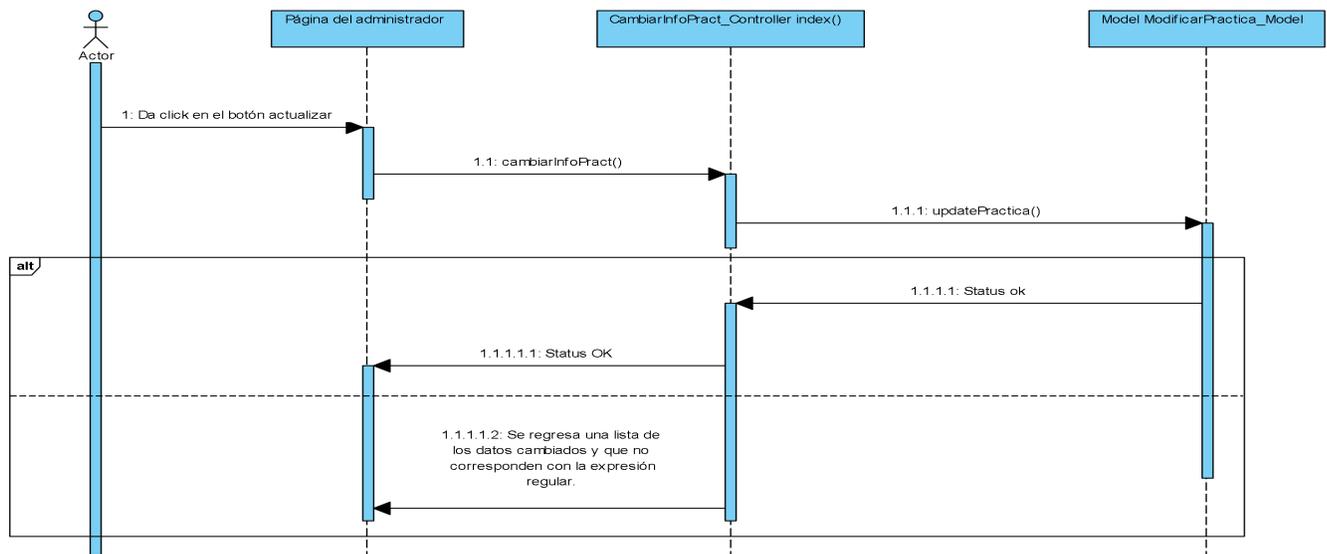
## Editar prácticas

En este apartado el administrador podrá visualizar la lista de prácticas que se encuentran en el sistema, al igual que editar por separado cada una de ellas, la siguiente imagen da muestra de cómo se desea este requerimiento:

Fecha de inscripción	Fecha para firmar	Fecha de llegada	Fecha de salida	Lugar de llegada	Lugar de salida	Cupo	Grupo	Profesor	Visita A	CASCO, CHALECO Y ZAPATO INDUSTRIAL	NO LLEVAR CELULAR	CARNET O SEGURO CORRESPONDIENTE ORIGINAL	CÁMARA	Cancelada
2020-12-05	2020-12-08	2020-02-15 14:00:00	2020-02-15 09:45:00	CALLE DE DELICIAS No 67, METRO SALTO DEL AGUA	CALLE DE DELICIAS No 67, METRO SALTO DEL AGUA	30	2	M.I. FRANCISCO GRANADOS VILLAFUERTE	PUESTO CENTRAL DE CONTROL DEL METRO	SI	SI	SI	SI	NO
2020-12-05	2020-12-05	2020-02-19 15:00:00	2020-02-19 07:00:00	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	15	4	DR. HUMBERTO MARENGO MOGOLLO N	LABORATORIO DE COMISION FEDERAL DE ELECTRICIDAD EN CUERNAVACA	SI	SI	SI	SI	SI

**Figura 2.21 Lista de prácticas en la página del administrador**

Al lado de cada práctica se mostrará un botón “editar” que permita mostrar un formulario con los datos de la práctica seleccionada para su edición.



**Figura 2.22 Diagrama secuencial para editar una práctica**

### Escenario principal

El administrador dará clic en el botón “editar” correspondiente a la práctica, posteriormente se le mostrará un formulario en un modal para modificar la información pertinente a la práctica y al último dará en el botón “actualizar” para guardar la nueva información.

### Ruta alterna

En el dado caso que los datos ingresados no contengan caracteres correspondientes al campo, se dará a conocer la información que se ingresó incorrectamente y que se deberá cambiar.

### Agregar o eliminar prácticas

En esta parte el administrador se le mostrará un formulario para seleccionar el archivo excel de su computadora, al dar clic en el botón enviar se cargarán las prácticas y debajo de este formulario un botón eliminar que eliminará por completo la información de las prácticas.

# Prácticas de campo

## Enviar

## Eliminar Prácticas

Figura 2.23 Apartado en la página del administrador para agregar o eliminar las prácticas

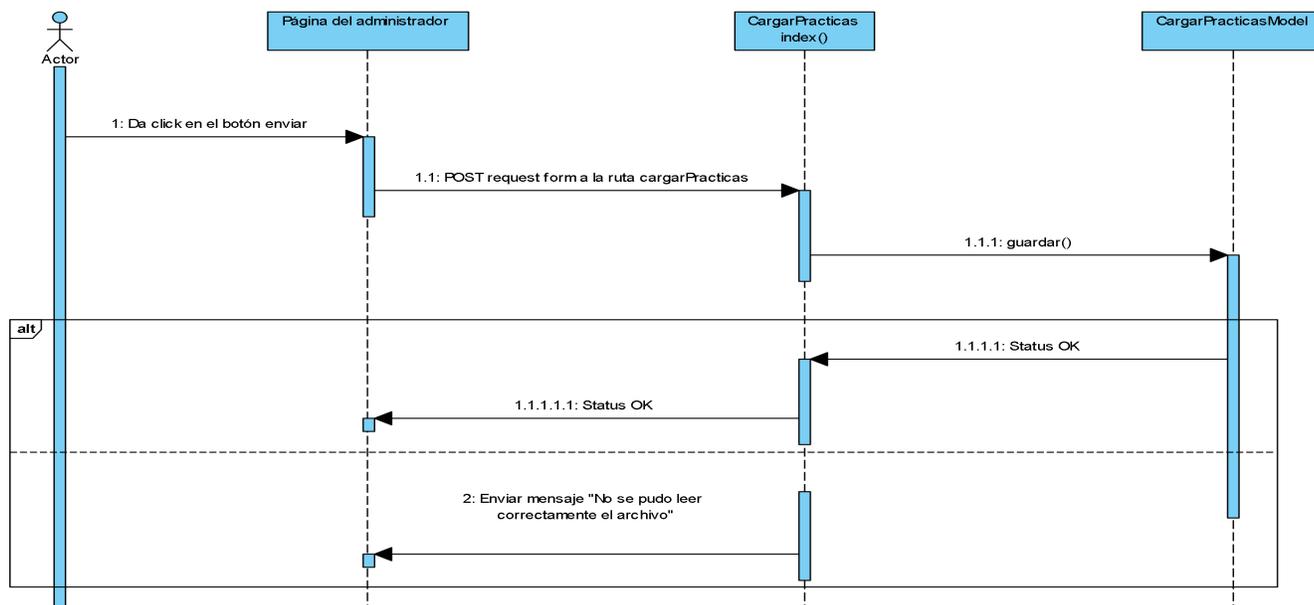


Figura 2.24 Diagrama para cargar prácticas

### Escenario principal

El administrador seleccionará un archivo excel donde se encuentre la información pertinente de las prácticas de campo, posteriormente dará clic en el botón enviar, haciendo esto el *backend* leerá cada fila del archivo hasta cargar todas las prácticas.

### Ruta alterna

El administrador no estructuró la información como se muestra en la figura 2.4 por lo que el programa no pudo leer correctamente la información y termina regresando el mensaje "No se pudo leer correctamente el archivo".

## Agregar o eliminar alumnos

El administrador podrá llenar las tablas `alumnos_siprac`, `materias_siprac`, `profesores`, `inscripciones` y `grupos_siprac` con los archivos excel que corresponden a un grupo por materia, este archivo es proporcionado por cada uno de los profesores al inicio del semestre después de altas y bajas definitivos. Para eliminar a los alumnos se hará clic en un botón y se eliminará la información pertinente.



Figura 2.25 Formulario para agregar o eliminar a los alumnos

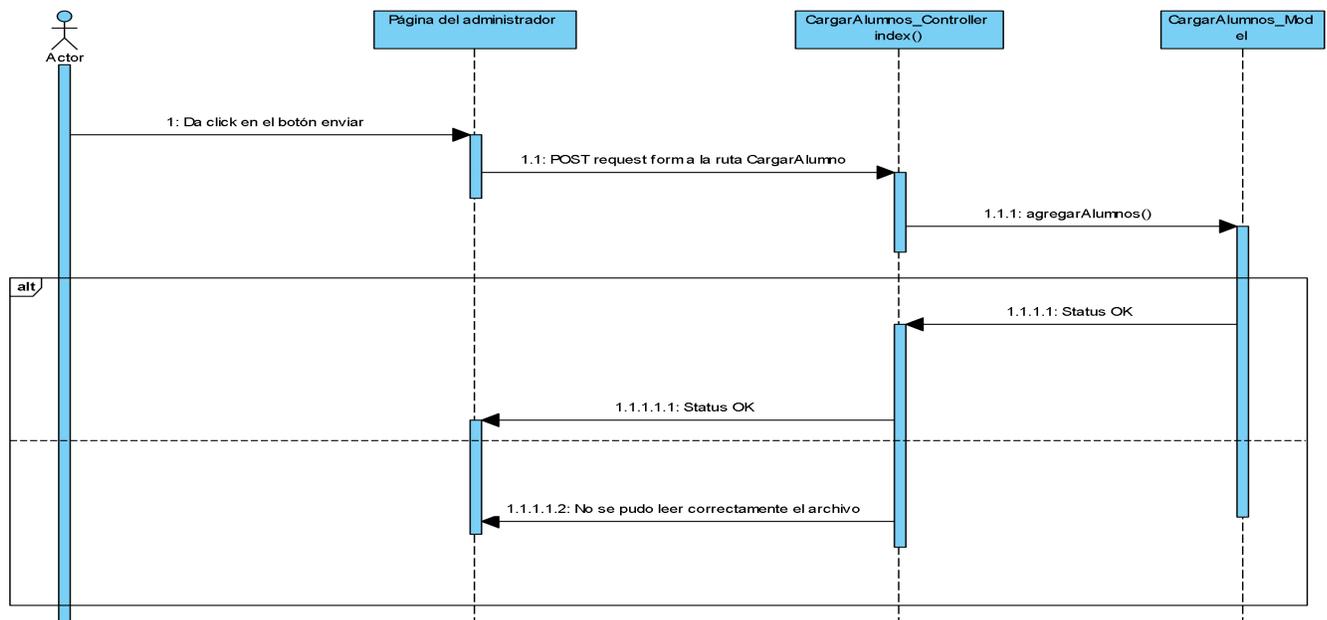


Figura 2.26 Diagrama secuencial para cargar a los alumnos

## Escenario principal

El administrador seleccionará un archivo de las listas proporcionadas por los maestros de cada grupo, posteriormente el sistema leerá y cargará los datos.

## Ruta alterna

En caso de que los datos no estén bien estructurados, esto puede pasar porque el sistema en donde los profesores obtienen sus listas cambió el formato excel que se les proporciona por cada grupo que se tenga, entonces se deberá mostrar el mensaje “No se pudo leer correctamente el archivo”.

Con la información que se tiene en los documentos “*Project charter*” y “*Project Scope*”, se puede obtener el “*Work breakdown structure*” del proyecto:

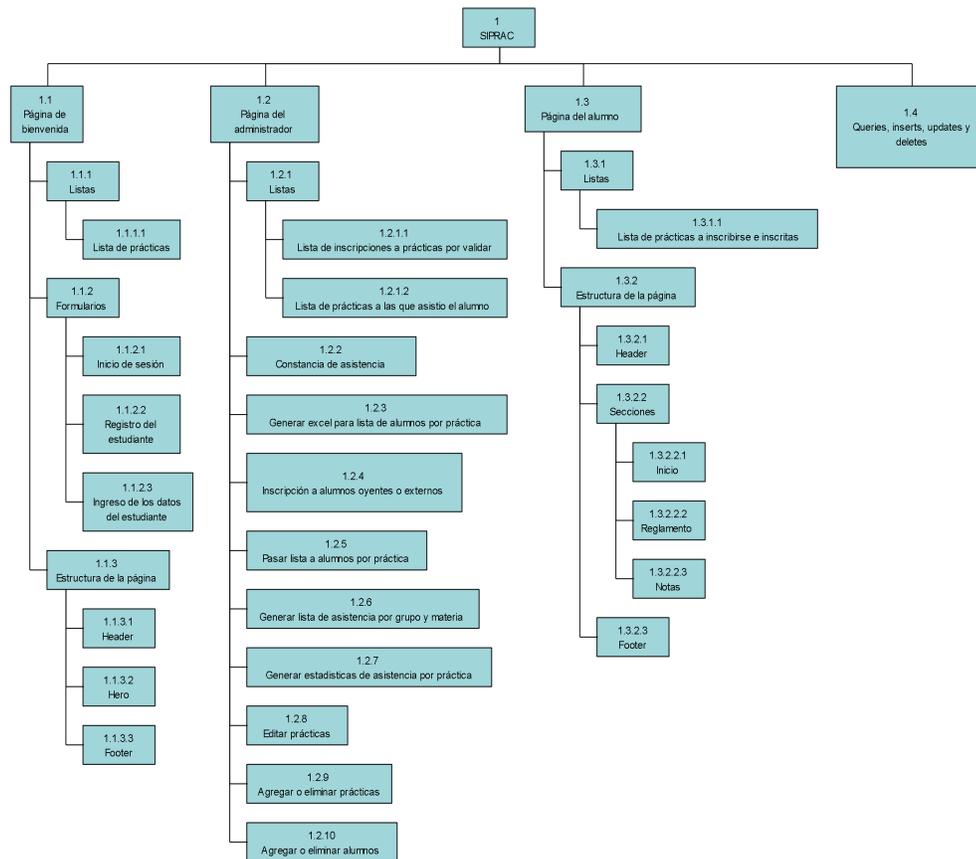


Figura 2.27 WBS SIPRAC

### 2.1.1.3 Specific Practice - Establecer estimaciones de productos de trabajo y atributos de tareas (Establish Estimates of Work Product and Task Attributes)

En todo proyecto debemos dar una estimación de entrega del proyecto finalizado ya que el cliente necesita saber el costo del proyecto y una de las maneras es el costo que genera el tener a la persona de crear el proyecto, en este caso el costo es de cuatro mil quinientos pesos mexicanos, es el monto que se le da a los becarios, cada mes por un tiempo de veinte horas por semana y así al tener el tiempo total estimado obtendremos el costo del proyecto que le costará a la DICyG el desarrollo de este sistema.

#### 2.1.1.3.1 Estimación del proyecto

Ahora, se mostrará la estimación del tamaño del sistema a desarrollar, haciendo uso de *IFPUG* 4.3.1<sup>4</sup>, para lograrlo, se debe de identificar en primera instancia el límite de nuestra aplicación, así como los archivos lógicos internos y externos del sistema que va a hacer medido, la siguiente imagen muestra el límite:

#### External inputs

Registrar alumno  
Inscribir práctica  
Validar práctica  
Pasar lista  
Inscribir alumno  
oyente o externo



#### External queries

Buscar prácticas  
Buscar prácticas inscritas  
Buscar prácticas a validar  
Constancia de asistencia  
Recuperar información de los alumnos  
Pasar lista

Figura 2.28 Límites

<sup>4</sup>AIAN Albrecht, Kim Albee, Maarten Barth, Andy Belden, Angela Benton, Mary Bradley, ... Grant Rule. (2010). Function Point Counting Practices Manual. United States of America: IFPUG.

Peter R. Hill. (2011). Practical Software Project Estimation A toolkit for Estimating Software Development Effort & Duration. United States of America: McGraw-Hill.

Steve McConnell. (2006). Software Estimation Demystifying the Black Art. United States of America: Microsoft Press

En la figura 2.28 se muestran los límites del sistema SIPRAC que se obtienen por medio de los requerimientos que proporcione los usuarios del sistema, un usuario es aquella persona que proporciona los requerimientos del sistema.

Los *External inputs* son aquellos elementos que el usuario va a utilizar para solicitar a los servidores alguna acción como por ejemplo el validar una de las prácticas inscritas. Esta es externa, ya que el usuario lo hace a través del navegador, donde se visualiza el sitio web, y este es externo a donde reside el sitio web los servidores.

Los *External queries* son aquellos elementos de la página que le permiten al usuario solicitar información a los servidores como ejemplo buscar las prácticas que se muestran en la página principal.

Los *Internal logic Files* son los elementos del sistema que almacenan la información. Para identificar los Archivos lógicos internos en nuestro sistema se utiliza el diagrama entidad relación, los elementos de un *ILF* son los siguientes:

Los *Data Element Type (DET)* son los atributos o columnas en cada tabla de la base de datos.

Los *Record Element Type (RET)* son los grupos de entidades que se tienen en la base de datos. Estos grupos se obtienen al ver si la asociación de una tabla con otra es:

- Un subtipo: Son entidades que identifican el tipo de otra, por ejemplo, se tiene una entidad llamada Cliente y hay dos tipos de clientes uno llamado Persona física y otro Persona moral.
- Una entidad asociativa: son tablas en la base de datos que apoya a realizar la relación mucho a muchos. Hay que tener en cuenta la cantidad de *DETs* que tiene la entidad, ya que, si solo tiene *Foreign Keys*, entonces no se cuenta como un *RET*, además si este tiene más atributos que no están en ninguna de las dos tablas, se debe contar este como un *ILF* a parte.
- Una entidad de atributo: Es aquella que permite extender la información de otra, como por ejemplo la entidad método de pago extiende la información de una entidad llamada cliente.

Para obtener la complejidad y tamaño de los *ILFs* se consultó dos tablas del documento de *IFPUG*, la cual se ilustra en la siguiente imagen:

**Tabla 2.3 Tamaño de *ILFs***

	<i>ILF</i>
<i>Low</i>	7
<i>Middle</i>	10
<i>High</i>	15

**Tabla 2.4 Complejidad de un *ILF***

		<i>DET</i>		
		[1 - 19]	[20 - 50]	> 50
	1	<i>Low</i>	<i>Low</i>	<i>Average</i>
<i>RET</i>	[2 - 5]	<i>Low</i>	<i>Average</i>	<i>High</i>
	> 5	<i>Average</i>	<i>High</i>	<i>High</i>

Con estos elementos se obtuvo la siguiente figura que muestra los *ILFs* del sitio web:

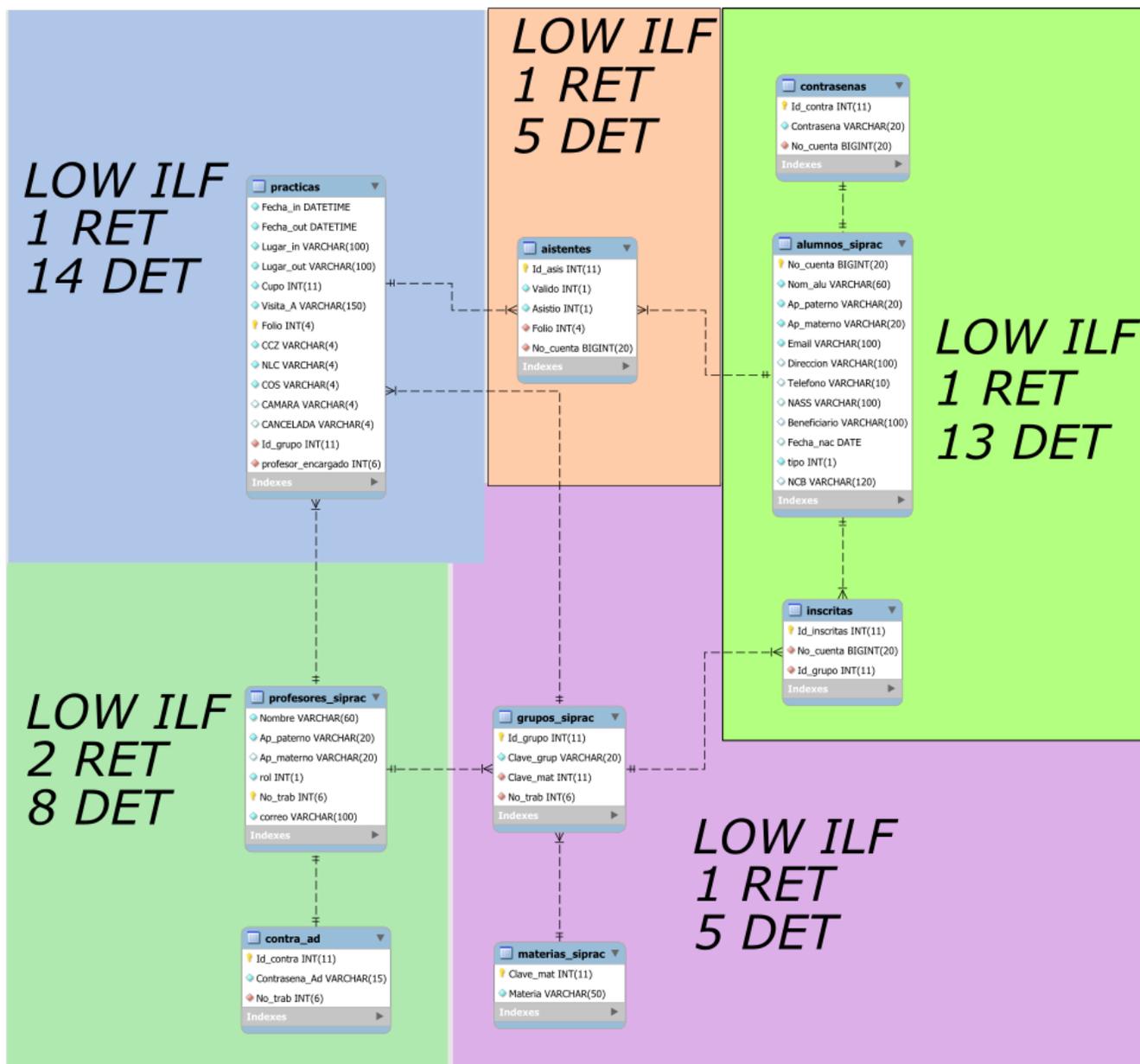


Figura 2.29 ILFs

**Tabla 2.5 Puntos de función**

<i>ILFs</i>	5x7=35 <i>FPs</i>
-------------	-------------------

Lo siguiente es obtener el tamaño de las funciones de transacción que comprenden Entradas externas (*EI*) que son en esencia los formularios utilizados por el alumno para almacenar información en la base de datos cuando la información llega al servidor, Consultas externas (*EQ*) son aquellos formularios que utiliza el alumno para consultar información de la base de datos para ser mostrados en el sitio web y Salidas Externas (*EO*) son formularios que le permiten al alumno generar archivos a partir de los datos almacenados en la base de datos.

Al igual que los *ILFs* se tienen *DETs* pero estos identifican los elementos que conforman al formulario y además no se utilizan los *RETs* sino que aquí se llaman *FTRs* (*File Types Referenced*) que son las tablas que se utilizarán para consultar o para agregar registros en ella. Las siguientes tablas apoyaron en dar la complejidad a los *EQs*, *EOs* y *EIs*:

**Tabla 2.6 Complejidad Funcional de una *EI***

			<i>DET</i>	
		[1 - 4]	[5 - 15]	>15
	[0 - 1]	<i>Low</i>	<i>Low</i>	<i>Average</i>
<i>FTR</i>	2	<i>Low</i>	<i>Average</i>	<i>High</i>
	> 2	<i>Average</i>	<i>High</i>	<i>High</i>

**Tabla 2.7 Complejidad Funcional de una *EO* y una *EQ***

			<i>DET</i>	
		[1 - 5]	[6 - 19]	>19
	[0 - 1]	<i>Low</i>	<i>Low</i>	<i>Average</i>
<i>FTR</i>	[2 - 3]	<i>Low</i>	<i>Average</i>	<i>High</i>
	> 3	<i>Average</i>	<i>High</i>	<i>High</i>

**Tabla 2.8 tamaño de una función transaccional**

	<i>EI</i>
<i>Low</i>	3
<i>Middle</i>	4
<i>High</i>	6

**Tabla 2.9 Formulario de inicio de sesión (EQ)**

<b><i>DETs</i></b>	<b><i>FTRs</i></b>
1 Usuario 2 Contraseña 3 Botón de enviar 4 Mensajes de respuesta	1 alumnos_siprac 2 contraseñas
La complejidad es <i>LOW</i>	
El tamaño es de 3 <i>FPs</i>	

**Tabla 2.10 Formulario para recuperar la contraseña (EO)**

<b><i>DETs</i></b>	<b><i>FTRs</i></b>
1 Número de cuenta 2 Botón de enviar 3 Mensajes de Respuesta	1 alumnos_siprac 2 contraseñas
La complejidad es <i>LOW</i>	
El tamaño es de 3 <i>FPs</i>	

**Tabla 2.11 Formulario para saber la existencia del estudiantes (EQ)**

<b>DETs</b>	<b>FTRs</b>
1 Número de cuenta 2 Nombres 3 Apellido paterno 4 Apellido materno 5 Botón de enviar 6 Mensajes de respuesta	1 alumnos_siprac
La complejidad es <i>LOW</i>	
El tamaño es de 3 <i>FPs</i>	

**Tabla 2.12 Formulario para ingresar los datos del estudiante (EI)**

<b>DETs</b>	<b>FTRs</b>
1 Calle 2 Número 3 Colonia 4 Alcaldía/Municipio 5 Entidad Federativa 6 Código postal 7 Número de afiliación a tu seguro social 8 Correo 9 Fecha de nacimiento 10 Apellido paterno del beneficiario 11 Apellido materno del beneficiario 12 Nombres del beneficiario 13 Teléfono del beneficiario 14 Botón de enviar 15 Mensajes de respuesta	1 alumnos_siprac
La complejidad es <i>LOW</i>	
El tamaño es de 3 <i>FPs</i>	

**Tabla 2.13 Lista de las prácticas en la página de bienvenida (EQ)**

<b>DETs</b>	<b>FTRs</b>
1 Folio 2 Fecha de inscripción 3 Fecha para firmar 4 Fecha (salida) 5 Fecha (llegada) 6 Grupo 7 Profesor 8 Visita a 9 Lugar de salida 10 Lugar de llegada 11 Cupo 12 Cancelada	1 prácticas
La complejidad es <i>LOW</i>	
El tamaño es de 3 <i>FPs</i>	

**Tabla 2.14 Inscripción a alguna práctica (EI)**

<b>DETs</b>	<b>FTRs</b>
1 Folio 2 Número de cuenta 3 Materia 4 Botón de enviar 5 Mensajes de respuesta	1 alumnos_siprac 2 inscritas 3 grupos_siprac 4 asistencias
La complejidad es <i>HIGH</i>	
El tamaño es de 6 <i>FPs</i>	

**Tabla 2.15 Lista de prácticas inscritas y a inscribirse (EQ)**

<b>DETs</b>	<b>FTRs</b>
1 Folio 2 Fecha de inscripción 3 Fecha para firmar 4 Fecha (salida) 5 Fecha (llegada) 6 Grupo 7 Profesor 8 Visita a 9 Lugar de salida 10 Lugar de llegada 11 Validación 12 Cancelado 13 Cupo 14 Mensajes de respuesta 15 CASCO, CHALECO Y ZAPATO INDUSTRIAL 16 NO LLEVAR CELULAR 17 CARNET O SEGURO CORREPONDIENTE ORIGINAL 18 CÁMARA	1 prácticas
La complejidad es <i>LOW</i>	
El tamaño es de 3 <i>FPs</i>	

**Tabla 2.16 Lista de validaciones de asistencia a las prácticas (EQ)**

<b>DETs</b>	<b>FTRs</b>
1 Número de cuenta 2 Nombre 3 Apellidos 4 Clave de la materia 5 Folio 6 Nombre de la materia	1 asistencias
La complejidad es <i>LOW</i>	
El tamaño es de 3 <i>FPs</i>	

**Tabla 2.17 Validar la asistencia a las prácticas (EI)**

<b>DETs</b>	<b>FTRs</b>
1 Número de cuenta 2 Folio 3 Botón de validar 4 Mensajes de respuesta 5 Nombre del alumno	1 asistencias
La complejidad es <i>LOW</i>	
El tamaño es de 3 <i>FPs</i>	

**Tabla 2.18 Lista de prácticas a las que asistió el alumno (EQ)**

<b>DETs</b>	<b>FTRs</b>
1 Visita a 2 Fecha (salida) 3 Fecha (llegada) 4 Folio 5 Materia 6 Número de cuenta 7 Botón de ver prácticas 8 Mensajes de respuesta	1 alumnos_siprac 2 asistencias 3 prácticas
La complejidad es <i>Average</i>	
El tamaño es de 4 <i>FPs</i>	

**Tabla 2.19 Constancia de asistencia a las prácticas (EO)**

<b>DETs</b>	<b>FTRs</b>
1 Folio 2 Número de cuenta 3 Nombre del alumno 4 Nombre de la práctica 5 Fecha de la práctica 6 Botón de enviar 7 Mensajes de respuesta	1 asistencias 2 prácticas 3 alumnos_siprac
La complejidad es <i>Average</i>	
El tamaño es de 5 <i>FPs</i>	

**Tabla 2.20 Dropdown de prácticas (EQ)**

<b>DETs</b>	<b>FTRs</b>
1 Folio 2 Nombre de la prácticas 3 Botón de Listas 4 Mensajes de respuesta	1 prácticas
La complejidad es <i>LOW</i>	
El tamaño es de 3 <i>FPs</i>	

**Tabla 2.21 Generar lista de prácticas (EO)**

<b>DETs</b>	<b>FTRs</b>
1 Folio 2 Nombre de la práctica 3 Botón de Generar lista 4 Mensajes de respuesta 5 Fecha de salida 6 No. De días 7 Destino 8 Nombre del asegurado 9 Fecha de nacimiento 10 No. De cuenta 11 No. Afiliación al IMSS 12 Domicilio 13 Teléfono 14 Nombre completo del beneficiario 15 Alumno/Estudiante 16 No. de integrantes	1 prácticas 2 alumnos_siprac 3 asistencias
La complejidad es de <i>Average</i>	
El tamaño es de 5 <i>FPs</i>	

**Tabla 2.22 Inscripción a oyentes o externos (EI)**

<b>DETs</b>	<b>FTRs</b>
1 Número de cuenta 2 Nombres del alumno 3 Apellido paterno del alumno 4 Apellido materno del alumno 5 Calle 6 Número 7 Colonia 8 Delegación/Municipio 9 Entidad Federativa 10 Código postal 11 Correo 12 Número de afiliación a su seguro social 13 Fecha de nacimiento 14 Nombres del beneficiario 15 Apellido paterno del beneficiario 16 Apellido materno del beneficiario 17 Teléfono del beneficiario 18 Tipo de estudiante 19 Botón de enviar 20 Mensajes de respuesta	1 alumnos_siprac
La complejidad es de <i>Average</i>	
El tamaño es de 4 <i>FPs</i>	

**Tabla 2.23 Inscribir alumno a una materia (EI)**

<b>DETs</b>	<b>FTRs</b>
1 Número de cuenta 2 Materia 3 Grupo 4 Botón de enviar 5 Mensajes de respuesta	1 inscritas 2 alumnos_siprac
La complejidad es de <i>Average</i>	
El tamaño es de 4 <i>FPs</i>	

**Tabla 2.24 Ver materias inscritas del alumno (EQ)**

<b>DETs</b>	<b>FTRs</b>
1 Número de cuenta 2 Botón de buscar 3 Mensajes de respuesta 4 Clave de la materia 5 Grupo	1 inscritas 2 grupos_siprac
La complejidad es de <i>LOW</i>	
El tamaño es de 3 <i>FPs</i>	

**Tabla 2.25 Lista de alumnos para pasar lista por práctica (EQ)**

<b>DETs</b>	<b>FTRs</b>
1 Folio 2 Nombre de la prácticas 3 Botón de enviar 4 Mensajes de respuesta 5 Número de cuenta 6 Nombre del alumno 7 Apellido paterno 8 Apellido materno 9 Asistencia	1 alumnos_siprac 2 asistencias
La complejidad es de <i>Average</i>	
El tamaño es de 4 <i>FPs</i>	

**Tabla 2.26 Lista de prácticas para Editar (EQ)**

DETs	FTRs
1 Folio 2 Nombre 3 Fecha de inscripción 4 Fecha para firmar 5 Fecha de llegada 6 Fecha de salida 7 Lugar de llegada 8 Lugar de salida 9 Cupo 10 Grupo 11 Profesores 12 Visita a 13 CASCO, CHALECO Y ZAPATO INDUSTRIAL 14 NO LLEVAR CELULAR 15 CARNET O SEGURO CORRESPONDIENTE ORIGINAL 16 CÁMARA 17 Cancelada 18 Clave de la materia 19 Botón Edición de prácticas 20 Mensajes	1 prácticas
La complejidad es de <i>Average</i>	
El tamaño es de 4 <i>FPs</i>	

**Tabla 2.27 Modificación de los datos de las prácticas (E1)**

<b>DETs</b>	<b>FTRs</b>
1 Folio 2 Nombre 3 Fecha de inscripción 4 Fecha para firmar 5 Fecha de llegada 6 Fecha de salida 7 Lugar de llegada 8 Lugar de salida 9 Cupo 10 Grupo 11 Profesores 12 Visita a 13 CASCO, CHALECO Y ZAPATO INDUSTRIAL 14 NO LLEVAR CELULAR 15 CARNET O SEGURO CORRESPONDIENTE ORIGINAL 16 CÁMARA 17 Cancelada 18 Clave de la materia 19 Botón Enviar 20 Mensajes de respuesta	1 practicas 2 asistencias 3 grupos_siprac
La complejidad es de <i>High</i>	
El tamaño es de 6 <i>FPs</i>	

**Tabla 2.28 Número de alumnos inscritos contra asistentes a cada una de las prácticas (EO)**

<b>DETs</b>	<b>FTRs</b>
1 Botón de enviar 2 Folio 3 Nombre de la prácticas 4 Asistieron 5 Inscritos en las prácticas 6 Mensajes de respuesta	1 asistencias 2 prácticas
La complejidad es de <i>Average</i>	
El tamaño es de 5 <i>FPs</i>	

**Tabla 2.29 Generar listas de asistencia por grupo y materia (EO)**

<b>DETs</b>	<b>FTRs</b>
1 Nombre de la práctica 2 Grupo 3 Botón de enviar 4 Mensajes de respuesta 5 Nombre del alumno 6 Lista de prácticas 7 Número de cuenta del alumno 8 Nombre de la materia	1 grupos_siprac\prácticas 2 asistencias 3 alumnos_siprac 4 materias_siprac
La complejidad es de <i>High</i>	
El tamaño es de 7 <i>FPs</i>	

**Conversion functionality****Tabla 2.30 Cargar las prácticas de campo desde un excel (EI)**

<b>DETs</b>	<b>FTRs</b>
1 Folio 2 Asignatura 3 Grupo 4 Profesores 5 Visita a 6 Fecha de salida 7 Fecha de llegada 8 Cupo 9 Clave de la materia 10 CASCO, CHALECO Y ZAPATO INDUSTRIAL 12 NO LLEVAR CELULAR 13 CARNET O SEGURO CORREPONDIENTE ORIGINAL 14 CÁMARA 15 Cancelada 16 Botón de enviar 17 Mensajes de respuesta	1 prácticas 2 profesores 3 grupos_siprac
La complejidad es de <i>HIGH</i>	
El tamaño es de 6 <i>FPs</i>	

**Tabla 2.31 Eliminar prácticas (EO)**

<b>DETs</b>	<b>FTRs</b>
1 Botón de eliminar 2 Mensajes de respuesta	1 Prácticas 2 Asistencias
La complejidad es de <i>LOW</i>	
El tamaño es de 4 <i>FPs</i>	

**Tabla 2.32 Cargar alumnos desde archivos excel (EI)**

<b>DETs</b>	<b>FTRs</b>
1 Nombre de la materia 2 Folio 3 Nombre del profesor 4 Número de cuenta del alumno 5 Nombre del alumno 6 Grupo	1 alumnos_siprac 2 materias_siprac 3 grupos_siprac\profesores_siprac
La complejidad es de <i>HIGH</i>	
El tamaño es de 6 <i>FPs</i>	

**Tabla 2.33 Eliminar alumnos (EO)**

<b>DETs</b>	<b>FTRs</b>
1 Botón de eliminar 2 Mensajes de respuesta	1 alumnos_siprac 2 inscritas 3 contraseñas 4 asistencias
La complejidad es de <i>Average</i>	
El tamaño es de 5 <i>FPs</i>	

Con esto lo que obtenemos es el peso de nuestro sistema, es decir cuando uno compra por ejemplo tortillas tu solicitas un kilo dos kilos y se va pesando las tortillas, la báscula te va dando el precio final de las tortillas agregadas, así es *IFPUG* obtienes la cantidad en “kilos” o en este caso *Function Points (FPs)* y estos al utilizar una “báscula” de *ISBSG* obtienes el tiempo que tardarás en desarrollar el sistema, esto último se verá más adelante.

### **2.1.1.3.2 Specific Practice - Estimar el esfuerzo y el costo (Estimate Effort and Cost)**

Después para estimar el tiempo que se tardará en desarrollar el sistema se utilizará una ecuación de regresión de *ISBSG* que se obtuvo del libro “*Practical Software Project Estimation: A Toolkit for Estimating Software Development Effort & Duration*” en el Apéndice C, por lo tanto, con el siguiente cálculo se obtendrá el tiempo requerido por cada punto de función:

$$Project\ Delivery\ Rate^5 = C * (Size)^{E1} * (MaxTeam)^{E2}$$

En donde C es igual a 37.48, E1 es igual a -0.496 y E2 es igual a 0.759, estos datos se obtuvieron de la tabla C-1-0 del mismo libro y size es el tamaño total del sistema en puntos de función que en este caso es de 140 *FPs*, al utilizar la ecuación se obtiene el siguiente resultado:

$$Project\ Delivery\ Rate = 37.48 * (140)^{-0.496} * (1)^{0.729} = 3.231 \text{ horas por punto de función}$$

Entonces la siguiente lista muestra el tiempo de desarrollo por cada función de transacción.

**Tabla 2.34 Tiempo de desarrollo por cada función de transacción**

<b>Función de transacción</b>	<b>Horas</b>
Formulario de inicio de sesión (EQ)	3 <i>FPs</i> x 3.231 = 9.693 Horas
Formulario para recuperar la contraseña (EO)	3 <i>FPs</i> x 3.231 = 9.693 Horas
Formulario para saber la existencia del estudiantes (EQ)	3 <i>FPs</i> x 3.231 = 9.693 Horas
Formulario para ingresar los datos del estudiante (EI)	3 <i>FPs</i> x 3.231 = 9.693 Horas
Lista de las prácticas en la página de bienvenida (EQ)	3 <i>FPs</i> x 3.231 = 9.693 Horas
Inscripción a alguna práctica (EI)	6 <i>FPs</i> x 3.231 = 19.386 Horas
Lista de prácticas inscritas y a inscribirse (EQ)	3 <i>FPs</i> x 3.231 = 9.693 Horas
Lista de validaciones de asistencia a las prácticas (EQ)	3 <i>FPs</i> x 3.231 = 9.693 Horas
Validar la asistencia a las prácticas (EI)	3 <i>FPs</i> x 3.231 = 9.693 Horas
Lista de prácticas a las que asistió el alumno (EQ)	4 <i>FPs</i> x 3.231 = 12.924 Horas
Constancia de asistencia a las prácticas (EO)	5 <i>FPs</i> x 3.231 = 16.155 Horas
Dropdown de prácticas (EQ)	3 <i>FPs</i> x 3.231 = 9.693 Horas
Generar lista de prácticas (EO)	5 <i>FPs</i> x 3.231 = 16.155 Horas
Inscripción a oyentes o externos (EI)	4 <i>FPs</i> x 3.231 = 12.924 Horas
Inscribir alumno a una materia (EI)	4 <i>FPs</i> x 3.231 = 12.924 Horas
Ver materias inscritas del alumno (EQ)	3 <i>FPs</i> x 3.231 = 9.693 Horas

<sup>5</sup> Peter R. Hill. (2011). *Practical Software Project Estimation A toolkit for Estimating Software Development Effort & Duration*. United States of America: McGraw-Hill.

Lista de alumnos para pasar lista por práctica (EQ)	4 FPs x 3.231 = 12.924 Horas
Lista de prácticas para Editar (EQ)	4 FPs x 3.231 = 12.924 Horas
Modificar datos de las prácticas (EI)	6 FPs x 3.231 = 19.386 Horas
Número de alumnos inscritos contra asistentes a cada una de las prácticas (EO)	5 FPs x 3.231 = 16.155 Horas
Generar listas de asistencia por grupo y materia (EO)	7 FPs x 3.231 = 22.617 Horas
Cargar las prácticas de campo desde un excel (EI)	6 FPs x 3.231 = 19.386 Horas
Eliminar prácticas (EO)	4 FPs x 3.231 = 12.924 Horas
Cargar alumnos desde archivos excel (EI)	6 FPs x 3.231 = 19.386 Horas
Eliminar alumnos (EO)	5 FPs x 3.231 = 16.155 Horas
Queries, Inserts y Updates (ILFs)	42 FPs x 3.231 = 135.702 Horas

Para la estructura principal del sistema cómo son los *headers*, el *hero* y el *footer*, se basó en experiencias anteriores creando este tipo de requerimientos:

*Header* de la página de bienvenida 20 minutos

*Hero* de la página de bienvenida 10 minutos

*Footer* de la página de bienvenida 10 minutos

*Header* de la página del alumno 20 minutos

Sección inicio de la página del alumno 10 minutos

Sección Notas de la página del alumno 10 minutos

Gracias a toda esta información se puede proceder con el desarrollo del plan del proyecto. Todos estos datos fueron obtenidos gracias al esfuerzo de leer la documentación pertinente como el documento *Counting Practices Manual (CPM)*, al igual que algunos libros *Practical Software Project Estimation: A toolkit for Estimating Software Development Effort & Duration* y además el libro *SOFTWARE ESTIMATION Demystifying the Black Art*.

### **2.1.2 Specific Goal - Desarrollar un plan de proyecto (Develop a Project Plan)**

En este punto del proyecto debemos empezar a obtener las fechas de entrega de los requerimientos por lo que se utilizará la información obtenida en el capítulo anterior para desarrollar el calendario y obtener una fecha aproximada para hacer la entrega del sistema web finalizado.

### 2.1.2.1 Specific Practice - Establish the Budget and Schedule & Specific Practice - Establish the Project Plan

Con estas estimaciones se creará el diccionario *WBS*, algunas de las fechas son muy largas para las horas que el requerimiento necesita, el motivo de ello es que se contemplaron las fechas de instalación de programas en los laboratorios de cómputo en la DICyG además de los días inhábiles en el calendario universitario, la siguiente tabla muestra el *WBS* dictionary:

**Tabla 2.35 WBS part 1**

Nombre y número de la tarea	Dueño de la tarea	Descripción de la tarea	Consideraciones	Critería de aceptación	Fecha de finalización
<b>1.3</b> <b>Queries, Inserts, updates y deletes.</b>	Alan Gabriel Barrios Muñoz	Crear consultas, inserts, updates y deletes que apoyarán a las transacciones a completarse.	Las consultas, inserts, updates y deletes deben ser compatibles con <i>MySQL</i> .	Las Consultas, <i>inserts, updates y deletes</i> deben manipular correctamente la información del estudiante.	Fecha de inicio 7 de Diciembre de 2015, Fecha de término 11 de Febrero de 2016, Duración 140 horas
<b>1.1.3.1</b> <b>Header</b>	Alan Gabriel Barrios Muñoz	El <i>Header</i> va a estar conformado por el nombre del sistema, acompañado de dos enlaces: uno que indique "Inicio de sesión" y el otro "Regístrate".	Se entregará un <i>Header</i> ya optimizado para su visualización en cualquier pantalla además de tener un tono azulado para su diseño.	Poder visualizarlo en cualquier dispositivo y navegador.	Fecha de inicio 15 de Febrero de 2016 y Fecha de término 15 de Febrero de 2016, Duración 20 minutos
<b>1.1.3.2</b> <b>Hero</b>	Alan Gabriel Barrios Muñoz	Simplemente es una imagen de un puente con un párrafo dándole la bienvenida al estudiante.	El <i>Hero</i> centrado con sus márgenes y bordes, después del <i>Header</i>	Poder visualizarlo en cualquier dispositivo y navegador.	Fecha de inicio 15 de Febrero de 2016 y Fecha de término 15 de Febrero de 2016, Duración 10 minutos.

**Tabla 2.36 WBS part 2**

Nombre y número de la tarea	Dueño de la tarea	Descripción de la tarea	Consideraciones	Critería de aceptación	Fecha de finalización
1.1.3.3 <i>Footer</i>	Alan Gabriel Barrios Muñoz	El <i>Footer</i> mostrará el <i>copyright</i> , nombre de la universidad, de la facultad y además los logos de <i>Twitter</i> y <i>Facebook</i> que tendrán <i>links</i> a las respectivas páginas de la DICyG en las respectivas plataformas.	Un <i>Footer</i> con el mismo tono azulado que el <i>Header</i> , con el texto acomodado y con el padding que lo visualicé mejor.	Su visualización deberá ser correcta en cualquier dispositivo y navegador.	Fecha de inicio 15 de Febrero de 2016 y Fecha de término 15 de Febrero de 2016, Duración 10 minutos.
1.1.2.1 Inicio de sesión	Alan Gabriel Barrios Muñoz	Formulario para el estudiante que le permitirá ingresar al sistema para visualizar e inscribirse a prácticas de campo	Formulario en <i>HTML5</i> , diseñado con <i>CSS3</i> y permitiendo el envío de la solicitud al <i>backend</i> y mostrando mensajes de error o redireccionando a la página del alumno. También se agregará un botón para mostrar un formulario para recuperar su contraseña.	El alumno deberá visualizar correctamente el formulario, ya sea en un monitor de escritorio o en su celular, además que el funcionamiento se ejecute correctamente no importa el navegador para celular o computadora.	Fecha de inicio 15 de Febrero de 2016 y Fecha de término 18 de Febrero de 2016, Duración 20 Horas
1.1.2.2 Registro del estudiante	Alan Gabriel Barrios Muñoz	El estudiante ingresará en el formulario su Número de cuenta, Nombres y apellidos para verificar que exista y posteriormente se le mostrará un formulario para ingresar los datos solicitados por la universidad para su asistencia a las prácticas de campo.	Formulario que se utiliza para enviar los datos solicitados y cómo respuesta se le muestra uno de los mensajes de error o de éxito.	Su funcionamiento y visualización es correcto en cualquier navegador y dispositivo.	Fecha de inicio 22 de Febrero de 2016 y Fecha de término 23 de Febrero de 2016, Duración 10 Horas.

**Tabla 2.37 WBS part 3**

Nombre y número de la tarea	Dueño de la tarea	Descripción de la tarea	Consideraciones	Critería de aceptación	Fecha de finalización
1.1.2.3 Ingreso de los datos del estudiante	Alan Gabriel Barrios Muñoz	El estudiante al mandar su número de cuenta junto con su nombre completo, se le mostrará otro formulario para ingresar los datos solicitados por la universidad para la asistencia de alumnos a prácticas de campo.	Formulario que el alumno utilizará para ingresar sus datos personales, los cuales, al enviarse, se verificarán que hayan sido llenados correctamente y si todo está correcto se generará la contraseña del alumno y se enviara por el correo que proporcione el alumno.	El formulario debe ser visualizado correctamente en cualquier dispositivo y su funcionamiento correcto en cualquier navegador.	Fecha de inicio 24 de Febrero de 2016 y Fecha de término 25 de Febrero de 2016, Duración 10 Horas
1.1.1.1 Lista de prácticas	Alan Gabriel Barrios Muñoz	Lista de prácticas que se le mostrará a los visitantes de la página de bienvenida.	Una tabla en <i>HTML5</i> con diseño implementado con Materialize.	La tabla debe ser visible correctamente en cualquier dispositivo sin importar el navegador.	Fecha de inicio 29 de Febrero de 2016 y Fecha de término 1 Marzo de 2016, Duración 10 Horas
1.3.2.1 <i>Header</i>	Alan Gabriel Barrios Muñoz	Mostrarle al usuario el nombre del sistema junto con su nombre al ingresar al sistema, junto con una opción para cerrar sesión.	Un <i>header</i> con el color azul presente en la página de bienvenida.	El <i>header</i> debe visualizarse correctamente en distintos dispositivos y en cualquier navegador.	Fecha de inicio 2 de Marzo de 2016 y Fecha de término febrero 2 de Marzo de 2016, Duración 20 minutos.

**Tabla 2.38 WBS part 4**

Nombre y número de la tarea	Dueño de la tarea	Descripción de la tarea	Consideraciones	Critería de aceptación	Fecha de finalización
1.3.2.2.1 Inicio	Alan Gabriel Barrios Muñoz	Sección que explica los diferentes apartados de la página del estudiante.	Párrafos y headers en <i>HTML5</i> , estilizados con <i>Materialize</i> .	El inicio deberá visualizarse en cualquier dispositivo y navegador.	Fecha de inicio 2 de Marzo de 2016 y Fecha de término 2 de Marzo de 2016, Duración 10 minutos.
1.3.2.2.2 Reglamento	Alan Gabriel Barrios Muñoz	Se mostrará el pdf del reglamento de asistencia a las prácticas de campo que se encuentra en la página de la DICyG.	Una sección que carga el pdf del reglamento de asistencia a las prácticas de campo.	El <i>PDF</i> deberá visualizarse correctamente en cualquier dispositivo y navegador.	Fecha de inicio 2 de Marzo de 2016 y Fecha de término 2 de Marzo de 2016, Duración 10 minutos.
1.3.2.2.3 Notas	Alan Gabriel Barrios Muñoz	Notas esenciales que deben seguir los estudiantes para poder finalizar su inscripción a las prácticas de campo.	Párrafos en <i>HTML5</i> dándoles estilo con <i>Materialize</i> .	Se debe visualizar correctamente en cualquier dispositivo y navegador.	Fecha de inicio 2 de Marzo de 2016 y Fecha de término 2 de Marzo de 2016, Duración 10 minutos.

**Tabla 2.39 WBS part 5**

Nombre y número de la tarea	Dueño de la tarea	Descripción de la tarea	Consideraciones	Critería de aceptación	Fecha de finalización
1.3.1.1 Lista de prácticas a inscribirse e inscritas	Alan Gabriel Barrios Muñoz	Al alumno se le mostrará dos secciones, en la primera, se le permitirá visualizar las prácticas por materia para poder inscribirse a alguna de ellas y en la otra sección se le enlistará las prácticas a las cuales se inscribió.	Dos tablas de prácticas de campo dándoles estilo con <i>Materialize</i> .	Se deben visualizar correctamente en cualquier navegador y dispositivo.	Fecha de inicio 2 de Marzo de 2016 y Fecha de término 10 de Marzo, Duración 29 horas.
1.2.1.1 Lista de inscripciones a prácticas por validar	Alan Gabriel Barrios Muñoz	Al administrador se le mostrará por materia las listas de inscripciones de los alumnos a las prácticas, por lo que solamente dará clic en el botón validar, cuando el alumno lo visite y de su peso del seguro.	Dos <i>dropdown</i> con tablas anidadas.	Las tablas deberán visualizarse correctamente en cualquier dispositivo y navegador.	Fecha de inicio 14 de Marzo de 2016 y Fecha de término 17 de Marzo de 2016, Duración 20 horas.
1.2.1.2 Lista de prácticas a las que asistió el alumno	Alan Gabriel Barrios Muñoz	El administrador ingresará el número de cuenta del alumno a generar su(s) constancia(s) a la(s) práctica(s) que asistió para visualizarlas en una lista.	Un formulario para ingresar un número de cuenta del alumno con un botón para solicitar las prácticas a las que asistió, el mensaje de respuesta del sistema será una lista en donde sus elementos sean las prácticas y en cada una de ellas un botón para solicitar la constancia correspondiente a la asistencia.	La visualización y el funcionamiento deberán ejecutarse correctamente en cualquier dispositivo y navegador.	Fecha de inicio 28 de Marzo de 2016 y Fecha de término 31 de Marzo de 2016, Duración 13 horas.

**Tabla 2.40 WBS part 6**

Nombre y número de la tarea	Dueño de la tarea	Descripción de la tarea	Consideraciones	Critería de aceptación	Fecha de finalización
1.2.2 Constancia de asistencia	Alan Gabriel Barrios Muñoz	El administrador al dar clic en el botón generar de alguna de las prácticas en las que asistió el alumno, solicitará al sistema generar un <i>PDF</i> con la información correspondiente.	Las funciones Javascript y de <i>PHP</i> para solicitar la información de asistencia a la práctica de campo.	El funcionamiento deberá ejecutarse correctamente en cualquier navegador.	Fecha de inicio 31 de Marzo de 2016, Fecha de término 6 de Abril de 2016, Duración 17 horas.
1.2.3 Generar excel para lista de alumnos por práctica	Alan Gabriel Barrios Muñoz	Son las listas en donde va la información del alumno junto con su firma para ser entregadas a la universidad junto con el pago del seguro.	Un formulario donde se seleccionará la práctica a la que se desea generar la lista de alumnos.	El funcionamiento deberá ejecutarse correctamente en cualquier navegador.	Fecha de inicio 7 de Abril de 2016, Fecha de término 14 de Abril de 2016, 26 horas.
1.2.4 Inscripción a alumnos oyentes o externos	Alan Gabriel Barrios Muñoz	Al administrador se le proporcionará dos formularios uno para registrar al alumno y el segundo para inscribir al alumno a una materia, para que éste pueda ingresar al sistema e inscribirse a las prácticas de campo que desee.	Dos formularios.	Los formularios deberán visualizarse y funcionar correctamente en cualquier dispositivo y navegador.	Fecha de inicio 18 de Abril de 2016, Fecha de término 26 de Abril de 2016, Duración 26 horas.

**Tabla 2.41 WBS part 7**

Nombre y número de la tarea	Dueño de la tarea	Descripción de la tarea	Consideraciones	Critería de aceptación	Fecha de finalización
1.2.5 Pasar lista a alumnos por práctica	Alan Gabriel Barrios Muñoz	El administrador seleccionará alguna de las prácticas para pasar lista a los alumnos y también ayudará a seleccionar quién de los alumnos tendrá constancia.	Un formulario para seleccionar la práctica y un botón para enviar la solicitud, posteriormente la respuesta del sistema será la lista de alumnos inscritos en la práctica y se le mostrará dos opciones asistió o no asistió y un botón para enviar la información al sistema y que ésta se guarde.	El formulario deberá visualizarse y funcionar correctamente en cualquier dispositivo y navegador.	Fecha de inicio 26 de Abril de 2016, Fecha de término 4 de Mayo de 2016, Duración 23 horas.
1.2.6 Generar lista de asistencia por grupo y materia	Alan Gabriel Barrios Muñoz	El administrador podrá entregar a cada profesor una lista de sus estudiantes con las prácticas a las que asistió.	Un formulario para seleccionar la materia y el grupo, además de un botón para enviar la solicitud al sistema, en respuesta se le enviará un archivo excel.	El formulario deberá visualizarse y funcionar correctamente en cualquier dispositivo y navegador.	Fecha de inicio 4 de Mayo de 2016, Fecha de término 11 de Mayo de 2016, Duración 23 horas.
1.2.7 Generar estadísticas de asistencia por práctica	Alan Gabriel Barrios Muñoz	El administrador podrá seleccionar la práctica y solicitar un excel con la siguiente información: Nombre y folio de la práctica, número de asistencias y número de alumnos inscritos.	Un formulario con un <i>dropdown</i> para seleccionar la práctica y un botón para solicitar el archivo excel al sistema, en respuesta se le descargará el archivo solicitado.	El formulario deberá visualizarse y funcionar correctamente en cualquier dispositivo y navegador.	Fecha de inicio 12 de Mayo de 2016, Fecha de término 18 de Mayo de 2016, Duración 17 horas.

**Tabla 2.42 WBS part 8**

Nombre y número de la tarea	Dueño de la tarea	Descripción de la tarea	Consideraciones	Critería de aceptación	Fecha de finalización
1.2.8 Editar prácticas	Alan Gabriel Barrios Muñoz	Lista de prácticas con toda su información y un botón para poder activar la edición.	Tabla en <i>HTML5</i> con estilo implementado con <i>Materialize</i> .	La tabla deberá visualizarse y en cualquier dispositivo y navegador.	Fecha de inicio 18 de Mayo de 2016, Fecha de término 31 de Mayo de 2016, Duración 33 horas.
1.2.9 Agregar o eliminar prácticas	Alan Gabriel Barrios Muñoz	El administrador tendrá la posibilidad de cargar un archivo excel con las prácticas de campo que se impartirán durante el semestre.	Formulario y botón para enviar el archivo, posteriormente en el <i>backend</i> se leerá el archivo para cargar los datos a la base de datos.	El formulario deberá visualizarse y funcionar correctamente en cualquier dispositivo y navegador.	Fecha de inicio 11 de agosto de 2016, Fecha de término 23 de agosto de 2016. Duración 33 horas.
1.2.10 Agregar o eliminar alumnos	Alan Gabriel Barrios Muñoz	El administrador tendrá la posibilidad de cargar un archivo excel por grupo con los datos de los alumnos.	Formulario y botón para enviar el archivo, posteriormente en el <i>backend</i> se leerá el archivo para cargar los datos a la base de datos.	El formulario deberá visualizarse y funcionar correctamente en cualquier dispositivo y navegador.	Fecha de inicio 23 de agosto de 2016, Fecha de término 7 de septiembre de 2016. Duración 36 horas.

El costo total del proyecto sería 4500 pesos/mes x 9 meses de duración es igual a 40500 pesos aproximadamente. Con esta información se procederá a crear el diagrama de *Gantt* utilizando el programa llamado *GanttProject*, la siguiente imagen lo muestra:

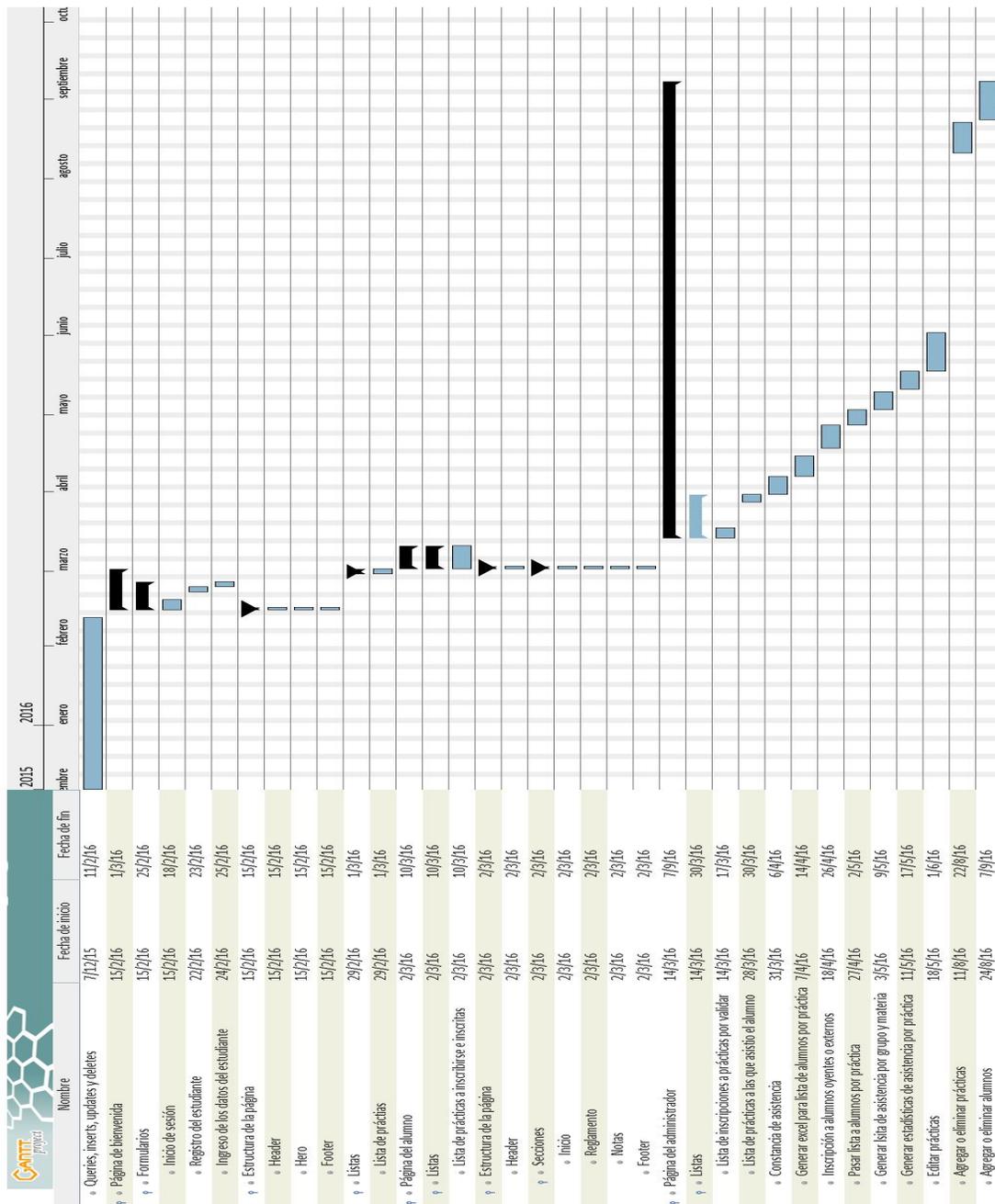


Figura 2.30 Diagrama de Gantt

En este caso el trabajo a realizar no conlleva a tener un presupuesto, ya que el puesto conlleva la realización de las siguientes actividades:

1. Administrar los sistemas existentes que contempla la carga de datos a la base de datos, manejo de errores, creación de nuevos sistemas.
2. Soporte técnico.
3. Instalación de software en el edificio solicitado por los profesores que imparten clases en los laboratorios de computación.

Por todo esto cada mes se le paga a la persona a cargo 4500 pesos, pero no se tiene un presupuesto para el tiempo requerido para hacer nuevos sistemas, sino que se va continuando mes con mes el desarrollo de los nuevos sistemas.

#### **2.1.2.2 *Specific Practice* - Identificar Riesgos del Proyecto (*Identify Project Risk*)**

Los principales riesgos que conlleva el realizar este proyecto son:

1. El tiempo estimado puede ser afectado por actividades de soporte técnico, ya que hay días en que se presentan varios problemas como la instalación de un programa, error en un programa de ofimática, que una computadora no prende, entre otras situaciones, por lo que en estos casos no nos damos abasto solo con las personas de servicio social entonces uno tiene que dejar de desarrollar o de mantener los otros sistemas para resolver esos problemas, tales problemas conllevan un tiempo desde media hora hasta tres horas en poder finalizar el soporte por lo que el tiempo de desarrollo del sitio web SIPRAC será afectado.
2. También se tiene el tiempo de instalación del software que se realiza cada intersemestral para los programas requeridos por los profesores además de los cursos impartidos, por ello ya se tiene establecido dentro de la planeación este tiempo.
3. Los recursos que se tienen para el desarrollo del sistema web es una máquina asignada, pero en cualquier caso de que falle alguno de los componentes como disco duro o tarjeta gráfica se tienen repuestos para su posible cambio.

#### **2.1.2.3 *Specific Practice* - Planificar la gestión de datos (*Data Management Plan*)**

Los datos que se recopilan en el sistema solo son accesibles para los ingenieros a cargo de las prácticas de campo y para la universidad, ya que es información necesaria para que se acepten las visitas. Por ello de ninguna forma se realizará la venta de la información recabada y generada en el sistema, así como el compartir la información con ninguna institución por parte de la Unidad de Computo (UC) de la DICyG.

Los datos recabados serán respaldados con fines de aclaraciones de la asistencia de algún alumno u estadísticas que se soliciten por parte de los administradores de las prácticas y de los profesores que lo soliciten, por ello cada fin de semestre se hará el respaldo de la información y no se visualizará más en el sistema.

La administración de los datos de las prácticas junto con la información pertinente a cada materia, profesor y alumnos se manejará cada semestre para hacer la carga de los datos, esta acción será realizada por el jefe del departamento por ello los profesores comparten sus listas con él, es posible esta situación ya que se solicitó esta funcionalidad para facilitar la administración en caso de no estar disponible el encargado de los sistemas.

La manipulación de los datos en el sitio web solo es posible con la cuenta del administrador y sólo es accesible dentro de la red interna de la DICyG, por ello en ningún momento será posible para ninguna persona externa acceder a esta parte del sistema fuera del edificio y que no tenga acceso a la red interna, con ello mejoramos la seguridad en los datos para que solo la persona encargada los administre y los visualice.

Las constancias generadas en el sistema son solamente para que el estudiante que las solicitó muestre en sus materias la asistencia y valide el requerimiento de su profesor para pasar la materia, la Unidad de Cómputo y la DICyG no se hacen responsables con la información que el estudiante solicitó.

#### **2.1.2.4 Specific Practice - Plan de Recursos del Proyecto (Plan the Project's Resources)**

Los recursos necesarios para que se desarrolle el sistema se proporcionaron con anterioridad, los cuales son:

1. Una computadora
2. Un teclado
3. Un mouse
4. Dos monitores
5. El sistema operativo que se seleccionó fue un *Ubuntu*.
6. El espacio en los cubículos es adecuado para un ambiente de trabajo.
7. la energía es proporcionada correctamente para que los dispositivos funcionen correctamente y la iluminación está bien. Con esto concluimos este apartado.

### **2.1.2.5 Specific Practice - Plan de conocimientos y habilidades necesarios (*Plan Needed Knowledge and Skills*)**

Los conocimientos necesarios para el desarrollo del sitio web SIPRAC son:

1. *PHP 5.6*
2. *Framework CodeIgniter*
3. *HTML5*
4. *CSS3*
5. *Materialize*
6. *MySQL*
7. *SQL*
8. *Framework PHPExcel*
9. *Framework PDFPHP*
10. *Postman*

Para el entrenamiento necesario para realizar este sitio web fue hecho “*in-house training*” y además por investigación propia de los frameworks implementados. En el caso del “*in-house*” se proporcionó clases para utilizar *CodeIgniter*, *HTML5*, *CSS3*, *Materialize* y *PHP*, los demás conocimientos fueron adquiridos por cuenta propia con aclaraciones de que *framework* se podrían utilizar. Ya que el desarrollo de este proyecto solo se llevó a cabo por un solo desarrollador no hará falta más clases o tutoriales para el desarrollo del sitio web o contratar a más personal.

### **2.1.2.6 Specific Practice - Planificar la participación de las partes interesadas (*Plan Stakeholder Involvement*)**

Las principales personas que manejarán el sistema son los ingenieros:

1. El jefe del departamento
2. Ayudantes en la administración de las prácticas de campo

Con los cuales vamos a tener juntas de acuerdo con cada “*Milestone*”, consideramos esto ya que representan avances importantes, por ello se necesita la aceptación del cliente del proceso y del diseño que se implementó en el sistema. Las juntas se harán en el cubículo donde se lleva el desarrollo del sistema, y se establecerá la fecha de revisión una vez se haya llegado a la “*Milestone*”.

### **2.1.2.7 Specific Practice - Comprometerse con el plan (*Obtain Commitment to the Plan*)**

Todas las modificaciones requeridas, los riesgos que se establecieron van a ser informadas a la jefa del departamento de la Unidad de Cómputo y a los ingenieros que solicitaron el sitio web, por lo que siempre se hará de manera física en sus cubículos. También las modificaciones se agregarán a los documentos para que reflejen los nuevos

requerimientos. Además, siempre se escuchará toda idea para implementarse en el sistema y se comentará para que sea agregado al proyecto dando a entender que esto aumentará el tiempo establecido en el desarrollo.

Con esto se finaliza este capítulo, toda documentación generada fue utilizada para llevar un orden, respetar los requerimientos además para tener listo a tiempo el sistema y solucionar la duda que toda persona tiene al desarrollar o solicitar un sistema que es el tiempo que se tardará en terminar el proyecto junto con una fecha estimada.

## Capítulo 3. Desarrollo de la base de datos

### 3.1 Bases de datos

En el desarrollo del sitio web SIPRAC, se utilizó un sistema de gestión de bases de datos relacionales (*RDBMS* por sus siglas en inglés), llamada *MYSQL*, desarrollada con los lenguajes de programación C y C++, la empresa encargada de su desarrollo es Oracle Corporation® y las conexiones a esta *RDBMS*<sup>6</sup>, que son una colección de programas que administran la estructura y controlan el acceso a las bases de datos, es muy fácil con varios lenguajes de programación como son: *PHP*, C, C++, *JAVA* y muchos lenguajes más.



Figura 3.1 bases de datos

La manera de almacenar los datos es a través de tablas, en las cuales se especifican los atributos o columnas. En el caso de este proyecto se tienen varias tablas, por ejemplo:

---

<sup>6</sup> *Relational Data Bases Management System*

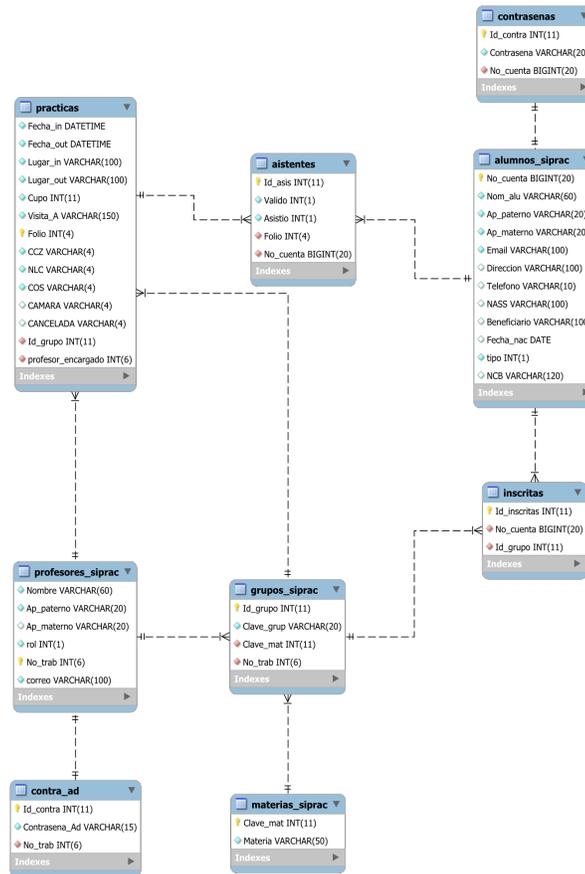


Figura 3.2 Diagrama Entidad Relación SIPRAC

### Modelo de bases de datos relacional:

Hoy en día se manejan gran cantidad de datos en los sistemas para almacenarla o procesarla, por ello en el ejemplo dado sobre cómo un profesor genera información a partir de los datos que tiene en su lista no es necesario un sistema para que el profesor la analice, a menos que tenga demasiados alumnos y grupos, en este caso solo se piensa en una pequeña cantidad, pero ¿qué pasa cuando todo esto se necesita hacer en grandes cantidades?, como es el caso de este proyecto que resuelve el problema de inscripciones a prácticas de campo:

- 1 Primero almacenando los datos de los alumnos para generar listas en las cuales se indica nombre completo, fecha de nacimiento, dirección, entre otros datos, con el fin de que firmen la hoja para su seguro. También se guardan los datos de cada práctica, por ejemplo, lugar, nombre, horario y profesor asignado. Inscripciones a cada práctica, por ejemplo, enlazar al alumno con su práctica.

- 2 En segundo lugar, procesando esos datos para su análisis, tomando la decisión de proporcionar o no una constancia de la práctica a la cual se asistió, por lo que en el sistema se debe de registrar su asistencia, en el caso de que sí se haya asistido, el sistema le generará la constancia.

En este caso estamos hablando de la administración de más de diez prácticas al semestre, lo que causa una gran cantidad de trabajo si no se implementara el sistema y no solo eso, también se realiza un análisis de cuantas personas se inscribieron y cuantas fueron, generando así información a partir de los datos almacenados, por lo que una base de datos relacional, es el modelo que facilitó el diseño e implementación del almacenamiento de los datos para SIPRAC.

Las bases de datos relacionales que han existido más de cuarenta años gracias a **Edgar Frank Codd** que puso los cimientos, éstas se definen como “una computadora estructurada, compartida e integrada que almacena una colección de lo siguiente:

- Datos del usuario final
- Metadatos que son datos del dato, es decir describen el contenido que puede ser almacenado, como por ejemplo un número entero que en *MYSQL* puede contener un número del -2147983648 hasta 2147983647, esto describe los límites que puedes almacenar en un campo que sea tipo entero.



**Figura 3.3 MYSQL**

La forma en que se almacenan los datos de una base de datos implementando el modelo relacional, se logra gracias a las estructuras matemáticas que se usaron para la creación de este modelo que son: la teoría de conjuntos y la lógica de primer orden. Las cuales ayudaron a almacenar los datos en estructuras llamadas tablas que almacenan atributos que nos ayudan en el proceso de negocio, un conjunto de estos atributos se llaman registros, tuplas o filas, cada uno de estos registros se identifican de manera única por un atributo en la tabla, agregando así la ventaja de solicitar los datos sin tener forzosamente que saber el lugar en el disco duro en donde están almacenados los datos, por lo que la forma de reunir los datos de una tabla junto con los de otra, se logra con las relaciones entre ellas, a diferencia de los modelos anteriores, el relacional logra hacer uno a uno, uno a muchos y de muchos a muchos, facilitando el diseño de la base de datos.

Cuando se habla sobre las relaciones de uno a uno, uno a muchos o de muchos a muchos, se refiere a la cantidad de datos que puedes obtener al relacionar un registro de una tabla con otra, al coincidir los identificadores (IDs) del registro junto con una de las columnas de la otra tabla, por ejemplo:

- 1 En el caso de la relación uno a uno, se tiene el ejemplo de dos tablas uno de los nombres de los estados de México y la otra tabla es de la capital de cada uno.
- 2 En la relación uno a muchos, el ejemplo que se tiene es de una tabla que almacena la información de los clientes de Amazon, principalmente sus nombre, edades y fechas de registro en el sistema, la otra tabla almacena las direcciones del cliente donde Amazon puede enviar las compras.
- 3 La última relación muchos a muchos, se refiere a que varios registros de una tabla pueden tener varias coincidencias con registros de la otra tabla, un ejemplo muy claro es el de los estudiantes y las materias, tal ejemplo muestra claramente que un alumno puede estar inscrito en varias materias al mismo tiempo y que varias materias pueden tener una cantidad considerable de alumnos.

Una base de datos almacena los datos de los usuarios finales, pero la estructura de la base de datos es almacenada en un conjunto de archivos, los datos que almacena la base de datos son guardados en bloques de datos que son mapeados a un sector de una pista en el disco duro. Estos bloques pueden almacenar tantos registros que quepan en el tamaño del bloque que puede ser en *MySQL* de 1KB, 16KB o 64KB.

La mayoría de *RDBMS* soportan multiusuario, es decir que más de un usuario puede estar conectado a la vez, por lo que puede llevar a una sobrecarga de trabajo en el servidor, para ello existen las llamadas bases de datos distribuidas, pero en este caso lo mejor es una base de datos centralizada ya que la carga de trabajo es menor en el sistema implementado. “La manera más común de clasificar a las bases de datos son en cómo se usan, por ejemplo, dando servicios para almacenar las compras de un cliente en el supermercado, o servicios para pagar tus compras en línea o en esencia operaciones críticas de día con día, por lo que estas bases de datos se les clasifican como una base de datos operacional, también conocidas como *online transaction processing* ó *OLTP*. Por otro lado, las bases de datos que se encargan de almacenar datos históricos o métricos de negocio usado para su análisis se realizan para toma de decisiones por ello son llamados *OLAP* ó *online analytical processing*”<sup>7</sup>.

---

<sup>7</sup> Jan L. Harrington. (2016). relational database design and implementation. United States of America: Morgan Kaufmann.

### 3.2 Diagrama Entidad Relación

En la División de Ingeniería Civil y Geomática se solicitó el desarrollo de un sistema web que facilitará la administración y generación de las listas de alumnos, así como las inscripciones de los alumnos en las prácticas, constancias de asistencia, además de otras funcionalidades. Por lo que se debe hacer uso de datos almacenados en la base de datos para llevar a cabo cada una de las funcionalidades y cómo se van a utilizar, para ello debemos mostrar los datos de acuerdo con lo que se requiere en el sistema y así cada usuario podrá seleccionar las prácticas que desea inscribirse, al igual que los datos involucrados como la fecha de ida y de regreso, como del equipo a llevar, entre otra información.

Para realizar lo anterior se debe realizar el diseño de la base de datos con la ayuda del siguiente modelo de datos, para poder entender la abstracción de la información del mundo real al ordenador y saber cómo se relacionan los datos de los alumnos con las prácticas y los profesores, por lo que podemos visualizar los datos necesarios para poder realizar los procesos de inscripción, generación de constancias, de impresión de alumnos inscritos en cada práctica y así poder crear a partir del modelo nuestra base de datos en cualquier *RDBMS*.

Lo que compone a un modelo de datos es la entidad, la cual nos proporcionara la abstracción de los objetos o cosas que queramos representar en la base de datos como, por ejemplo: carros, casas o en este caso un estudiante, también podemos agregar una entidad para representar las inscripciones de los estudiantes a las prácticas de campo. También lo componen los atributos, estos lo que proporcionan son las características que deseamos almacenar ya sea en el caso de los estudiantes sus nombres, fechas de almacenamiento y en el caso de las inscripciones se almacena quién se inscribió, a que práctica, su fecha de inscripción y otros atributos.

Otro elemento importante en la base de datos son las relaciones de los datos que están en las entidades, para así obtener la información necesaria, por ejemplo se desea obtener el número de alumnos inscritos en la práctica "LABORATORIO DE COMISION FEDERAL DE ELECTRICIDAD EN CUERNAVACA " y de asistentes, en este caso se tendría dos entidades inscripciones y Materias, cuyos atributos para relacionar sus registros son el id de la inscripción y el id de la materia respectivamente, en este caso el atributo en común en las dos entidades es el id de la materia, ya que está presente en las dos entidades, por lo que los requerimientos extras se solucionan agregando a la consulta los predicados necesarios, que son los que limitan el número de registros a obtener.

Por último, los *constraints* que permitirán revisar la correcta existencia de registros en las tablas, al igual que la relación que hay entre entidades, por ejemplo, el obligar que exista la práctica en la entidad practicas cuando el usuario quiera inscribirse a una y así no se almacenen datos incoherentes.

Una parte muy importante para tener un modelo de datos son las reglas de negocio, las cuales indica el uso de los datos en el sistema. Unos ejemplos en relación con el sistema de Inscripciones a Prácticas de Campo son:

Unos ejemplos con relación al sistema de Inscripciones a Prácticas de Campo son:

- I Los estudiantes se pueden inscribir a más de una práctica de campo con un límite de cinco.
- II Los estudiantes de una misma materia, pero diferente grupo se pueden inscribir a una misma práctica si así lo indican los datos en la hoja de excel.
- III La inscripción a una práctica no asegura una constancia, el estudiante debe asistir y pasar lista.
- IV Los alumnos pueden estar inscritos en más de una materia.

En estas reglas de negocio nos indican varias entidades que conforman al modelo de datos de SIPRAC las cuales son:

- Estudiantes
- Materias
- Prácticas
- Inscripciones

Claro que no todas las reglas de negocio aplican para obtener el modelo de datos.

Una de las principales cosas que se debe hacer es identificar las relaciones entre las entidades las cuales obtenemos al preguntarnos lo siguiente:

- I “¿Cuántas instancias de “Estudiantes” están relacionadas con una instancia de “Materias”?”
- II “¿Cuántas instancias de “Prácticas” están relacionadas con una instancia de “Inscripciones”?”

Tenemos las siguientes entidades Estudiantes y Prácticas, ¿Cómo se relacionan?, para responder, las siguientes preguntas ejemplifican una de las tantas relaciones en la base de datos:

- A ¿Cuántas instancias de “Estudiantes” están relacionadas con una instancia de “Prácticas”?

R = Varios “Estudiantes” pueden inscribirse a varias “Prácticas” al mismo tiempo y las “Prácticas” pueden tener varias inscripciones, por lo que tenemos una relación muchos a muchos;

B ¿Cuántas instancias de “Materias” están relacionadas con una instancia de “estudiantes”?<sup>8</sup>

R = Las “Materias” pueden tener varias inscripciones por parte de los alumnos y los Alumnos están inscritos a varias materias, por lo que tenemos una relación de muchos a muchos.

Cuando tenemos dos entidades que se relacionan entre sí, éstas pueden leerse de izquierda a derecha y de derecha a izquierda, en este caso tenemos que todos los estudiantes pueden estar inscritos en varias prácticas y que todas las prácticas pueden tener varios participantes, por lo que nos da una relación muchos a muchos.

El siguiente elemento de un modelo de datos son los atributos que describen a las entidades, en esencia tenemos a los estudiantes cuyas características que los identifican y son necesarias para el sistema son su nombre, número de cuenta, fecha de nacimiento entre otros datos, uno de los atributos más importantes es el identificador de cada uno de los registros, es decir para asegurar que cada registro sea un estudiante el cual no debe repetirse en esta identidad se utiliza el id (en este sistema el número de cuenta) y por si se ingresa otro estudiante con el mismo número de cuenta, no lo permita el *RDBMS*, es aquí donde se presenta un *constraint* el *PRIMARY KEY*, el cual nos asegura que el atributo (el id) debe tener un valor diferente de nulo y de acuerdo a su subdominio, es decir en este caso que solo contenga números y no otro carácter, además será único para cada registro.

Después de saber qué conforma un modelo de datos debemos representarlo gráficamente, para facilitararlo se utiliza un diagrama entidad relación, hay dos representaciones más usadas:

- 1 Chen nos muestra los elementos de la base de datos, tal diagrama permite a todas las personas involucradas en el desarrollo del sistema ver los datos que se manejan en el desarrollo del sistema, por ejemplo, la persona encargada del frontend proporcionara una vista donde involucre la muestra de la información de las prácticas en las cuales ingreso al sistema, hay otra persona que se encarga de la programación de la lógica del sistema que es en el *backend* y por último el diseñador de la base de datos para crear el modelo físico de la base de datos en la *RDBMS*.

Para ello chen representa a las entidades en un rectángulo el cual tiene en el centro su nombre, también los atributos se muestran en este diagrama por medio de una elipse, en la cual se tiene en el centro el nombre del atributo, alrededor de la entidad se ubican sus atributos y para determinar que atributos le pertenece a que entidad, estos se unen a cada una con una línea recta con una característica que es el

---

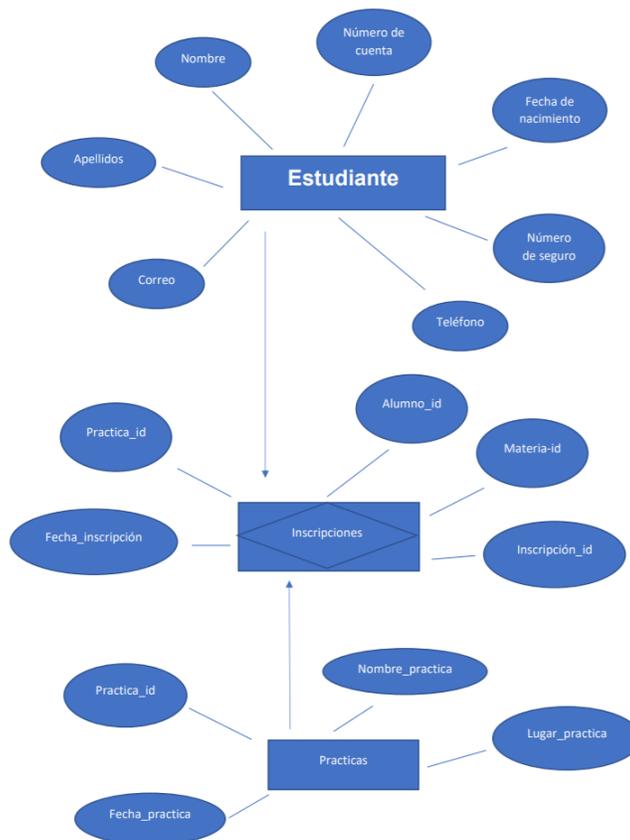
<sup>8</sup> Jan L. Harrington. (2016). relational database design and implementation. United States of America: Morgan Kaufmann.

subrayado del id de cada entidad (que es el que identifica de manera única a cada instancia de la entidad).



**Figura 3.4 En esta imagen se muestra el diagrama de una entidad con sus atributos.**

Para indicar las relaciones entre las entidades en el diagrama se usa una flecha para indicar que una instancia de una entidad llamada "Estudiante" solamente está relacionada con una instancia de la entidad "Practicas", pero la alternativa es con el número uno para solo una relación y M para muchas.



**Figura 3.5 Relación entre dos entidades muchos a muchos**

El diagrama entidad relación Chen es bueno para mostrar el diseño de la base de datos, pero no es un estándar en el *software*, poniendo un ejemplo *MySQL Workbench* que te permite crear el diagrama de la base de datos desde cero u obtener el diagrama de la base de datos de una ya implementada en su motor, pero ese diagrama no se muestra en formato chen sino que se muestra con el formato *crow's foot*, el cual es mucho más estructurado, no ocupa mucho espacio y está implementado en la mayoría del *software* de generación de diagramas de base de datos.

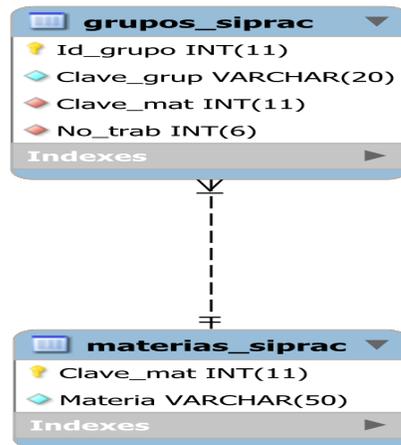
- 2 El diagrama entidad relación formato *crow's foot* nos muestra las entidades en un rectángulo el cual en la parte superior se encuentra el nombre del objeto a representar en la base de datos junto con sus atributos, separando uno de ellos el cual es el ID que diferencia de manera única a cada una de las instancias en esa tabla, como se muestra en la siguiente imagen:



**Figura 3.6 Representación de una tabla en *MySQL Workbench***

La figura 3.6 muestra la representación de una entidad en un diagrama *crow's foot* con los datos de los estudiantes a almacenar, en este caso particular se solicitó el nombre del alumno, apellidos, teléfono entre otros atributos para poder generar las listas en donde firman los alumnos para su seguro.

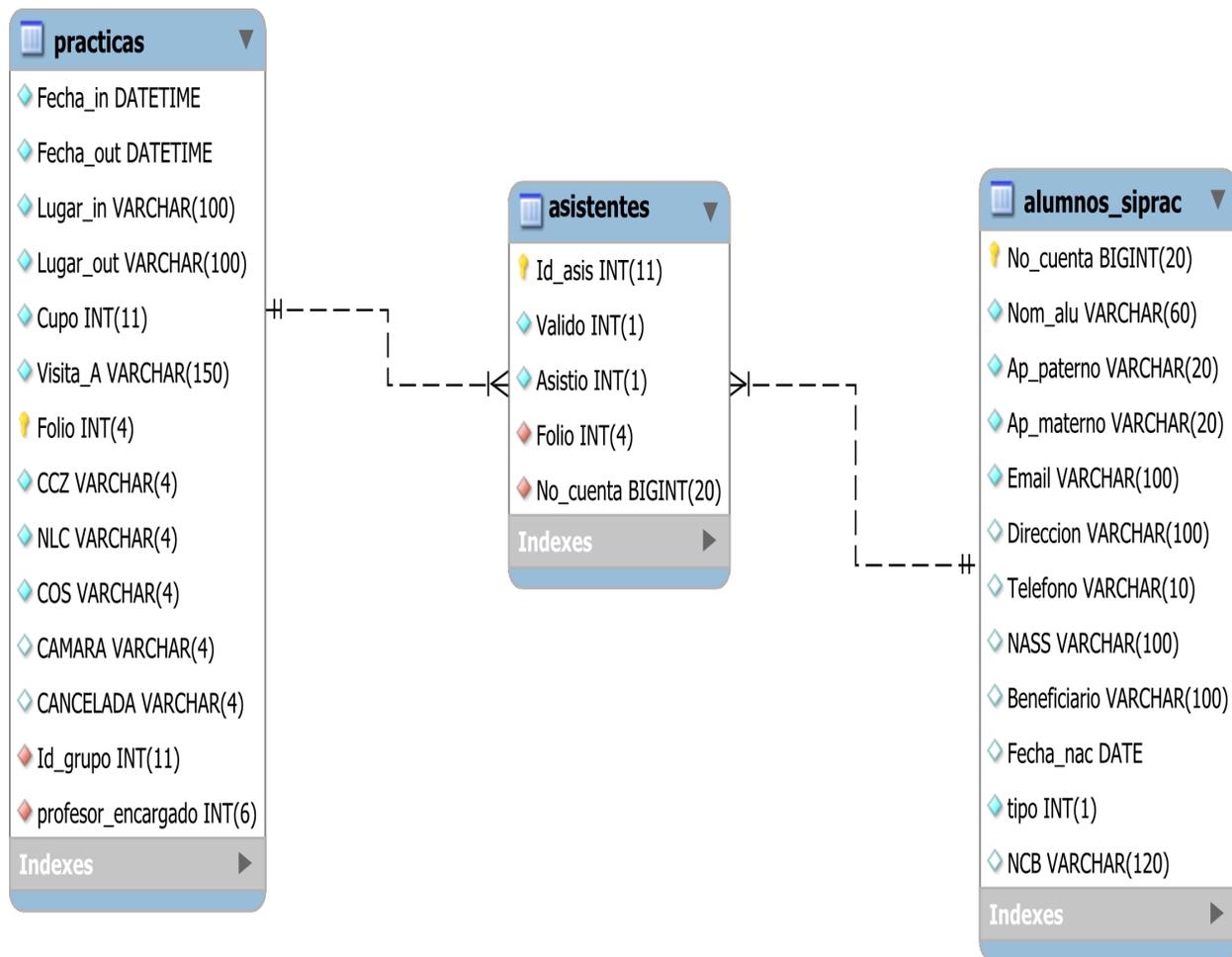
En este tipo de diagrama al igual que Chen, hay una forma de representar las relaciones entre las entidades, un ejemplo para mostrar la representación de las entidades junto con sus relaciones en *crow's foot* es la de una entidad de materias y una de grupos, la entidad de materias se une con una línea la cual en uno de sus extremos tiene una línea perpendicular a ella que representa la relación de una de sus instancias con una de grupos, el otro extremo se muestra como el nombre del diagrama lo indica con una pata de pollo que muestra que una materia puede tener varios grupos. Por lo que se leería que una materia puede tener varios grupos y que un grupo sólo pertenece a una materia. La siguiente imagen muestra el diagrama entidad relación formato *crow's foot* de las dos tablas:



**Figura 3.7 Relación de dos tablas uno a muchos**

En la figura 3.7 se muestra la relación identificativa de una entidad llamada Materias que es una relación de uno a muchos, nos indica que un grupo solamente pertenece a una sola materia y que una materia tiene varios grupos.

Para entidades que tienen una relación mucho a muchos, se crea una tabla intermedia que almacena los datos como los IDs de las dos entidades involucradas, entre otros atributos, un ejemplo de una relación muchos a muchos es entre la entidad Alumnos y Prácticas, la cual tiene una entidad intermedia que se crea porque cada alumno puede tener varias prácticas inscritas y varias prácticas pueden ser seleccionadas por varios alumnos, la relación es no identificativa, lo que indica que los IDs en la entidad intermedia no serán utilizadas para identificar de manera única a las instancias y que por lo tanto se tendrá otro atributo como ID, esta entidad intermedia también involucra atributos que describen alguna propiedad de la inscripción como la fecha y hora a la cual se inscribió, también podemos almacenar si el alumno válido, si asistió a la práctica, entre otros. El siguiente diagrama muestra la relación explicada:



**Figura 3.8 Relación entre varias tablas**

Cada una de las relaciones involucran el identificador de la tabla padre esté presente en la tabla hija, por lo que debe haber un constraint que verifique que el registro ingresado en nuestro ejemplo anterior que el alumno exista cuando se inserte una instancia en la entidad asistentes y también asegura que no se elimine ninguna instancia en la entidad alumnos si existe algún registro que pertenezca a ese alumno en asistentes.

La implementación del diseño de la base de datos en un *RDBMS* se logra con el lenguaje de consulta estructurado que nos permite crear nuestras entidades que en el motor son llamadas tablas, los atributos son mostradas en columnas, y muestra los registros como fueron ingresados, la siguiente imagen muestra un ejemplo:

No_cuenta	Nom_alu	Ap_paterno	Ap_materno
109001863	JULIAN	MOLINA	PADILLA
310063689	SERGIO ENRIQUE	CRUZ	LUGO
310270740	EMMANUEL	HERNANDEZ	LOPEZ

**Figura 3.9 Registro de alumnos en la tabla alumnos\_siprac**

### 3.3 SQL

Hay varios lenguajes dentro de *SQL* que nos permiten manipular nuestras estructuras, datos, estos son lenguajes principalmente que siguen un estándar para poder ser utilizados en los motores de bases de datos existentes, aunque claro varios de ellos a pesar de que utilizan un estándar no siempre son implementados tal cual, sino que agregan sus propias opciones al lenguaje.

#### Lenguaje de Definición de Datos (*Data Definition Language*)

Este primer lenguaje nos ayudará a crear las estructuras donde se almacenarán los datos. La cual tiene sentencias para crear nuestra base de datos y específicamente las tablas, estas dos son llamadas objetos de la base de datos. Lo que nos proporciona es el almacenamiento de varias bases de datos, una base de datos no ayuda a almacenar cada tabla, a su vez una tabla almacenar los datos requeridos para nuestro sistema, los datos son almacenados en las tablas como registros, los cuales son un conjunto de columnas que especifican cada dato que queremos almacenar, por ejemplo, el nombre, fecha de nacimiento, fecha de registro en el sistema y mucho más. Cada tabla puede almacenar tantos registros que se desea, considerando el tamaño de almacenamiento disponible.

Las sentencias en orden que ayudan a crear los objetos de la base de datos son los siguientes:

- I Primero debemos ingresar al servidor, en este caso se realizó la prueba en un servidor *MySQL* local, por lo tanto, no es necesario agregar la opción `-host 127.0.0.1` para especificar donde conectarnos, ni tampoco el puerto `-P 3036`, ya que el motor de la base de datos lo utiliza por *default*, si no se ingresan estas opciones, las opciones que tomas son el puerto por default y se intenta conectar a un servidor que este en la misma máquina.
- II Luego al ingresar a la base de datos lo que nos proporciona es la facilidad de crear una base de datos para después seleccionarla.
- III Por último, lo que se hace es crear la tabla o tablas donde se almacenara los datos del sistema a desarrollar.

Todos estos pasos se muestran en la siguiente imagen:

```
Command Prompt - mysql.exe -u root -p
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Gabriel>cd C:\xampp\mysql\bin

C:\xampp\mysql\bin>mysql.exe -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.4.6-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database siprac;
Query OK, 1 row affected (0.056 sec)

MariaDB [(none)]> use siprac;
Database changed
MariaDB [siprac]> create table alumnos_siprac(
  -> alumno_id int(10) auto_increment primary key,
  -> nombre varchar(60) not null,
  -> edad int(3) not null,
  -> CONSTRAINT ChecarEdad CHECK (edad BETWEEN 15 and 90)
  -> );
Query OK, 0 rows affected (0.306 sec)

MariaDB [siprac]>
```

Figura 3.10 SQL para crear la tabla alumnos\_siprac

### 3.3.1 Lenguaje de manipulación de datos

#### Data Manipulation Language

Este lenguaje nos permite consultar, actualizar y eliminar los datos almacenados en la base de datos, al igual que nos permite insertar nuevos datos, para ello los siguientes ejemplos de cada operación:

##### 3.3.1.1 Insertar los datos en una tabla

Para ingresar los datos en una tabla debemos indicar las palabras reservadas “*INSERT INTO*” en *MySQL*, después se teclea el nombre de alguna tabla existente en la base de datos, después se indica la palabra reservada “*VALUES*” por último entre paréntesis debemos ingresar los valores separados por coma y en orden de especificación de las columnas a la hora de creación de la tabla, finalizando la sentencia con un punto y coma, este es un solo ejemplo de cómo se debe de realizar un *INSERT*, pero en realidad se tiene diferentes maneras de cambiar la sentencia para poder ingresar los datos deseados. La siguiente imagen muestra un ejemplo de acuerdo con la anterior explicación:

```
MariaDB [siprac]> create table alumnos_siprac(
-> alumno_id int(10) auto_increment primary key,
-> nombre varchar(60) not null,
-> edad int(3) not null,
-> CONSTRAINT ChecarEdad CHECK (edad BETWEEN 15 and 90)
-> );
Query OK, 0 rows affected (0.306 sec)

MariaDB [siprac]> INSERT INTO alumnos_siprac VALUES (1,'Gabriel',25);
Query OK, 1 row affected (0.291 sec)
```

### 3.11 SQL para ingresar un registro en la tabla alumnos\_siprac

#### 3.3.1.2 Consultar los datos ingresados

Para consultarlos debemos formar una sentencia que comienza con la palabra reservada “*SELECT*” la cual le prosigue un conjunto de atributos perteneciente a la o las tablas de las cuales se desea obtener los datos deseados, en seguida se utiliza “*FROM*” y por último el nombre de la tabla junto con un punto y coma para finalizar. Todo esto se muestra en la siguiente imagen:

```
MariaDB [siprac]> SELECT alumno_id, nombre, edad FROM alumnos_siprac
-> ;
+-----+-----+-----+
| alumno_id | nombre | edad |
+-----+-----+-----+
|          1 | Gabriel |    25 |
+-----+-----+-----+
1 row in set (0.078 sec)
```

### 3.12 Consulta para leer los datos en la tabla alumnos\_siprac

#### 3.3.1.3 Modificar datos

Para modificar los datos el motor *MySQL* proporciona un comando que empieza con “*UPDATE*” después el nombre de la tabla donde se encuentran las columnas a modificar su valor, luego tenemos otra palabra reservada “*SET*”, junto a ella un conjunto de nombre de los atributos con un símbolo igual acompañado del valor deseado en cada uno, separados por una coma y por último un punto y coma. Al igual que los comandos anteriormente explicados este tiene varias formas para poder modificar los datos en una tabla. El siguiente ejemplo muestra un ejemplo:

```
+-----+-----+-----+
| alumno_id | nombre | edad |
+-----+-----+-----+
|          1 | Gabriel | 25 |
+-----+-----+-----+
1 row in set (0.078 sec)

MariaDB [siprac]> update alumnos_siprac set nombre="Gabriel", edad=34;
Query OK, 1 row affected (0.104 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [siprac]> select * from alumnos_siprac;
+-----+-----+-----+
| alumno_id | nombre | edad |
+-----+-----+-----+
|          1 | Gabriel | 34 |
+-----+-----+-----+
1 row in set (0.000 sec)
```

### 3.13 SQL para actualizar los datos de un alumno

#### 3.3.2 Data Definition Language (DDL)

En un sistema de gestión de bases de datos relacional se puede tener varias bases de datos que se desean y dentro de cada una se debe de crear los objetos que almacenarán los datos en este caso las tablas por ello en este apartado se mostrarán las sentencias que facilitan la creación de una base de datos y algunas tablas, pero todo esto es fácilmente creado a partir del software llamado “MySQL Workbench”.

##### 3.3.2.1 Crear una base de datos en MySQL

La creación de una base de datos nos permite almacenar datos que están interrelacionados utilizando las tablas, la siguiente sentencia crea una base de datos llamada siprac2:

```
Setting environment for using XAMPP for Windows.
Gabriel@DESKTOP-0N5JQRO j:\xampp2
# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.38-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database siprac2;
Query OK, 1 row affected (0.01 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| photo_app |
| phpmyadmin |
| siprac |
| siprac2 |
| test |
+-----+
8 rows in set (0.01 sec)

MariaDB [(none)]>
```

**Figura 3.14 Creación de una base de datos**

En esta base de datos crearemos una tabla que almacenara la información pertinente a los estudiantes:

```
MariaDB [siprac]> use siprac2;
Database changed
MariaDB [siprac2]> CREATE TABLE IF NOT EXISTS alumnos_siprac (
  ->   No_cuenta BIGINT(20) NOT NULL,
  ->   Nom_alu VARCHAR(60) NOT NULL,
  ->   Ap_paterno VARCHAR(20) NOT NULL,
  ->   Ap_materno VARCHAR(20) NOT NULL,
  ->   Email VARCHAR(100) NOT NULL,
  ->   Direccion VARCHAR(100) NULL DEFAULT NULL,
  ->   Telefono VARCHAR(10) NULL DEFAULT NULL,
  ->   NASS VARCHAR(100) NULL DEFAULT NULL,
  ->   Beneficiario VARCHAR(100) NULL DEFAULT NULL,
  ->   Fecha_nac DATE NULL DEFAULT NULL,
  ->   tipo INT(1) NOT NULL,
  ->   NCB VARCHAR(120) NULL DEFAULT NULL,
  ->   PRIMARY KEY (No_cuenta));
Query OK, 0 rows affected (0.26 sec)

MariaDB [siprac2]>
```

**Figura 3.15 Creación de la tabla para almacenar los datos de los alumnos**

De esta manera se crea los objetos (tablas) en el sistema de gestión de bases de datos relacionales, pero una forma eficiente es a través del software “*MySQL Workbench*” que nos permite a través del diagrama entidad relación generar un archivo sql que contiene todas las instrucciones para crear las tablas de la base de datos junto con sus constraints (*Foreign keys*), la siguiente figura muestra cómo se carga en la base de datos el archivo:

```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p siprac2 < C:\Users\Gabriel\Desktop\baseSiprac.sql
Enter password: *****
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)

C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p siprac2 < C:\Users\Gabriel\Desktop\baseSiprac.sql
Enter password: *****

C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p siprac2
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 8.0.23 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show tables;
+-----+
| Tables_in_siprac2 |
+-----+
| asistentes        |
| alumnos_siprac   |
| contra_ad        |
| contraseñas      |
| grupos_siprac    |
| inscritas        |
| materias_siprac  |
| practicas        |
| profesores_siprac|
+-----+
9 rows in set (0.01 sec)

mysql>
```

**Figura 3.16 Creación de las tablas en la base de datos siprac2**

### 3.3.3 Control de concurrencia en la base de datos *MySQL*

En el sistema, los alumnos se inscriben mínimo una práctica hasta cinco prácticas, por lo que el número de asientos en cada una es limitado, por ello cuando dos o más alumnos quieren ir a la misma práctica, pero esta ya no tiene más que un lugar disponible, entonces se hace el control de concurrencia en la base de datos, al aplicar un nivel de aislamiento llamado “Lectura confirmada” o “*Read-Committed*” y utilizando en una consulta un *FOR UPDATE*, se logra que solo haya un estudiante ganador del último lugar en la práctica.

El nivel de aislamiento de Lectura confirmada resuelve el problema de Lecturas sucias, por ejemplo, si no aplicáramos este nivel sucedería lo siguiente en el sistema a la hora de que dos estudiantes quisieran inscribirse en una práctica que tiene un solo cupo:

**Tabla 3.1 Análisis para saber el cupo en una práctica con dos transacciones concurrentes con lecturas no confirmadas.**

Tx A	Tx B	
Select cupo into readcupo from practicas where folio = 1;		La transacción A lee si todavía hay cupo en la práctica con folio 1, en dado caso prosigue a guardar el cambio.
	Select cupo from practicas where folio = 1;	La transacción B lee si todavía hay cupo en la práctica con folio 1, por lo cual guarda el cupo en la variable readcupo y prosigue a actualizar.
Update practicas set cupo = readcupo -1 where folio = 1 and cupo>0;		La transacción A actualiza el cupo de la práctica con folio 1, quedando el cupo en ceros.
	Update practicas set cupo = cupo -1 where folio = 1 and cupo>0;	La transacción B no actualiza porque el cupo de la práctica no es mayor a uno es cero, por lo tanto, termina, esto pasa porque la Tx B puede leer los cambios de Tx A a pesar de no haber hecho commit o rollback; Por lo que la Tx B debió de obtener el lugar y no avisarle al alumno que ya no hay lugares.
Rollback;		
	Commit;	

Para resolver este problema se utilizó el bloqueo de registros desde la consulta, entonces el ejemplo anterior y su problema que conlleva se resuelve agregando un *for update* en la consulta, esto permite que la práctica (o registro) involucrada en la consulta se bloquee hasta que la transacción A de *commit* o *rollback*, además las lecturas y escrituras de las demás transacciones, solamente en esta práctica, se pondrán en modo de espera.

Por lo tanto, se debe activar *Read-Committed* en *MySQL*, el cual se hace con el comando *-transaction-isolation = READ-COMMITTED*, el ejemplo anterior quedaría de la siguiente manera:

**Tabla 3.2 Análisis para saber el cupo en una práctica con dos transacciones concurrentes con lecturas confirmadas.**

Tx A	Tx B	
Select cupo from practicas where folio = 1 for update;		La transacción A lee si todavía hay cupo en la práctica con folio 1, se bloquea ese registro hasta que Tx A haga commit o rollback;
	Select cupo from practicas where folio = 1 for update;	La transacción B trata de leer si todavía hay cupo en la práctica con folio 1, pero se queda en modo de espera hasta que la Tx A de commit o rollback, cuando la Tx A termine, la Tx B seguirá con la consulta.
Update practicas set cupo = <u>cupo</u> -1 where folio = 1 and cupo>0;		La transacción A actualiza el cupo de la práctica con folio 1.
	Update practicas set cupo = cupo -1 where folio = 1 and cupo>0;	La transacción B actualiza porque la TxA hizo rollback, por lo tanto, se queda con el último cupo de la práctica, pero esto sucede hasta que la Tx A dio rollback y la Tx B continuo a partir de la consulta y así no perdería la Tx B el lugar.
Rollback;		
	Commit;	

## Capítulo 4. Implementación

En la implementación del sistema se utilizó el patrón de desarrollo Modelo-Vista-Controlador (MVC) que permitió separar la lógica del sistema, el procesamiento de los datos y lo visual para facilitar así el desarrollo del sistema y tener un mayor orden. Este patrón no se escogió, simplemente es el que implementa *CodeIgniter*.

Cada sigla significa

Modelo: Es la parte del código que se encarga de ingresar, eliminar, actualizar o leer datos de la base de datos, es decir, las consultas que se requieren.

Vista: Son los archivos que combinan *HTML5*, *CSS3* y *PHP* para poder entregarle al usuario cada una de las páginas.

Controlador: Es el archivo que lleva la lógica del sistema y las reglas de negocio para que el sistema funcione como se desee.

### 4.1 Front

Para diseñar el front del sistema, es decir, lo que ve el usuario en su pantalla, se debería de empezar por el color del texto, su tamaño, también se debería de elegir el tamaño de los botones al igual que los iconos que se muestren; Esto comúnmente se realizaría con una propuesta hecha en *CSS3* y *HTML5*, pero *Materialize* ya tiene por defecto diseños ya establecidos, como, por ejemplo:

Las tablas tienen varios estilos ya establecidos

#### Un *highlight table*

Name	Item Name	Item Price
Alvin	Eclair	\$0.87
Alan	Jellybean	\$3.76
Jonathan	Lollipop	\$7.00
Shannon	KitKat	\$9.99

Figura 4.1 tabla con diseño *highlight* de *Materialize*

## Ó un *Striped table*

Name	Item Name	Item Price
Alvin	Eclair	\$0.87
Alan	Jellybean	\$3.76
Jonathan	Lollipop	\$7.00
Shannon	KitKat	\$9.99

**Figura 4.2** tabla con diseño *striped* de *Materialize*

Tales estilos se utilizan donde sea conveniente, pero también se tiene lo que se llaman modales, que son cuadros que se ponen como otra capa en la página que se está visualizando y muestra información, formularios o lo que se desee. En este sistema, principalmente se utilizó para mostrar los formularios de inicio de sesión y de registro, cómo se muestra en la siguiente imagen:



**Figura 4.3** Formulario de inicio de sesión

Y así sucesivamente podremos obtener elementos que se necesite con un diseño ya predeterminado, pero lo que se cambió fue el color principal que viene de ser un naranja a un azul oscuro. El color que utiliza *Materialize* por defecto es #ee6e73.

Pero en nuestro caso se utilizó el color azul fuerte para ser exactos #0e0e0e que es el color en hexadecimal.

Para elegir el lugar de cada elemento en las páginas del sitio web, se realizaron propuestas, sin funcionalidad alguna, cuando estas propuestas fueron aprobadas se comenzó con la programación de los formularios para enviar los datos, la recuperación de los datos de las prácticas de campo, entre otras funcionalidades que fueron necesarias.

Para crear las vistas que se entregaran al usuario final, se debe de crear un archivo en la carpeta *application/views/* la cual, nos permitirá el ingreso de *HTML5* y *CSS3* combinado con *PHP*, esto con el fin de entregar el diseño y la información. Un ejemplo de una página sería la principal, simplemente se enseñará parte de ella:

```
<!DOCTYPE html>
<html>
<head>
  <title>SIPRAC</title>
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

  <link rel="stylesheet" type="text/css" href="<?=CSS?>materialize.min.css">
  <link rel="stylesheet" type="text/css" href="<?=CSS?>index.css">
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
</head>
<body>
  <nav>
    <div class="nav-wrapper">
      <div class="container">
        <div class="row">
          <a href="#" class="brand-logo" style="font-size: 17px">Sistema de Inscripción a Prácticas de Campo</a>
          <a href="#" data-target="mobile-demo" class="sidenav-trigger"><i class="material-icons">menu</i></a>
          <ul class="right hide-on-med-and-down">
            <li><a class="modal-trigger" href="#modal1">Iniciar sesión</a></li>
            <li><a class="modal-trigger" href="#modal4">Crear cuenta</a></li>
          </ul>
        </div>
      </div>
    </div>
  </nav>
```

La parte que se le muestra al usuario final con el código de arriba es:

**Figura 4.4 Header**

Esta parte simplemente carga *Materialize* y el *Header* del sistema, pero el siguiente código utiliza la variable de *PHP*, llamada *materias* que con un *foreach* se imprime el nombre de cada materia con las prácticas relacionadas y además se puede agregar *HTML5* con clases para agregarle diseño:

```
foreach ($materias as $materia) {
    if($materia->Clave_mat == $i){
        echo '
            <li>
                <div class="collapsible-header">
                    <i class="material-icons">subject
                </i>
                    '.$materia->Nombre.'
                </div>
                <div class="collapsible-body">
                    <table class="muestralo">
                        <thead>
                            <tr>
                                <th>#</th>
                                <th>FECHA DE INSCRIPCIÓN</th>
                                <th>FECHA PARA FIRMAR</th>
                                <th>FECHA (SALIDA)</th>
                                <th>FECHA (LLEGADA)</th>
                                <th>GRUPO</th>
                                <th>PROFESOR</th>
                                <th>VISITA A</th>
                                <th>LUGAR DE SALIDA</th>
                                <th>LUGAR DE LLEGADA</th>
                                <th>CUPO</th>
                                <th>Cancelada</th>
                            </tr>
                        </thead>
                        <tbody>
                            ;
                        </tbody>
                    </table>
                </div>
            </li>
        ';
    }
    if($materia->Clave_mat == $i){
        echo '<tr>
            <td>'.$materia -> Folio .'</td>
            <td>'.$materia -> Fins.'</td>
            <td>'.$materia -> Ffirma.'</td>
            <td>'.$materia -> Fecha_out.'</td>
            <td>'.$materia -> Fecha_in.'</td>
            <td>'.$materia -> Grupo.'</td>
            <td>'.$materia -> Profesor.'</td>
        </tr>';
    }
}
```

```

        <td>'.$materia -> Visita_A.'</td>
        <td>'.$materia -> Lugar_out.'</td>
        <td>'.$materia -> Lugar_in.'</td>
        <td>
            '$materia -> Cupo.'</td>
        <td>
            '$materia -> CANCELADA.'
        </td>
    </tr>';
}

```

Este es parte del código para mostrar las prácticas por materia, se utiliza un elemento de *Materialize* para exponer el nombre de la materia y cuando alguna de ellas sea seleccionada, se muestran las prácticas que están relacionadas. La siguiente imagen muestra lo que visualiza el usuario final:

#	FECHA DE INSCRIPCIÓN	FECHA PARA FIRMAR	FECHA (SALIDA)	FECHA (LLEGADA)	GRUPO	PROFESOR	VISITA A	LUGAR DE SALIDA	LUGAR DE LLEGADA	CUPO	Cancelada
2	2020-12-05	2020-12-05	2020-02-19 07:00:00	2020-02-19 15:00:00	4	DR. HUMBERTO MARENGO MOGOLLON	LABORATORIO DE COMISION FEDERAL DE ELECTRICIDAD EN CUERNAWACA	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	15	SI
7	2020-12-20	2020-12-27	2020-12-29 06:00:00	2020-12-29 19:00:00	4	DR. HUMBERTO MARENGO MOGOLLON	CENTRAL HIDROELÉCTRICA DE ZIMAPÁN, HIDALGO	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	18	NO
9	2020-12-05	2020-12-05	2020-04-03 06:00:00	2020-04-06 21:00:00	4	ING. VICTOR MANUEL VALDEZ RUBIO	FABRICACIÓN DE PLATAFORMAS PETROLERAS EN TAMPICO, TAMPS.	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	15	SI
11	2020-12-05	2020-12-05	2020-04-10 06:00:00	2020-04-13 20:00:00		DR. HUMBERTO MARENGO MOGOLLON	CENTRALES HIDROELÉCTRICAS CAJON Y LA YESCA	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	18	SI

**Figura 4.5 Parte de la página de bienvenida para mostrar las prácticas disponibles en el semestre corriente.**

**4.2 Backend**

El *backend* es el código que lleva la lógica del sistema que permite responder al usuario final cuando solicite una página web, cuando se desea almacenar la información que se ingresó en un formulario, verificar que el alumno al inscribirse a una práctica de campo no esté inscrito en más de cinco prácticas entre otras funciones. Es lo que permite que el sitio web sea dinámico, es decir, si se tiene una sección donde se muestran las prácticas a las que se inscribió el alumno, esta sección mostrará diferentes prácticas dependiendo del alumno que haya iniciado sesión.

El siguiente es un ejemplo:

La página de inicio

El usuario al abrir su navegador e ingresar a la página del sitio web <https://siprac.test> lo primero que se le mostrará es la siguiente página:

Sistema de Inscripción a Prácticas de Campo Iniciar sesión / Crear cuenta

Bienvenido al SIPRAC Como parte complementaria de las asignaturas teóricas, el departamento de Ingeniería en Sistemas, Planeación y Transporte de la DICyS pone a tu disposición el catálogo de prácticas de campo correspondientes al semestre 2021-2. Te invitamos a que revises y te inscribas a las prácticas que te correspondan. Solo necesitas registrarte.

**Prácticas correspondientes al semestre 2021-2**

#	FECHA DE INSCRIPCIÓN	FECHA PARA FIRMAR	FECHA (SALIDA)	FECHA (LLEGADA)	GRUPO	PROFESOR	VISITA A	LUGAR DE SALIDA	LUGAR DE LLEGADA	CUPO	Cancelada
2	2020-12-05	2020-12-05	2020-02-19 07:00:00	2020-02-19 15:00:00	4	DR. HUMBERTO MARENGO MOGOLLON	LABORATORIO DE COMISION FEDERAL DE ELECTRICIDAD EN CUERNAVACA	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	15	SI
7	2020-12-20	2020-12-27	2020-12-29 06:00:00	2020-12-29 19:00:00	4	DR. HUMBERTO MARENGO MOGOLLON	CENTRAL HIDROELECTRICA DE ZIMAPAÁN, HIDALGO	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	18	NO
9	2020-12-05	2020-12-05	2020-04-03 06:00:00	2020-04-06 21:00:00	4	ING. VICTOR MANUEL VALDEZ RUBIO	FABRICACIÓN DE PLATAFORMAS PETROLERAS EN TAMPICO, TAMPIS.	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	15	SI
11	2020-12-05	2020-12-05	2020-04-10 06:00:00	2020-04-13 20:00:00		DR. HUMBERTO MARENGO MOGOLLON	CENTRALES HIDROELECTRICAS CAJON Y LA YESCA	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	18	SI

Universidad Nacional Autónoma de México, Facultad de Ingeniería  
División de Ingenierías Civil y Geomática @ 2021  
Ciudad Universitaria, México CDMX, Av. Universidad 3000.

Redes sociales

Copyright © 2021 Todos los derechos reservados.

**Figura 4.6** Página de bienvenida

Para ello el *framework* busca en el archivo `application/config/routes.php` la ruta por *default* para contestar a la petición del usuario, en este caso sería:

```
$route['default_controller'] = 'HomeController';
```

Tal ruta tiene definido un controlador llamado *HomeController* que en su interior tiene una función llamada *index*, el controlador está definido en el archivo `HomeController.php`, y *CodeIgniter* se encargará de ejecutar el código que está en la función, por motivos de seguridad no se enseñará todo el código en este archivo:

```
<?php
class HomeController extends CI_Controller
{
    public function index()
    {
        $this->load->model('Practicas_Model');
        $materias = $this->Practicas_Model->materias();
        $data['materias'] = $materias;
        $this->load->view('index',$data);
    }
}
```

En esta función del controlador se tiene la recuperación de las materias con sus prácticas y así mostrarlas correctamente. Primero se carga el modelo que contiene la función para recuperar las prácticas. Después se ejecuta la función llamada materias, esta regresa las materias con sus respectivas prácticas y se almacenan en una variable llamada \$materias.

Para que se puedan pasar variables a las vistas, se debe de hacer por medio de un arreglo asociativo en donde cada elemento tiene un nombre, este lo utiliza *CodeIgniter* para crear una variable de cada elemento y tener acceso a la información.

Por ejemplo:

Si \$data tiene los elementos \$data['materias'] y \$data['alumnos'], entonces en la vista que se cargue, se tendrá acceso a dos variables \$materias y \$alumnos.

Finalmente se carga la vista que vera el usuario final con los datos recuperados de la base de datos.

El modelo en este caso tiene la función materias que se utiliza para recuperar las materias con las practicas que tienen, el siguiente código muestra la función:

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Practicas_Model extends CI_Model
{
    function materias(){
        $query = "select * from materias_siprac ms, practicas p where ms.Clave_mat
                = p.Clave_mat order by p.Clave_mat, p.Folio";
        return $this->db->query($query)->result();
    }
}
```

Lo que hace es almacenar la consulta en una variable y luego hace que se ejecute en la base de datos y el resultado lo regresa al controlador que ejecuto la función.

Por último, la vista en este caso está en el archivo **application/views/index.php** hace un foreach en la variable \$materias, en este caso el arreglo asociativo que se envió a esta vista en el controlador tiene un elemento llamado \$data['materias'] que internamente CodeIgniter toma y en la vista simplemente se tiene que escribir \$materias para tener acceso a su valor. La vista tiene el siguiente código para imprimir los nombres de las materias y las prácticas relacionadas:

```

foreach ($materias as $materia) {
    if($materia->Clave_mat == $i){
        echo '
            <li>
                <div class="collapsible-header">
                    <i class="material-icons">subject
                    </i>
                    '.$materia->Nombre.'
                </div>
                <div class="collapsible-body">
                    <table class="muestralo">
                        <thead>
                            <tr>
                                <th>#</th>
                                <th>FECHA DE INSCRIPCIÓN</th>
                                <th>FECHA PARA FIRMAR</th>
                                <th>FECHA (SALIDA)</th>
                                <th>FECHA (LLEGADA)</th>
                                <th>GRUPO</th>
                                <th>PROFESOR</th>
                                <th>VISITA A</th>
                                <th>LUGAR DE SALIDA</th>
                                <th>LUGAR DE LLEGADA</th>
                                <th>CUPO</th>
                                <th>Cancelada</th>
                            </tr>
                        </thead>
                        <tbody>
                            ;
                        </tbody>
                    </table>
                </div>
            </li>
        '
    }
    if($materia->Clave_mat == $i){
        echo '<tr>
            <td>'.$materia -> Folio .'</td>
            <td>'.$materia -> Fins.'</td>
            <td>'.$materia -> Ffirma.'</td>
            <td>'.$materia -> Fecha_out.'</td>
            <td>'.$materia -> Fecha_in.'</td>
            <td>'.$materia -> Grupo.'</td>
            <td>'.$materia -> Profesor.'</td>
            <td>'.$materia -> Visita_A.'</td>
            <td>'.$materia -> Lugar_out.'</td>
            <td>'.$materia -> Lugar_in.'</td>
            <td>
                '.$materia -> Cupo.'</td>
        '
    }
}

```

```
        <td>
            '.$materia -> CANCELADA.'
        </td>
    </tr>';
}
```

No se muestra todo el código porque sería demasiadas líneas.

Una vez terminado de generarse la vista se envía al usuario final para que pudiese ver la página de inicio del sitio web.

Así es cómo se trabaja con el Modelo-Vista-Controlador, lo que se considera *backend* aquí es la parte del modelo y el controlador.

#### 4.4 Seguridad del sistema

Los atacantes van a buscar varias formas de ingresar al sistema para obtener la información de los alumnos, o hacer la maldad de borrar o modificar la información en la base de datos y de interceptar la información del usuario al enviar algún formulario. Para esto se hablará de *SQL injection*, Certificados de seguridad y Cifrado en las conexiones.

##### 4.4.1 SQL Injection

El *SQL Injection* es tratar de cambiar los *queries* de un sistema para modificar, eliminar u obtener más información de la que se proporciona, esto se logra al ingresar en los formularios, datos que modifiquen los *queries*, por ejemplo, en la mayoría de los sistemas se utiliza el método de ingresar un usuario y la contraseña para poder ingresar al sistema, pero si el atacante ingresa por ejemplo en este sistema la siguiente contraseña **hola' or 1=1 limit 1; #**, entonces el atacante va a poder ingresar sin ningún problema como el primer usuario registrado en el sistema.



**Figura 4.7 Parte de la página del alumno, sección inicio**

Esto sucediera si no se tuvieran filtros que revisen que exista la contraseña, que esta fuese máximo de once caracteres y además que fuese alfanumérica, es decir, solo acepta contraseñas que contengan las letras del alfabeto en mayúsculas o minúsculas y sin acento o números, por lo tanto el atacante no podrá ingresar una comilla sencilla('), un

símbolo de igual (=), un punto y coma (;) o el símbolo del gato (#) que se utiliza para poner comentarios y así evitar que se ejecute lo que este a la derecha del *query*.

En el caso del usuario, es obligatorio que tenga nueve caracteres de largo, que solo tenga números y que sea obligatorio, entonces en el formulario de iniciar sesión, no se podrá agregar nada que modifique el *query* que busca al usuario.

Estas verificaciones suceden en cada uno de los formularios para evitar la modificación de los *queries* y así no podrán ingresar al sistema o eliminar o modificar información.

Otro elemento de seguridad para evitar *SQL Injection* son los llamados *parameterized query*, en esencia lo que hace en el caso de un *select*, primero crea el *query plan* y posteriormente utiliza los valores para buscar la información y a pesar de que tenga una unión o un valor como **hola' or 1=1 limit 1; #**, el motor de la base de datos ya no le va a hacer caso al *limit*, sino que simplemente lo tomara como el valor de la contraseña y buscara un registro, en este caso sin éxito, por ejemplo:

```
$sql = "select a.Nom_alu,a.Ap_materno,a.Ap_paterno from alumnos_siprac a, contraseñas c where a.No_cuenta = c.No_cuenta and c.Contrasena = '?' and a.No_cuenta=?";  
$this->db->query($sql, array(314458996, 'hola' or 1=1 limit 1; #'));
```

En este caso a pesar de que el atacante intente ingresar al sistema utilizando *SQL Injection*, no podrá, ya que el parametrizado de la consulta no lo permitirá y por lo tanto no encontrará ningún registro. Así el sistema mostrará un mensaje de que la información ingresada es incorrecta, la siguiente imagen lo muestra:

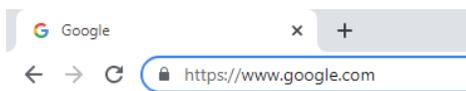


The image shows a web-based login interface. At the top, there is a dark header with the word "ACCESO" in white. Below the header, a message says "Favor de introducir tu usuario y tu contraseña para acceder al SIPRAC". There are two input fields: "Usuario" with the value "145875241" and "Contraseña" which is masked with dots. Below the fields is a blue error message bar that says "Usuario o contraseña incorrecta". At the bottom, there are four buttons: a green "ENVIAR" button with a right-pointing arrow, a green "BORRAR" button with a trash icon, a teal "RECUPERAR CONTRASEÑA" button, and a teal "CERRAR" button.

**Figura 4.8 Formulario de inicio de sesión**

#### 4.4.2 Certificados de seguridad

A la hora de entrar a un sitio web debemos cerciorarnos de que tenga el icono del candado, lo que nos permitirá saber que el sitio es quién dice ser y además que la información que enviamos a través de sus formularios estará encriptada de extremo a extremo. La siguiente imagen muestra un ejemplo:

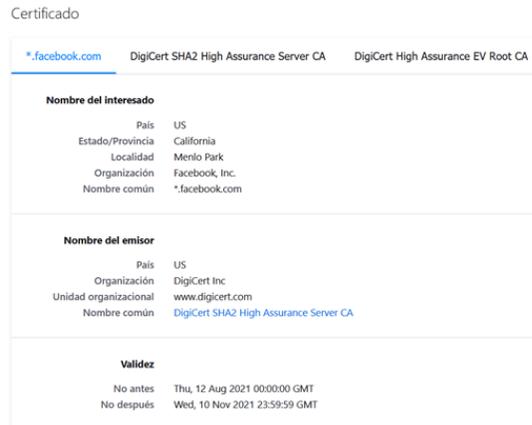


**Figura 4.9 Parte de la ventana de *Google Chrome* donde se muestra el candado de seguridad**

Esto significa que el sitio web al que se ingreso está utilizando un certificado *SSL (Secure Socket Layer)*, este permite saber si el sitio al que se está conectando es el real y no la página clonada por un hacker, que su objetivo principal es robar datos, para ello el certificado incluye el nombre del *Certificate authority (CA)* junto con otra información como los dominios disponibles, el nombre de la empresa que es dueña del sitio web, la llave publica del servidor y la llave publica del servidor encriptada por la llave publica del *CA*, entre otra información, que utiliza el navegador web para verificar el sitio web. Esta organización (*CA*) es la que emite el certificado y es lo que permite tener un internet más seguro.

No solo es necesario saber que un sitio web es el que dice ser, sino que los mensajes entre el cliente y el servidor no van a ser vistos por terceros no deseados. Esto se logra gracias a que el servidor envía la primera vez su llave publica junto con su certificado, después el navegador verificará que sea válido el certificado y si lo es generará una llave llamada *symmetric key* que será encriptada por la llave publica y desencriptada con la llave privada que el servidor tiene. Ahora toda información que se envíe del servidor al cliente y del cliente al servidor será encriptada y desencriptada con la misma llave (*symmetric key*) y así ningún tercero podrá ver en realidad la información que comparte el cliente con el servidor.

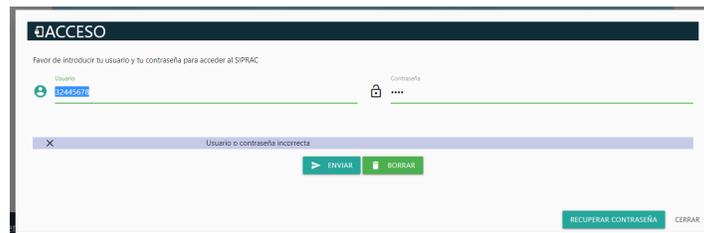
En la siguiente imagen se muestra el certificado de *Facebook* y su *CA* es *DigiCert*



**Figura 4.10 Parte del Certificado SSL de Facebook**

Para probar este punto se utilizará dos programas en *Windows* que permite tener un servidor *Apache*, uno con *http* (*XAMMP*) y otro con *https* (*Laragon*), para así con el programa *wireshark* poder analizar la red y poder obtener la información que se envía del cliente al servidor en el caso del *http* y en *https* no se podrá ver la información, ya que está encriptada.

En la siguiente imagen se muestra el formulario de inicio de sesión en el cual se ingresó el usuario y la contraseña, pero el sitio web no tiene un certificado SSL, es decir se está utilizando *http*



**Figura 4.11 Formulario de inicio de sesión**

Antes de enviar la información se inició el programa *wireshark* y se inicializo el análisis de paquetes en la red, por lo que, al enviar el usuario y la contraseña, se pudo identificar el paquete con la información:

4	0.000157	::1	::1	TCP	1460	64540 → 80	[PSH, ACK] Seq=1 Ack=1 Win=2618880 Len=1396 [TCP segment of a reassembled PDU]
5	0.000168	::1	::1	TCP	64	80 → 64540	[ACK] Seq=1 Ack=1397 Win=2618880 Len=0
6	0.000178	::1	::1	HTTP	89	POST /SIPRAC2/Login_Controller/iniSesion	HTTP/1.1 (application/x-www-form-urlencoded)
7	0.000182	::1	::1	TCP	64	80 → 64540	[ACK] Seq=1 Ack=1422 Win=2618880 Len=0

**Figura 4.12 Parte del programa *wireshark* donde se resalta el paquete que contiene la información que envió el usuario.**

En esta parte del programa podemos ver que una de las solicitudes es para iniciar sesión, pero si le damos clic se puede ver la información del paquete, en la cual incluye la contraseña del alumno

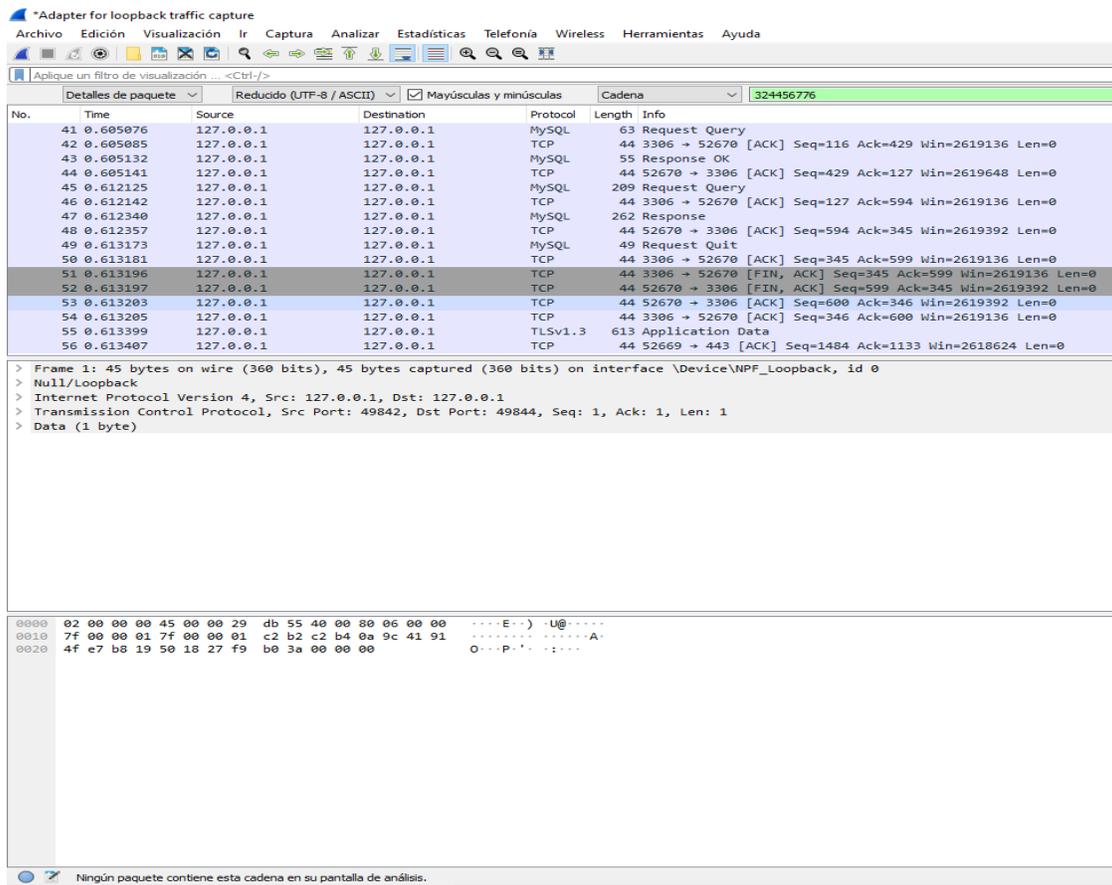
```

> Frame 6: 89 bytes on wire (712 bits), 89 bytes captured (712 bits) on interface \Device\NPF_{Loopback}, id 0
> Null/Loopback
> Internet Protocol Version 6, Src: ::1, Dst: ::1
> Transmission Control Protocol, Src Port: 64540, Dst Port: 80, Seq: 1397, Ack: 1, Len: 25
> [2 Reassembled TCP Segments (1421 bytes): #4(1396), #6(25)]
> Hypertext Transfer Protocol
  HTML Form URL Encoded: application/x-www-form-urlencoded
    Form item: "iuser" = "32445678"
      Key: iuser
      Value: 32445678
    Form item: "ipass" = "hola"

```

**Figura 4.13** Parte del programa *wireshark* para mostrar la información del paquete seleccionado

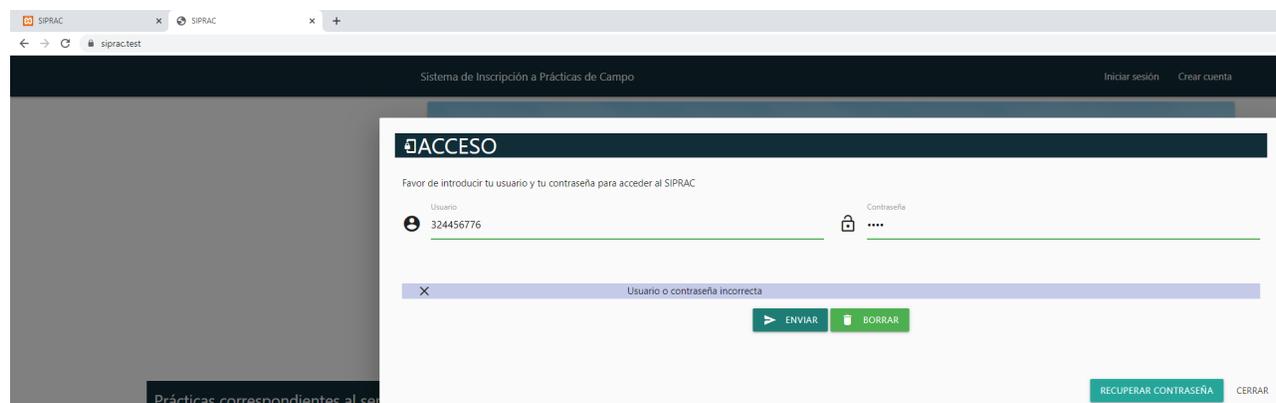
Por lo tanto, no es recomendable utilizar páginas http donde se ingresa información sensible, ya que un tercero podría interceptar el paquete y ver la información que la persona envía al servidor y ahora se hace el mismo ejercicio con el sitio web utilizando *https*, es decir, tiene certificado *SSL*, entonces no se podrá ver la información que se envía.



**Figura 4.14** Programa *wireshark*

En este punto podemos ver que a pesar de buscar en los paquetes el número de cuenta del usuario, ya no se encuentra nada, principalmente porque todo se encuentra encriptado,

la siguiente imagen muestra el candado en la *URL* que nos indica que el sitio web está utilizando *https*.



**Figura 4.15** Parte de la página de bienvenida utilizando *https*

#### 4.4.2 Seguridad en las contraseñas

Para el sistema se decidió que el usuario no eligiese la contraseña, ya que la mayoría de las veces, se escoge una muy débil y fácil de recordar, por lo que, el alumno al registrarse hará que el sistema genere una contraseña aleatoria y además encriptada con ayuda de la función `password_hash()`. Esta función regresa la contraseña encriptada y que permite generar una salt de manera automática y fuerte.

La salt previene que un atacante tenga una lista de hashes de passwords y pueda saber con antelación qué contraseña está guardada en la base de datos, si es que tiene acceso a ella. Por lo que se indicó, la salt previene que a pesar de que se utilice una y otra vez la misma contraseña, la cadena de caracteres que se genera no es la misma, por ejemplo:

```
>>> echo password_hash("123", PASSWORD_DEFAULT);
$2y$10$56XCxie/BZsDSCo5xe3g3OyYd622NDHXC011TPdYaTmw8Mc06LU0.↵
>>> echo password_hash("123", PASSWORD_DEFAULT);
$2y$10$uNJth0Oo6rUOYL90IIA7GeDVaSpc9c6ZbqYOHMUDTjzuhG.vbnyoS↵
```

Con esto vemos que, a pesar de utilizar la misma contraseña débil, se generan dos hashes de password diferentes y así el atacante no podrá buscar en una lista el hash de la contraseña 123.

Pero en este caso el atacante lo tiene mucho más difícil, ya que se le genera a cada usuario una contraseña aleatoria y además se encripta, entonces es muy difícil saber la contraseña sin encriptar.

# Capítulo 5. Experiencia de usuario y pruebas del sistema

En este capítulo se mostrarán imágenes del sistema implementado en los servidores de producción y además una explicación de las pruebas que se le hizo a algunas rutas del sistema para verificar su correcto funcionamiento, es decir, que la ruta regrese un *JSON* de acuerdo a los datos ingresados en un formulario.

## Página de bienvenida

La página de bienvenida se muestra en la siguiente imagen mostrando las prácticas de campo disponibles durante el semestre además del *header*, el *footer* y el *Hero*:

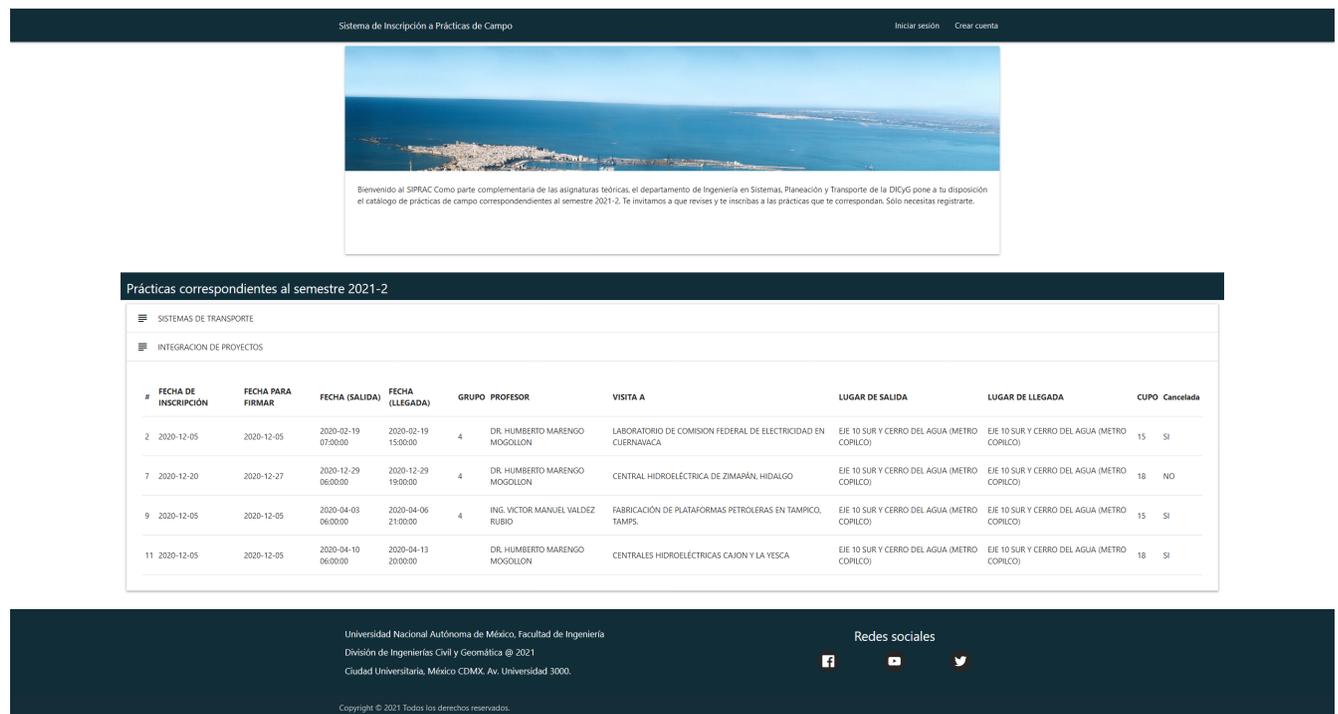


Figura 5.1 Página de bienvenida

En esta página se tiene dos enlaces que abren dos *“modals”* que permiten al alumno visualizar un formulario, el primero es para iniciar sesión:

ACCESO

Favor de introducir tu usuario y tu contraseña para acceder al SIPRAC

Usuario

Contraseña

ENVIAR BORRAR RECUPERAR CONTRASEÑA CERRAR

Prácticas correspondientes al semestre 2021-2

#	FECHA DE INSCRIPCIÓN	FECHA PARA FIRMAR	FECHA (SALIDA)	FECHA (LLEGADA)	GRUPO	PROFESOR	VISITA A	LUGAR DE SALIDA	LUGAR DE LLEGADA	CUPO	Cancelada
2	2020-12-05	2020-12-05	2020-02-19 07:00:00	2020-02-19 15:00:00	4	DR. HUMBERTO MARENGO MOGOLLÓN	LABORATORIO DE COMISION FEDERAL DE ELECTRICIDAD EN CUERNAVACA.	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	15	SI
7	2020-12-20	2020-12-27	2020-12-29 06:00:00	2020-12-29 19:00:00	4	DR. HUMBERTO MARENGO MOGOLLÓN	CENTRAL HIDROELECTRICA DE ZIMAPAN, HIDALGO	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	18	NO
9	2020-12-05	2020-12-05	2020-04-03 06:00:00	2020-04-06 21:00:00	4	ING. VICTOR MANUEL VALDEZ RUBIO	FABRICACION DE PLATAFORMAS PETROLERAS EN TAMPICO, TAMPS.	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	15	SI
11	2020-12-05	2020-12-05	2020-04-10 06:00:00	2020-04-13 20:00:00		DR. HUMBERTO MARENGO MOGOLLÓN	CENTRALES HIDROELECTRICAS CAJON Y LA YESCA	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	18	SI

Universidad Nacional Autónoma de México, Facultad de Ingeniería  
División de Ingenierías Civil y Geomática © 2021  
Ciudad Universitaria, México CDMX, Av. Universidad 3000.

Redes sociales

Copyright © 2021 Todos los derechos reservados.

Figura 5.2 Formulario para inicio de sesión

Este formulario tiene tres botones:

1. Uno para enviar las credenciales e ingresar al sistema
2. Otro para borrar los datos ingresados en cada campo
3. Y finalmente uno para abrir otro "modal" para recuperar la contraseña el cual se muestra a continuación:

Contraseña

Número de cuenta

ENVIAR BORRAR RECUPERAR CONTRASEÑA CERRAR

Prácticas correspondientes al semestre 2021-2

#	FECHA DE INSCRIPCIÓN	FECHA PARA FIRMAR	FECHA (SALIDA)	FECHA (LLEGADA)	GRUPO	PROFESOR	VISITA A	LUGAR DE SALIDA	LUGAR DE LLEGADA	CUPO	Cancelada
2	2020-12-05	2020-12-05	2020-02-19 07:00:00	2020-02-19 15:00:00	4	DR. HUMBERTO MARENGO MOGOLLÓN	LABORATORIO DE COMISION FEDERAL DE ELECTRICIDAD EN CUERNAVACA.	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	15	SI
7	2020-12-20	2020-12-27	2020-12-29 06:00:00	2020-12-29 19:00:00	4	DR. HUMBERTO MARENGO MOGOLLÓN	CENTRAL HIDROELECTRICA DE ZIMAPAN, HIDALGO	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	18	NO
9	2020-12-05	2020-12-05	2020-04-03 06:00:00	2020-04-06 21:00:00	4	ING. VICTOR MANUEL VALDEZ RUBIO	FABRICACION DE PLATAFORMAS PETROLERAS EN TAMPICO, TAMPS.	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	15	SI
11	2020-12-05	2020-12-05	2020-04-10 06:00:00	2020-04-13 20:00:00		DR. HUMBERTO MARENGO MOGOLLÓN	CENTRALES HIDROELECTRICAS CAJON Y LA YESCA	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EJE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	18	SI

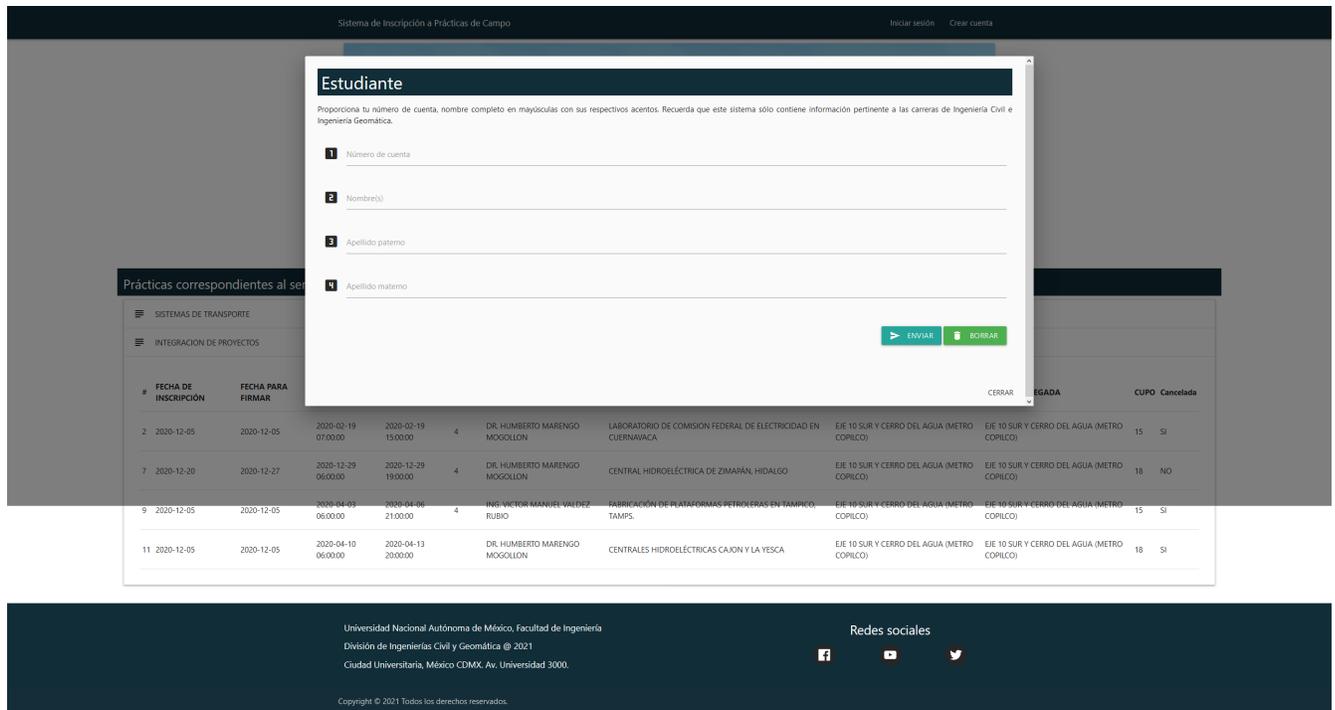
Universidad Nacional Autónoma de México, Facultad de Ingeniería  
División de Ingenierías Civil y Geomática © 2021  
Ciudad Universitaria, México CDMX, Av. Universidad 3000.

Redes sociales

Copyright © 2021 Todos los derechos reservados.

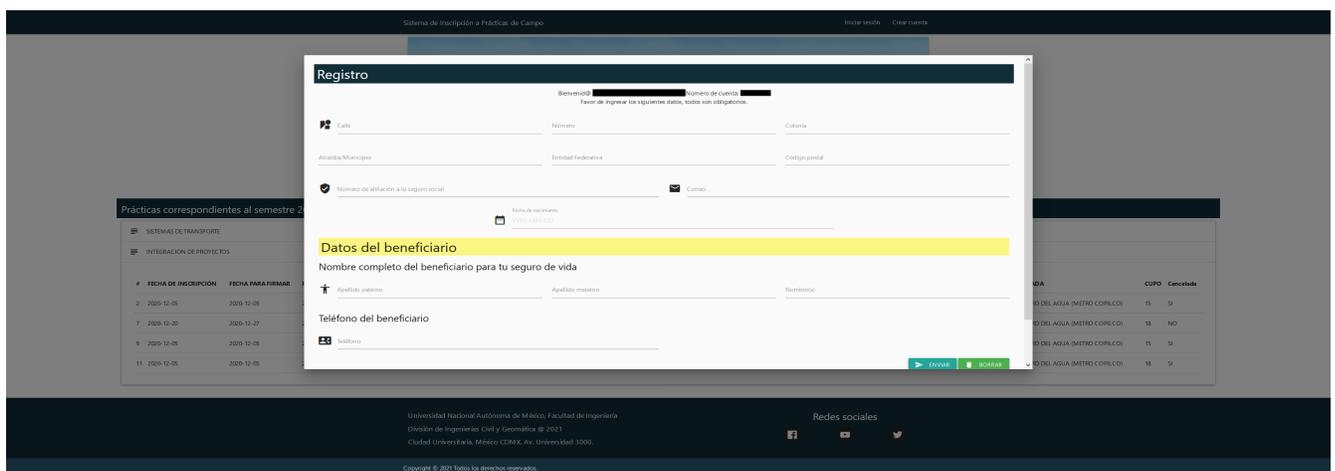
Figura 5.3 Formulario para recuperar contraseña

Finalmente, el segundo enlace en el *Header* es para mostrar un “*modal*” con un formulario para registrarse:



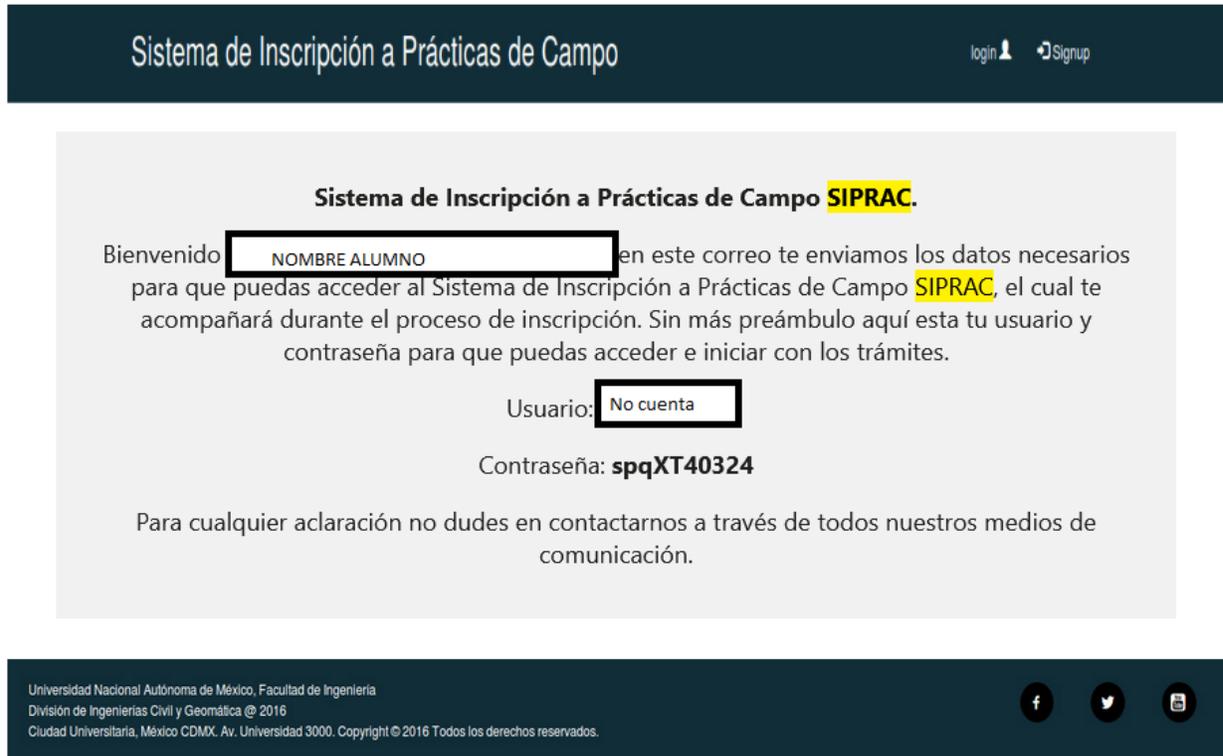
**Figura 5.4 Inicio del registro del alumno**

El alumno al enviar sus datos y éstos estar correctos se desencadena la acción de abrir un nuevo “*modal*” con un formulario que permite ingresar los datos necesarios para asistir a las prácticas de campo:



**Figura 5.5 Registro del alumno**

Posteriormente el alumno recibirá un correo con su usuario y contraseña cómo se muestra en la siguiente imagen:



[Responder](#) | [Reenviar](#)

**Figura 5.6 Correo que recibe el alumno al registrarse**

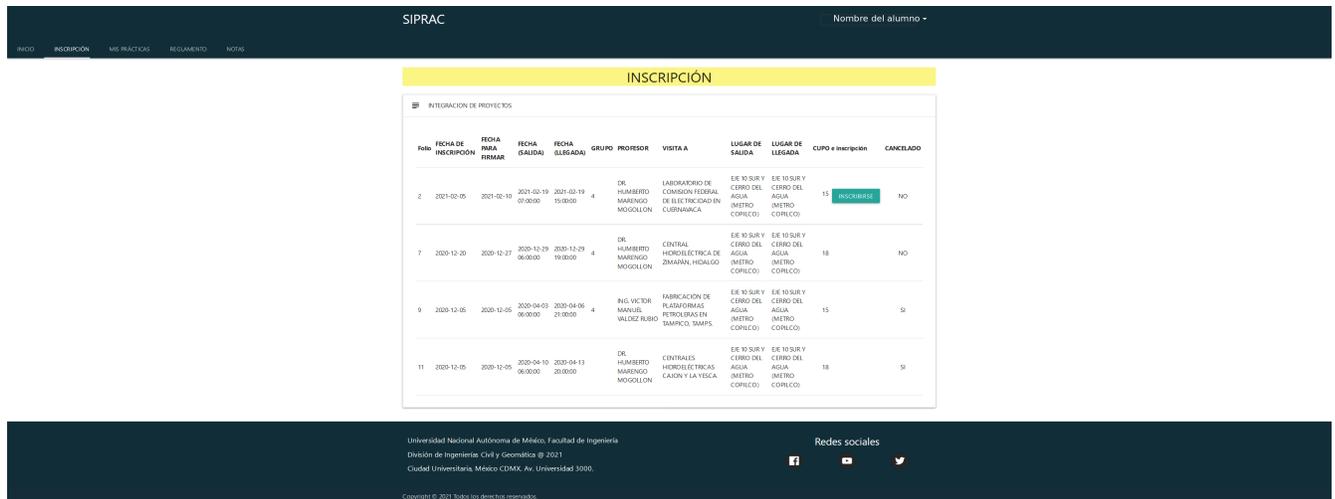
### **Página del alumno**

Por lo tanto, podrá ingresar al sistema, lo que se le mostrará es la siguiente página web:



**Figura 5.7** Página del alumno al iniciar sesión

La siguiente pestaña es “INSCRIPCIÓN”, ésta permitirá visualizar las prácticas de campo por materia:



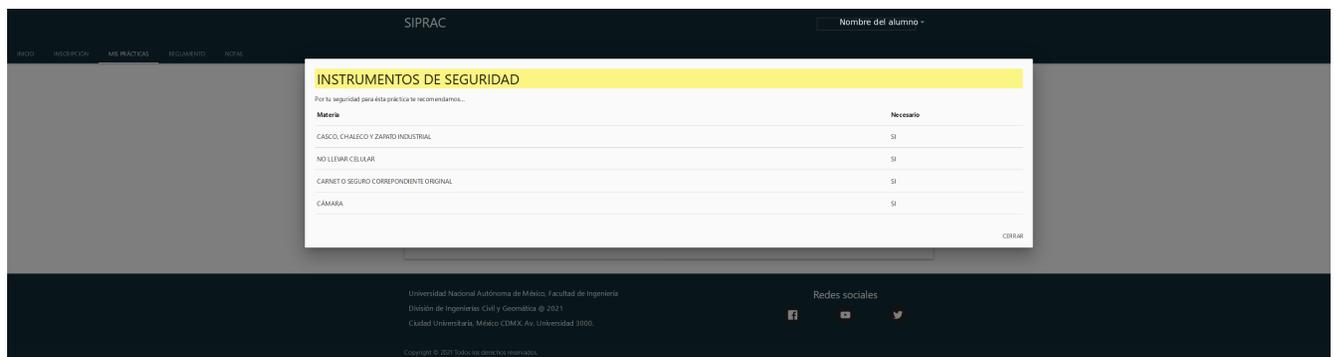
**Figura 5.8** Lista de prácticas para inscribirse

La penúltima pestaña “MIS PRÁCTICAS” muestra las prácticas por materia a las cuales está inscrito el alumno:



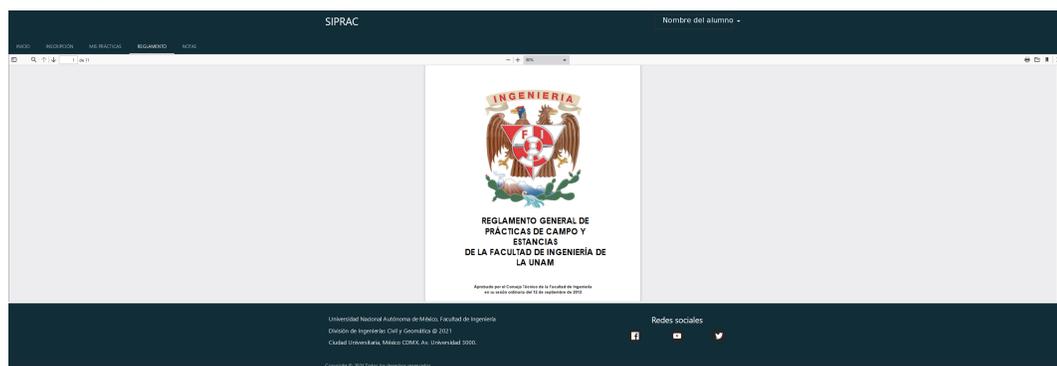
**Figura 5.9 Lista de prácticas inscritas**

También al dar clic en el botón “MÁS” se mostrarán los materiales a llevar a la práctica de campo:



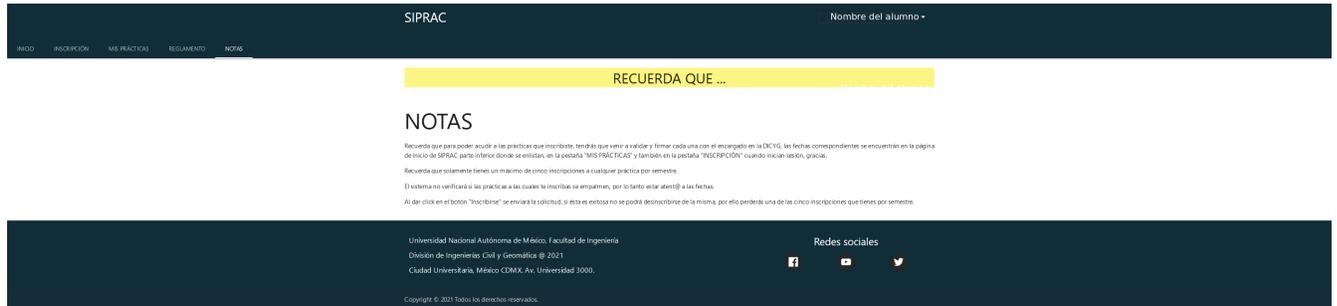
**Figura 5.10 Lista de instrumentos de seguridad a llevar a la práctica**

En penúltimo lugar tenemos el reglamento que el estudiante debe leer antes de presentarse a alguna de las prácticas:



**Figura 5.11 Reglamento general**

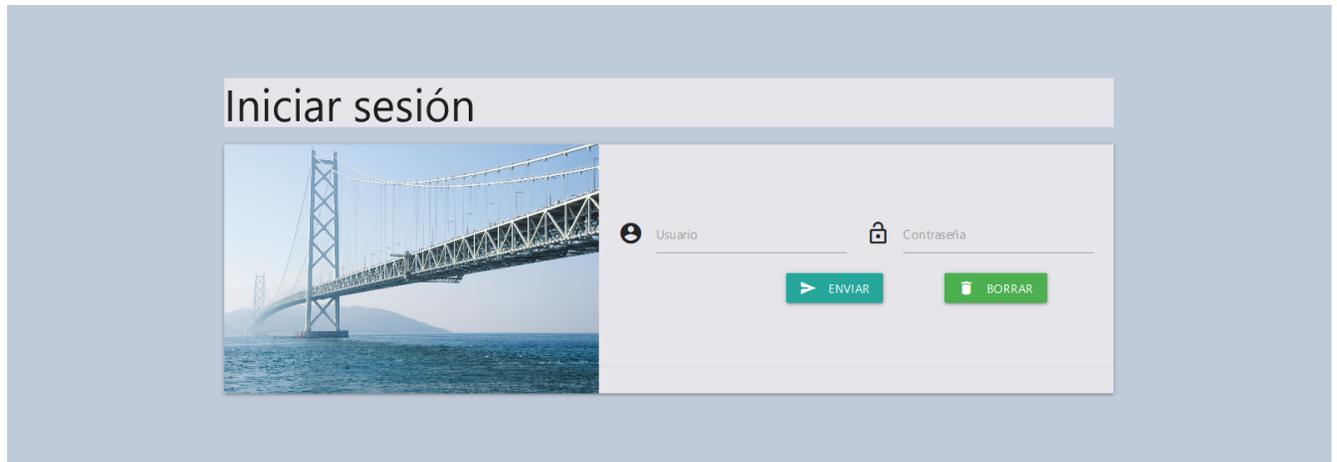
Por último, se tiene la pestaña “NOTAS”, la cual presenta información importante para que el alumno tome en cuenta al inscribirse a una práctica:



**Figura 5.12 Notas importantes que el alumno debe leer**

## Página del administrador

En el caso del administrador se tiene dos páginas web, una para inicio de sesión, la cual se muestra en la siguiente imagen:



**Figura 5.13 Formulario de inicio de sesión para el administrador.**

Al iniciar sesión el administrador visualizará una página con un menú lateral para realizar un cambio en el sistema, la siguiente imagen muestra la página web:

Validación
Constancias de Asistencia
Reglamento
Listas
Inscripción a oyentes o externos
Ver materias inscritas del alumno
Pasar lista alumnos
Generar lista de asistencia
Generar Estadísticas
Edición de prácticas
Agregar o eliminar prácticas
Agregar o eliminar alumnos

**Figura 5.14** Página del administrador al iniciar sesión

En la primera opción del menú permite visualizar las inscripciones a las prácticas y validar las inscripciones:

Sistema de Inscripción a Prácticas de Campo ING. ERNESTO RENÉ MENDOZA SÁNCHEZ ▾

Validación
Constancias de Asistencia
Reglamento
Listas
Inscripción a oyentes o externos
Ver materias inscritas del alumno
Pasar lista alumnos
Generar lista de asistencia
Generar Estadísticas
Edición de prácticas
Agregar o eliminar prácticas
Agregar o eliminar alumnos

**Validación**

☁ INTEGRACION DE PROYECTOS

Número de cuenta	Nombre	Apellidos	Clave	Folio	Validación
310270740	EMMANUEL	HERNANDEZ LOPEZ	1966	2	<a href="#" style="background-color: #008080; color: white; padding: 2px 5px; text-decoration: none;">VALIDAR</a>

**Figura 5.15** Opción validación

Sistema de Inscripción a Prácticas de Campo ING. ERNESTO RENÉ MENDOZA SÁNCHEZ ▾

Validación

- Constancias de Asistencia
- Reglamento
- Listas
- Inscripción a oyerentes o externos
- Ver materias inscritas del alumno
- Pasar lista alumnos
- Generar lista de asistencia
- Generar Estadísticas
- Edición de prácticas
- Agregar o eliminar prácticas
- Agregar o eliminar alumnos

Número de cuenta  
31 02 70 740

Constancias

[VER PRÁCTICAS INSCRITAS](#)

Vista a	Fecha (SALIDA)	Fecha (LLEGADA)	Folio	Materia	Enviar
LABORATORIO DE COMISION FEDERAL DE ELECTRICIDAD EN CUERNAVACA	2021-02-19 07:00:00	2021-02-19 15:00:00	2	INTEGRACION DE PROYECTOS	<a href="#">GENERAR PDF</a>

**Figura 5.16 Se muestra la lista de prácticas a las que el alumno asistió**

La segunda opción es para generar las constancias de asistencia:

Sistema de Inscripción a Prácticas de Campo ING. ERNESTO RENÉ MENDOZA SÁNCHEZ ▾

Validación

- Constancias de Asistencia
- Reglamento
- Listas
- Inscripción a oyerentes o externos
- Ver materias inscritas del alumno
- Pasar lista alumnos
- Generar lista de asistencia
- Generar Estadísticas
- Edición de prácticas
- Agregar o eliminar prácticas
- Agregar o eliminar alumnos

FACULTAD DE INGENIERÍA  
DIVISIÓN DE INGENIERÍA CIVIL Y GEOMÁTICA  
DEPARTAMENTO DE SISTEMAS DE SISTEMAS,  
PLANEACIÓN Y TRANSPORTE

Asunto: CONSTANCIA DE ASISTENCIA PRÁCTICAS

AL SEÑOR COMISARIO,  
PRESENTE.

Por medio del presente, se hace constar la asistencia a las prácticas que se indican.

Nombre del alumno (o): ERNESTO RENÉ MENDOZA SÁNCHEZ  
Número de cuenta: 310270740

Práctica a la que asistió:  
LABORATORIO DE COMISION FEDERAL DE ELECTRICIDAD EN CUERNAVACA

Fecha de la práctica:  
2021-02-19

Se extiende la presente constancia a petición del interesado para los fines académicos que correspondan.

Atentamente,  
POR ME RAZA HABLA EL PRESIDENTE  
Cedro Cuernavaca, C.F. No. 412 de febrero del 2021.  
EL JEFE DEL DEPARTAMENTO

Ing. Ernesto René Mendoza Sánchez

**Figura 5.17 En esta figura se muestra la constancia generada en un archivo PDF que puede ser descargado.**

La tercera opción muestra el reglamento:



**Figura 5.18 Opción ver Reglamento general**

La cuarta es la opción para generar las listas de cada práctica a entregarse a la Facultad para permitir realizar la práctica y también para hacer el pago de su seguro:



Sistema de Inscripción a Prácticas de Campo
ING. ERNESTO RENÉ MENDOZA SÁNCHEZ ▾

- Validación
- Constancias de Asistencia
- Reglamento
- Listas
- Inscripción a oyentes o externos
- Ver materias inscritas del alumno
- Pasar lista alumnos
- Generar lista de asistencia
- Generar Estadísticas
- Edición de prácticas
- Agregar o eliminar prácticas
- Agregar o eliminar alumnos

### Agregar alumno

☰ Número de cuenta

Nombre del alumno	<input type="text"/>	<input type="text"/>	<input type="text"/>
Calle	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Correo	Número de afiliación a su seguro social	Fecha de nacimiento del alumno	
Nombre(s) del beneficiario	<input type="text"/>	<input type="text"/>	<input type="text"/>
Teléfono del beneficiario	Tipo Escoge el tipo de estudiante <input type="text"/>		

▶ ENVIAR
🗑 Borrar

### Inscribir alumno a una materia

☰ Número de cuenta

☰ Materia

👤 Grupo

▶ ENVIAR
🗑 Borrar

**Figura 5.20 Formularios para inscribir a un alumno externo u oyente**

La opción “Ver materias inscritas del alumno” es para apoyar al administrador en caso de que uno de los alumnos indique que tiene problemas al visualizar las prácticas de la materia indicada, la siguiente imagen muestra un ejemplo de la lista de materias a las que está inscrito el alumno:

Sistema de Inscripción a Prácticas de Campo
ING. ERNESTO RENÉ MENDOZA SÁNCHEZ ▾

- Validación
- Constancias de Asistencia
- Reglamento
- Listas
- Inscripción a oyentes o externos
- Ver materias inscritas del alumno
- Pasar lista alumnos
- Generar lista de asistencia
- Generar Estadísticas
- Edición de prácticas
- Agregar o eliminar prácticas
- Agregar o eliminar alumnos

### Ver materias inscritas del alumno

Número de cuenta:

310270740

BUSCAR

Materias del alumno

Numero de cuenta	Materia	Grupo
310270740	1966	4

**Figura 5.21 Opción ver materias inscritas del alumno**

Pasar lista a alumnos es una opción que permite seleccionar la práctica en la cual se desea pasar asistencia:

**Figura 5.22 Seleccionar práctica para pasar lista**

Folio de la Práctica	Número de cuenta	Nombre	Apellido paterno	Apellido materno	Asistió	Faltó
2	310270740	EMMANUEL	HERNANDEZ	LOPEZ	<input type="radio"/>	<input checked="" type="radio"/>

**Figura 5.23 Pasar Lista**

“Generar lista de asistencia” permite obtener un excel para dárselo al profesor y que éste verifique la asistencia de los alumnos para asignarles el porcentaje proporcional en su calificación final:

Sistema de Inscripción a Prácticas de Campo ING. ERNESTO RENÉ MENDOZA SÁNCHEZ

**Generar listas de asistencia por grupo**

INTEGRACION DE PROYECTOS 4

**ENVIAR**

- Validación
- Constancias de Asistencia
- Reglamento
- Listas
- Inscripción a oyentes o externos
- Ver materias inscritas del alumno
- Pasar lista alumnos
- Generar lista de asistencia
- Generar Estadísticas
- Edición de prácticas
- Agregar o eliminar prácticas
- Agregar o eliminar alumnos

**Figura 5.24** Seleccionar materia y grupo para generar lista de asistencia a prácticas de campo

	A	B	C	D	E	F
1						
2			Asignatura		1966	
3			Grupo		4	
4			Profesor	MARENGO MOGOLLON HUMBERTO JUAN FRANCISCO DR.		
5						
6						
7						
8	No.	Número de cuenta	Nombre		2	7 9
9	1	310270740	HERNANDEZ LOPEZ EMMANUEL		E	
10						
11						
12			2 LABORATORIO DE COMISION FEDERAL DE ELECTRICIDAD EN CUERNAVACA			
13			7 CENTRAL HIDROELÉCTRICA DE ZIMAPÁN, HIDALGO			
14			9 FABRICACIÓN DE PLATAFORMAS PETROLERAS EN TAMPICO, TAMP.			

**Figura 5.25** Archivo excel generado de la asistencia a prácticas de campo

“Generar estadísticas” simplemente genera un archivo excel con una lista de todas las prácticas con sus inscripciones y asistencias:

Validación
Constancias de Asistencia
Reglamento
Listas
Inscripción a oyentes o externos
Ver materias inscritas del alumno
Pasar lista alumnos
Generar lista de asistencia
Generar Estadísticas
Edición de prácticas
Agregar o eliminar prácticas
Agregar o eliminar alumnos

## Generar estadísticas

Número de alumnos inscritos contra asistentes a cada una de las prácticas.

ENVIAR

Figura 5.26 Seleccionar opción para generar estadísticas

	A	B	C	D
1	Folio	Práctica	Asistieron	Inscribieron
2	2	LABORATORIO DE COMISION FEDERAL DE ELECTRICIDAD EN CUERNAVACA	0	1
3	14	INSTITUTO MEXICANO DEL TRANSPORTE, TERMINAL INTERMODAL LOGISTICA DE HIDALGO Y AEROPUERTO INTERNACIONAL DE QUERÉTARO	1	1

Figura 5.27 Archivo excel generado de las estadísticas

“Edición de prácticas” permite visualizar y cambiar información de las prácticas, una de las más importante es cancelar la práctica:

Sistema de Inscripción a Prácticas de Campo										ING. ERNESTO RENÉ MENDOZA SÁNCHEZ ▾									
Validación	Folio	Nombre	Fecha de Inscripción	Fecha para firmar	Fecha de llegada	Fecha de salida	Lugar de llegada	Lugar de salida	Capo	Grupo	Profesor	Vuelta A	CASCO CHALECO Y ZAPATO INDUSTRIAL	NO LLEVAR CELULAR	CABINETE O SEGURO CORRESPONDIENTE ORIGINAL	CÁMARA	Canceleda	Clave de la materia	Aplicar Modificación
Constancias de Asistencia	1	SISTEMAS DE TRANSPORTE	2020-12-05	2020-12-06	2020-02-15 14:00:00	2020-02-15 08:45:00	CALLE DE DELICIAS No 67, METRO SALTO DEL AGUA	CALLE DE DELICIAS No 67, METRO SALTO DEL AGUA	30	2	MA. FRANCISCO GRANADOS VILARDETE	PUESTO CENTRAL DE CONTROL DEL METRO	SI	SI	SI	SI	SI	781	EDITAR
Reglamento	2	INTEGRACION DE PROYECTOS	2021-02-05	2021-02-10	2021-02-19 15:00:00	2021-02-19 07:00:00	EE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	14	4	DR. HUMBERTO MARENGO MOGOLLON	LABORATORIO DE COMISION FEDERAL DE ELECTRICIDAD EN CUERNAVACA	SI	SI	SI	SI	NO	1966	EDITAR
Listas	3	SISTEMAS DE TRANSPORTE	2020-12-05	2020-12-05	2020-02-22 14:00:00	2020-02-22 08:45:00	ESTACION BUENAVISTA	ESTACION BUENAVISTA	30	5	M.I. MAURO TERAN HUERTA	TREN SUBURBANO	SI	SI	SI	SI	SI	781	EDITAR
Inscripción a oyentes o externos	4	SISTEMAS DE TRANSPORTE	2020-12-05	2020-12-05	2020-05-01 16:00:00	2020-05-28 07:00:00	EE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	30	1	ING. GUILLERMO ESCOBAR CASTAÑEDA	AEROPUERTO INTERNACIONAL DE CUERNAVACA, MORELOS	SI	SI	SI	SI	SI	781	EDITAR
Ver materias inscritas del alumno	5	SISTEMAS DE TRANSPORTE	2020-12-05	2020-12-05	2020-03-14 14:00:00	2020-05-14 08:45:00	CALLE DE DELICIAS No 67, METRO SALTO DEL AGUA	CALLE DE DELICIAS No 67, METRO SALTO DEL AGUA	30	4	ING. GUILLERMO ESCOBAR CASTAÑEDA	PUESTO CENTRAL DE CONTROL DEL METRO	SI	SI	SI	SI	NO	781	EDITAR
Pasar lista alumnos	7	INTEGRACION DE PROYECTOS	2020-12-20	2020-12-27	2020-12-29 12:00:00	2020-12-29 06:00:00	EE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	18	4	DR. HUMBERTO MARENGO MOGOLLON	CENTRAL HIDROELECTRICA DE ZIMAHUÁN, HIDALGO	SI	SI	SI	SI	NO	1966	EDITAR
Generar lista de asistencia	8	SISTEMAS DE TRANSPORTE	2020-12-17	2020-12-20	2020-12-22 16:00:00	2020-12-22 07:00:00	EE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	28	3	DR. ALEJANDRO LEO VARGAS	AEROPUERTO INTERNACIONAL DE PUERBLA, PUE.	SI	SI	SI	SI	NO	781	EDITAR
Generar Estadísticas	9	INTEGRACION DE PROYECTOS	2020-12-05	2020-12-05	2020-04-06 21:00:00	2020-04-20 06:00:00	EE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	15	4	ING. VICTOR MANUEL VALDEZ RUBIO	FABRICACION DE PLATAFORMAS PETROLERAS EN TAMPUCO, TAMPS.	SI	SI	SI	SI	SI	1966	EDITAR
Edición de prácticas	10	SISTEMAS DE TRANSPORTE	2020-12-05	2020-12-05	2020-05-28 16:00:00	2020-05-28 07:00:00	EE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	30	1	ING. GUILLERMO ESCOBAR CASTAÑEDA	AEROPUERTO INTERNACIONAL DE CUERNAVACA, MORELOS	SI	SI	SI	SI	SI	781	EDITAR
Agregar o eliminar prácticas	11	INTEGRACION DE PROYECTOS	2020-12-05	2020-12-05	2020-04-13 20:00:00	2020-04-13 06:00:00	EE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	18		DR. HUMBERTO MARENGO MOGOLLON	CENTRALES HIDROELECTRICAS CAJON Y LA VESCA	SI	SI	SI	SI	SI	1966	EDITAR
Agregar o eliminar alumnos	12	SISTEMAS DE TRANSPORTE	2020-12-05	2020-12-05	2020-04-27 14:00:00	2020-04-25 14:00:00	EE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	EE 10 SUR Y CERRO DEL AGUA (METRO COPILCO)	40		VARIOS	PUESTO Y TRABAJOS DE AMPLIACION DEL PUESTO DE VERACRUZ, AEROPUERTO INTERNACIONAL DE	SI	SI	SI	SI	SI	781	EDITAR

Figura 5.28 Lista de prácticas para poder editarlas

y para editar la práctica deseada solo se debe dar clic en el botón editar, por lo que se mostrará un “modal” con un formulario:



**Figura 5.29 Formulario para editar los datos de la práctica seleccionada**

La penúltima opción es “Agregar o eliminar prácticas” la siguiente imagen muestra lo que vera el administrador:



**Figura 5.30 Formularios para agregar o eliminar prácticas**

La última opción es “Agregar o eliminar alumnos”, esta opción se muestra en la siguiente imagen:



**Figura 5.31 Formularios para agregar o eliminar alumnos**

### Pruebas de *URLs* con *Postman*

El *framework CodeIgniter* te permite tener *URLs* amigables, por ejemplo:

“http://localhost/siprac/registrarAlumno”

“http://localhost/siprac/iniciarSesion”

Las rutas se implementaron para que respondieran con una página web, es decir *HTML*, *CSS3*, etc., pero también hay rutas que responden con *JSON*, éstas en específico son las rutas que se probaron con *Postman* para verificar que las respuestas sean las correctas, un ejemplo claro es cuando el alumno trata de iniciar sesión, él/ella puede ingresar mal el usuario o la contraseña por lo que la ruta deberá regresar un *JSON* con la siguiente información:

```
{
  "estatus": "2",
  "msg": "Usuario o contrase\u00f1a incorrecta"
}
```

En el caso de que alguna de las credenciales no se hayan ingresado la ruta responderá con el siguiente *JSON*:

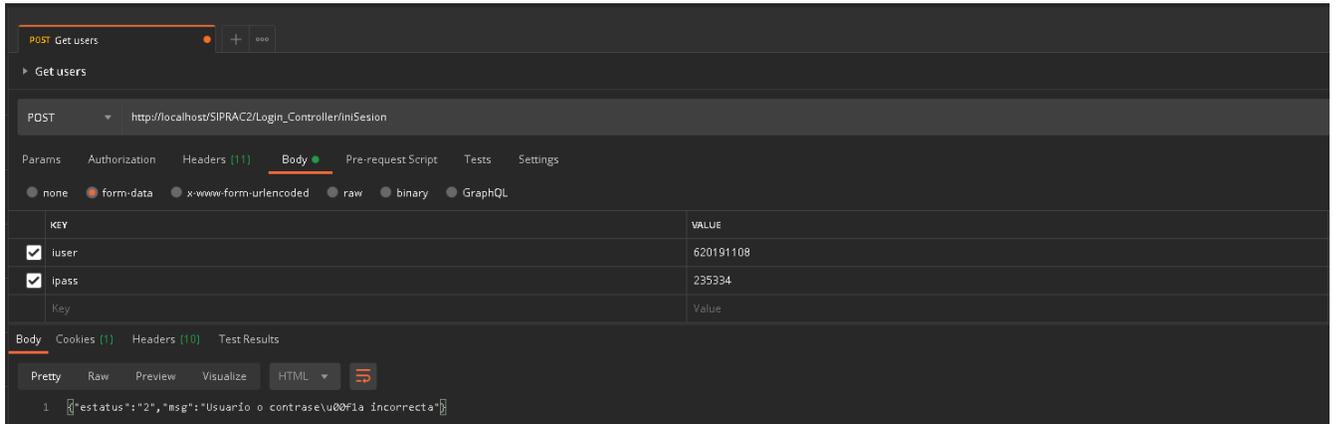
```
{
  "estatus":"0",
  "msg":{
    "iuser":"El usuario es obligatorio.",
    "ipass":"La contrase\u00f1a es obligatoria."
  }
}
```

, en caso contrario el usuario ingreso correctamente las credenciales y deberá obtener el siguiente *JSON*:

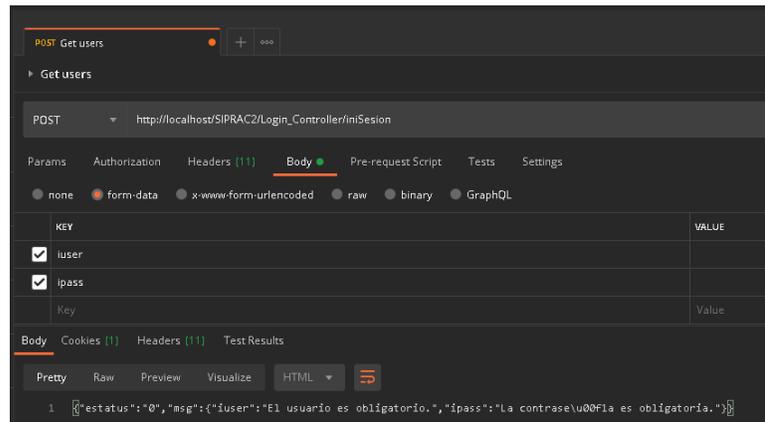
```
{
  "estatus":"1",
  "msg":"Bienvenido"
}
```

En este caso la solicitud se realiza cuando el alumno da clic en el botón enviar, esto provoca que se active una función *Javascript* que utiliza *Jquery* para enviar la solicitud, por lo que *Jquery* está atento a la respuesta, si contiene estatus igual a dos mostrará el mensaje de “Usuario o contraseña incorrecta”, en caso de que las credenciales estén bien hace una redirección a la página del alumno o en caso que no haya ingresado alguna de las credenciales modificará la página de bienvenida para que muestre los mensajes de “El usuario es obligatorio” o “La contraseña es obligatoria”.

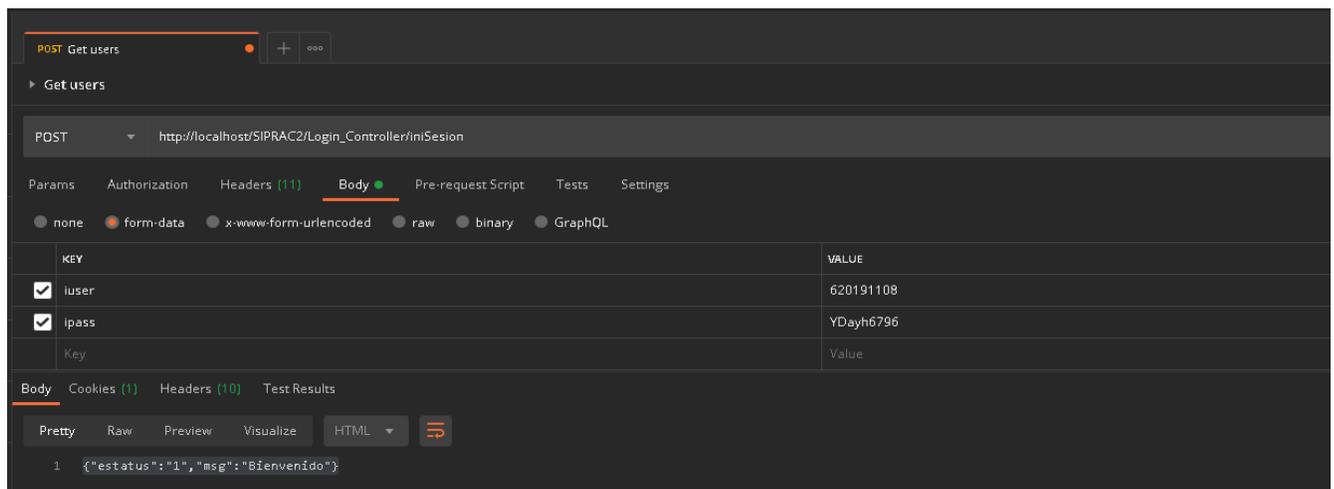
## Imágenes de *Postman* con los resultados



**Figura 5.32** Credenciales incorrectas



**Figura 5.33** Las credenciales no fueron ingresadas



**Figura 5.34** Las credenciales son correctas

Por lo tanto, cómo las respuestas son correctas la prueba con esta ruta ha finalizado y así se realiza con cada una de las rutas restantes.

## Capítulo 6. Validación del proyecto y conclusiones

Al finalizar el desarrollo de **SIPRAC** se enseñó cómo utilizar el sistema al administrador tanto la parte del alumno cómo la que utilizará para hacer cambios en la información, este proceso se llevó a cabo en el cubículo del ingeniero.



### 6.1 Enseñando el sistema finalizado

En primera instancia se mostró cómo el alumno se registra y llena sus datos requeridos para la asistencia a cualquier práctica, después por celular se les enseñó el correo que recibirá cada uno de los alumnos al registrarse con las credenciales, en seguida efectuamos el ingreso al sistema donde el alumno puede visualizar las prácticas a inscribirse y las inscritas, entre otros datos.

Por último, se les mostró cada una de las opciones en la página del administrador, pero la que más les llamó la atención es la de cargar a los alumnos y las prácticas, ya que le permite al administrador no depender tanto de la persona a cargo de los sistemas y en caso de la ausencia de éste poder hacer a tiempo cambios y carga de datos, con esto quedaron fascinados con el sistema y con ello fue aceptado el sistema.

## Conclusiones

El conocimiento adquirido al desarrollar desde cero un sistema permitió mejorar la comunicación que se tenía para dar ideas y recibirlas, ya que se pasó su proceso manual de inscripción a las prácticas de campo a un sitio web que hace lo mismo pero con menos trabajo, por lo que dejó escuchar los requerimientos de cada uno de los ingenieros a cargo y combinarla con los conocimientos en el desarrollo del sitios web para darles un sistema que les ayude y que se adecue a sus necesidades, también permitió visualizar cómo un sistema va cambiando con el paso del tiempo, es decir que la mayoría de los sistemas desarrollados se tenía una visualización inicial de ellos pero con el tiempo se va mejorando o cambiando requerimientos iniciales por otros y el sistema con el que se finalizó es el deseado por los ingenieros y que cumple con su función.

Una de las dificultades que se presentaron durante el desarrollo del sistema fue en esencia que las dos personas encargadas de las inscripciones a las prácticas de campo discernían en el proceso de validación ya que la idea de uno de los encargados era que solamente vinieran una sola vez a firmar y ya no a validar su inscripción, la otra persona a cargo concordaba con nosotros ya que la idea es que el alumno venga una primera vez para asegurar que el alumno sí quiere asistir a la práctica porque el problema principal es que si algún alumno no firma el documento la Facultad no lo acepta, pero en este caso si no valida el alumno su inscripción entonces no aparecerá en la lista y la segunda vez que visiten el cubículo de la persona a cargo simplemente sea para firmar el documento y reforzando que esto se cumpla, se les informa a los estudiantes desde el principio del semestre la que si se inscriben y validan alguna práctica tienen que asistir ya que sino habría represalias por ello, al último se explicó por qué esta última opción es la correcta, finalmente se llegó a un acuerdo y se implementó, éste es un claro ejemplo de aclarar las ideas, tener comunicación y hay veces que las ideas chocan pero hay que resolver el conflicto para que el sistema siga su desarrollo, la experiencia adquirida fue una de las mejores que he tenido además fue agradable ir viendo cómo progresaba el desarrollo del sistema.

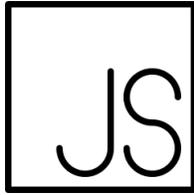
Lo aprendido en la escuela junto con el nuevo conocimiento adquirido te hace ver que finalizando la escuela no significa que el estudio termina sino que hay mucho por aprender, por actualizarse y por desarrollar, ya que continuamente van agregando nuevas funcionalidades a los lenguajes de programación por ejemplo *JAVA* que fue uno de mis primeros lenguajes de programación orientados a objetos lo aprendí en la preparatoria, pero continuamente van sacando versiones como por ejemplo la número ocho donde agregaron funciones lambda que facilita aún más el desarrollo del software y sé que *PHP* no es la excepción por ello tengo que seguir leyendo la documentación de las nuevas versiones que saquen al igual que los frameworks, las bases de datos, *CMMI-DEV* y

*IFPUG*, por mi parte es un gusto cumplir con mi deber y regresar un poco de lo que la universidad me ha ofrecido durante mi estadía en ella, al igual que el pueblo que me dió la oportunidad de finalizar mi carrera y además seguiré apoyando con lo que este a mi alcance, gracias.

## Referencias

- Peter R. Hill. (2011). Practical Software Project Estimation A toolkit for Estimating Software Development Effort & Duration. United States of America: McGraw-Hill.
- Steve McConnell. (2006). Software Estimation Demystifying the Black Art. United States of America: Microsoft Press.
- Jan L. Harrington. (2016). relational database design and implementation. United States of America: Morgan Kaufmann.
- Carnegie Mellon University. (2010). CMMI for Development, Version 1.3. United States of America: Software Engineering Institute.
- AllAN Albrecht, Kim Albee, Maarten Barth, Andy Belden, Angela Benton, Mary Bradley, ... Grant Rule. (2010). Function Point Counting Practices Manual. United States of America: IFPUG.
- Eric S. Norman, Shelly A. Brotherton & Robert T. Fried. (2008). Work Breakdown Structures: Foundation for project management excellence. Hoboken, New Jersey & Canada: John Wiley & Sons, Inc..

## Referencia de imágenes



javascript by Ben Davis from the Noun Project

<https://thenounproject.com/search/?q=javascript&i=980458>

CCBY



PHP by icon 54 from the Noun Project

<https://thenounproject.com/term/php/396500/>

CCBY



<https://thenounproject.com/search/?q=python&i=1375869>

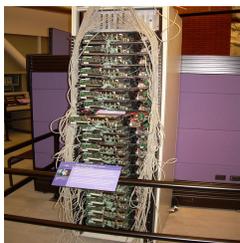
python by Danil Polshin from the Noun Project

CCBY



<https://www.flickr.com/photos/torkildr/3462607995>

Torkild Retvedt



<https://www.flickr.com/photos/torkildr/>

CCBYSA

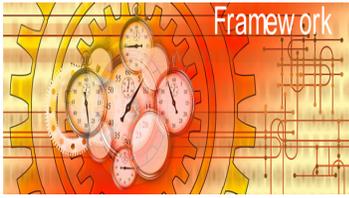
Carlo Nardone from Roma, Italy, CC BY-SA 2.0

<<https://creativecommons.org/licenses/by-sa/2.0/>>, via Wikimedia Commons



Eric Gaba, Wikimedia Commons user Sting, CC BY-SA 3.0

<<https://creativecommons.org/licenses/by-sa/3.0/>>, via Wikimedia Commons



<https://pixabay.com/es/photos/cron%C3%B3metro-engranajes-trabajo-3699318/>



Fire by Mihail Zhukov from the Noun Project



database by Komkrit Noenpoempisut from the Noun Project



<https://pxhere.com/es/photo/754216> CC0



Foto de Moe Magners: <https://www.pexels.com/es-es/foto/gente-sentado-habitacion-mirando-7495611/>

## Referencia de videos



<https://www.youtube.com/watch?v=XZmGGAhHqa0>

# Glosario

**Internal logical files (ILF):** Entidades persistentes que son mantenidas por el software de la aplicación.<sup>9</sup>

**External Interface files (EIF):** Entidades persistentes que son solamente referenciadas por el software de la aplicación.<sup>10</sup>

**External inputs (EI):** Procesos elementales del negocio donde los datos se envían hacia la aplicación para ser almacenados en un *ILF* o para asegurar el cumplimiento de los requerimientos comerciales del usuario.<sup>11</sup>

**External outputs (EO):** Procesos elementales del negocio que resultan en datos dejando los límites de la aplicación para cumplir un requerimiento del usuario que involucra al menos una de las siguientes opciones:

- Un calculo
- La actualización de uno o más *ILFs*.
- Controlar el comportamiento del *software*
- Derivación de los datos<sup>12</sup>

**Datos derivados:** El proceso de creación de un valor a partir de uno o más valores a través de un algoritmo.<sup>13</sup>

**External queries (EQ):** Procesos elementales del negocio que resultan en datos dejando los límites de la aplicación para cumplir un requerimiento del usuario, que involucra la recuperación de datos de al menos un *ILF* o *EIF* y no involucra las siguientes opciones:

---

<sup>9</sup> Peter R. Hill. (2011). Practical Software Project Estimation A toolkit for Estimating Software Development Effort & Duration. United States of America: McGraw-Hill, página 155.

<sup>10</sup> Peter R. Hill. (2011). Practical Software Project Estimation A toolkit for Estimating Software Development Effort & Duration. United States of America: McGraw-Hill, página 156.

<sup>11</sup> Peter R. Hill. (2011). Practical Software Project Estimation A toolkit for Estimating Software Development Effort & Duration. United States of America: McGraw-Hill, página 156.

<sup>12</sup> Peter R. Hill. (2011). Practical Software Project Estimation A toolkit for Estimating Software Development Effort & Duration. United States of America: McGraw-Hill, página 156.

<sup>13</sup> Definición obtenida a partir de la página web <http://www.grcdi.nl/dqglossary/data%20derivation.html>

- Un calculo
- La actualización de uno o más *ILFs*.
- Controlar el comportamiento del software
- Derivación de los datos<sup>14</sup>

**Function Points:** Son una unidad de medida estandarizada para representar el tamaño del software.<sup>15</sup>

**Hero:** Es una imagen con una descripción de lo que estás ofreciendo en tu sitio web, especialmente debe de llamar la atención del usuario.

**Backend:** Es la parte de un sistema computacional o aplicación que no es directamente accedida por el usuario, normalmente es la encargada e almacenar y manipular los datos.<sup>16</sup>

**Frontend:** Es la parte del sistema que visualiza e interactua el usuario en su dispositivo.

---

<sup>14</sup> Peter R. Hill. (2011). Practical Software Project Estimation A toolkit for Estimating Software Development Effort & Duration. United States of America: McGraw-Hill, página 156.

<sup>15</sup> Definición obtenida a partir de la página web <https://www.ifpug.org/faqs-2/>

<sup>16</sup> Al hacer la consulta en Google del siguiente texto “meaning of backend” se obtuvo el significado y el diccionario de inglés de Google es proporcionado por Oxford Languages.