



UNIVERSIDAD POLITECNICA DE PUEBLA
INGENIERÍA EN INFORMÁTICA

PROYECTO DE ESTADÍA PROFESIONAL

MODULO HORARIO DE LABORATORIOS

OMAR TOXQUI TOLAMA

ASESOR TÉCNICO
MTRA. REBECA RODRÍGUEZ HUESCA

ASESOR ACADÉMICO
MTRA. REBECA RODRÍGUEZ HUESCA

Juan C. Bonilla, Puebla.

ÍNDICE

INTRODUCCION	1
CAPITULO I	2
1.1 Misión	2
1.2 Visión	2
1.3 Organigrama	2
1.5 Función departamento de informática	4
CAPITULO II	5
2.1 Metodología	5
2.2 Herramientas	6
Capitulo III	8
3.1 Análisis de requerimientos	8
3.2 Diseño	9
3.3 Implementación	14
3.4 Lanzamiento	15
3.5 Prototipo	19
3.6 Evaluación	20
Conclusión	21
Glosario	22
Anexos	23
Bibliografía	31

INTRODUCCION

Hoy en día nos encontramos en la época de evolución en la tecnología, donde el objetivo es hacer la vida de todos más fácil de forma automatizada, las personas se van adaptando a ello acostumbrándose a realiza tareas optimizadas de una manera más rápida diariamente.

En este documento se explican todos los pasos que se siguieron para el desarrollo del proyecto comprendiendo la institución en la que se va a trabajar, explicando la metodología que se siguió los pasos para instalar las herramientas necesarias y el desarrollo de la aplicación.

Este proyecto consiste en crear un sistema para la administración de los laboratorios que tiene la Uppue, para una asignación más ordenada, eficaz y más rápido ya que será una aplicación web y se podrá acceder desde internet todo esto para tener un mejor control.

CAPITULO I

La Universidad Politécnica de Puebla, es una Institución Pública que mediante un modelo de educación basado en competencias, prepara profesionales, con una sólida formación científica, técnica y social, conscientes del contexto económico, político y sociocultural del Estado y del país; además, impulsa la investigación aplicada y la innovación tecnológica, asociadas a las necesidades de las empresas y de la sociedad. [1]

1.1 Misión

Formar íntegramente profesionales competentes que atiendan necesidades de los sectores productivo y social, mediante el desarrollo tecnológico, la innovación y la investigación aplicada, promoviendo una cultura ambiental y de equidad de género.

1.2 Visión

Ser una Universidad acreditada nacional e internacionalmente por su modelo educativo, sus servicios y la calidad de sus egresados(as), en un marco de sustentabilidad, de respeto al medio ambiente y fomento a la equidad de género, que consolida permanentemente alianzas estratégicas con los sectores productivo y social. [2]

1.3 Organigrama

La institución está compuesta de la siguiente manera (Figura 1. Organigrama) [3]

Universidad Politécnica de Puebla

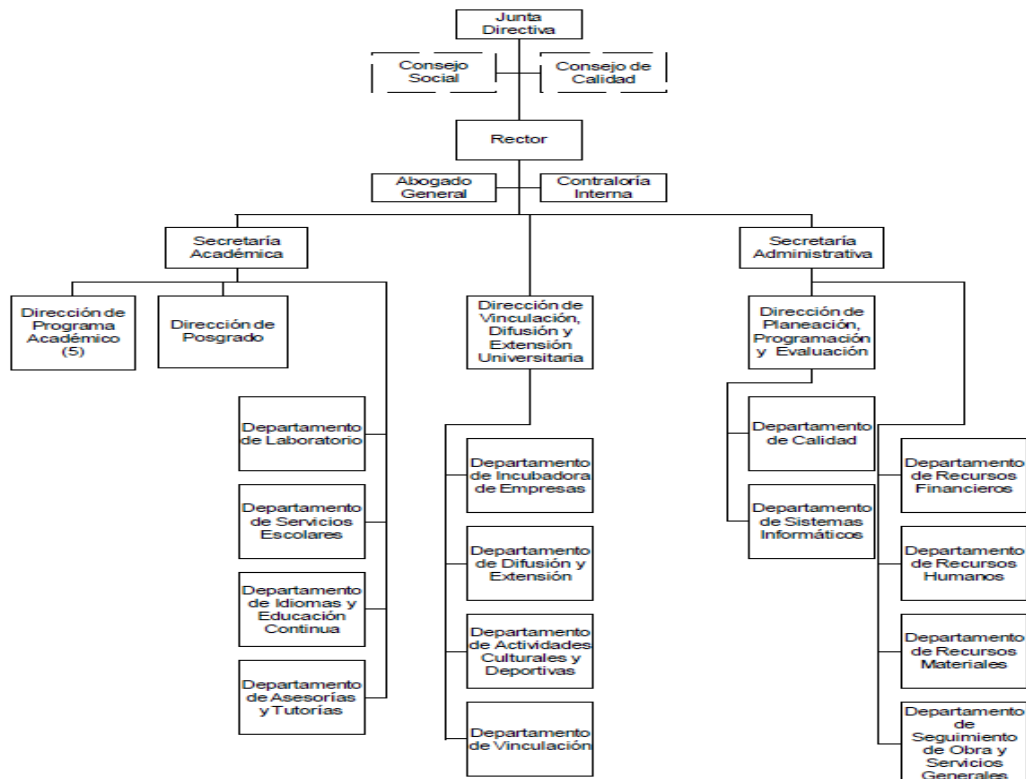


Figura 1. Control de Acceso al sistema

1.4 Funciones

ARTÍCULO 4.- Para el cumplimiento de su objeto, la "Universidad" tendrá las atribuciones siguientes:

- I.- Fomentar el desarrollo de la investigación en el sector público y privado;
- II.- Contribuir a la adopción y asimilación de tecnologías de vanguardia en las empresas del sector público y privado, que les permitan mejorar su competitividad;
- III.- Impulsar en forma permanente mecanismos de evaluación de calidad, docencia, investigación y desarrollo tecnológico, a través de evaluaciones internas y externas a fin de lograr los más altos estándares de calidad;
- IV.- Reglamentar la selección, ingreso, estancia y egreso de los estudiantes;
- V.- Establecer los términos del ingreso, promoción y permanencia del personal académico, así como la selección, admisión, ascenso del personal administrativo, apoyada en la reglamentación correspondiente;
- VI.- Impulsar la certificación de procesos estratégicos de gestión de los servicios y programas que apoyan las actividades académicas con el objeto de asegurar la calidad de la gestión institucional;
- VII.- Promover y desarrollar programas y proyectos académicos de intercambio, cooperación, coordinación y colaboración de beneficio institucional con organismos e instituciones de los diversos sectores social, público y privado nacionales e internacionales;
- VIII.- Elaborar programas educativos con base en competencias profesionales de buena calidad con una amplia aceptación social para la sólida formación tecnológica y en valores de sus egresados;
- IX.- Planear y programar la enseñanza superior que imparta en un modelo curricular flexible, dentro de los lineamientos señalados en materia de educación superior;
- X.- Emitir certificados de estudio, certificados de competencias laborales, así como otorgar diplomas, títulos o grados académicos a los alumnos que hayan concluido estudios, de conformidad con los requisitos establecidos en los planes y programas de estudios correspondientes. Los títulos o grados académicos serán expedidos por el Gobierno del Estado y suscritos por el Secretario de Educación Pública del mismo;
- XI.- Realizar los trámites de equivalencia de estudios del mismo tipo educativo cursados en otras instituciones de enseñanza superior nacionales y extranjeras, en términos de la normatividad aplicable;
- XII.- Crear las instancias necesarias de vinculación con los sectores público, privado y social, que deberán ser distintas y diferenciadas de los órganos de gobierno de la "Universidad";
- XIII.- Promover y organizar programas de prestación del servicio social, estadías, intercambios u otras modalidades de vinculación entre la sociedad y la "Universidad" acordes a los objetivos de los programas educativos;
- XIV.- Celebrar convenios, contratos, acuerdos y demás actos jurídicos que se requieran para el funcionamiento de la "Universidad";
- XV.- Implementar mecanismos de apoyo financiero;

XVI.- Instituir anualmente el calendario escolar en función de los programas de trabajo aprobados por los órganos competentes, de modo que pueda cumplir de manera eficaz las actividades académicas programas;

XVII.- Conferir grados honoríficos, distinciones, reconocimientos y estímulos, y

XVIII.- Las demás que le confieran las normas y disposiciones reglamentarias de la "Universidad".

1.5 Función departamento de informática

Formar profesionales competentes, con conocimientos, habilidades y actitudes para especificar, diseñar, desarrollar, implementar y administrar soluciones de negocios basadas en computadora, con una amplia visión de la ciencia y las nuevas tecnologías de la información. [2]

CAPITULO II

2.1 Metodología

La metodología a utilizar será MP lu+a (Figura 2. Modelo de Proceso de la Ingeniería de la usabilidad y de la accesibilidad)[5] con las siguientes etapas:

- **Análisis de Requisitos:**
El punto de vista que de la fase del Análisis de Requisitos realiza la Ingeniería del Software, establece los servicios que el sistema debe proporcionar y las restricciones bajo las cuales debe operar. Se especifican las condiciones que determinan qué debe hacer el sistema
- **Diseño:**
Se llega a esta fase tras realizar actividades relacionadas con el Análisis de Requisitos que proporcionan información necesaria para que el equipo de desarrollo sea capaz de modelar el sistema para, posteriormente, proceder a su codificación.
- **Implementación**
La fase de implementación es conocida también como fase de codificación, pues supone todo el proceso de escribir el código software necesario que hará posible que el sistema finalmente implementado cumpla con las especificaciones establecidas en la fase de análisis de requisitos y responda al diseño del sistema descrito en la fase anterior.
- **Lanzamiento**
Es el momento en que se ven concretadas en mayor o menor grado las expectativas puestas en el producto. El usuario final del producto tiene un peso específico enorme a la hora de indicar si el producto será aceptado o no.
- **Prototipado**
Los prototipos, que son documentos, diseños o sistemas que simulan o tienen implementadas partes del sistema final, constituyen una herramienta muy útil para, como es nuestra intención, hacer participar al usuario en el desarrollo y poder evaluar el producto desde las primeras fases del desarrollo.
- **Evaluación**
Evaluar consiste en probar algo. Tanto para saber si funciona correctamente como no, si cumple con las expectativas o no, o simplemente para conocer cómo funciona una determinada herramienta o utensilio.

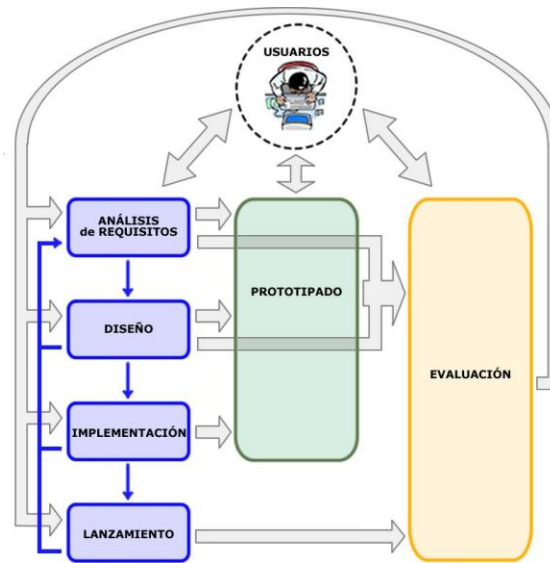


Figura 2. MPLu+a

2.2 Herramientas

Las herramientas a utilizar son:

1.- Sistema operativo DEBIAN 8:

- Ventajas

Este sistema operativo es adecuado ya que ofrece estabilidad para servidores que trabajan durante más de un año seguido sin reiniciarse, rápido y ligero en memoria y tiene buena seguridad del sistema.

- Desventajas

Se necesita un previo conocimiento en Linux para poder usar con comodidad el Sistema Operativo [4]

2.- Ruby 2.1 y RubyGems 2.6

- Ventaja

Permite desarrollar soluciones a bajo costo, software libre y multiplataforma.

- Desventaja:

No soporta polimorfismo de funciones, no es tipado fuerte.

1.- PostgreSQL

- Ventajas

Es ideal para tecnologías web, ofrece estabilidad, optimización y manejo de errores, ya que cuando un proceso llega a fallar, no afecta el resto.

- Desventaja

Es fácil de vulnerar sin protección adecuada, reducida cantidad de tipo de datos sin experticia suele ser un caos de configurar.

Capítulo III

3.1 Análisis de requerimientos

Sistema horario de laboratorios.

La Universidad Politécnica de Puebla tiene la necesidad de contar con un sistema que permita llevar un mejor control, que a su vez sea fácil de manejar; el control consiste en llevar a cabo un registro de todos los laboratorios con los que se cuenta en las diferentes carreras y llevar un horario para asignarlos.

Caso de uso: Crear horario (Figura 3. Caso de uso crear horario)

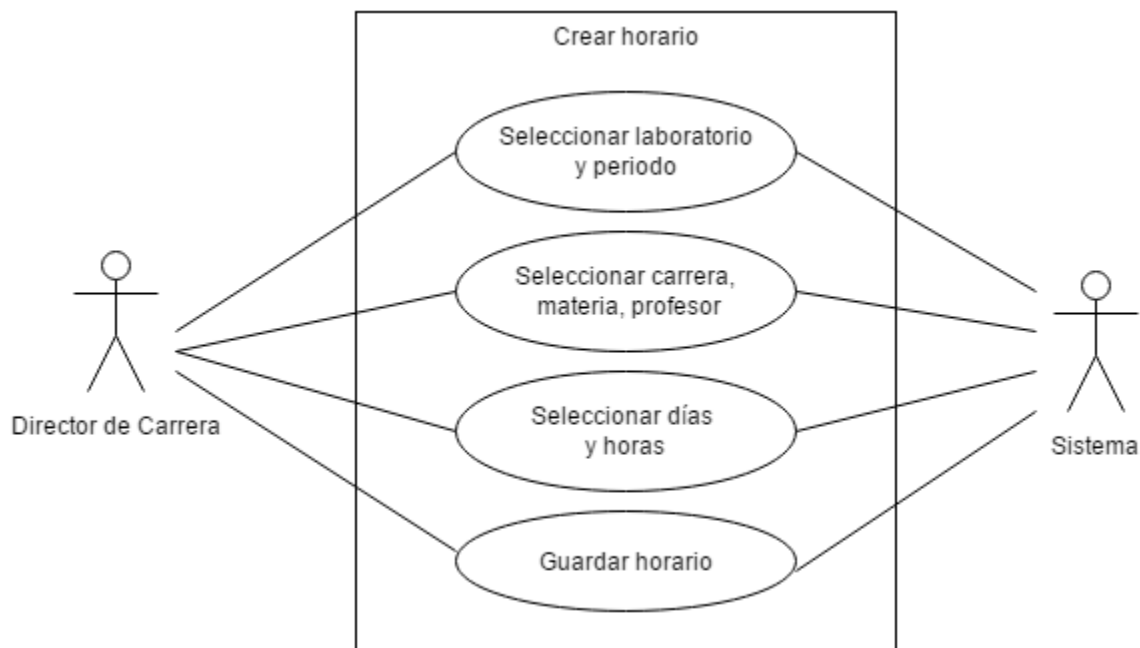


Figura 3. Caso de uso crear horario

Nombre caso de uso	Crear horario
Descripción	En este módulo el usuario puede consultar la disponibilidad de un laboratorio para crear un horario de clases que será impartido dentro de este ingresando la información requerida.
Actores	Director de carrera
Descripción paso a paso	1. Seleccionar carrera, materia y profesor a) El sistema muestra una lista de carreras. b) El usuario selecciona una carrera

	<p>c) El usuario selecciona una materia y el nombre de profesor correspondiente.</p> <p>2. Seleccionar laboratorio y periodo</p> <p>a) El sistema muestra una lista de laboratorios disponibles</p> <p>b) El usuario selecciona un laboratorio</p> <p>c) El usuario selecciona el periodo</p> <p>d) El sistema muestra el horario del laboratorio</p> <p>3. Seleccionar día y horas</p> <p>a) El sistema muestra un formato con los días y horas disponibles.</p> <p>b) El usuario ingresa el día y hora necesaria en que ocupará el laboratorio tantas veces sea necesario para cubrir las horas planeadas, usando el mismo registro que se usó al seleccionar laboratorio y periodo.</p> <p>4. Guardar horario</p> <p>a) El sistema valida que todos los campos estén completos</p> <p>b) El sistema almacena los datos</p>
--	---

3.2 Diseño

En esta etapa los productos que se generaron fueron:

- Diseño de la base de datos.
- Diccionario de datos.
- Diseño de pantallas

A continuación se muestra el diseño de la BD (Figura 4. Diseño BD), Se muestran las tablas que se crearon así como sus relaciones, la tabla horarios es el encabezado del horario del laboratorio donde se almacena toda la información como programa educativo, profesores, laboratorios, asignatura, periodo y grupo, todo esto heredándola de las respectivas tablas donde se encuentra dicha información, también se tiene la tabla detalle de horario donde se hereda la clave del encabezado del horario y se guardan los días y la hora inicio y fin que se solicita el laboratorio seleccionado.

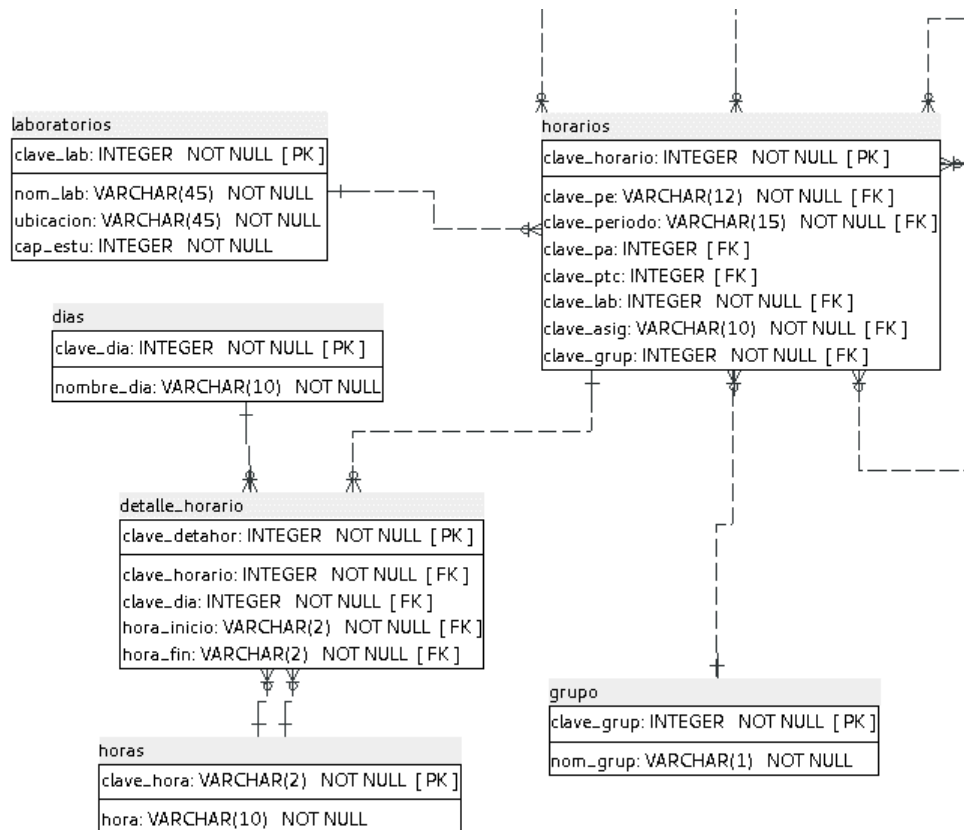


Figura 4. Diseño BD

Diccionario de datos

Este es el diccionario de datos de las tablas y relaciones que se crearon.

laboratorios

(Nombre físico: laboratorios)

Contiene el registro de los laboratorios que se tienen

Nombre Columna	Lógico Nombre columna	físico Tipo	PK	Nulo
clave_lab (PK)	clave_lab	INTEGER	PK	NOT NULL
Clave asignada a cada laboratorio				
nom_lab	nom_lab	VARCHAR(45)		NOT NULL
Asigna un nombre a un laboratorio				
ubicacion	ubicacion	VARCHAR(45)		NOT NULL
Indica la ubicación de un laboratorio en un edificio en específico				
cap_estu	cap_estu	INTEGER		NOT NULL

Indica la capacidad total de alumnos para cada laboratorio

Referenciado por

- horarios hace referencia a (clave_lab)

dias

(Nombre físico: dias)

Contiene el registro de los días de la semana

Nombre Columna	Lógico Nombre columna	físico Tipo	PK	Nulo
clave_dia (PK)	clave_dia	INTEGER	PK	NOT NULL
Clave del día				
nombre_dia	nombre_dia	VARCHAR(10)		NOT NULL
Indica el nombre del día				

Referenciado por

- detalle_horario hace referencia a (clave_dia)

horas

(Nombre físico: horas)

Contiene el registro de los días de las horas

Nombre Columna	Lógico Nombre columna	físico Tipo	PK	Nulo
clave_hora (PK)	clave_hora	VARCHAR(2)	PK	NOT NULL
Clave con la que se identifica una hora				
hora	hora	VARCHAR(10)		NOT NULL
Indica la hora				

Referenciado por

- detalle_horario hace referencia a (clave_hora)
- detalle_horario hace referencia a (clave_hora)

grupo

(Nombre físico: grupo)

Contiene el registro de los grupos

Nombre Columna	Lógico Nombre columna	físico Tipo	PK	Nulo
clave_grup (PK)	clave_grup	INTEGER	PK	NOT NULL
Indica la clave del grupo				
nom_grup	nom_grup	VARCHAR(1)		NOT NULL
Indica el nombre del grupo				

Referenciado por

- horarios hace referencia a (clave_grup)

horarios

(Nombre físico: horarios)

Contiene la información de los horarios creados para los laboratorios

Nombre Columna	Lógico Nombre columna	físico Tipo	PK	Nulo
clave_horario (PK)	clave_horario	INTEGER	PK	NOT NULL
Indica la clave del horario de laboratorio				
clave_pe (FK)	clave_pe	VARCHAR(12)		NOT NULL
Clave del programa educativo				
clave_periodo (FK)	clave_periodo	VARCHAR(15)		NOT NULL
Indica la clave del periodo				
clave_pa (FK)	clave_pa	INTEGER		
Es la clave asignada por el sistema a un PA				
clave_ptc (FK)	clave_ptc	INTEGER		
Es la clave asignada por el sistema a un PTC				
clave_lab (FK)	clave_lab	INTEGER		NOT NULL
Clave asignada a cada laboratorio				
clave_asig (FK)	clave_asig	VARCHAR(10)		NOT NULL
Clave asignada a la asignatura				
clave_grup (FK)	clave_grup	INTEGER		NOT NULL

Indica la clave del grupo

Referencias

- programas_educativos a través de (clave_pe)
- periodos a través de (clave_periodo)
- ptcs a través de (clave_ptc)
- profesores_asignatura a través de (clave_pa)
- asignatura a través de (clave_asig)
- laboratorios a través de (clave_lab)
- grupo a través de (clave_grup)

Referenciado por

- detalle_horario hace referencia a (clave_horario)

detalle_horario

(Nombre físico: detalle_horario)

Contiene la información del detalle de horario

Nombre Lógico	Columna	Nombre físico	Tipo	PK	Nulo
clave_detahor (PK)	clave_detahor	clave_detahor	INTEGER	PK	NOT NULL
Clave detalle horario					
clave_horario (FK)	clave_horario	clave_horario	INTEGER		NOT NULL
Indica la clave del horario de laboratorio					
clave_dia (FK)	clave_dia	clave_dia	INTEGER		NOT NULL
Clave del día					
hora_inicio (FK)	hora_inicio	hora_inicio	VARCHAR(2)		NOT NULL
Clave de la hora inicio					
hora_fin (FK)	hora_fin	hora_fin	VARCHAR(2)		NOT NULL
Clave de la hora final					

Referencias

- horarios a través de (clave_horario)

- días a través de (clave_dia)
- horas a través de (hora_inicio)
- horas a través de (hora_fin)

Listado de pantallas:

Número de pantalla	Ruta	Descripción	Permisos			
			Agregar	Modificar	Eliminar	Consultar
300	/reg_horariolab	Se puede solicitar un laboratorio dependiendo del periodo y el laboratorio mostrando los días disponibles y ocupados	SI	NO	NO	SI

3.3 Implementación

1.- Se inicia la implementación desde una plantilla que se estableció para seguir el patrón de colores estilo y estructura que ya se tenía establecido. Después se creó la vista reg_horariolaboratorio.erb dentro de esta vista se puede solicitar un laboratorio.

2.- Se implementaron select y radio button (Figura 5. Selección de información) dentro de esta vista para poder seleccionar los distintos datos que se necesitan para solicitar el laboratorio. La información se carga desde la base de datos donde se encuentra dicha información para este formulario.

Programa educativo:

Periodo:

Profesor: T. completo Asignatura

Laboratorio:

Grupo:

Asignatura:

Figura 5. Selección de información

3.- Estos select están bloqueados al inicio y sólo el programa educativo está activo, conforme se vaya seleccionando la información, el select que sigue será desbloqueado y generado automáticamente conforme a la información seleccionada por ejemplo el programa educativo mostrará sólo los profesores y materias que le pertenezcan. Estas

consultas se llevan a cabo por medio de JavaScript Ajax que hacen automático la petición de llenado de cada select por el método get (en el apartado de anexo se muestra el código que se implementó para este funcionamiento).

4.- Para la parte de seleccionar el día la hora de inicio y fin se genera una tabla de horarios dependiendo del periodo y el laboratorio seleccionado previamente, en esta tabla es donde se puede seleccionar el día y la hora que desea por medio de un checkbox permitiendo de esta forma solicitar el laboratorio, para los días y hora que ya esté ocupado dicho laboratorio consultando desde la base de datos se muestra un checkbox seleccionado y deshabilitado (en el apartado de anexo se muestra el código que se implementó para este funcionamiento), de esta manera se representa que ese día ya está ocupado y no podrá ser seleccionarlo (Figura 6. Tabla horario)

	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado
08:00 - 09:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
09:00 - 10:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10:00 - 11:00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11:00 - 12:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12:00 - 13:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13:00 - 14:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14:00 - 15:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15:00 - 16:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
16:00 - 17:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
17:00 - 18:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
18:00 - 19:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19:00 - 20:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 6. Tabla horario

5.- Después de tener toda la información necesaria para realizar la solicitud de laboratorio se envía la información por el método post y se verificada que todos los campos estén llenos para poder guardarlos en la base de datos, si todo fue exitoso se manda un mensaje “Registro guardado” y se guarda toda la información en la base de datos, en otro caso si faltó algún campo se manda un mensaje “Dejo un campo vacío”, en caso de que ocurra un error la información no se guardara y se manda un mensaje con el error. [8]

3.4 Lanzamiento

Instalación y configuración de las herramientas

Para el desarrollo de este proyecto es necesario contar con algunas herramientas en el servidor, para ello la instalación y configuración de estas es indispensables.

Instalación de S.O debían:

1.- Como primer paso se llevó a cabo la instalación de este S.O descargando una imagen del sistema en un medio de almacenamiento USB. Se siguieron todos los pasos del instalador, en la instalación cuando se está detectando el hardware de red pide un paquete de firmware para que pueda funcionar correctamente las interfaces de red, se tiene que descargar el fichero firmware-bnx2_0.43_all directamente desde la página de debían y depositarlo en un medio USB para que el instalador de debían instale este fichero. Continuando con la instalación se estableció como contraseña de superusuario “adminserver” y el usuario es “UPP” con la contraseña “2017”.

2.- Se presentó el problema de que no accedía a internet para ello se configuro el archivo nos logueamos como superusuario en una terminal y ejecutamos

Nano /etc/network/interfaces agregando lo siguiente:

```
#Interface eth0
auto eth0
allow-hotplug eth0
iface eth0 inet dhcp
```

Después de haber agregado este texto el archivo se guarda y se reinicia la pc para que acceda con la nueva configuración, esto es para obtener una dirección ip por dhcp

3.- Actualizar repositorios de debian 8 editar el archivo siguiendo logueado como superusuario ejecutamos nano /etc/apt/sources.list agregando lo siguiente

```
###DEBIAN JESSIE ESTABLE###
## Repositorio Oficial
deb http://http.us.debian.org/debian/ jessie main contrib non-free
deb-src http://http.us.debian.org/debian/ jessie main contrib non-free

## Repositorio de Seguridad
deb http://security.debian.org/ jessie/updates main contrib non-free
deb-src http://security.debian.org/ jessie/updates main contrib non-free
```

Se debe de comentar la línea que hace referencia al CD, ejecutamos los comandos:

```
apt update
```

Y por último se ejecuta

apt upgrade

Listo, tendremos el sistema actualizado con los repositorios oficiales debían 8.

Instalación Ruby

Para instalar Ruby se deberá ejecutar el siguiente comando en consola

```
sudo apt-get install ruby
```

Instalar RubyGems:

De la página <http://rubygems.org/pages/download> descargamos el archivo .zip o .tgz y lo descomprimos. Si se descarga el archivo rubygems-1.8.10.tgz, al descomprimirlo se crea un directorio rubygems-1.8.10, hay que cambiarse a ese directorio y ejecutar

```
sudo ruby setup.rb
```

Instalar Sinatra

Para instalar Sinatra se deberá ejecutar el siguiente comando en consola

```
sudo gem install Sinatra
```

Instalación Thin

Primero se deberá instalar el los paquetes ruby1.9.1-dev y build-essential con el comando

```
sudo apt-get install ruby1.9.1-dev build-essential
```

Después ejecutamos el siguiente comando para instalar Thin

```
sudo gem install thin
```

Instalar PostgreSQL

Para instalar Postgresql se deberá ejecutar el siguiente comando en consola

```
sudo apt-get install postgresql postgresql-contrib libpq-dev
```

Debemos poner un password al usuario postgres

```
sudo passwd postgres
```

Nos cambiamos al usuario postgres y asignamos un password al usuario postgres dentro de PostgreSQL

```
su postgres
psql -c "ALTER USER postgres WITH PASSWORD 'password_aqui' -d template1"
```

Aparece ALTER ROLE y nos muestra nuevamente el prompt

Todavía como el usuario postgres crear un nuevo usuario en PostgreSQL. Para crear el usuario en PostgreSQL vamos a indicar que sea superusuario (-s), que PostgreSQL nos pida indicar el password para dicho usuario (-P) y nos muestre en pantalla los mensajes que esto genere (-e de -echo)

El usuario se llamará sysdba y password la palabra s3cret

```
createuser -P -s -e sysdba
```

Aparece lo siguiente:

```
Ingrese la contraseña para el nuevo rol:
Ingrésela nuevamente:
CREATE ROLE sysdba PASSWORD 'md51767410185836193fcba975fe021e85b'
SUPERUSER CREATEDB CREATEROLE INHERIT LOGIN;
```

Salimos del usuario postgres con exit

Editamos el archivo /etc/postgresql/8.4/main/pg_hba.conf para indicar que el usuario que acabamos de crear puede acceder a la línea de comandos de PostgreSQL (psql)

```
sudo nano /etc/postgresql/8.4/main/pg_hba.conf
```

```
Agregamos la linea
local all sysdba md5
```

En la sección # Database administrative login by UNIX sockets

Y queda algo parecido a esto

```
# Database administrative login by UNIX sockets
local all postgres ident
local all sysdba md5
```

Reiniciamos el servidor PostgreSQL

```
sudo /etc/init.d/postgresql restart
```

Aparece

```
Restarting PostgreSQL 8.4 database server: main. [7]
```

Instalar el gem para conectarse desde Ruby a PostgreSQL

Para instalar la gema ejecutamos el siguiente comando en consola

```
sudo gem install pg
```

Instalar el adaptador que permite a Datamapper comunicarse con la base de datos

Para instalar la gema ejecutamos el siguiente comando en consola

```
sudo gem install dm-postgres-adapter
```

Si se está instalando una versión más nueva de PostgreSQL (como la 9.4) y/o una versión más nueva de las gemas de ruby (como rubygems-2.6.7), es posible que al instalar la gema dm-postgres-adapter marque un error como “Failed to build gem native extension.”. La solución es instalar el paquete postgresql-server-dev-9.4 con el siguiente comando en consola

```
sudo apt-get install postgresql-server-dev-9.4
```

Después de esto, intentar nuevamente la instalación de la gema dm-postgres-adapter y ya debe instalarse sin problemas. Ya tenemos instalado todo lo necesario para desarrollar aplicaciones web con acceso a base de datos utilizando Sinatra.

3.5 Prototipo

A continuación se muestra el prototipo que se tiene diseñado sobre la interfaz del sistema (Figura 7. Diseño interfaz), aquí es donde se selecciona toda la información necesaria para poder solicitar un laboratorio, así mismo se sigue el orden dependiendo del programa educativo, periodo y laboratorio para poder ir desglosando toda la información necesaria y a su vez mostrar los días que estén libres y ocupados. Se diseñó de esta forma para facilitar la solicitud procurando evitar errores de información y empalmes, asíéndolo más fácil y rápido.

Crear horario laboratorios

Horario

Programa educativo: INGENIERIA EN ELECTRÓNICA *

Periodo: SEP-DIC 2010 *

Profesor: T. completo Asignatura Mario Espinoza Tlaxcaltecati *

Asignatura: Álgebra Lineal *

Laboratorio: Lab - A *

Grupo: A *

	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado
08:00 - 09:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
09:00 - 10:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10:00 - 11:00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11:00 - 12:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12:00 - 13:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13:00 - 14:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14:00 - 15:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15:00 - 16:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
16:00 - 17:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
17:00 - 18:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
18:00 - 19:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19:00 - 20:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Guardar Cancelar

Figura 7. Diseño interfaz

3.6 Evaluación

A continuación se presentan las pruebas que se le hicieron a la interfaz para poder llevar acabo su evaluación, las pruebas de evaluación que se hicieron son:

- Prueba de compatibilidad con browserstack
- Prueba de seguridad con w3af

En la siguiente imagen (Figura 8. Prueba compatibilidad) se muestra los resultados que se obtuvieron en el test y los navegadores que es compatible la aplicación.




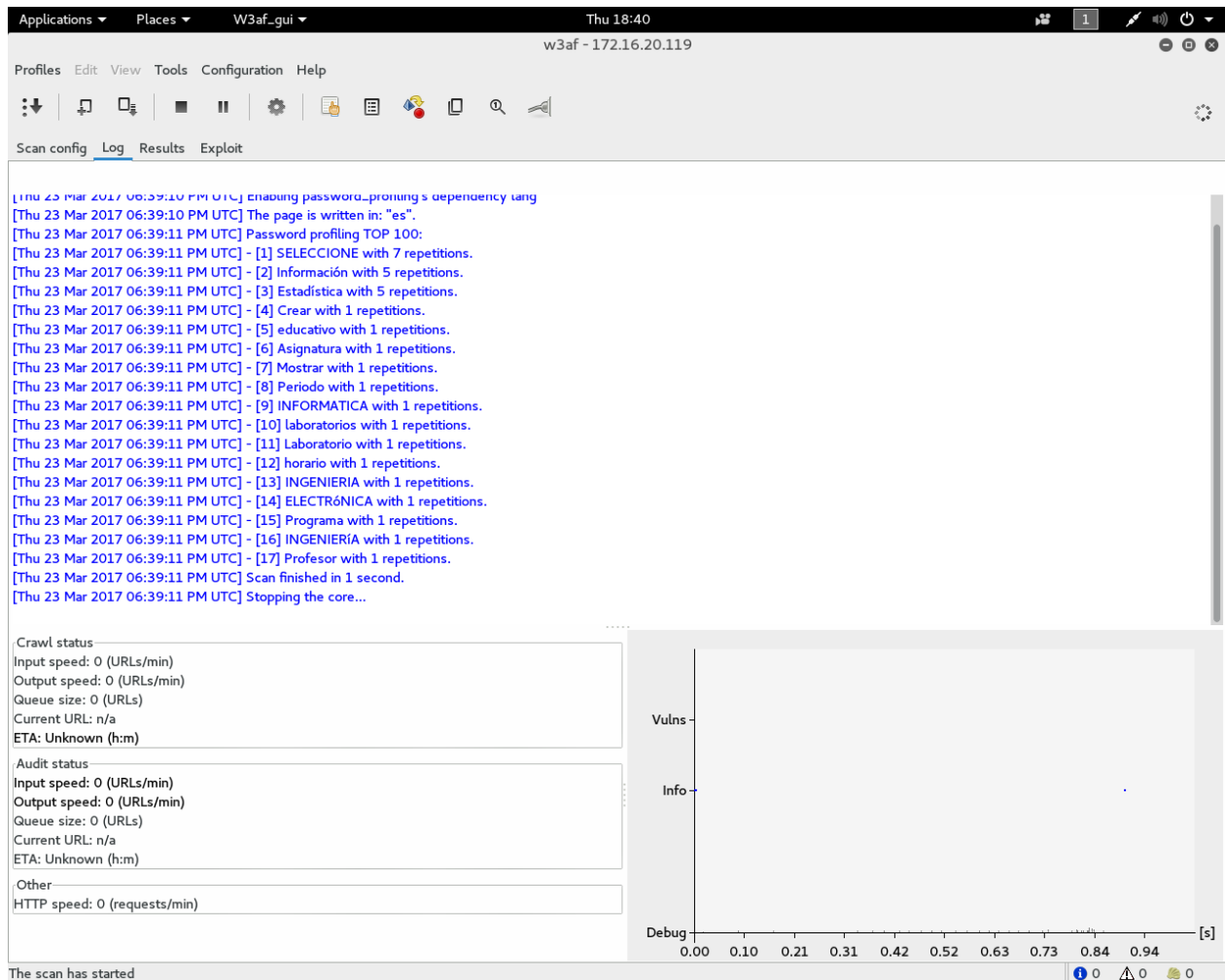
S.O.	 Chrome 36	 Firefox 30, 3.6	 Safari 8	 Opera 12.15	 Ie 8, 9, 10, 11
Windows 7, 8.1, 10	X	X		X	X
OS X Yosemite, Mavericks	X		X		
Linux		X			

Figura 8. Prueba compatibilidad

En la siguiente imagen (Figura 9. Prueba de seguridad w3af) mostramos el resultado que se obtuvo del test de seguridad con la herramienta w3af a la aplicación web.



Conclusión

En el desarrollo de este proyecto se obtuvo como resultado un módulo del sistema para poder solicitar un laboratorio y así crear un horario, este proyecto me ayudo a obtener conocimientos en el lenguaje Ruby desarrollando aplicaciones web en Sinatra y también aprender a manejar postgresql, este sistema beneficia a la Institución ayudándola a hacer más rápido, tener un mejor control y evitar empalmes en la solicitud de laboratorios de una forma automatizada. Como recomendación para desarrollar en este lenguaje es, si se tiene la lógica es cuestión de sólo aprender la sintaxis, recomiendo libro "Sinatra up and Running".

Glosario

Ruby: es un lenguaje de programación interpretado, reflexivo y orientado a objetos, creado por el programador japonés Yukihiro "Matz" Matsumoto

Sintara: es un DSL para construir páginas Web, servicios Web o aplicaciones Web en Ruby. Sinatra está basado en el enfoque minimalista de desarrollo, es decir no existe una estructura ni una manera "correcta" de hacer las cosas, solo una "mejor práctica" de cómo realizarlo.

Postgresql: es un Sistema de gestión de bases de datos relacional orientado a objetos y libre.

Select: El elemento select (<select>) de HTML representa un control que muestra un menú de opciones. Las opciones contenidas en el menú son representadas por elementos <option>.

Checkbox: nos permiten seleccionar varias opciones marcando el cuadrado. El cuadrado pulsado equivale a un "sí" y sin pulsar a un "no" o, lo que es lo mismo, a "true" o "false".

Anexos

Anexo 1

Código para solicitar un laboratorio

```
get '/HorarioLab' do
  @opcion_menu = 1

  @consultar = true
  @activa = true

  if pantalla_activa?(30)
    @activa = true
    if permiso?(30, "consulta")
      @consultar = true
    end
  end
end

if @activa
  if @consultar
    @pes = Programa_Educativo.all :order => :clave_pe, :activo =>true
    @periodos = Periodo.all :order => :fecha_inicio, :activo =>true
    @labs = Laboratorio.all :order => :clave_lab
    @grup = Grupos.all :order => :clave_grup
    erb :horario_lab
  else
    flash[:error] = "Ud. No tiene permiso de consulta para esa pantalla"
    redirect '/'
  end
else
  flash[:error] = "Esa pantalla no está; activa en este momento, intente
mã;s tarde"
  redirect '/'
end
end
```

Anexo 2

Código para llenar los select del formulario y desbloquearlos

```
$(document).ready(function() {
  //Funcion para ptc
  $("#seletc_ptc").click(function() {

    $("#ptc").load("/g_ptc/"+$("#p_educativo
option:selected").val(),function(responseTxt,statusTxt,xhr) {
      /* if(statusTxt=="success") alert("External content loaded
successfully!"); */
      if(statusTxt=="error")
```

```

        alert("Error: "+xhr.status+": "+xhr.statusText);
    });
});
//funcion para PA
$("#seletc_prof").change(function(){
    var str_esc=escape($("#periodo option:selected").val());
    $("#s_pa").load("/g_pa/"+$("#p_educativo
option:selected").val()+"/"+str_esc,function(responseTxt,statusTxt,xhr){
        /* if(statusTxt=="success") alert("External content loaded
successfully!"); */
        if(statusTxt=="error")
            alert("Error: "+xhr.status+": "+xhr.statusText);
    });
});
//funcion para asignaturas
$("#p_educativo").change(function(){
    $("#asignatura").load("/g_asignatura/"+$("#p_educativo
option:selected").val(),function(responseTxt,statusTxt,xhr){
        /* if(statusTxt=="success") alert("External content loaded
successfully!"); */
        if(statusTxt=="error")
            alert("Error: "+xhr.status+": "+xhr.statusText);
    });
});
// funcion para generar el horario
$("#muestra").click(function(){
    var str=escape($("#periodo option:selected").val());
    $("#mu_tabla").load("/g_tabla/"+$("#p_educativo
option:selected").val()+"/"+$("#laboratorio
option:selected").val()+"/"+str,function(responseTxt,statusTxt,xhr){
        /* if(statusTxt=="success") alert("External content loaded
successfully!"); */
        if(statusTxt=="error")
            alert("Error: "+xhr.status+": "+xhr.statusText);
    });
});
});
function bloqueaall(){
    document.getElementById("p_educativo").disabled = true ;
    document.getElementById("periodo").disabled = true ;
    document.getElementById("seletc_ptc").disabled = true ;
    document.getElementById("seletc_prof").disabled = true ;
    document.getElementById("ptc").disabled = true ;
    document.getElementById("s_pa").disabled = true ;

```

```

document.getElementById("asignatura").disabled = true ;
document.getElementById("laboratorio").disabled = true ;
document.getElementById("grupo").disabled = true ;
}

function enableall(){
    document.getElementById("p_educativo").disabled = false ;
    document.getElementById("periodo").disabled = false ;
    document.getElementById("seletc_ptc").disabled = false ;
    document.getElementById("seletc_prof").disabled = false ;
    document.getElementById("ptc").disabled = false ;
    document.getElementById("s_pa").disabled = false ;
    document.getElementById("asignatura").disabled = false ;
    document.getElementById("laboratorio").disabled = false ;
    document.getElementById("grupo").disabled = false ;
}

function enableperiodo() {
    document.getElementById("periodo").disabled = false ;
}

function enableprofesor() {
    document.getElementById("seletc_ptc").disabled = false ;
    document.getElementById("seletc_prof").disabled = false ;
}

function enableasignatura(){
    document.getElementById("asignatura").disabled = false
}

function enablelaboratorio(){
    document.getElementById("laboratorio").disabled= false
}

function enablegrupo(){
    document.getElementById("grupo").disabled = false
}

function enablemostrar(){
    document.getElementById("muestra").disabled = false
}

function mostrar(){

```

```
document.getElementById('oculto').style.display = 'block';
document.getElementById('link_mostrar').style.display = 'none';}
```

```
function mostrar_ptc(){
document.getElementById('ptc').style.display = 'inline'
document.getElementById('s_pa').style.display = 'none'
;}
```

```
function mostrar_pa(){
document.getElementById('s_pa').style.display = 'inline'
document.getElementById('ptc').style.display = 'none'
;}
```

Anexo 3

Código para enviar la información a los select

```
#get para obtener ptc dependiendo del programa educativo
get '/g_ptc/:clave_peducativo' do
if !campo_vacio(params[:clave_peducativo])
@ptcs= Ptc.all :order => :clave_ptc, :activo =>true, :clave_pe =>
params[:clave_peducativo]
$cadena = "<option selected value="">-- SELECCIONE --</option>"
@ptcs.each do |ptc|
$cadena+= "<option value=\"#{ptc.clave_ptc}\">#{ptc.nombre_ptc}
#{ptc.ap_ptc} #{ptc.am_ptc}</option>"
end
if request.xhr?
return $cadena
else
"<h1>Not an Ajax request!</h1>"
end
end
end

# get para obtener pa dependiendo del programa educativo
get '/g_pa/:clave_peducativo/:periodo_clave' do
@profe_asig = Profesor_Asignatura.all :order => :clave_pa, :activo =>true
@periodo_pa = Pa_Periodo.all :order => :clave_pa, :clave_pe =>
params[:clave_peducativo], :clave_periodo => params[:periodo_clave]
$cadena = "<option selected value=\"nothing\">-- SELECCIONE --</option>"
@profe_asig.each do |profe_asig|
@periodo_pa.each do |periodo_pa|
if periodo_pa.clave_pa == profe_asig.clave_pa
$cadena+= "<option
value=\"#{profe_asig.clave_pa}\">#{profe_asig.nombre_pa} #{profe_asig.ap_pa}
#{profe_asig.am_pa}</option>"
end
end
end
```

```

        end
    end
    if request.xhr?
        return $cadena
    else
        "<h1>Not an Ajax request! #{@url}</h1>"
    end
end

#get para obtener la asignatura dependiendo del programa educativo
get '/g_asignatura/:clave_peducativo' do
    #code...
    @mapa_curri = Mapa_curricular.all :order => :clave_mapa, :clave_pe =>
params[:clave_peducativo]
    @ma_detalle = Detalle_ma.all :order => :clave_mapa
    @asig = Asignaturas.all :order => :clave_asig
    $cadena = "<option selected value="">-- SELECCIONE --</option>"
    @mapa_curri.each do |mapa_curri|
        @ma_detalle.each do |ma_detalle|
            @asig.each do |asig|
                if mapa_curri.clave_mapa == ma_detalle.clave_mapa and
ma_detalle.clave_asig == asig.clave_asig
                    $cadena+= "<option
value=\"#{asig.clave_asig}\">#{asig.nom_asig}</option>"
                end
            end
        end
    end
end
    if request.xhr?
        return $cadena
    else
        "<h1>Not an Ajax request!</h1>"
    end
end
end

```

Anexo 4

Código para generar la tabla de horario

```

#obtiene el horario para los laboratorios
get '/g_tabla/:clave_peducativo/:clave_laboratorio/:periodo_clave' do
    @hors = Horas.all :order => :clave_hora
    @det = Detalle_Horario.all :order => :clave_detahor
    @hora_lab = Horario_Lab.all :order => :clave_horario, :clave_periodo =>
params[:periodo_clave], :clave_lab => params[:clave_laboratorio]

    $cadena=%q{<table>
        <tr>

```

```

        <th class="tabla"    ></th>
        <th class="tabla" WIDTH="110" ><h1>Lunes</h1></th>
        <th class="tabla" WIDTH="110" ><h1>Martes</h1></th>
        <th class="tabla" WIDTH="110" ><h1>Miércoles</h1></th>
        <th class="tabla" WIDTH="110" ><h1>Jueves</h1></th>
        <th class="tabla" WIDTH="110" ><h1>Viernes</h1></th>
        <th class="tabla" WIDTH="110" ><h1>Sábado</h1></th>
    </tr>}
@fila = 0
$hora_indice = $hora_inicio
$hora_valor = $hora_inicio
vec1 = []
vec2 = []
indice = 0
@hors.each do |hors|
    vec1[indice] = hors.hora
    vec2[indice] = hors.clave_hora
    indice= indice+1
end
indice = 0
c = 0
for i in 1..$horas do
    if (@fila%2 == 0)
        @tipo = "r0"
    else
        @tipo = "r1"
    end
    @fila += 1

    $cadena+= "<tr class=\"#{@tipo}\" align=\"center\" ><th
class=\"tabla\"><h2>#{vec1[indice]} - #{vec1[indice+1]}</h2></th>"
    for b in 1..6
        k=0
        @det.each do |det|
            @hora_lab.each do |hora_lab|
                if det.clave_dia == b and det.hora_inicio == vec2[indice] and
det.hora_fin == vec2[indice+1] and det.clave_horario == hora_lab.clave_horario
                    $cadena+= %q{<td><input type="checkbox" disabled="true" checked>
</td>}
                k=1
            end
        end
    end
    if k==0
        $cadena+= "<td ><input type=\"checkbox\" name=\"check[#{c}]\"
value=\"#{b}, #{vec2[indice]}, #{vec2[indice+1]}\"></td>"
    end
end

```

```

        end
        c += 1
    end
    indice+= 1
    $hora_valor+=1
end

$cadena+= "</tr></table>"
return $cadena
end

```

Anexo 4

Código para guardar la información del horario y el detalle

```

# Guardar informacion de Horario laboratorio
post '/regHorario' do

    if params['check'] ||= false and !campo_vacio(params[:p_educativo]) and
!campo_vacio(params[:periodo]) and !campo_vacio(params[:laboratorio]) and
!campo_vacio(params[:grupo]) and !campo_vacio(params[:asignatura]) and (
!campo_vacio(params[:ptc]) || !campo_vacio(params[:s_pa]))
        # Calcula el próximo valor para el campo clave
        @nueva_clave = Horario_Lab.count > 0 ?
Horario_Lab.max(:clave_horario)+1 : 1
        if !campo_vacio(params[:ptc])
            @lab_horario = Horario_Lab.new(:clave_horario => @nueva_clave,
:clave_pe => params[:p_educativo].strip.upcase, :clave_periodo =>
params[:periodo].strip.upcase, :clave_ptc => params[:ptc], :clave_lab =>
params[:laboratorio].strip.upcase, :clave_grup => params[:grupo].strip.upcase,
:clave_asig => params[:asignatura])

        else
            @lab_horario = Horario_Lab.new(:clave_horario => @nueva_clave,
:clave_pe => params[:p_educativo].strip.upcase, :clave_periodo =>
params[:periodo].strip.upcase, :clave_pa => params[:s_pa], :clave_lab =>
params[:laboratorio].strip.upcase, :clave_grup => params[:grupo].strip.upcase,
:clave_asig => params[:asignatura])
        end

        if params['check'] == false
            flash[:error] = "No selecciono el día, por favor verifique!"
            redirect '/HorarioLab'
        end

        if @lab_horario.save
            x = 0

```



```

j = 1
z = 2
token1 = []
  params['check'].each do |e|
    token1 = e[j].split(", ")
    @nueva_clavedeta = Detalle_Horario.count > 0 ?
Detalle_Horario.max(:clave_detahor)+1 : 1
    @horario_deta = Detalle_Horario.new(:clave_detahor =>
@nueva_clavedeta, :clave_horario => @nueva_clave, :clave_dia => token1[x],
:hora_inicio => token1[j], :hora_fin => token1[z])
    @horario_deta.save
  end

  flash[:notice] = "Registro guardado correctamente"
  redirect '/HorarioLab'
else
  flash[:error] = @lab_horario.errors.to_a
  redirect '/HorarioLab'
end

if @deta_horario.save
  flash[:notice] = "Registro guardado correctamente"
  redirect '/HorarioLab'
else
  flash[:error] = @lab_horario.errors.to_a
  redirect '/HorarioLab'
end

else
  flash[:error] = "Dejó vacío un campo requerido, por favor
verifique!"
  redirect '/HorarioLab'
end

end
end

```

Bibliografía

- [1] URL: <http://www.uppuebla.edu.mx/> Página Principal de la Universidad politécnica de Puebla, en ella se puede consultar información acerca de las carreras y servicios que ofrece.
- [2] URL: http://www.transparencia.puebla.gob.mx/index.php?option=com_k2&view=itemlist&layout=category&task=category&id=3636&Itemid=3181 Página donde se puede consultar la misión visión y funciones de la Uppue.
- [3] URL: http://www.transparencia.puebla.gob.mx/index.php?option=com_content&view=frontpage&Itemid=59 Página donde se puede consultar el organigrama de la Uppue.
- [4] URL: <https://www.debian.org/> Página donde se puede descargar consultar información acerca del S.O Debían
- [5] URL: <http://www.grihotools.udl.cat/mpiua/fases-mpiua/> Página donde se puede consultar información acerca de la metodología MPIu+a.
- [6] URL: <http://www.sinatrarb.com/> Página donde se puede consultar documentación acerca de Sinatra.
- [7] URL: <https://www.postgresql.org/docs/9.2/static/app-psql.html> Página donde se puede consultar documentación acerca de postgresql.
- [8] Alan Harris, Konstantin haase, "Sinatra Up and Running" Editorial O'RELLY