

# **UNIVERSIDAD ORT**

## **FACULTAD DE INGENIERÍA**

### **Obligatorio**

### **Taller de Programación**



Romina Areco – 206874



Sebastián González – 208938

<b>Manual de Instalación y Configuración:</b>	<b>4</b>
<b>Manual de Uso:</b>	<b>9</b>
Landing Page:	9
Registro:	10
Login:	11
Ingreso de Usuario:	11
Ingreso de Cliente:	12
Ingreso de Administrador:	12
Logout:	13
Alta de Instructor:	13
Aprobación de Cliente:	14
Confirmación de Libreta:	14
Listado de Clases:	15
<b>Base de Datos:</b>	<b>16</b>
<b>Fuentes:</b>	<b>16</b>
Consultas:	16
Approve User:	16
Book Class:	17
Confirm License:	19
Information:	20
Information Get:	22
Login:	24
Logout:	25
Register Instructor:	25
Register User:	27
PHP:	29
Index:	29
Templates:	31
Confirm Driving License:	31
Index:	32
Information Page:	33
Login Menu:	35
Menu Admin:	36
Menu User:	37
Register Instructor:	38
Register User:	40

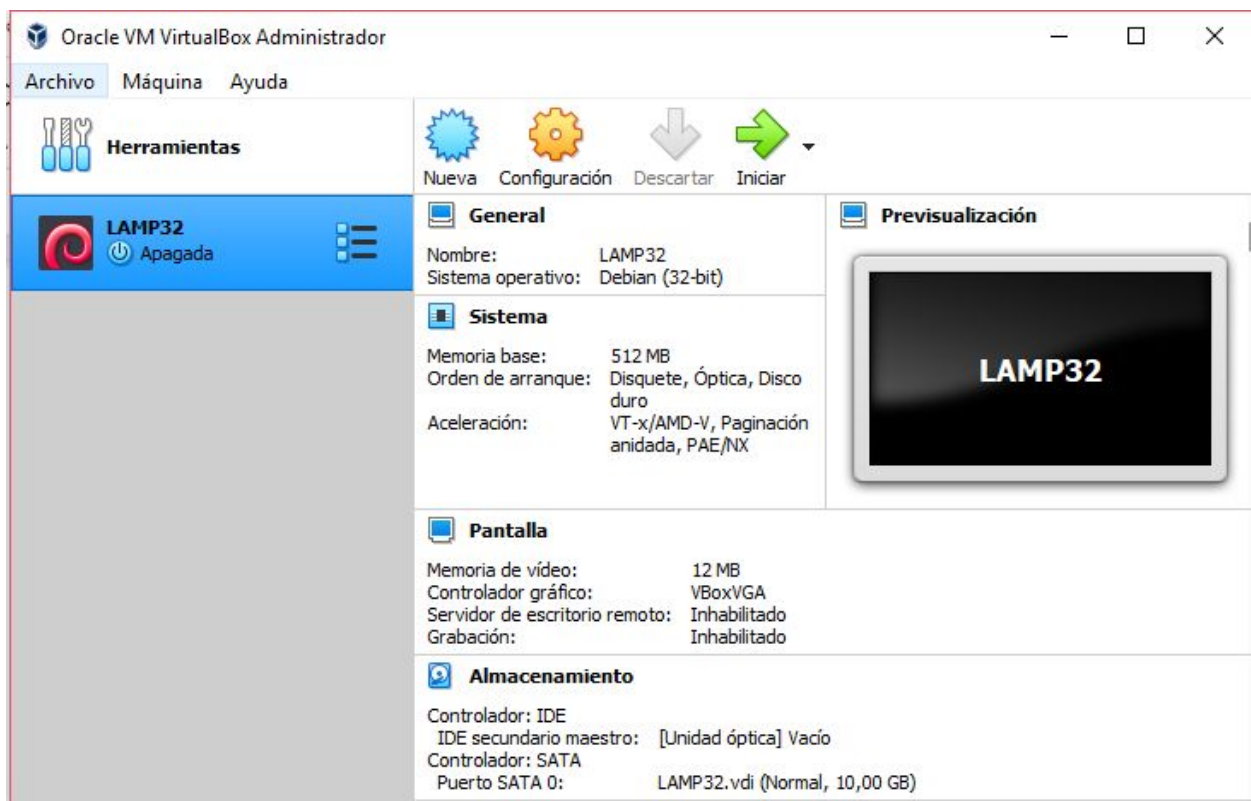
Approve Client:	42
Book Class:	42
Class List:	44
Config:	48
Data Base:	48
Smarty:	48
Javascript:	49

# Manual de Instalación y Configuración:

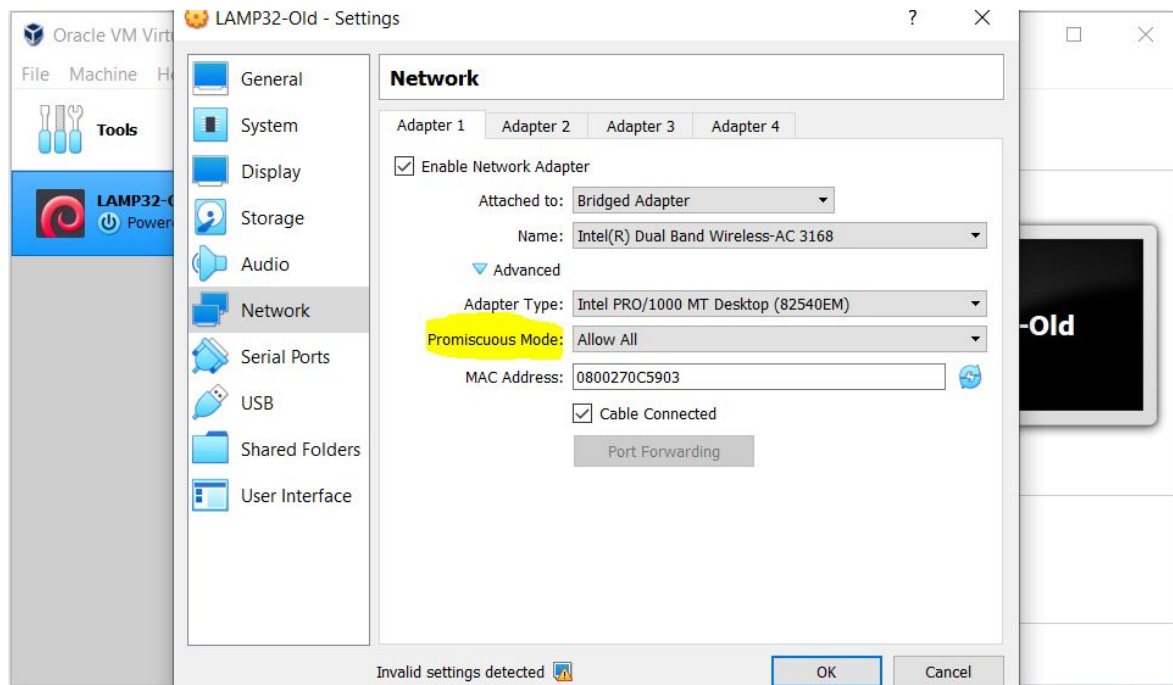
Descargar e instalar VirtualBox de: <https://www.virtualbox.org/wiki/Downloads> En función del sistema operativo que se tenga. Proceso de instalación:

<https://www.virtualbox.org/manual/ch01.html#introinstalling>

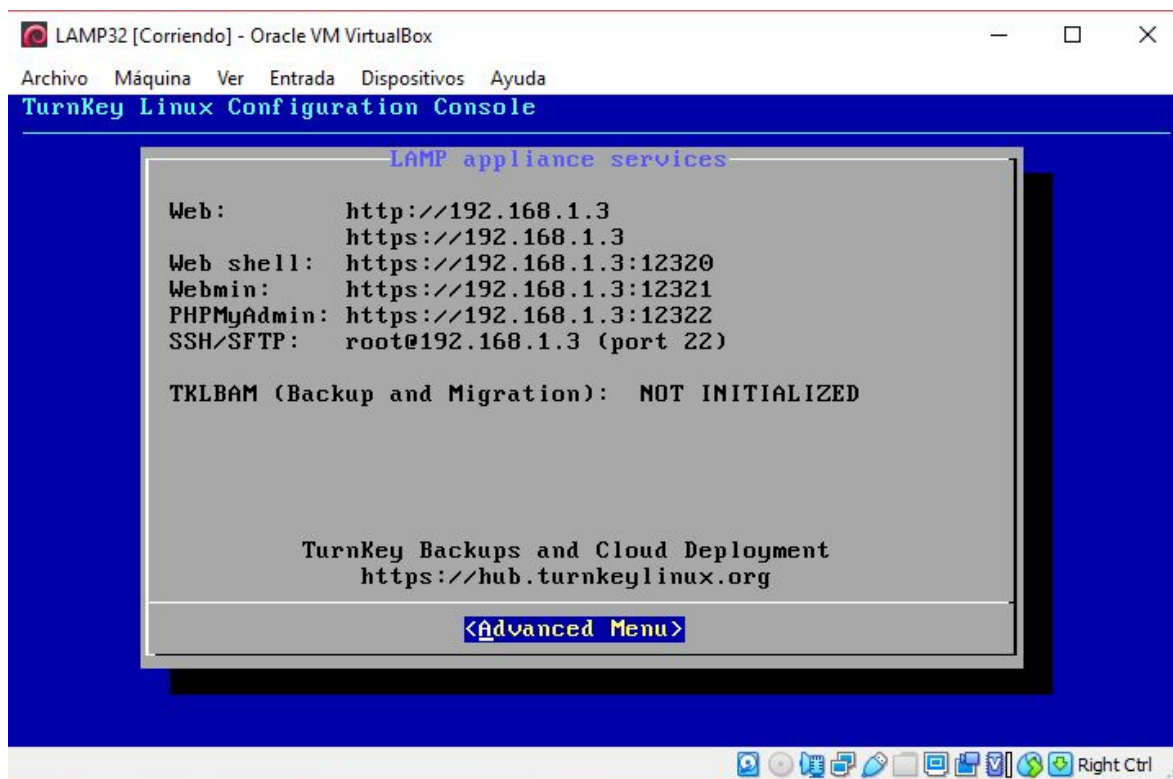
Descomprimir en una carpeta el comprimido de la máquina virtual utilizado en clase y en VirtualBox, dentro del menú Máquina, seleccionar la opción Agregar y buscar dentro de la carpeta extraída el archivo LAMP.vbox. A partir de este momento ya disponemos de una máquina virtual llamada LAMP en el inventario.



Se debe validar que se tenga el modo Promiscuo activo (dentro de Configuración, Redes, Avanzado) como se observa en la siguiente imagen:



Mediante el botón Iniciar sobre la máquina LAMP, iniciamos la misma y esta finaliza cuando vemos en pantalla los datos de acceso a la misma.



Luego, chequeamos la IP que nos muestra en pantalla y accedemos a través del navegador.

En el ejemplo de la imagen seria a traves accedemos a traves de <https://192.168.1.3/>  
Allí se nos presenta la siguiente pantalla

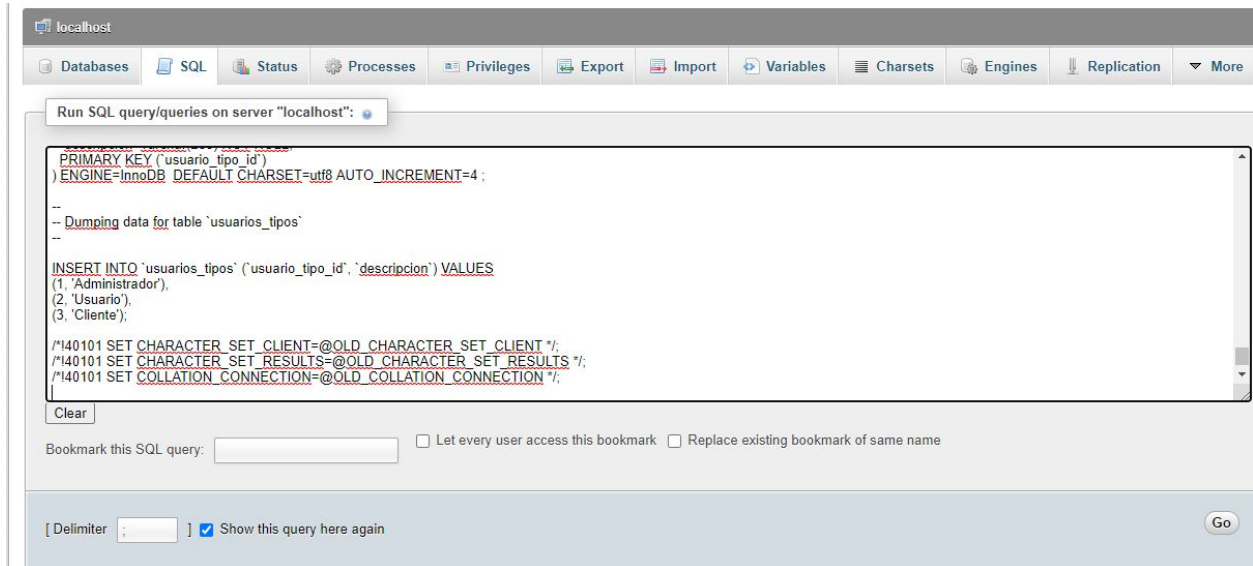


### Instalación de base de datos

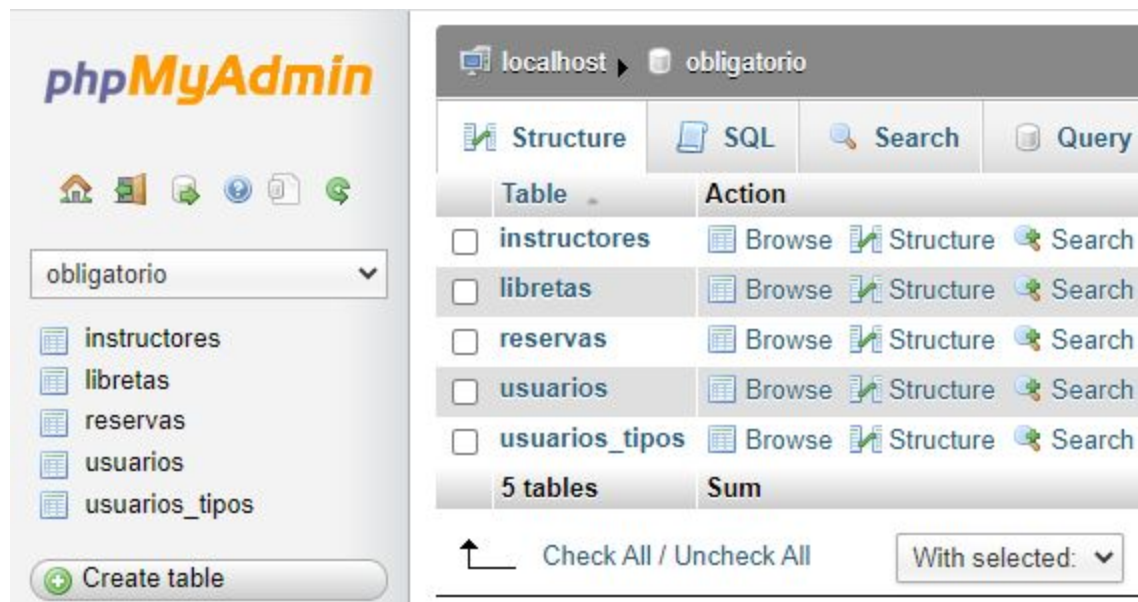
Para la instalación de base de datos, debemos realizar click en PHPMyAdmin e ingresamos las siguientes credenciales:

- Usuario: root
- Password: root

Allí, una vez adentro, nos dirigimos a la pestaña SQL, pegamos el contenido del script adjunto, y clickeamos en el botón go.



Luego podemos verificar en las bases de datos, que se ha creado correctamente la base obligatorio con toda su estructura

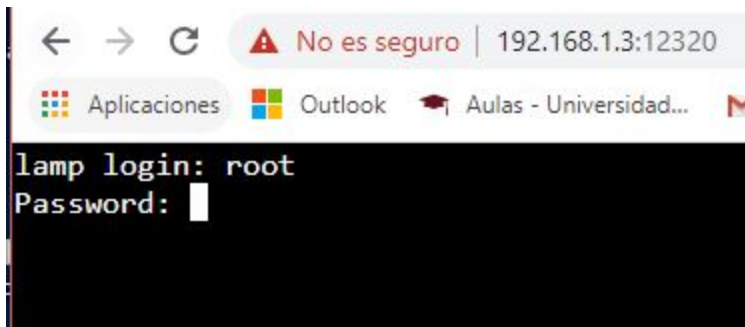


## Instalación del aplicativo

Primero debemos llevar el código al ambiente. Para eso es requerido contar con un cliente SFTP encargado de llevar el código fuente a la VM.

Debemos conectarnos a la misma ip con la que veníamos trabajando. En el caso de los ejemplos, es a la 192.168.1.3. Allí debemos dirigirnos hacia la carpeta var/www y copiar la carpeta.

Una vez copiado, debemos asegurarnos de dar permisos para acceder. Para eso, volvemos al menú inicial escribiendo en el browser la ip (192.168.1.3) y seleccionar la opción Web Shell.



Una vez dentro nos pedirá usuario y contraseña. Deberán ingresar root en ambos. Luego de validado les mostrara la siguiente imagen.

```
lamp login: root
Password:
Last login: Mon Aug 17 22:48:54 UTC 2020 from 192.168.1.6 on pts/2
Linux lamp 3.2.0-4-686-pae #1 SMP Debian 3.2.51-1 i686
Welcome to Lamp, TurnKey Linux 13.0 / Debian 7.2 Wheezy

System information (as of Wed Aug 19 22:57:51 2020)

System load:  0.01          Memory usage: 36%
Processes:   85            Swap usage:   0%
Usage of /:   9.8% of 8.15GB IP address for eth0: 192.168.1.3

TKLBAM (Backup and Migration): NOT INITIALIZED

To initialize TKLBAM, run the "tklbam-init" command to link this
system to your TurnKey Hub account. For details see the man page or
go to:

    http://www.turnkeylinux.org/tklbam

You have mail.
root@lamp ~#
```

Una vez allí, deberán ejecutar los comandos tal como se muestran a continuación:

Para posicionarnos en la carpeta donde se encuentra el proyecto.

```
cd ../var/www
```

Con **ls** pueden listar todas las carpetas para chequear el nombre de su proyecto, y finalmente ejecutar **cd [nombreProyecto]** para posicionarse en él.



```
root@lamp /var/www# ls
ObligatorioTaller cgi-bin css images index.php js phpinfo.php
root@lamp /var/www# cd ObligatorioTaller
```

Una vez posicionados deberán ejecutar el siguiente comando para dar permisos full sobre el directorio.

```
chmod -R 777 ./
```

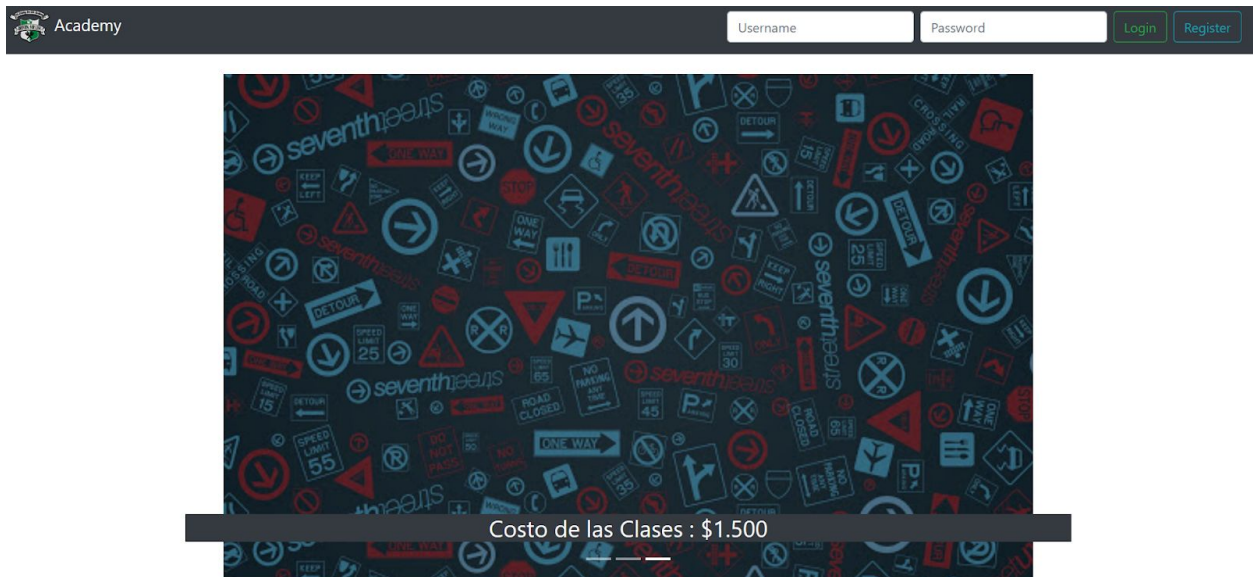
Una vez realizado esto, pueden probar desde el navegador acceder al sitio a través de la ip\nombreProyecto que hayan ingresado. En nuestro caso es a traves de

<http://192.168.1.3/ObligatorioTaller/>

## Manual de Uso:

A continuación se mencionan las diferentes pantallas que brinda el sistema, con sus respectivas funcionalidades.

Landing Page:



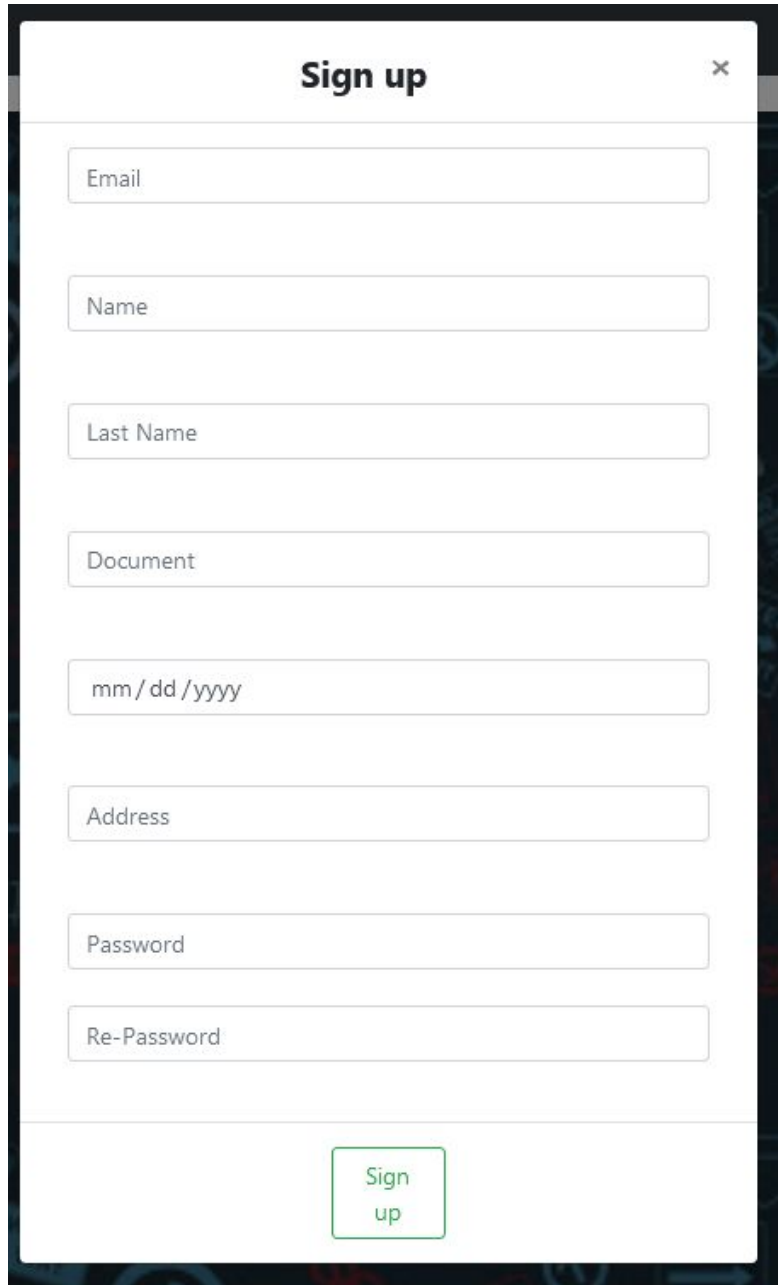
Esta es la pantalla que se nos presenta al ingresar al sistema, la cual posee una slide con información sobre la academia como:

- Costos de la clase.
- Cantidad de clientes que ya reservaron clases y cantidad de libretas obtenidas.
- Calendario con información sobre los horarios.
  - Los días que no poseen ninguna etiqueta son aquellos los cuales no poseen reservas.
  - Las etiquetas verdes indican los días que poseen pocas reservas.

- Las etiquetas amarillas indican los días que se encuentran reservados en un 50% o más.
- Las etiquetas rojas indican los días los cuales no poseen horarios disponibles.

Además de brindarnos información sobre la academia, esta página nos permite registrarnos o realizar el login en caso de que ya poseamos usuarios.

Registro:

A screenshot of a 'Sign up' form popup. The form is white with a black border and a close button (X) in the top right corner. It contains eight input fields: Email, Name, Last Name, Document, mm/dd/yyyy (for date), Address, Password, and Re-Password. At the bottom center is a green 'Sign up' button.

**Sign up** ×

Email

Name

Last Name

Document

mm / dd / yyyy

Address

Password

Re-Password

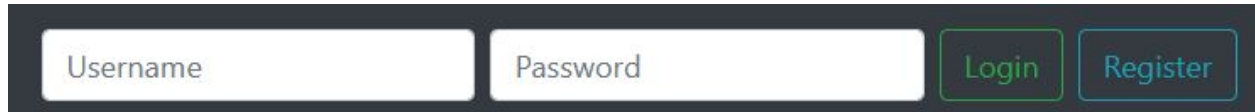
Sign up

El registro es un Popup que solicita la información necesaria para registrarnos a la academia. También luego de realizar el registro, realiza el login automáticamente.

Para este caso, solo el campo de dirección no es requerido y se realizan las siguientes validaciones:

- Mayor de 18 años.
- Cedula numerica.
- Email unico.
- La contraseña indicada debe ser igual en Password y Re-Password.

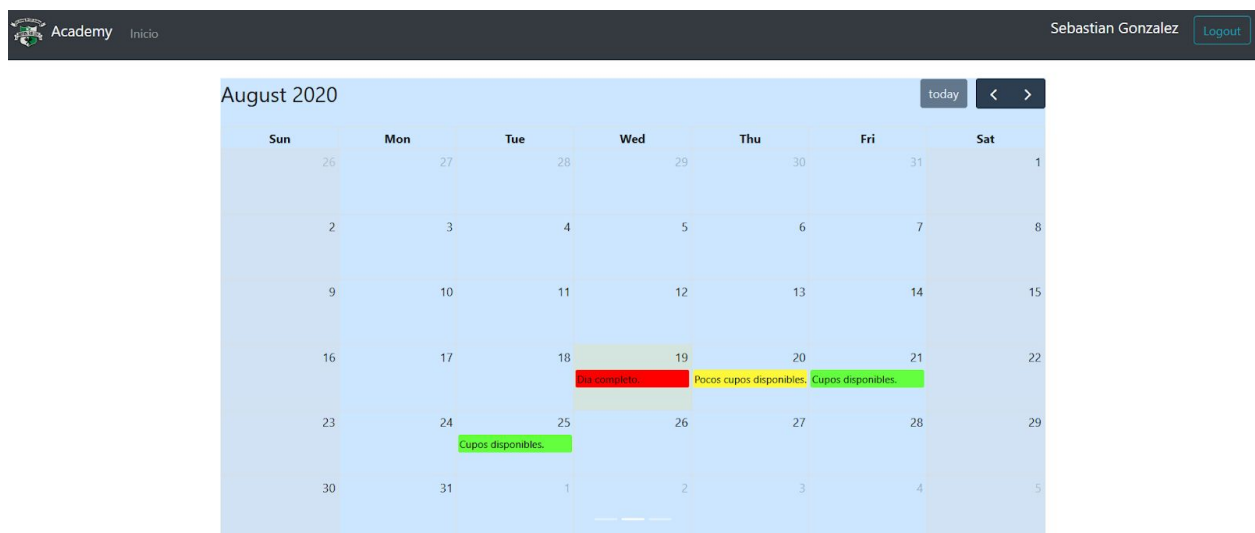
Login:



A dark-themed login and registration form. It features two input fields: 'Username' and 'Password'. To the right of these fields are two buttons: 'Login' (green border) and 'Register' (blue border).

Como mencionamos anteriormente, la landing page nos permite realizar el login a la pagina, ingresando mediante nuestro usuario (email) y contraseña.

Ingreso de Usuario:

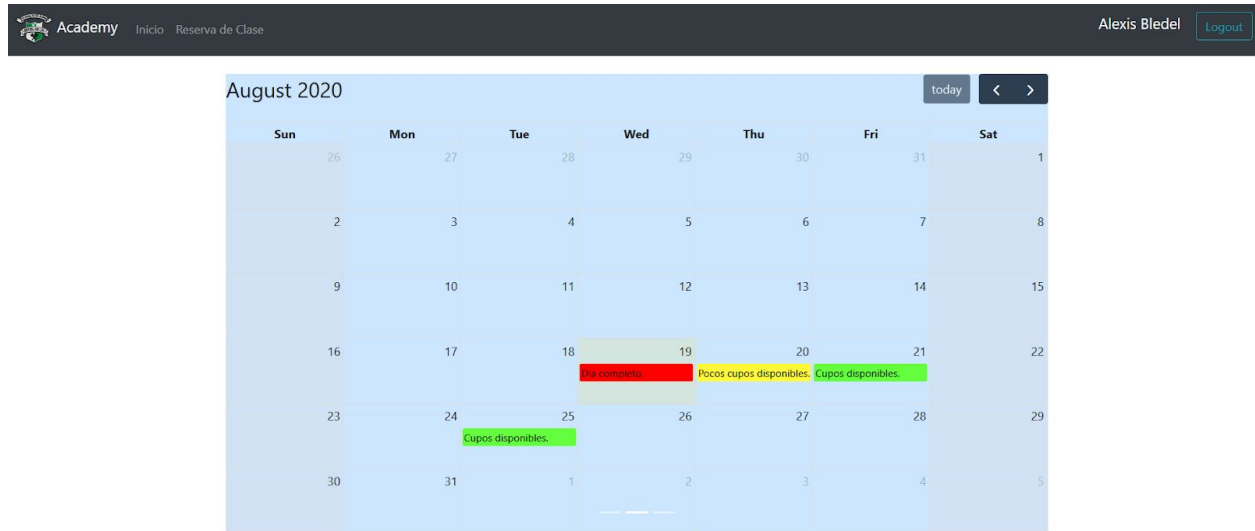


A screenshot of a web application interface. At the top, a dark header bar contains the 'Academy Inicio' logo on the left and the user name 'Sebastian Gonzalez' next to a 'Logout' button on the right. Below the header is a calendar for August 2020. The calendar grid shows dates from 26 to 5. Specific dates are highlighted with colored boxes and text: August 19 is red with 'No disponible', August 20 is yellow with 'Pocos cupos disponibles', August 21 is green with 'Cupos disponibles', August 24 is green with 'Cupos disponibles', and August 25 is green with 'Cupos disponibles'. Navigation arrows and a 'today' button are visible at the top right of the calendar.

Al logearnos a la página, la misma exhibirá nuestro Nombre y Apellido en la sección superior derecha, al lado del botón de Logout.

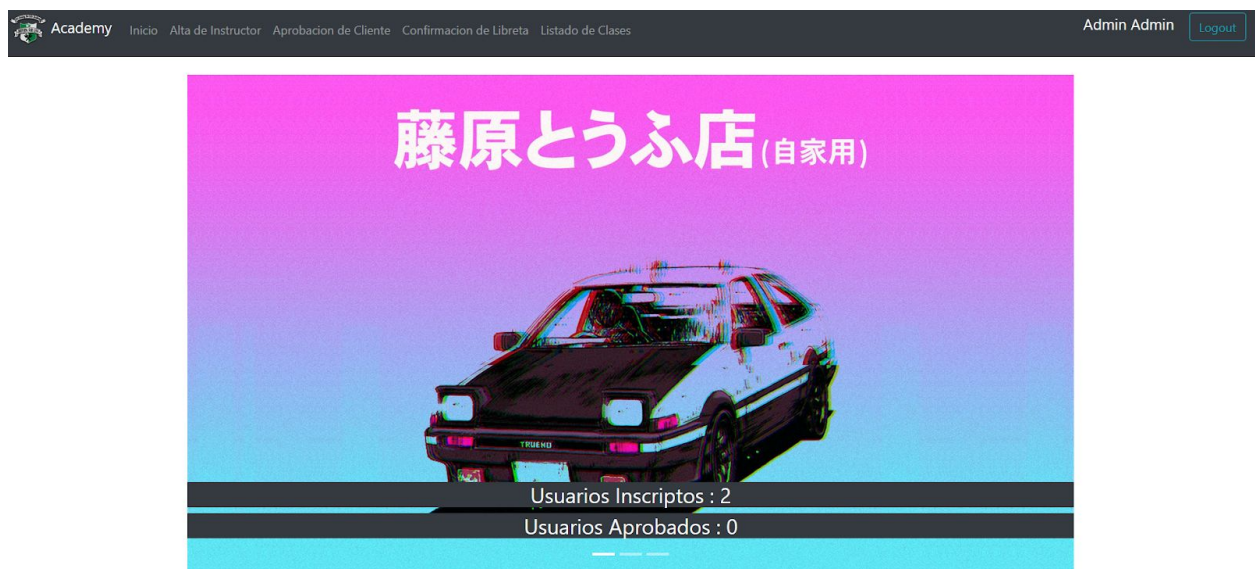
Para el caso de que los usuarios no hayan sido admitidos en la academia (no son clientes), no hay opciones a las cuales ingresar, solo se exhibe la Landing Page.

## Ingreso de Cliente:



La única diferencia que posee con el Login de Usuario es que a los Clientes (usuarios ya admitidos a la academia) se les permite Reservar clases.

## Ingreso de Administrador:



Los administradores poseen una gran variedad de acciones posibles, como lo son:

- Alta de un nuevo Instructor.
- Aprobación de Clientes.
- Confirmación de Libretas.
- Listado de Clases.

## Logout:

El botón de Logout nos permite cerrar la sección que se encuentra activa al momento, dirigiéndose nuevamente a la Landing Page.

## Alta de Instructor:

[Inicio](#) [Alta de Instructor](#) [Aprobacion de Cliente](#) [Confirmacion de Libreta](#) [Listado de Clases](#)

### Alta de instructor

Nombre	<input type="text"/>
Apellido	<input type="text"/>
Cedula	<input type="text"/>
Fecha de nacimiento	<input type="text"/>
Fecha de vencimiento de libreta	<input type="text"/>

El alta de Instructor, como su nombre lo indica, nos permite ingresar un nuevo instructor a la Base de Datos.

Para este caso, todos los campos son requeridos y se realizan las siguientes validaciones:

- Mayor de 18 años.
- Cedula numerica.
- Libreta Vigente.

## Aprobación de Cliente:

Alta de Instructor Aprobacion de Cliente Confirmacion de Libreta Listado de Clases

### Usuarios inscriptos para clase de manejo

<input type="radio"/>	CI	21321312
	Nombre	Maria Stevez
<input type="radio"/>	CI	4819638
	Nombre	Sebastian Gonzalez
<input type="radio"/>	CI	15862354
	Nombre	Charlie Damelio

Aprobar clientes

Esta pantalla nos brinda una lista de los usuarios que se encuentran inscriptos en la Academia pero aún no fueron aceptados como Clientes.

Al seleccionar a uno de los mismos y presionar el botón, se acepta dicho usuario como Cliente.

## Confirmación de Libreta:

Alta de Instructor Aprobacion de Cliente Confirmacion de Libreta Listado de Clases

### Usuarios con mas de 15 clases


<input type="radio"/>	CI	7522152
	Nombre	Alexis BledeI
<input type="radio"/>	CI	15862354
	Nombre	Charlie Damelio

Confirmar libretas

Esta pantalla nos brinda un listado de todos aquellos Clientes los cuales a la fecha tomaron más de 15 clases.

Al seleccionar uno de los mismos y presionar el botón, se le otorgará la libreta de conducir al mismo.

## Listado de Clases:

 Academy

[Inicio](#) [Alta de Instructor](#) [Aprobación de Cliente](#) [Confirmación de Libreta](#) [Listado de Clases](#)

Admin Admin [Logout](#)

### Listado de clases por instructor

Instructor

Jose Fernandez

Fecha

20/08/2020

Aplicar filtro

Remover filtro

Exportar lista a PDF

#### Listado de Clases

Fecha	2020-08-20
Turno	7
Id Instructor	13
Nombre	Alexis Bledel
Direccion	
Fecha	2020-08-20
Turno	8
Id Instructor	13

En esta pantalla se exhibe información sobre las reservas, filtrando la misma a partir del Instructor y Fecha seleccionado.

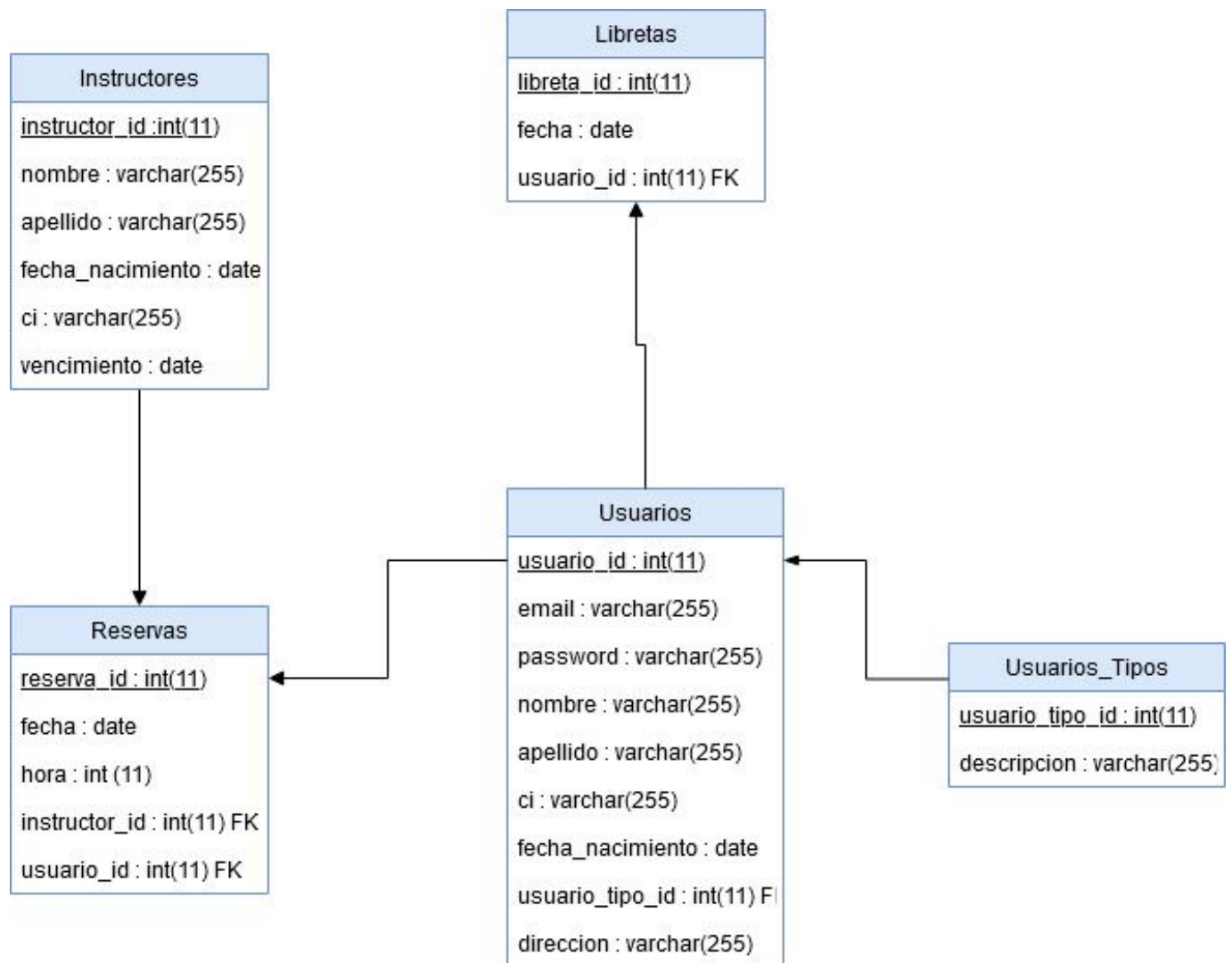
La información a visualizar corresponde a:

- Fecha.
- Turno (siendo 7 el de 7:00 - 8:00 y así sucesivamente).
- Id Instructor.
- Nombre Completo del Cliente.
- Direccion del Cliente.

Si deseamos visualizar las clases de otro instructor u otra fecha, solo basta con cambiar el filtro y presionar “Aplicar Filtro”.

También se nos brinda la opción de Exportar a PDF la información que se exhibe en la parte inferior.

## Base de Datos:



## Fuentes:

## Consultas:

Approve User:

<?php



```

session_start();
include("../lib/Config/config_SMARTY.php");

$doc = $_POST["document"];

unset($_POST["document"]);

$conn = new ConexionBD(MOTOR, SERVIDOR, BASE, USUARIO, CLAVE);
$conn->conectar();

if ($conn) {
    $sql = "UPDATE usuarios SET usuario_tipo_id = :type WHERE ci = :document";

    $parametros = array();
    $parametros[0] = array("type", 3, "int");
    $parametros[1] = array("document", $doc, "string");

    $result = $conn->consulta($sql, $parametros);

    if ($result) {
        $_SESSION["successMessage"] = "Usuario actualizado con exito.";
    } else {
        $_SESSION["errorMessage"] = "Error al actualizar usuario";
    }
} else {
    $_SESSION["errorMessage"] = "Error de Conexion";
}

header('location: ../index.php');
$conn->desconectar();

```

## Book Class:

<?php

```

session_start();
include("../lib/Config/config_SMARTY.php");

$date = $_POST["date"];
$time = $_POST["time"];

$instructor = $_POST["instructor"];

```

```

$dateName = date("w", strtotime($date));

unset($_POST["date"]);
unset($_POST["time"]);
unset($_POST["instructor"]);

$conn = new ConexionBD(MOTOR, SERVIDOR, BASE, USUARIO, CLAVE);
$conn->conectar();

if ($conn) {
    if ($dateName == 0 || $dateName == 6) {
        $_SESSION["errorMessage"] = "No se dictan clases los fines de semana.";
    } else {

        if (strtotime($date) < strtotime(date("Y-m-d"))) {
            $_SESSION["errorMessage"] = "La fecha de reserva no puede ser anterior al día actual.";
        } else {
            $sql = "SELECT * FROM reservas WHERE fecha = :date AND hora = :time AND instructor_id =
:inst";

            $parametros = array();
            $parametros[0] = array("date", $date, "string");
            $parametros[1] = array("time", intval($time), "int");
            $parametros[2] = array("inst", intval($instructor), "int");

            $result = $conn->consulta($sql, $parametros);

            if (!$result || $conn->cantidadRegistros() > 0) {
                $_SESSION["errorMessage"] = "El instructor ya posee una reserva para esa fecha.";
            } else {
                $sql = "INSERT INTO reservas (fecha, hora, instructor_id, usuario_id) ";
                $sql .= "VALUES (:date, :hour, :instructor, :user)";

                $parametros = array();
                $parametros[0] = array("date", $date, "string");
                $parametros[1] = array("hour", intval($time), "int");
                $parametros[2] = array("instructor", intval($instructor), "int");
                $parametros[3] = array("user", intval($_SESSION['userLogged']['usuario_id']), "int");

                $result = $conn->consulta($sql, $parametros);

                if($result)
                    $_SESSION["successMessage"] = "Reserva almacenada con éxito.";
            }
        }
    }
}

```

```

        else
            $_SESSION["errorMessage"] = "Error al realizar la reserva.";
        }
    }
} else {
    $_SESSION["errorMessage"] = "Error de Conexion";
}

//header('location: ../../index.php');
$conn->desconectar();

```

## Confirm License:

```

<?php

session_start();
include("../lib/Config/config_SMARTY.php");

$doc = $_POST["document"];

unset($_POST["document"]);

$conn = new ConexionBD(MOTOR, SERVIDOR, BASE, USUARIO, CLAVE);
$conn->conectar();

if ($conn) {

    $sql = "SELECT usuario_id as id FROM usuarios WHERE ci = :document";
    $parametros = array();
    $parametros[0] = array("document", $doc, "string");

    $result = $conn->consulta($sql, $parametros);
    $user = $conn->siguienteRegistro();

    $sql = "INSERT INTO libretas (fecha, usuario_id) ";
    $sql .= "VALUES (:todayDate, :userId)";

    $parametros = array();
    $parametros[0] = array("todayDate", date("Y-m-d"), "string");
    $parametros[1] = array("userId", intval($user["id"]), "int");
}

```

```

$result = $conn->consulta($sql, $parametros);

if ($result) {
    $_SESSION["successMessage"] = "Licencia otorgada con exito.";
} else {
    $_SESSION["errorMessage"] = "Error al otorgada licencia";
}
} else {
    $_SESSION["errorMessage"] = "Error de Conexion";
}

header('location: ../../index.php');
$conn->desconectar();

```

## Information:

```
<?php
```

```
session_start();
```

```
include_once("../lib/Config/config_SMARTY.php");
```

```

$conn = new ConexionBD(MOTOR, SERVIDOR, BASE, USUARIO, CLAVE);
$conn->conectar();

```

```
if ($conn) {
```

```
    //CANTIDAD DE USUARIOS CURSANDO
```

```
    $sql = "SELECT count(distinct(usuario_id)) reservs FROM reservas WHERE fecha >= :todayDate";
```

```
    $parametros = array();
```

```
    $parametros[0] = array("todayDate", date("Y-m-d"), "string");
```

```
    $result = $conn->consulta($sql, $parametros);
```

```
    if ($result)
```

```
        $_SESSION["countRegisteredUsers"] = $conn->siguienteRegistro()['reservs'];
```

```
    else
```

```
        unset($_SESSION["countRegisteredUsers"]);
```

```

//CANTIDAD DE USUARIOS QUE APROBARON
$sql = "SELECT count(*) as libs FROM libretas ";

$result = $conn->consulta($sql, $parametros);

if ($result)
    $_SESSION["countApprovedUsers"] = $conn->siguienteRegistro()["libs"];
else
    unset($_SESSION["countApprovedUsers"]);

//CANTIDAD CLIENTES A APROVAR
$sql = "SELECT * FROM usuarios ";
$sql .= "WHERE usuario_tipo_id = :userType ";

$parametros = array();
$parametros[0] = array("userType", 2, "int");

$result = $conn->consulta($sql, $parametros);

if ($result)
    $_SESSION["pendingClients"] = $conn->restantesRegistros();
else
    unset($_SESSION["pendingClients"]);

//LICENCIAS PENDIENTES
$sql = "SELECT u.* FROM reservas r ";
$sql .= "INNER JOIN usuarios u ON u.usuario_id = r.usuario_id ";
$sql .= "WHERE r.usuario_id NOT IN ( SELECT usuario_id FROM libretas ) ";
$sql .= "GROUP BY u.usuario_id ";
$sql .= "HAVING COUNT(*) >= :countClass ";

$parametros = array();
$parametros[0] = array("countClass", 15, "int");

$result = $conn->consulta($sql, $parametros);

if ($result)
    $_SESSION["pendingLicences"] = $conn->restantesRegistros();
else
    unset($_SESSION["pendingLicences"]);

//INSTRUCTORES
$sql = "SELECT * FROM instructores ";

```

```

$parametros = array();

$result = $conn->consulta($sql, $parametros);

if ($result)
$_SESSION["allInstructors"] = $conn->restantesRegistros();
else
unset($_SESSION["allInstructors"]);

//CLASES DE INSTRUCTORES
$sql = "SELECT * FROM reservas r ";
$sql .= "INNER JOIN usuarios u ON u.usuario_id = r.usuario_id ";
$sql .= "ORDER BY fecha, hora";

$result = $conn->consulta($sql);

if ($result)
$_SESSION["clasesUsuarios"] = $conn->restantesRegistros();
else
unset($_SESSION["clasesUsuarios"]);

} else {
    $_SESSION["errorMessage"] = "Error de Conexion";
}

$conn->desconectar();

```

## Information Get:

```
<?php
```

```
session_start();
```

```
date_default_timezone_set("America/Santiago");
```

```
include("../lib/Config/config_SMARTY.php");
```

```

$conn = new ConexionBD(MOTOR, SERVIDOR, BASE, USUARIO, CLAVE);
$conn->conectar();

if ($conn) {

    // Obtenemos el ultimo id insertado
    $sql = "SELECT fecha, COUNT( reserva_id ) total FROM reservas WHERE Fecha BETWEEN (
    SELECT DATE_ADD( DATE_ADD( LAST_DAY( NOW( ) ) , INTERVAL 1
    DAY ) , INTERVAL -1
    MONTH ) AS first_day
    )
    AND LAST_DAY( NOW( ) )
    GROUP BY fecha
    LIMIT 0 , 30";

    $datos = array();
    $result = $conn->consulta($sql, $datos);

    if ($result) {
        //guardamos en un array multidimensional todos los datos de la consulta
        $i = 0;

        while ($row = $conn->siguienteRegistro()) { // realizamos un ciclo while para traer los agenda
encontrados en la base de dato
            // Alimentamos el array con los datos de los agenda
            $datos[$i] = $row;
            $i++;
        }

        $instructores = 0;

        $sql = "SELECT count(*) total FROM instructores";
        $result = $conn->consulta($sql);
        if ($result) {
            $instructores = $conn->siguienteRegistro();
        }

        // Transformamos los datos encontrado en la BD al formato JSON
        $respuesta = array(
            "status" => "OK",
            "data" => $datos,
            "instructores" => $instructores
        );
    }
}

```

```

        echo json_encode($respuesta);
    } else {
        // Si no existen agenda mostramos este mensaje.
        $respuesta = array(
            "status" => "OK"
        );

        echo json_encode($respuesta);
    }

    $conn->desconectar();
} else {
    $respuesta = array(
        "status" => "ERROR"
    );

    echo json_encode($respuesta);
}

```

Login:

```
<?php
```

```
session_start();
```

```
include("../lib/Config/config_SMARTY.php");
```

```
$email = $_POST["user"];
```

```
$pass = $_POST["password"];
```

```
unset($_POST["user"]);
```

```
unset($_POST["password"]);
```

```
$conn = new ConexionBD(MOTOR, SERVIDOR, BASE, USUARIO, CLAVE);
```

```
$conn->conectar();
```

```
if ($conn) {
```

```
    $sql = "SELECT * FROM usuarios u ";
```

```
    $sql .= "INNER JOIN usuarios_tipos ut ON u.usuario_tipo_id = ut.usuario_tipo_id ";
```

```
    $sql .= "WHERE u.email = :mail AND u.password = :pass";
```



```

$parametros = array();
$parametros[0] = array("mail", $email, "string");
$parametros[1] = array("pass", md5($pass), "string");

$result = $conn->consulta($sql, $parametros);

if ($result && $conn->cantidadRegistros() > 0) {
    $_SESSION["userLogged"] = $conn->siguienteRegistro();
} else {
    unset($_SESSION["userLogged"]);
    $_SESSION["errorMessage"] = "Usuario / Contraseña incorrecto.";
}
} else {
    $_SESSION["errorMessage"] = "Error de Conexion";
}

header('location: ../../index.php');
$conn->desconectar();

```

Logout:

```

<?php

session_start();

session_destroy();

header('location: ../../index.php');

```

Register Instructor:

```

<?php

session_start();
include("../../lib/Config/config_SMARTY.php");

$formName = $_POST['name'];
$formLastName = $_POST['lastname'];
$formDueDate = $_POST['dueDate'];
$formDocument = $_POST['document'];
$formBirthdate = $_POST['birthDate'];

```

```

unset($_POST["name"]);
unset($_POST["lastname"]);
unset($_POST["dueDate"]);
unset($_POST["document"]);
unset($_POST["birthDate"]);

$conn = new ConexionBD(MOTOR, SERVIDOR, BASE, USUARIO, CLAVE);
$conn->conectar();

if ($conn) {

    if (!is_numeric($formDocument)) {
        $_SESSION['errorMessage'] = 'El documento ingresado no es valido.';
    } else {
        $age = floor(abs(strtotime($formBirthdate) - strtotime(date("Y-m-d")))) / (365 * 60 * 60 * 24));
        if ($age < 18) {
            $_SESSION['errorMessage'] = 'El Instructor debe tener al menos 18 años.';
        } else {
            if (strtotime($formBirthdate) > strtotime($formDueDate)) {
                $_SESSION['errorMessage'] = 'La fecha de Vencimiento no puede ser menor a la fecha de
Nacimiento.';
            } else {
                if (strtotime($formDueDate) < strtotime(date("Y-m-d"))) {
                    $_SESSION['errorMessage'] = 'La libreta se encuentra vencida.';
                } else {
                    $sql = "INSERT INTO instructores (nombre, apellido, fecha_nacimiento, ci, vencimiento)
";
                    $sql .= "VALUES (:name, :lastName, :birthdate, :document, :duedate)";

                    $parametros = array();
                    $parametros[0] = array("name", $formName, "string");
                    $parametros[1] = array("lastName", $formLastName, "string");
                    $parametros[2] = array("birthdate", $formBirthdate, "string");
                    $parametros[3] = array("document", $formDocument, "string");
                    $parametros[4] = array("duedate", $formDueDate, "string");

                    $result = $conn->consulta($sql, $parametros);
                    $_SESSION['successMessage'] = "Instructor creado con exito.";
                }
            }
        }
    }
} else {

```

```
        $_SESSION['errorMessage'] = "Error de conexion a la Base de Datos.";
    }
}
```

```
header('location: ../../index.php');
$conn->desconectar();
```

## Register User:

```
<?php
```

```
session_start();
```

```
include("../../lib/Config/config_SMARTY.php");
```

```
$formEmail = $_POST["formEmail"];
$formName = $_POST["formName"];
$formLastName = $_POST["formLastName"];
$formDocument = $_POST["formDocument"];
$formBirthdate = $_POST["formBirthdate"];
$formAddress = $_POST["formAddress"];
$formPass = $_POST["formPass"];
$formRePass = $_POST["formRePass"];
$formType = 2;
```

```
unset($_POST["formEmail"]);
unset($_POST["formName"]);
unset($_POST["formLastName"]);
unset($_POST["formDocument"]);
unset($_POST["formBirthdate"]);
unset($_POST["formAddress"]);
unset($_POST["formPass"]);
unset($_POST["formRePass"]);
```

```
$conn = new ConexionBD(MOTOR, SERVIDOR, BASE, USUARIO, CLAVE);
$conn->conectar();
```

```
if ($conn) {

    if (!is_numeric($formDocument)) {
        $_SESSION['errorMessage'] = 'El documento ingresado no es valido.';
    } else {
```

```

$age = floor(abs(strtotime($formBirthdate) - strtotime(date("Y-m-d")))) / (365 * 60 * 60 * 24));
if ($age < 18) {
$_SESSION['errorMessage'] = 'El usuario debe ser mayor a 18 años.';
} else {

if ($formPass != $formRePass) {
$_SESSION['errorMessage'] = 'Las contraseñas no coinciden.';
} else {
$sql = "SELECT * FROM usuarios WHERE email = :email";
$params = array();
$params[0] = array("email", $formEmail, "string");

$result = $conn->consulta($sql, $params);

if (!$result || $conn->cantidadRegistros() > 0) {
    $_SESSION['errorMessage'] = 'El email ya se encuentra registrado.';
} else {
    //Register
    $sql = "INSERT INTO usuarios (email, password, nombre, apellido, ci, direccion,
usuario_tipo_id, fecha_nacimiento) ";
    $sql .= "VALUES (:email, :password, :name, :lastName, :document, :address, :type,
:birthdate)";

    $params[0] = array("email", $formEmail, "string");
    $params[1] = array("password", md5($formPass), "string");
    $params[2] = array("name", $formName, "string");
    $params[3] = array("lastName", $formLastName, "string");
    $params[4] = array("document", $formDocument, "string");
    $params[5] = array("address", $formAddress, "string");
    $params[6] = array("type", $formType, "int");
    $params[7] = array("birthdate", $formBirthdate, "string");

    $result = $conn->consulta($sql, $params);

    //Login
    $sql = "SELECT * FROM usuarios u ";
    $sql .= "INNER JOIN usuarios_tipos ut ON u.usuario_tipo_id = ut.usuario_tipo_id ";
    $sql .= "WHERE u.email = :mail";

    $params = array();
    $params[0] = array("mail", $formEmail, "string");

    $result = $conn->consulta($sql, $params);

```

```

        $_SESSION["userLogged"] = $conn->siguienteRegistro();
        $_SESSION["successMessage"] = 'Usuario creado con exito.';
    }
}
}
}
} else {
    $_SESSION['errorMessage'] = 'Error de conexion a la Base de Datos.';
}

```

```

header('location: ../../index.php');
$conn->desconectar();

```

## PHP:

### Index:

```

<?php

```

```

session_start();
include_once("../lib/Config/config_SMARTY.php");

if (!isset($_SESSION['userLogged'])) {
    $smarty->assign("menu", "../loginMenu.tpl");
    $smarty->assign("pages", $unregisteredPages);
} else {
    $smarty->assign("fullName", $_SESSION['userLogged']['nombre'] . " " .
$_SESSION['userLogged']['apellido']);
    $isAdmin = $_SESSION['userLogged']['descripcion'] == 'Administrador';
    $isClient = $_SESSION['userLogged']['descripcion'] == 'Cliente';

    if ($isAdmin) {
        $smarty->assign("menu", "../menuAdmin.tpl");
        $smarty->assign("pages", $adminPages);
    } else {
        $smarty->assign("pages", $userPages);
        $smarty->assign("menu", "../menuUser.tpl");
        if($isClient)
            $smarty->assign("isClient", "");
    }
}

```

```

        else
            $smarty->assign("isClient", "hidden");
        }
    }

    include './DB/Queries/information.php';

    $smarty->assign("registerUsers", $_SESSION['countRegisteredUsers']);
    $smarty->assign("approvedUsers", $_SESSION['countApprovedUsers']);
    $smarty->assign("successHidden", 'hidden');
    $smarty->assign("errorHidden", 'hidden');

    if (isset($_SESSION['errorMessage'])) {
        $smarty->assign("errorMessage", $_SESSION['errorMessage']);
        $smarty->assign("errorHidden", "");
        unset($_SESSION['errorMessage']);
    }

    if (isset($_SESSION['successMessage'])) {
        $smarty->assign("successMessage", $_SESSION['successMessage']);
        $smarty->assign("successHidden", "");
        unset($_SESSION['successMessage']);
    }

    if (isset($_SESSION['pendingClients'])) {
        $smarty->assign("pendingClients", $_SESSION['pendingClients']);
    }

    if (isset($_SESSION['pendingLicences'])) {
        $smarty->assign("pendingLicences", $_SESSION['pendingLicences']);
    }

    if (isset($_SESSION['allInstructors'])) {
        $smarty->assign("allInstructors", $_SESSION['allInstructors']);
    }

    if (isset($_SESSION['classesUsuarios'])) {
        $smarty->assign("classesUsuarios", $_SESSION['classesUsuarios']);
    }

    $smarty->display("index.tpl");

```

## Templates:

### Confirm Driving License:

{\* Smarty \*}

<!DOCTYPE html>

<html>

<div class="container shadow p-3 mb-5 bg-white rounded" id="divConfirmDrivingLic" hidden>  
<h2>Usuarios con mas de 15 clases</h2>

<form method="POST" action="./DB/Queries/confirmLicense.php">

<ul id="ulClients" class="list-group">

{foreach from=\$pendingLicences item=client}

<div class="input-group mb-3">

<div class="input-group-prepend">

<div class="input-group-text">

<input name="document" type="radio" value="{ \$client['ci'] }">

</div>

</div>

<div class="form-control" style="height: auto;">

<label class="col-sm-2 col-form-label col-form-label-sm">CI</label>

<label class="col-sm-5 col-form-label col-form-label-sm">{ \$client['ci'] }</label>

<br>

<label class="col-sm-2 col-form-label col-form-label-sm">Nombre</label>

<label class="col-sm-5 col-form-label col-form-label-sm">{ \$client['nombre'] }

{ \$client['apellido'] }</label>

</div>

</div>

{/foreach}

</ul>

<input type="submit" value="Confirmar libretas" class="btn btn-outline-success col-sm-2"/>

</form>

</div>

</html>

Index:

```
{* Smarty *}
```

```
<!DOCTYPE html>
```

```
<html>
```

```
    <head>
```

```
        <meta charset="UTF-8">
```

```
        <title>Obligatorio</title>
```

```
        <link rel="stylesheet" href="/css/bootstrap/bootstrap.min.css" />
```

```
        <link href='/css/main.css' rel='stylesheet' />
```

```
        <script src="/js/jquery/jquery-3.5.1.min.js" type="text/javascript"></script>
```

```
        <script src='/js/jquery/main.js'></script>
```

```
        <script src='/js/scripts/script.js'></script>
```

```
        <script src="/js/bootstrap/bootstrap.min.js"></script>
```

```
        <script src="/js/jquery/moment.min.js" type="text/javascript"></script>
```

```
        <script src="/js/jquery/fullcalendar.min.js" type="text/javascript"></script>
```

```
        <script src="/js/jquery/jspdf.debug.js" type="text/javascript"></script>
```

```
        <link rel="icon" type="image/vnd.microsoft.icon" href="/imgs/icon/icon.ico" />
```

```
    </head>
```

```
    <body>
```

```
        {include file="{ $menu }"}
```

```
        <div id="errorAlert" class="alert alert-danger alert-dismissible fade show" role="alert"
```

```
{ $errorHidden }>
```

```
        { $errorMessage }
```

```
        <button type="button" class="close" data-dismiss="alert" aria-label="Close">
```

```
            <span aria-hidden="true">&times;</span>
```

```
        </button>
```

```
    </div>
```

```
        <div id="successAlert" class="alert alert-success alert-dismissible fade show" role="alert"
```

```
{ $successHidden }>
```

```
        { $successMessage }
```

```
        <button type="button" class="close" data-dismiss="alert" aria-label="Close">
```

```
            <span aria-hidden="true">&times;</span>
```

```
        </button>
```

```
    </div>
```



```

<br>
<div id="container">
{foreach from=$pages item=page}
{include file="{ $page}"}
{/foreach}
</div>
</body>
</html>

```

## Information Page:

```
{* Smarty *}
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script>
```

```

function procesoResultado(datos) {
if (datos["status"] == "OK") {
    var data = datos["data"];
} else {
    alert("Error al consultar las reservas")
}
}

```

```
document.addEventListener('DOMContentLoaded', function () {
```

```

var calendar = new FullCalendar.Calendar(document.getElementById('calendar'), {
    editable: false,
    selectable: false,
    businessHours: true,
    dayMaxEvents: true,
    aspectRatio: 2,
    events: function (fetchInfo, successCallback, failureCallback) {
    jQuery.ajax({
    url: "./DB/Queries/informationGet.php",
    type: "POST",
    dataType: "JSON",

```

```

    success: function (data) {
    var events = [];
    if (!!data.data) {
        var i = 0;
        $.map(data.data, function (r) {
        var event = {
        id: i,
        start: r.fecha,
        end: r.fecha
        };
        if (((data.instructores.total * 15) / 2) > r.total) {
        event.title = "Cupos disponibles.";
        event.color = "#64ff3e";
        } else if ((data.instructores.total * 15) == r.total) {
        event.title = "Dia completo.";
        event.color = "red";
        } else {
        event.title = "Pocos cupos disponibles.";
        event.color = "#fff93a";
        }
        events.push(event);
        i++;
        });
    }
    successCallback(events);
    },
    error: function (error) {
    failureCallback();
    }
    });
    });
    calendar.updateSize();
    calendar.render();
    });
</script>

</head>
<div id="divInfoMainPage" class="carousel slide" data-ride="carousel">

<ol class="carousel-indicators">
<li data-target="#divInfoMainPage" data-slide-to="0" ></li>
<li data-target="#divInfoMainPage" data-slide-to="1" class="active"></li>

```

```

<li data-target="#divInfoMainPage" data-slide-to="2"></li>
</ol>

<div class="carousel-inner">

<div class="carousel-item ">

<div class="carousel-caption d-none d-md-block">
    <h3 class="text-white bg-dark"> Usuarios Inscriptos : {$registerUsers} </h3>
    <h3 class="text-white bg-dark"> Usuarios Aprobados : {$approvedUsers} </h3>
</div>
</div>
<div class="carousel-item active" >
<div id='calendar' width="70%">
</div>
</div>

<div class="carousel-item">

<div class="carousel-caption d-none d-md-block">
    <h3 class="text-white bg-dark"> Costo de las Clases : $1.500 </h3>
</div>
</div>
</div>
</div>
</html>

```

Login Menu:

```
{* Smarty *}
```

```
<!DOCTYPE html>
```

```

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    </head>
    <body>
        <nav class="navbar navbar-dark bg-dark justify-content-between">
            <a class="navbar-brand" href="#">

```

```

        
        Academy
    </a>

    <form class="form-inline" method="POST" action="./DB/Queries/login.php" >
        <input name="user" class="form-control mr-sm-2" type="text" placeholder="Username"
required>
        <input name="password" class="form-control mr-sm-2" type="password"
placeholder="Password" required>
        <button type="submit" class="btn btn-outline-success mr-sm-2">Login</button>
        <button type="button" class="btn btn-outline-info mr-sm-2" data-toggle="modal"
data-target="#registerModal">Register</button>
    </form>
</nav>

    {include file="./registerUser.tpl"}

</body>
</html>

```

## Menu Admin:

{\* Smarty \*}

```
<!DOCTYPE html>
```

```

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    </head>
    <body>

        <nav class="navbar navbar-expand-lg navbar-dark bg-dark ">

            <a class="navbar-brand" href="#">
                
                Academy
            </a>

            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav mr-auto">
                    <li class="nav-item">

```

```

        <a class="nav-link" target="_self" href="javascript:changeContainer('mainPage')"> Inicio
    </a>
    </li>
    <li class="nav-item">
        <a class="nav-link" target="_self" href="javascript:changeContainer('addInstructor')">
Alta de Instructor </a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="javascript:changeContainer('approveClient')"> Aprobacion de
Cliente </a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="javascript:changeContainer('confirmDrivingLicense')">
Confirmacion de Libreta </a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" target="_self" href="javascript:changeContainer('classList')"> Listado
de Clases </a>
                </li>
            </ul>
        </div>

        <form action="./DB/Queries/logout.php" >
        <p class="navbar-brand"> {$fullName} </p>
        <button type="submit" class="btn btn-outline-info mr-sm-2">Logout</button>
        </form>
    </nav>

    </body>
</html>

```

Menu User:

```
{* Smarty *}
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
</head>
```

```
<body>
```

```

<nav class="navbar navbar-expand-lg navbar-dark bg-dark justify-content-between">

<a class="navbar-brand" href="#">

Academy
</a>

<div class="collapse navbar-collapse" id="navbarSupportedContent">
<ul class="navbar-nav mr-auto">
    <li class="nav-item">
        <a class="nav-link" target="_self" href="javascript:changeContainer('mainPage')"> Inicio
    </li>
    <li class="nav-item">
        <a class="nav-link" target="_self" href="javascript:changeContainer('bookClass')">
{$isClient}> Reserva de Clase </a>
    </li>
</ul>
</div>

<form action="./DB/Queries/logout.php" >
<p class="navbar-brand"> {$fullName} </p>
<button type="submit" class="btn btn-outline-info mr-sm-2">Logout</button>
</form>
</div>
</nav>

</body>
</html>

```

## Register Instructor:

```
{* Smarty *}
```

```
<!DOCTYPE html>
```

```
<html>
```

```

<div id="divRegisterInstructor" class="container shadow p-3 mb-5 bg-white rounded" hidden>
<h2>Alta de instructor</h2>

<form action="./DB/Queries/registerInstructor.php" method="POST">
<div class="form-group row">

```

```

        <label for="name" class="col-sm-2 col-form-label col-form-label-sm">Nombre</label>
        <div class="col-sm-10">
            <input type="text" class="form-control form-control-sm" name="name"
aria-describedby="name" placeholder="Nombre" required>
        </div>
    </div>
    <div class="form-group row">
        <label for="lastname" class="col-sm-2 col-form-label col-form-label-sm">Apellido</label>
        <div class="col-sm-10">
            <input type="text" class="form-control form-control-sm" name="lastname"
aria-describedby="lastname" placeholder="Apellido" required>
        </div>
    </div>
    <div class="form-group row">
        <label for="document" class="col-sm-2 col-form-label col-form-label-sm">Cedula</label>
        <div class="col-sm-10">
            <input type="text" class="form-control form-control-sm" name="document"
aria-describedby="document" placeholder="Cedula" required>
        </div>
    </div>
    <div class="form-group row">
        <label for="birthDate" class="col-sm-2 col-form-label col-form-label-sm">Fecha de
nacimiento</label>
        <div class="col-sm-10">
            <input type="date" class="form-control form-control-sm" name="birthDate"
aria-describedby="birthDate" placeholder="Fecha de nacimiento" required>
        </div>
    </div>
    <div class="form-group row">
        <label for="dueDate" class="col-sm-2 col-form-label col-form-label-sm">Fecha de vencimiento
de libreta</label>
        <div class="col-sm-10">
            <input type="date" class="form-control form-control-sm" name="dueDate"
aria-describedby="dueDate" placeholder="Fecha de vencimiento de libreta" required>
        </div>
    </div>

    <div class="form-group">
        <input class="btn btn-outline-secondary col-sm-2" name="reset" type="reset" value="Limpiar"
/>
        <input class="btn btn-outline-success col-sm-2" name="register" type="submit"
value="Registrar" />
    </div>

```

```

        </form>
    </div>
</html>

```

## Register User:

```

{* Smarty *}
<!DOCTYPE html>

```

```

<html>
    <!-- Modal -->
    <div class="modal fade" id="registerModal" tabindex="-1" role="dialog" aria-hidden="true">
        <div class="modal-dialog" role="document">
            <div class="modal-content">
                <div class="modal-header text-center">
                    <h4 class="modal-title w-100 font-weight-bold">Sign up</h4>
                    <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                        <span aria-hidden="true">&times;</span>
                    </button>
                </div>

                <form action="/DB/Queries/registerUser.php" method="POST">
                    <div class="modal-body mx-3">
                        <div class="md-form mb-5">
                            <i class="fas fa-user prefix grey-text"></i>
                            <input type="email" name="formEmail" class="form-control validate"
placeholder="Email" required>
                        </div>

                        <div class="md-form mb-5">
                            <i class="fas fa-envelope prefix grey-text"></i>
                            <input type="text" name="formName" class="form-control validate"
placeholder="Name" required>
                        </div>

                        <div class="md-form mb-5">
                            <i class="fas fa-envelope prefix grey-text"></i>
                            <input type="text" name="formLastName" class="form-control validate"
placeholder="Last Name" required>
                        </div>

```



```

        <div class="md-form mb-5">
        <i class="fas fa-envelope prefix grey-text"></i>
        <input type="text" name="formDocument" class="form-control validate"
placeholder="Document" required>
        </div>

        <div class="md-form mb-5">
        <i class="fas fa-envelope prefix grey-text"></i>
        <input type="date" name="formBirthdate" class="form-control validate" required>
        </div>

        <div class="md-form mb-5">
        <i class="fas fa-envelope prefix grey-text"></i>
        <input type="text" name="formAddress" class="form-control validate"
placeholder="Address">
        </div>

        <div class="md-form mb-4">
        <i class="fas fa-lock prefix grey-text"></i>
        <input type="password" name="formPass" class="form-control validate"
placeholder="Password" minlength="6" required>
        </div>

        <div class="md-form mb-4">
        <i class="fas fa-lock prefix grey-text"></i>
        <input type="password" name="formRePass" class="form-control validate"
placeholder="Re-Password" minlength="6" required>
        </div>
        </div>

        <div class="modal-footer d-flex justify-content-center">
        <button type="submit" class="btn btn-outline-success col-sm-2 ">Sign up</button>
        </div>
    </form>
</div>
</div>
</div>
</div>
</html>

```

## Approve Client:

{\* Smarty \*}

<!DOCTYPE html>

<html>

<div class="container shadow p-3 mb-5 bg-white rounded" id="divApproveClient" hidden>

<h2>Usuarios inscritos para clase de manejo</h2>

<form method="POST" action="./DB/Queries/approveUser.php">

<ul id="ulClients" class="list-group">

{foreach from=\$pendingClients item=client}

<div class="input-group mb-3">

<div class="input-group-prepend">

<div class="input-group-text">

<input name="document" type="radio" value="{ \$client['ci'] }">

</div>

</div>

<div class="form-control" style="height: auto;">

<label class="col-sm-2 col-form-label col-form-label-sm">CI</label>

<label class="col-sm-5 col-form-label col-form-label-sm">{ \$client['ci'] }</label>

<br>

<label class="col-sm-2 col-form-label col-form-label-sm">Nombre</label>

<label class="col-sm-5 col-form-label col-form-label-sm">{ \$client['nombre'] }

{ \$client['apellido'] }</label>

</div>

</div>

{/foreach}

</ul>

<div class="form-group">

<input type="submit" value="Aprobar clientes" class="btn btn-outline-success col-sm-2" />

</div>

</form>

</div>

</html>

## Book Class:

{\* Smarty \*}

<!DOCTYPE html>

```

<html>
  <div class="container shadow p-3 mb-5 bg-white rounded" id="divBookClass" hidden>
    <h2>Reservar Clase</h2>

    <form method="POST" action="./DB/Queries/bookClass.php">

      <div class="form-group row">
        <label for="date" class="col-sm-2 col-form-label col-form-label-sm">Fecha</label>
        <div class="col-sm-10">
          <input type="date" class="form-control form-control-sm" name="date"
aria-describedby="date" placeholder="Fecha" required>
        </div>
      </div>

      <div class="form-group row">
        <label for="time" class="col-sm-2 col-form-label col-form-label-sm">Horario</label>
        <div class="col-sm-10">
          <select name="time" class="form-control form-control-sm" required>
            <option value="7">7:00 - 8:00 hrs</option>
            <option value="8">8:00 - 9:00 hrs</option>
            <option value="9">9:00 - 10:00 hrs</option>
            <option value="10">10:00 - 11:00 hrs</option>
            <option value="11">11:00 - 12:00 hrs</option>
            <option value="12">12:00 - 13:00 hrs</option>
            <option value="13">13:00 - 14:00 hrs</option>
            <option value="14">14:00 - 15:00 hrs</option>
            <option value="15">15:00 - 16:00 hrs</option>
            <option value="16">16:00 - 17:00 hrs</option>
            <option value="17">17:00 - 18:00 hrs</option>
            <option value="18">18:00 - 19:00 hrs</option>
            <option value="19">19:00 - 20:00 hrs</option>
            <option value="20">20:00 - 21:00 hrs</option>
            <option value="21">21:00 - 22:00 hrs</option>
          </select>
        </div>
      </div>

      <div class="form-group row">
        <label for="instructor" class="col-sm-2 col-form-label col-form-label-sm">Instructor</label>
        <div class="col-sm-10">
          <select name="instructor" class="form-control form-control-sm" required>
            {foreach from=$allInstructors item=instructor}

```

```

        <option value="{ $instructor['instructor_id']}">{$instructor['nombre']}
{$instructor['apellido']}</option>
    {/foreach}
</select>
</div>
</div>

<div class="form-group">
<input type="submit" value="Reservar" class="btn btn-outline-success col-sm-2" />
</div>

</form>
</div>
</html>

```

## Class List:

```
{* Smarty *}
```

```
<!DOCTYPE html>
```

```

<html>
    <div class="container shadow p-3 mb-5 bg-white rounded" id="divClassList" hidden>
    <script>
    function myFunction() {
    var dateSelected = document.getElementById('dateSelected').value;
    var input = document.getElementById('instructorSelector');
    var instructor = input.options[input.selectedIndex].value;

    var li = document.getElementById('ulClases').getElementsByTagName('li');

    for (i = 0; i < li.length; i++) {

        var allLabel = li[i].getElementsByTagName("label");

        var fechaValue = allLabel[1].textContent;
        var instructorValue = allLabel[7].textContent;

        if (instructorValue == instructor && fechaValue == dateSelected) {
            li[i].style.display = "";
        } else {
            li[i].style.display = "none";
        }
    }
    }
    </script>
    </div>
</html>

```

```
}  
}  
}
```

```
function cleanFilter() {  
  var li = document.getElementById("ulClases").getElementsByTagName('li');  
  
  for (i = 0; i < li.length; i++) {  
    li[i].style.display = "";  
  }  
}
```

```
function generatePDF() {  
  var doc = new jsPDF('portrait');  
  var verticalOffset = 20;
```

```
  doc.text(20, verticalOffset, 'Listado de clases');  
  verticalOffset += 10;
```

```
  if (document.getElementById('instructorSelector').selectedIndex >= 0) {  
    var input = document.getElementById('instructorSelector');  
    var dateSelected = document.getElementById('dateSelected').value;  
    var instructor = input.options[input.selectedIndex].value;
```

```
    var li = document.getElementById("ulClases").getElementsByTagName('li');
```

```
    doc.setFontSize(12);
```

```
    // Loop through all list items, and hide those who don't match the search query  
    for (i = 0; i < li.length; i++) {  
      var allLabel = li[i].getElementsByTagName("label");
```

```
      var fechaValue = allLabel[1].textContent;  
      var horaValue = allLabel[4].textContent;  
      var instructorValue = allLabel[7].textContent;  
      var usuarioValue = allLabel[10].textContent + ' ' + allLabel[11].textContent ;  
      var direccionValue = allLabel[13].textContent;
```

```
      if (instructorValue == instructor && fechaValue == dateSelected) {  
        doc.text(20, verticalOffset, 'Fecha: ' + fechaValue + ' Turno: ' + horaValue + ' - Instructor:  
' + instructorValue);  
        verticalOffset += 6;  
        doc.text(20, verticalOffset, 'Usuario: ' + usuarioValue);
```

```

        verticalOffset += 6;
        doc.text(20, verticalOffset, 'Direccion: ' + direccionValue);
    } else if (li[i].style.display == "") {
        doc.text(20, verticalOffset, 'Fecha: ' + fechaValue + ' ' + horaValue + ' - Instructor: ' +
instructorValue);
        verticalOffset += 6;
        doc.text(20, verticalOffset, 'Usuario: ' + usuarioValue);
        verticalOffset += 6;
        doc.text(20, verticalOffset, 'Direccion: ' + direccionValue);
    }

    verticalOffset += 10;
}
}

```

```

doc.save('listadoClases.pdf');
}

```

```

Date.prototype.toDateInputValue = (function () {
var local = new Date(this);
local.setMinutes(this.getMinutes() - this.getTimezoneOffset());
return local.toJSON().slice(0, 10);
})();
</script>

```

<h2>Listado de clases por instructor</h2>

```

<div id="divInstructores">
<div class="form-group row">
<label for="instructor" class="col-sm-2 col-form-label col-form-label-sm">Instructor</label>
<div class="col-sm-10">
    <select name="instructor" id="instructorSelector" class="form-control form-control-sm"
required>
        {foreach from=$allInstructors item=instructor}
        <option value="{${instructor['instructor_id']}" >{${instructor['nombre']}}
{${instructor['apellido']}</option>
        {/foreach}
    </select>
</div>
</div>
<div class="form-group row">
<label class="col-sm-2 col-form-label col-form-label-sm">Fecha</label>
<div class="col-sm-10">

```

```

        <input type="date" id="dateSelected" name="formBirthdate" class="form-control
validate" required>
    </div>
</div>
<div class="form-group">
    <input type="button" value="Aplicar filtro" onclick="myFunction()" class="btn
btn-outline-success col-sm-2" />
    <input type="button" value="Exportar lista a PDF" onclick="generatePDF()" class="btn
btn-outline-info col-sm-2" />
</div>
</div>

<div id="divClasesInstructor">
<h3>Listado de Clases</h3>

<ul id="ulClases" class="list-group">
{foreach from=$clasesUsuarios item=clase}
    <li class="input-group mb-3" id="liClase">
        <div class="input-group mb-3">
            <label class="col-sm-2 col-form-label col-form-label-sm">Fecha</label>
            <label class="col-sm-5 col-form-label col-form-label-sm">{$clase['fecha']}</label>
            <label class="col-sm-5 col-form-label col-form-label-sm"></label>
            <br>
            <label class="col-sm-2 col-form-label col-form-label-sm">Turno</label>
            <label class="col-sm-5 col-form-label col-form-label-sm">{$clase['hora']}</label>
            <label class="col-sm-5 col-form-label col-form-label-sm"></label>
            <br>
            <label class="col-sm-2 col-form-label col-form-label-sm">Id Instructor</label>
            <label class="col-sm-5 col-form-label
col-form-label-sm">{$clase['instructor_id']}</label>
            <label class="col-sm-5 col-form-label col-form-label-sm"></label>
            <br>
            <label class="col-sm-2 col-form-label col-form-label-sm">Nombre</label>
            <label class="col-sm-5 col-form-label col-form-label-sm">{$clase['nombre']}</label>
            <label class="col-sm-5 col-form-label col-form-label-sm">{$clase['apellido']}</label>
            <br>
            <label class="col-sm-2 col-form-label col-form-label-sm">Direccion</label>
            <label class="col-sm-5 col-form-label col-form-label-sm">{$clase['direccion']} </label>
            <label class="col-sm-5 col-form-label col-form-label-sm"></label>
        </div>
    </li>
{foreach}
</ul>

```

```
</div>

</div>
</html>
```

## Config:

### Data Base:

```
<?php

define("MOTOR", "mysql");
define("SERVIDOR", "localhost");
define("BASE", "obligatorio");
define("USUARIO", "root");
define("CLAVE", "root");

$unregisteredPages = array('./informationPage.tpl');

$adminPages = array('./informationPage.tpl', './approveClient.tpl', 'registerInstructor.tpl',
'./confirmDrivingLicense.tpl', 'classList.tpl');
$userPages = array('./informationPage.tpl', './bookClass.tpl');
?>
```

### Smarty:

```
<?php

//Incluyo configuraciones
require_once($_SERVER['DOCUMENT_ROOT'].'/ObligatorioTaller/DB/config.php');
require_once($_SERVER['DOCUMENT_ROOT'].'/ObligatorioTaller/lib/Smarty/Smarty.class.php');
require_once($_SERVER['DOCUMENT_ROOT'].'/ObligatorioTaller/lib/DB/ConexionDB.php');

//Defino directorios
define('TEMPLATE_DIR', $_SERVER['DOCUMENT_ROOT'].'/ObligatorioTaller/templates/');
define('COMPILER_DIR', $_SERVER['DOCUMENT_ROOT'].'/ObligatorioTaller/tmp/templates_c/');
define('CONFIG_DIR', $_SERVER['DOCUMENT_ROOT'].'/ObligatorioTaller/tmp/configs/');
define('CACHE_DIR', $_SERVER['DOCUMENT_ROOT'].'/ObligatorioTaller/tmp/cache/');
```



```
//Inicializo Smarty
$smarty = new Smarty;
//Designacion de Directorios
$smarty->template_dir = TEMPLATE_DIR;
$smarty->compile_dir = COMPILER_DIR;
$smarty->config_dir = CONFIG_DIR;
$smarty->cache_dir = CACHE_DIR;

?>
```

## Javascript:

```
function changeContainer(pageName) {
    if (pageName == "mainPage")
        $("#divInfoMainPage").removeAttr('hidden');
    else
        $("#divInfoMainPage").attr('hidden', "");

    if (pageName == "addInstructor")
        $("#divRegisterInstructor").removeAttr('hidden');
    else
        $("#divRegisterInstructor").attr('hidden', "");

    if (pageName == "approveClient")
        $("#divApproveClient").removeAttr('hidden');
    else
        $("#divApproveClient").attr('hidden', "");

    if (pageName == "confirmDrivingLicense")
        $("#divConfirmDrivingLic").removeAttr('hidden');
    else
        $("#divConfirmDrivingLic").attr('hidden', "");

    if (pageName == "classList")
        $("#divClassList").removeAttr('hidden');
    else
        $("#divClassList").attr('hidden', "");

    if (pageName == "bookClass")
        $("#divBookClass").removeAttr('hidden');
    else
        $("#divBookClass").attr('hidden', "");
}
```

}