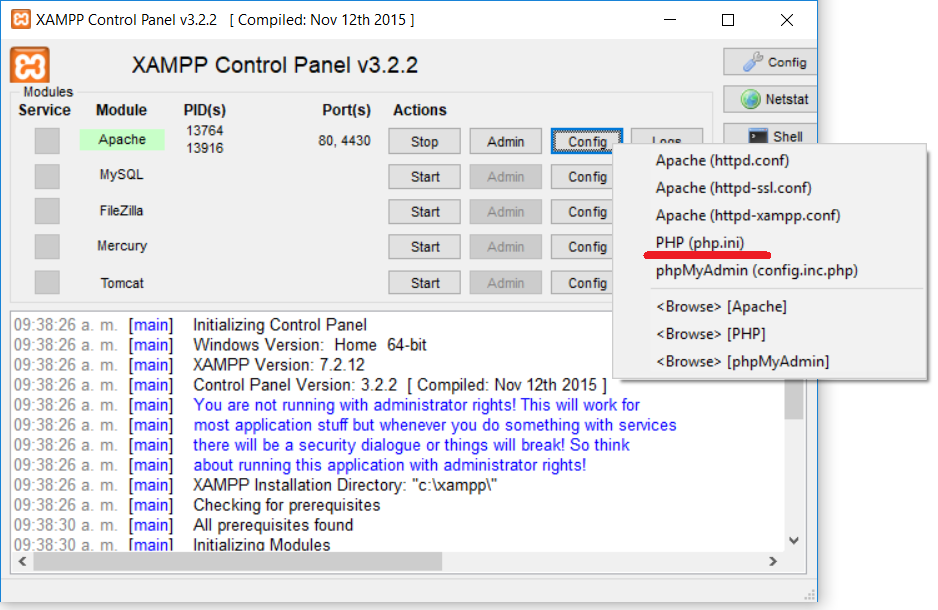
**Instalar el Gestor de Base de Datos PostgreSQL**



**Instalar Servidor Web Xampp**



Ahora vamos habilitar los drivers de Postgresql, para que este trabaje con el apache, para hacer esto vamos a editar el archivo php.ini:



y quitar el **;** de las siguientes líneas:

;extension=pdo\_odbc

;extension=pdo\_pgsql

quedando:

extension=pdo\_odbc

extension=pdo\_pgsql

enseguida guarde los cambios y reinicie el apache.

Creación de A/B/C del catálogo Clasif

Vamos a crear una Base de Datos en PostgreSQL con del nombre **PWeb** y un esquema de nombre **cuerpo**. Dentro de este esquema vamos a crear las siguientes tablas:

Para ejemplificar esto vamos a crear la tabla clasif, con la siguiente estructura:

Create Table cuerpo.rol(

idrol SERIAL,

nomrol varchar(100),

constraint pkrol primary key (idrol)

);

Insert into cuerpo.rol(nomrol) values ('Administrador');

Insert into cuerpo.rol(nomrol) values ('Usuario');

Create Table cuerpo.usuario(

idusuario SERIAL,

correousr varchar(100) not null UNIQUE,

nomusr varchar(100),

contrasena varchar(100),

idrol Int not null,

session varchar(60),

constraint pkusr primary key (idusuario),

constraint fkusr foreign key (idrol) references cuerpo.rol(idrol)

);

Insert into cuerpo.usuario(correousr,nomusr,contrasena,idrol,session)

values ('agz@hotmail.com','Alex Lora','$2y$10$LFEKkshlceG0b19R4NER5uMvQw2th.had.z8UvWej05rUn2V8Rxwi',1,'')

Create Table cuerpo.opcmenu(

idopcmenu SERIAL,

idmenu varchar(40) not null,

nommenu varchar(100),

orden Int,

constraint pkopc primary key (idopcmenu)

);

Insert into cuerpo.opcmenu(idmenu,nommenu,orden) Values ('opcClasif','Clasificaciones',10);

Insert into cuerpo.opcmenu(idmenu,nommenu,orden) Values ('opcRol','Roles',10);

Insert into cuerpo.opcmenu(idmenu,nommenu,orden) Values ('opcUsuario','Usuario',20);

Insert into cuerpo.opcmenu(idmenu,nommenu,orden) Values ('opcMenu','Menu',30);

Insert into cuerpo.opcmenu(idmenu,nommenu,orden) Values ('opcPerfil','Perfil',40);

Create Table cuerpo.perfil(

idrol Int not null,

idopcmenu Int not null,

fechahora timestamp default(now()),

constraint pkperfil primary key (idrol,idopcmenu),

constraint fkrol foreign key (idrol) references cuerpo.rol(idrol),

constraint fkmenu foreign key (idopcmenu) references cuerpo.opcmenu(idopcmenu)

);

Insert into cuerpo.perfil(idrol,idopcmenu) values (1,1);

Insert into cuerpo.perfil(idrol,idopcmenu) values (1,2);

Insert into cuerpo.perfil(idrol,idopcmenu) values (1,3);

Insert into cuerpo.perfil(idrol,idopcmenu) values (1,4);

Insert into cuerpo.perfil(idrol,idopcmenu) values (1,5);

Create Table cuerpo.clasif(

idclasif SERIAL,

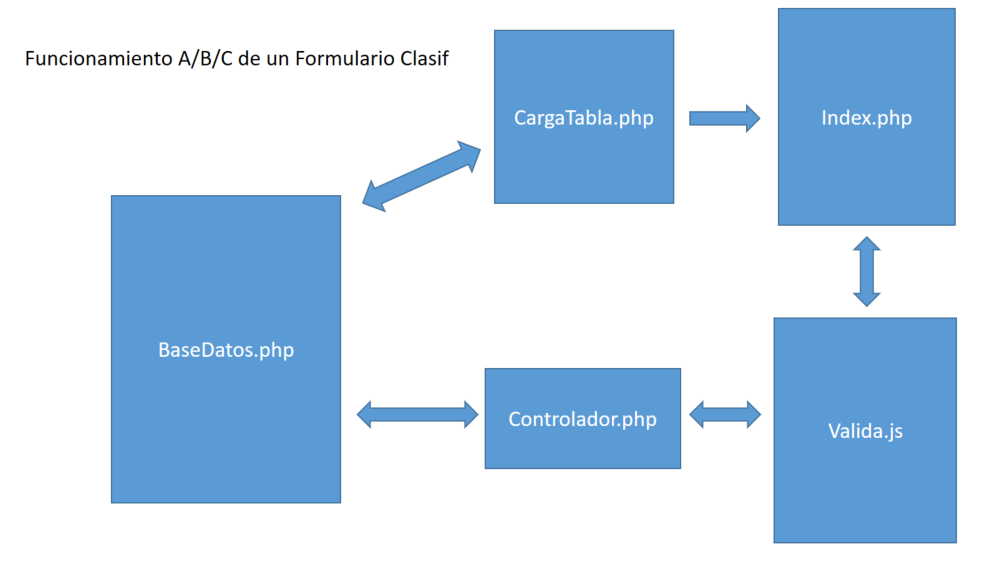
nomclasif varchar(40),

constraint pkClasif primary key (idclasif)

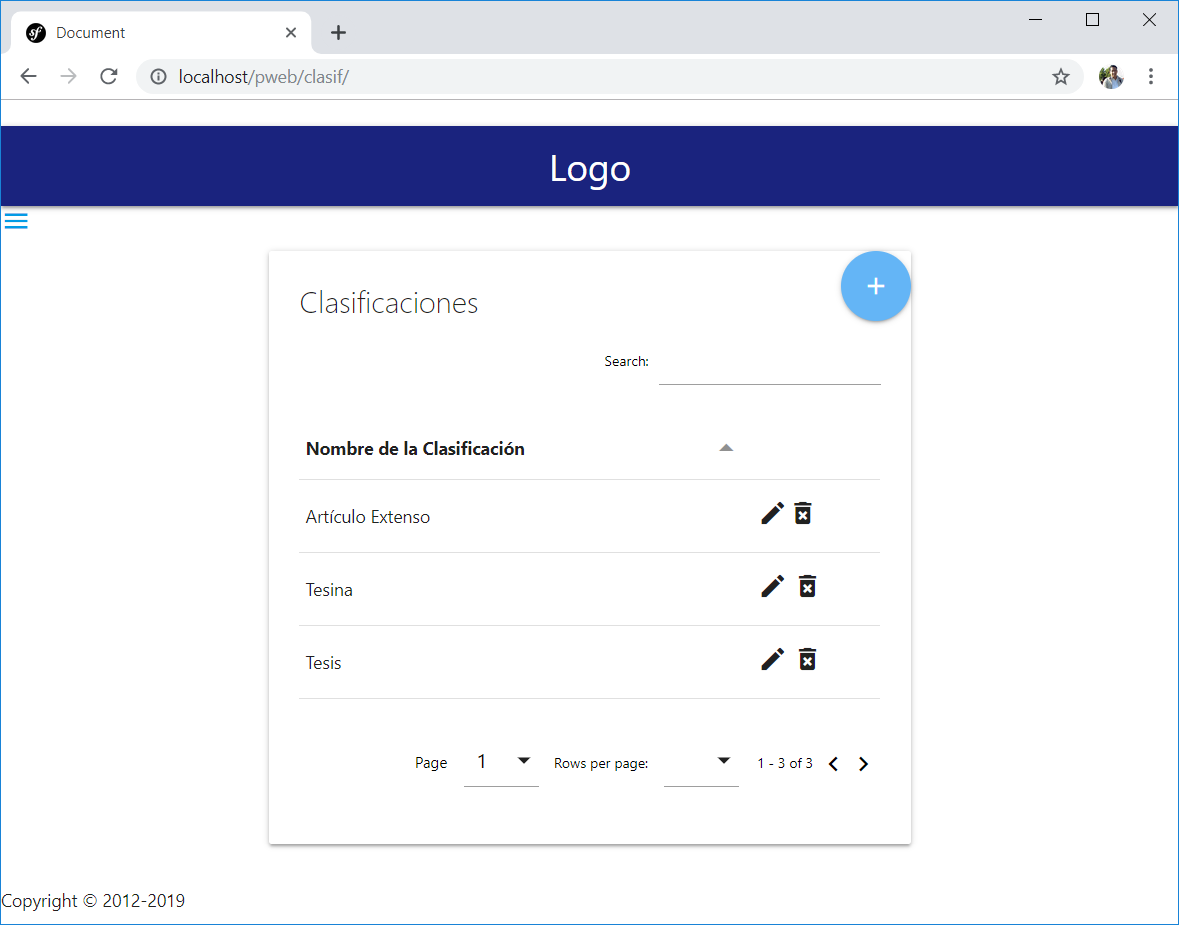
);

Insert into cuerpo.clasif(nomclasif) values('Herramientas');

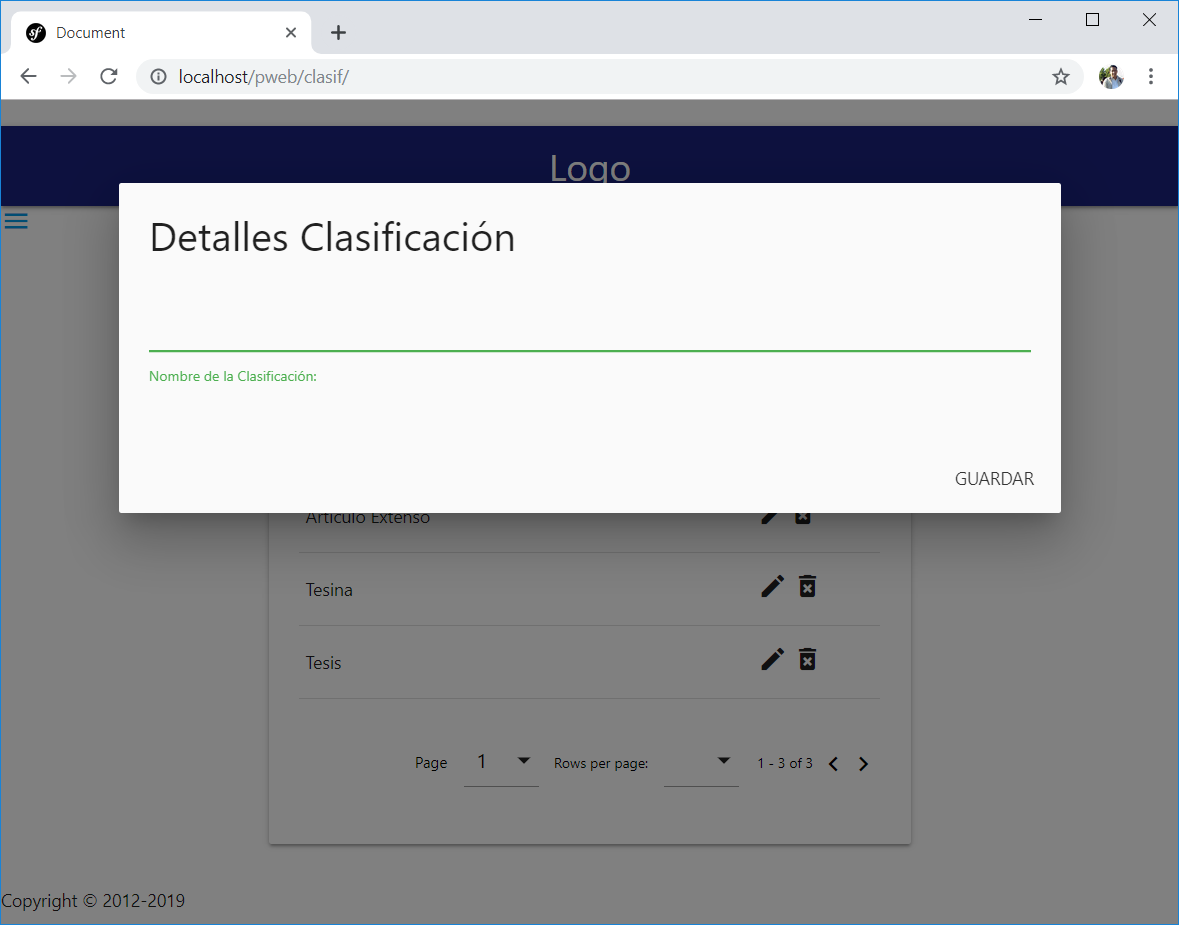
insert into cuerpo.clasif(nomclasif) values('Pinturas');



Vamos a crear una página web, con la siguiente distribución:



Al momento de dar clic en el icono circular de + se abrirá la ventana modal:



El código de la página **Index.php** se muestra a continuación:

|  |
| --- |
| <!DOCTYPE html>  <html lang="es">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <meta http-equiv="X-UA-Compatible" content="ie=edge">      <title>Clasificaciones</title>      <!-- Importa librerias de estilos de Materialize, DataTable e Iconos -->      <link type="text/css" rel="stylesheet" href="../css/materialize.min.css" media="screen,projection"/>      <link type="text/css" rel="stylesheet" href="../css/dataTables.materialize.css"/>      <link type="text/css" rel="stylesheet" href="../css/default.css"/>      <link rel="icon" type="image/x-icon" href="../fonts/favicon.ico" />  </head>  <body>  <?php         include\_once("../Utilerias/BaseDatos.php");         $idopc = "opcClasif";         $res = ValidaOpcion($idopc);         if ($res == "")              header("location:../Acceso/");         else              include\_once("../resources/html/header.php");  ?>  <!-- Colocar su código a partir de este comentario -->  <div class="container">      <div class="row">          <div class="col s12 m8 offset-m2">              <div class="card">                  <a id="add" class="btn-floating btn-large waves-effect waves-light right blue lighten-2">                      <i class="material-icons">add</i>                  </a>                  <div class="card-content">                      <span class="card-title">Clasificaciones</span>                      <table id="dtTable" class="highlight bordered dataTable">                          <thead>                              <tr><th>Nombre de la Clasificación</th><th></th></tr>                          </thead>                          <tbody>                              <?php                                  include\_once("./CargaTabla.php");                              ?>                          </tbody>                      </table>                  </div>              </div>          </div>      </div>  </div>  <!--                        Ventana Modal               -->  <div id="ventanaModal" class="modal">      <div class="modal-content">          <h4>Detalles Clasificación</h4>          <br>          <form id="formulario" method="post">              <div class="row">                  <div class="input-field\_col\_s12">                      <input type="hidden" name="idc" id="idc">                      <input type="text" name="nomc" id="nomc" class="validate">                      <label for="nomc">Nombre de la Clasificación:</label>                  </div>              </div>          </form>      </div>      <div class="modal-footer">          <a id="guardar" class="modal-action waves-effect waves-green btn-flat">Guardar</a>      </div>  </div>  <!-- La siguiente linea incluye el footer de nuestra página web -->  <?php include\_once("../resources/html/footer.php"); ?>  <!-- Importa librerias de JavaScript de Jquery, Jaquery Validate, DataTable       y Materialize                                                       -->  <script type="text/javascript" src="../js/jquery-3.0.0.min.js"></script>      <script type="text/javascript" src="../js/materialize.min.js"></script>      <script type="text/javascript" src="../js/jquery.dataTables.min.js"></script>      <script type="text/javascript" src="../js/dataTables.materialize.js"></script>      <script type="text/javascript" src="../js/jquery.validate.min.js"></script>      <script type="text/javascript" src="./Valida.js"></script>      <script type="text/javascript">      $(document).ready(function(){          $('select').formSelect();          $('.sidenav').sidenav();          $(".dropdown-trigger").dropdown();       });      </script>  </body>  </html> |

El archivo de **CargaTabla.php** llena el cuerpo de la tabla creada en Index.php, el código se muestra a continuación:

|  |
| --- |
| <?php        include\_once("../Utilerias/BaseDatos.php"); // Carga modelo de BD        $clasif = consultaClasif();         // Consulta la tabla de Clasif        foreach ($clasif as $tupla){        // Recorre los registro que retorno            $idclasif = $tupla['idclasif'];            $nomclasif = $tupla['nomclasif'];            //   Este id es el identificador para agregar y eliminar registros del dadaTable            //         v            echo "<tr id='$idclasif'><td>$nomclasif</td><td>  <i class='material-icons edit' data-id='$idclasif' data-nom='$nomclasif'>create</i>  <i class='material-icons delete' data-id='$idclasif'>delete\_forever</i></td></tr>";           // En los data-campo='valor' especiificamos los valores a recuperar cuando editamos           // El numero de columnas del <tr> deberá ser el mismo que el <tr> del encabezado de           // Tabla        }  ?> |

El archivo de **BaseDatos.php**, se muestra a continuación y contiene la conexión a la base de datos de postgresql y las funciones serán invocadas, dependiendo de la funcionalidad.

|  |
| --- |
| <?php  //---------------------------------------------------------------------  // Utilerias de Bases de Dato  // Alejandro Guzmán Zazueta  // Septiembre 2019  //---------------------------------------------------------------------  try{          $Cn = new PDO('pgsql:host=localhost;port=5432;dbname=PWeb;user=postgres;password=hola');          //$Cn = new PDO('mysql:host=localhost; dbname=bdalumnos','root','');          $Cn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);          $Cn->exec("SET CLIENT\_ENCODING TO 'UTF8';");          //$Cn->exec("SET CHARACTER SET utf8"); // MYSQL  }catch(Exception $e){      die("Error: " . $e->GetMessage());  }  // Función para ejecutar consultas SELECT  function Consulta($query)  {      global $Cn;      try{          $result =$Cn->query($query);          $resultado = $result->fetchAll(PDO::FETCH\_ASSOC);          $result->closeCursor();          return $resultado;      }catch(Exception $e){          die("Error en la LIN: " . $e->getLine() . ", MSG: " . $e->GetMessage());      }  }  // Función que recibe un insert y regresa el consecutivo que le genero en la llave primaria  // por ejemplo: Insert Into cuerpo.clasif (nomclasif) values ('Articulo en Extenso')  // Returning idclasif, nomclasif;  function EjecutaConsecutivo($sentencia, $llave){      global $Cn;      try {          $result = $Cn->query($sentencia);          $resultado = $result->fetchAll(PDO::FETCH\_ASSOC);          $result->closeCursor();          //var\_dump($resultado);         // die("AAA");          return $resultado[0][$llave];      } catch (Exception $e) {          die("Error en la linea: " + $e->getLine() + " MSG: " + $e->GetMessage());          return 0;      }  }  // Sirve para ejecutar una sentencia INSERT, UPDATE O DELETE  function Ejecuta ($sentencia){      global $Cn;      try {          $result = $Cn->query($sentencia);          $result->closeCursor();          return 1; // Exito      } catch (Exception $e) {          //die("Error en la linea: " + $e->getLine() + " MSG: " + $e->GetMessage());          return 0; // Fallo      }  }  //----------------------------------PERFIL------------------------------  function opcAsignadas($idrol)  {      $query = "select b.\* from cuerpo.perfil a inner join cuerpo.opcmenu b on (a.idopcmenu = b.idopcmenu) where a.idrol=$idrol order by nommenu";      return Consulta($query);  }  function opcDisponibles($idrol)  {      $query = "select a.\* from cuerpo.opcmenu a where a.idopcmenu not in (select b.idopcmenu from cuerpo.perfil b where b.idrol = $idrol)      order by nommenu";      return Consulta($query);  }  function eliminaPerfil($post){      $idopc = $post['ido'];      $idrol = $post['idr'];      $sentencia = "DELETE FROM cuerpo.perfil WHERE idopcmenu=$idopc and idrol=$idrol";      return Ejecuta($sentencia);  }  function insertarPerfil($post){      $idopc = $post['ido'];      $idrol = $post['idr'];      $sentencia = "INSERT INTO cuerpo.perfil(idrol,idopcmenu) values($idrol,$idopc)";      return Ejecuta($sentencia);  }  //---------------------------------ROL-----------------------------------  function consultaRol()  {      $query = "SELECT idrol,nomrol FROM cuerpo.rol ORDER BY nomrol";      return Consulta($query);  }  function InsertarRol(&$post){      $idr = $post['idr'];      $nom = $post['nomr'];      $sentencia = "INSERT INTO cuerpo.rol(nomrol) values('$nom') RETURNING idrol";      $id = EjecutaConsecutivo($sentencia,"idrol");      $post['idr']=$id;      return $id;  }  function ActualizarRol($post){      $idr = $post['idr'];      $nom = $post['nomr'];      $sentencia = "UPDATE cuerpo.rol SET nomrol='$nom' WHERE idrol=$idr";      return Ejecuta($sentencia);  }  function EliminarRol($post){      $idr = $post['idr'];      $sentencia = "DELETE FROM cuerpo.rol WHERE idrol=$idr";      return Ejecuta($sentencia);  }  //-------------------------------USUARIO--------------------------------  function ValidaOpcion($idopc){      if  (!isset($\_SESSION))          session\_start();      $idSess = session\_id();      $corr = $\_SESSION["correo"];      $query = "SELECT  idmenu FROM cuerpo.usuario A inner Join cuerpo.perfil B On (A.idrol = B.idrol)           Inner Join cuerpo.opcmenu C On (B.idopcmenu = C.idopcmenu)      WHERE correousr='$corr' and session= '$idSess' and C.idmenu='$idopc'";      $res = Consulta($query);      $opc = "";      foreach ($res as $tupla)      {          $opc = $tupla["idmenu"];      }      return $opc;  }  function ValidaUsr($post, $ids){      $corr = $post['usr'];      $contra = $post['contra'];      $query = "SELECT idusuario,correousr,nomusr,contrasena,idrol      FROM cuerpo.usuario Where correousr= '$corr'";      $res =  Consulta($query);      $pwdEnc = "";      $idrol = 0;      foreach ($res as $tupla)      {          $pwdEnc = $tupla['contrasena'];          $idrol = $tupla['idrol'];      }      if (password\_verify($contra, $pwdEnc) ){          $sentencia = "Update cuerpo.usuario set session='$ids' where correousr= '$corr'";          return Ejecuta($sentencia);      }      else{          return 0;      }  }  function consultaUsr()  {      $query = "SELECT idusuario,correousr,nomusr,contrasena,A.idrol,nomrol FROM cuerpo.usuario A Inner Join cuerpo.rol B on (A.idrol = B.idrol) ORDER BY nomusr";      return Consulta($query);  }  function InsertarUsuario(&$post){      $corr = $post['corr'];      $nom = $post['nom'];      $contra = $post['contra'];      $pwdEnc = password\_hash($contra, PASSWORD\_DEFAULT);      $idrol = $post['idrol'];      $sentencia = "INSERT INTO cuerpo.usuario(correousr,nomusr,contrasena,idrol) values('$corr','$nom','$pwdEnc',$idrol) RETURNING idusuario";      $id = EjecutaConsecutivo($sentencia,"idusuario");      $post['idusr']=$id;      $post['contra']=$pwdEnc;      return $id;  }  function ActualizarUsuario($post){      $idusr = $post['idusr'];      $corr = $post['corr'];      $nom = $post['nom'];      $contra = $post['contra'];      $idrol = $post['idrol'];      $sentencia = "UPDATE cuerpo.usuario SET nomusr='$nom', contrasena='$contra', idrol=$idrol, correousr='$corr' WHERE idusuario=$idusr";      return Ejecuta($sentencia);  }  function EliminarUsuario($post){      $idusr = $post['idusr'];      $sentencia = "DELETE FROM cuerpo.usuario WHERE idusuario=$idusr";      return Ejecuta($sentencia);  }  //----------------------------MENU-------------------------------  function consultaMenu()  {      $query = "SELECT idopcmenu,idmenu,nommenu,orden FROM cuerpo.opcmenu ORDER BY orden";      return Consulta($query);  }  function InsertarMenu(&$post){      $idm = $post['idm'];      $nomm = $post['nomm'];      $orden = $post['orden'];      $sentencia = "INSERT INTO cuerpo.opcmenu(idmenu,nommenu,orden) values ('$idm','$nomm',$orden) RETURNING idopcmenu";      $id = EjecutaConsecutivo($sentencia,"idopcmenu");      $post['idopc']=$id;      return $id;  }  function ActualizarMenu($post){      $idopc = $post['idopc'];      $idm = $post['idm'];      $nomm = $post['nomm'];      $orden = $post['orden'];      $sentencia = "UPDATE cuerpo.opcmenu SET idmenu='$idm', nommenu='$nomm', orden=$orden WHERE idopcmenu=$idopc";      return Ejecuta($sentencia);  }  function EliminarMenu($post){      $idopc = $post['idopc'];      $sentencia = "DELETE FROM cuerpo.opcmenu WHERE idopcmenu=$idopc";      return Ejecuta($sentencia);  }  //--------------------------CLASIF-------------------------------------  function consultaClasif()  {      $query = "SELECT idclasif,nomclasif FROM cuerpo.clasif ORDER BY nomclasif";      return Consulta($query);  }  function InsertarClasif(&$post){      $nom = $post['nomc'];      $sentencia = "INSERT INTO cuerpo.clasif(nomclasif) values('$nom') RETURNING idclasif";      $id = EjecutaConsecutivo($sentencia,"idclasif");      // En el idc colocamos el valor que nos regresa el Returning      // que es el Consecutivo que se genero por ser de tipo SERIAL      $post['idc']=$id;      return $id;  }  function ActualizarClasif($post){      $idc = $post['idc'];      $nom = $post['nomc'];      $sentencia = "UPDATE cuerpo.clasif SET nomclasif='$nom' WHERE idclasif=$idc";      return Ejecuta($sentencia);  }  function EliminarClasif($post){      $idc = $post['idc'];      $sentencia = "DELETE FROM cuerpo.clasif WHERE idclasif=$idc";      return Ejecuta($sentencia);  }  //--------------------------Sensado-------------------------------  function consultaSensado()  {      $query = "SELECT idsen,nomsensor,valor FROM cuerpo.sensado ORDER BY nomsensor";      return Consulta($query);  }  function InsertarSensado(&$post){      $noms = $post['noms'];      $val = $post['valor'];      $sentencia = "INSERT INTO cuerpo.sensado(nomsensor,valor) values('$noms', $val) RETURNING idsen";      $id = EjecutaConsecutivo($sentencia,"idsen");      $post['ids']=$id;      return $id;  }  function ActualizarSensado($post){      $ids = $post['ids'];      $noms = $post['noms'];      $val = $post['valor'];      $sentencia = "UPDATE cuerpo.sensado SET nomsensor='$noms', valor=$val WHERE idsen=$ids";      return Ejecuta($sentencia);  }  function EliminarSensado($post){      $ids = $post['ids'];      $sentencia = "DELETE FROM cuerpo.sensado WHERE idsen=$ids";      return Ejecuta($sentencia);  }    //------------------------------------------------------------  function validaSess(&$correo, &$tu, &$idsess){      $correo = $correo;      $sql = 'SELECT idsession,tipousr FROM "Escuela".usuario  WHERE correo like ' . "'" . $correo . "'";      $res = Consulta($sql);      $tipo = 0;      foreach ($res as $tupla )      {          $idsess = $tupla['idsession'];          $tu = $tupla['tipousr'];      }      return 0;  } |

El siguiente archivo es **Valida.js**, este archivo de javascript contiene el llenado del DataTable, la validación del formulario clasif y el clic de los botones. La afectación de la base de datos se hace usando AJAX, con la finalidad de evitar que se recargue totalmente la página web.

**AJAX** es el acrónimo de *Asynchronous Javascript and XML*, es decir: **Javascript y XML Asincrono**.

En esencia, **AJAX permite que una página web que ya ha sido cargada solicite nueva información al servidor**. Dicho así, no supondría en realidad ningún invento novedoso. Una página web que contiene un enlace permite que se solicite al servidor nueva información cada vez que se pincha dicho enlace. Una página web que contiene un formulario envía información al servidor y recibe de él nueva información, normalmente la respuesta ante los datos que se han enviado. En ambos casos hay una conexión entre el cliente y el servidor.

¿Cuál es la diferencia cuando usamos AJAX? La diferencia es que con AJAX no es necesario recargar toda la página web, como ocurre cuando pinchamos en un enlace o cuando pulsamos el botón submit de un formulario.

El código del archivo **Valida.js**, se muestra a continuación:

|  |
| --- |
| $(init);  var table = null;  var controlador = "../Utilerias/Controlador.php";  function init(){      //Configuración del DataTable      table = $('#dtTable').DataTable({          "aLengthMenu" : [[10,25,50,75,100],[10,25,50,75,100]],          "iDisplayLength" : 15      });      // Inicializa el formulario modal y la validación del formulario      $('#ventanaModal').modal();      validateForm();      // Clic del boton circular +      $('#add').on("click", function(){          $('#idc').val('');          $('#nomc').val('');          $('#ventanaModal').modal('open');          $('#nomc').focus();      });      // Clic del boton Guardar del formulario modal      $('#guardar').on("click", function(){          $('#formulario').submit();      });      // Clic del icono Editar del DataTable      $(document).on("click", '.edit', function(){          var id = $(this).attr("data-id");          var nom = $(this).attr("data-nom");          $('#idc').val(id);          $('#nomc').val(nom);          $('#ventanaModal').modal('open');          $('#nomc').focus();      });      // Clic del icono eliminar del DataTable      $(document).on("click", '.delete', function(){          var idc = $(this).attr("data-id");          var parametros = "idc=" + idc + "&accion=Eliminar&tabla=Clasif";          $.ajax({              type: "post",              url: controlador,              dataType: 'json',              data: parametros,              success: function(respuesta){                  if (respuesta['status']==1){                      var data = respuesta['data'];                      if (idc>0){                          // Elimina el registro del DataTable                          table.row('#' + data.idc).remove().draw();                      }                      M.toast({html: 'Clasificación Eliminada', classes: 'rounded blue lighten-2'});                  }              }          });// fin de AJAX      });  }  // Código de la validación del formulario modal  function validateForm(){      $('#formulario').validate({          rules:{              nomc:{required:true, minlength:4, maxlength:80},          },          messages: {              nomc:{required:"No puedes dejar este campo vacío",minlength:"Debes ingresar al menos 4 caracteres", maxlength:"No puedes ingresar más de 80 caracteres"},          },          errorElement: "div",          errorClass: "invalid",          errorPlacement: function(error, element){              error.insertAfter(element)          },          submitHandler: function(form){              saveData();          }      });  }  // Una vez validado el formulario si dio clic en Guardar del formulario  // agrega el registro nuevo o lo actualiza usando AJAX  function saveData(){      var idc = $("#idc").val();      var parametros = $("#formulario").serialize() + "&tabla=Clasif";      if (idc > 0){          parametros = parametros + "&accion=Actualizar";      }      else{          parametros = parametros + "&accion=Insertar";      }      $.ajax({          type: "post",          url: controlador,          dataType: 'json',          data: parametros,          success: function(respuesta){              if (respuesta['status']==1){                  $('#idc').val('');                  $('#nomc').val('');                  $('#ventanaModal').modal('close');                  var data = respuesta['data'];                  if (idc >0){                      // Elimina el registro que se modifico del DataTable                      table.row('#' + data.idc).remove().draw();                  }                  // Agrega el registro ya modificado o nuevo al DataTable                  var row = table.row.add([                      data.nomc,                      '<i class="material-icons edit" data-id="' + data.idc + '" data-nom="' + data.nomc +'">create</i><i class="material-icons delete" data-id="' + data.idc + '">delete\_forever</i>'                  ]).draw().node();                  $(row).attr('id',data.idc);                  M.toast({html: 'Clasificación Guardada', classes: 'rounded blue lighten-2'});              }          }      });  } |

Como se puede observar en este archivo al momento de invocar el clic de guardar, se valida el formulario y se ejecuta el archivo saveData(), en este dependiendo del valor que tenga la caja de texto **idc**, si la caja de texto esta vacía vamos a insertar un nuevo registro, por lo que ejecutamos **Controlador.php** y si la caja **idc** tiene un valor quiere decir que está editando y vamos a ejecutar **Controlador.php**, el código de estos se muestra a continuación:

**Controlador.php**

|  |
| --- |
| <?php        include\_once("../Utilerias/BaseDatos.php");        $post = $\_POST;        $accion = $post["accion"];        $tabla = $post["tabla"];        $funcion = $accion.$tabla;        $result = $funcion($post);        if ($result){              $response['status']=1; // Regresamos 1 por que se agrego con exito              // En data regresamos el contenido de las cajas de texto y el consecutivo              $response['data']=$post;        }        else{              $response['status']=0; // Regresamos 0 por que fallo el insert de clasif              $response['data']=$post;        }        // Convertimos el arreglo response en formato de JSON, para        // poder manipularlo en JAVASCRIPT        echo json\_encode($response);  ?> |