

SISTEMA WEB

MANUAL TÉCNICO

ENVIO DE COMPROBANTES

EXÁMEN UNIDAD 2

YURIDIA GUADALUPE MONTELONGO PADILLA

TECNOLOGÍAS Y APLICACIONES WEB

M.S.I. MARIO HUMBERTO RODRÍGUEZ CHÁVEZ

25 DE JUNIO DEL 2018

INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN
UNIVERSIDAD POLITÉCNICA DE VICTORIA

INTERFACES

Danzlife

[Inicio](#) [Lugares](#) [Ingresar](#)

Formulario de envío de Comprobantes
Festival Verano 2018

Grupo
Seleccione un grupo ▼

Alumna
▼

Nombre de la madre
Nombre: Apellidos:

Fecha de pago

Comprobante de folio
 Ningún archivo seleccionado

Folio de autorización

Inicio. Interfaz que se muestra al inicio de la página web y que contiene el formulario principal para registrar un nuevo comprobante.

Danzlife

[Inicio](#) [Lugares](#) [Ingresar](#)

Orden de envíos de Comprobantes Festival Verano 2018

Show 10 ▼ entries

Search:

Lugar	Folio	Nombre Alumna	Grupo	Nombre Mama	Fecha de Pago	Fecha de Envío
1	9847387	Miguel Hernandez	1B	Marcela Rodríguez	1995-06-11	22-06-2018 21:58:18
2	89347	Mariana Hinojosa	1C	Magdalena Tijerina	2017-11-11	22-06-2018 21:58:56
3	984798743	Dulce Montelongo	1B	Maria del Carmen Padilla	2018-06-22	23-06-2018 02:05:37

Showing 1 to 3 of 3 entries

Previous Next

Lugares. Interfaz que muestra una tabla con los comprobantes registrados en el orden en que se registraron.

Danzlife

Iniciar Sesión

Usuario

Contraseña

Ingresar

Log in de administrador. Interfaz para ingresar a las interfaces del administrador.

Danzlife

Alumnas Registradas

Alumnas Grupos Pagos Salir

Show 10 entries

Search:

Id	Nombre	Apellidos	Grupo	Fecha de Nacimiento		
4	Mariana	Hinojosa	1C	1997-10-17		
6	Maria	Padilla	1C	1998-07-16		
7	Miguel	Hernandez	1B	1995-06-11		
8	Dulce	Montalvo	1B	2000-04-24		

Registro de Alumna

Nombre(s)

Apellidos

1B

Fecha de Nacimiento

dd/mm/aaaa

Aceptar

Alumnas. Interfaz para mostrar las alumnas, modificarlas y eliminarlas. Así como un botón para poder registrarlas.

Grupos Registrados

Show 10 entries

Search:

Id	Nombre		
2	1B		
3	1C		
4			

Registro de Grupo

Nombre del grupo

Aceptar

Grupos. Interfaz que muestra los grupos de alumnas registrados, modificarlos y eliminarlos. Así como un botón para registrarlos.

Pagos Realizados

Show 10 entries

Search:

Id	Alumna	Grupo	Mamá	Fecha de pago	Fecha de envío	Imagen	Folio		
1	Miguel Hernandez	1B	Marcela Rodríguez	1995-06-11	22-06-2018 21:58:18		9847387		
2	Mariana Hinojosa	1C	Magdalena Tijerina	2017-11-11	22-06-2018 21:58:56		89347		
3	Dulce Montelongo	1B	Maria del Carmen Padilla	2018-06-22	23-06-2018 02:05:37		984798743		

Showing 1 to 3 of 3 entries

Previous 1 Next

Pagos. Interfaz que muestra todos los pagos y que permite ver la imagen, editarlos y eliminarlos.

CONTROLADOR

```
//Clase general que controla la comunicación entre la vista y el modelo
Class MvcController{

    //Funcion que carga el template en primera instancia
    public function pagina(){
        include "views/template.php";
    }
}
```

Clase Controller. Contiene todas las funciones que comunican las interfaces con el modelo.

Función pagina(). Incluye el template que contendrá todas las interfaces de l sitio.

```
//Funcion que obtiene la action de la pagina para poder dirigirse a la pagina adecuada con la ayud
enlaces
public function enlacesPaginasController(){
    //Comprueba que el action este en la url de la pagina
    if(isset( $_GET['action'])){
        //Almacena el action en una variable
        $enlaces = $_GET['action'];
    }else{

        //Si no hay action se asigna la palabra index al enlace
        $enlaces = "index";
    }
    //Se manda a llamar el modelo de enlaces para obtener la url completa a la que se va a dirigir
    $respuesta = Paginas::enlacesPaginasModel($enlaces);
    //Carga la url
    include $respuesta;
}
```

Clase enlacesPaginasController(). Contiene la validación de las páginas mediante una variable URL llamada “action”. Una condición comprueba que la variable “action” exista, si esta exista, la guarda en una variable llamada “enlaces”, si no existe la variable “action”, la variable “enlaces” toma como valor la palabra “index” que dirige a la pagina principal de la página que es el registro de comprobantes.

Después de haber cargado la variable “enlaces” se almacena el resultado de un modelo que se encuentra en la clase “Paginas”, llamado **enlacesPaginasModel()** que ejecuta el siguiente código:

```

:php
//Clase ue controla la navegacion del sitio
class Paginas{
    //unico metodo que verifica que action se ejecuto
    public function enlacesPaginasModel($enlaces){

        if($enlaces=="index"){
            $module = "views/modules/inicio.php";
        }else if($enlaces=="fallo"){
            $module = "views/modules/ingresar.php";
        }else{
            $module = "views/modules/".$enlaces.".php";
        }

        //se retorna la pagina a la que se dirige
        return $module;
    }
}

```

Esta función recibe como parámetro el nombre de la variable “action” de la URL, dependiendo de esta variable guardará la ruta de la página a la que se quiere ingresar y lo retornará.

Una vez retornado lo incluirá en el template y lo mostrará.

```

#IDENTIFICACION DE USUARIOS (LOGIN)

```

```

//Funcion controlador que autentica a un usuario
public function ingresoUsuarioController(){
    //Se comprueba que el boton se haya presionado
    if(isset($_POST["ingresar"])){
        //Se almacena el usuario y la contraseña ingresada por el usuario en un array
        $datosController = array ("usuario" => $_POST["usuario"],
                                   "contra" => $_POST["contra"]);
        //Se manda a llamar el modelo que identifica si los datos que ingreso el usuario
        $respuesta = Datos::ingresoUsuarioModel($datosController, "usuarios");
        //Si los datos son encontrados en el modelo se inicia la sesion
        if($respuesta){
            //Se inicia la sesion
            session_start();
            //Se crean las variables de sesion necesarias
            $_SESSION["validar"] = true;
            //Se dirige a la pagina de alumnas
            header("location:index.php?action=alumnas");
        }else{
            //Si los datos no son correctos se dirige al login de nuevo
            header("location:index.php?action=fallo");
        }
    }
}
}

```

La función **ingresoUsuarioController()** valida el acceso de un usuario a las interfaces de un administrador. Primero se valida que el botón de ingresar se haya seleccionado y almacena los valores que se tienen en la caja de texto de usuario y en la de contraseña. Esos datos se mandan como parámetro, junto con el nombre de la tabla en la que buscara el modelo esa cuenta de usuario. Si el modelo encuentra ese registro inicia la sesión y crea una variable de sesión con valor true. Y por último se dirige a la página de alumnas. Si los datos son incorrectos se manda la url con el action igual a "fallo" y este lo captura el modelo de paginas y lo dirige al login de nuevo.

```
//Funcion que ejecuta el modelo que registra un grupo a la b
public function registrarGrupoController(){
    //Se comprueba que se haya seleccionado el boton de acep
    if(isset($_POST["aceptar"])){
        //Se asigna lo ingresado en una variable
        $nombreG = $_POST["nombreG"];
        //Se manda a llamar un modelo que registra el grupo
        $respuesta = Datos::registroGrupoModel($nombreG);
        //Si se ejecuta el modelo se dirige a la pagina de p
        if($respuesta){
            //Se dirige a la pagina de grupos
            header("location:index.php?action=grupos");
        }
    }
}
```

La función **registrarGrupoController()** contiene una primera condición que valida si el botón se seleccionó para tomar lo que contiene la caja de texto del nombre del grupo nuevo y se manda como parámetro al modelo. Si se registro con éxito se guardará un true en una variable y se validará en otra condición. Si entra en esta, se direccionará a la página de grupos.

```
//Funcion que muestra todos los grupos existentes en una tabla
public function vistaGrupoController(){
    //Se manda a llamar el modelo que trae los resultados de la consulta de la tabla grupos
    $respuesta = Datos::vistaGrupoModel();
    //Si el modelo se ejecuta correctamente se imprimen todos los grupos en una tabla
    if($respuesta){
        //Ciclo que recorre el resultado del modelo
        foreach ($respuesta as $fila) {
            //Fila por cada grupo
            echo '<tr>
                <td>'.$fila["id_grupo"].'</td>
                <td>'.$fila["nombre_grupo"].'</td>
                <td><a href="index.php?action=modificargrupo&id='.$fila["id_grupo"].'"class="btn btn-info"><i class="fa fa-edit"></i></a></td>
                <td><a href="index.php?action=eliminargrupo&idBorrar='.$fila["id_grupo"].'"class="btn btn-danger"><i class="fa fa-times"></i></a></td>
            </tr>';
        }
    }
}
```

En la función **vistaGrupoController()** se cargan los registros de la tabla grupos existentes lógicamente en la base de datos y se almacenan en una variable. Si este modelo se ejecuta correctamente se recorre el array asociativo que se genera con el ciclo foreach y dependiendo del número de filas que se hayan encontrado se imprime en forma de fila de una tabla html.

```
//Funcion que elimina un grupo logicamente de la base de datos
public function eliminarGrupoController(){
    //Se comprueba que las variables id y contra esten en la url
    if(isset($_GET["id"]) && isset($_GET["contra"])){
        //Se asigna la variable id a una variable
        $id = $_GET["id"];
        //Se ejecuta el modelo que actualiza el estado de eliminado
        $respuesta = Datos::eliminarGrupoModel($id);
        //Si se ejecuta el modelo se dirige a la pagina principal de
        if($respuesta){
            header("location:index.php?action=grupos");
        }
    }
}
```

En la función **eliminarGrupoController()** se toma el id del grupo en una variable de URL y se manda como parámetro a un modelo que busca y elimina el registro lógicamente, si este modelo se ejecuta correctamente se redirecciona a la página de alumnos.

```
//Funcion que modifica los datos de un grupo
public function modificarGrupoController(){
    //Se comprueba que el boton se haya seleccionado
    if(isset($_POST["modificar"])){
        //Se asigna el id de la url y los datos a modificar en variables
        $id = $_GET["id"];
        $nombre = $_POST["nombreG"];
        //Se ejecuta el modelo que hace el update de los datos
        $respuesta = Datos::modificarGrupoModel($id, $nombre);
        //Si se ejecuta el modelo se dirige a la pagina principal de grupo
        if($respuesta){
            header("location:index.php?action=grupos");
        }
    }
}
```

En la función **modificarGrupoController()** se comprueba que se haya seleccionado el botón, toma el valor de la variable id de la URL, además del valor del texto del input. Después se ejecuta el modelo que actualiza esos campos y si se ejecuta correctamente se dirige a la página de grupos.


```
//Funcion que muestra los datos a editar de un grupo
public function editarGrupoController(){
    //Se verifica que el id del grupo este en la url
    if(isset($_GET["id"])){
        //Se asigna el id en una variable
        $id = $_GET["id"];
        //Se ejecuta el modelo que busca el registro en la tabla grupos con ese id
        $respuesta = Datos::buscarGrupoModel($id);
        //Si se ejecuta el modelo correctamente se imprime el formulario
        if($respuesta){
            echo '<input type="text" name="nombreG" id="nombreG" placeholder="Nombre del grupo" value="'.$respuesta["nombre_grupo"].'" required>
                <input type="submit" class="button tiny" name="modificar" id="modificar" value="Modificar" style="width: 100%;
                background-color: #853BBE" onclick="mod();">';
        }
    }
}
```

En la función **editarGrupoController()** se cargan los datos a la interfaz con el formulario y los datos del grupo que se seleccionó.

```
//Funcion que muestra todas las alumnas registradas en la base datos
public function vistaAlumnasController(){
    //Se ejecuta el modelo que carga todos los registros existentes de alumnas de la base de datos
    $respuesta = Datos::vistaAlumnasModel();
    //Si el modelo se ejecuta correctamente se imprimen los registros en una tabla
    if($respuesta){
        //Ciclo que imprime cada registro como una fila en una tabla
        foreach ($respuesta as $fila) {
            //Se ejecuta el modelo que busca los datos de un grupo en especifico
            $grupo = Datos::buscarGrupoModel($fila["grupo_alumna"]);
            //Se imprimen las alumnas como filas
            echo '<tr>
                <td>'.$fila["id_alumna"].'</td>
                <td>'.$fila["nombre_alumna"].'</td>
                <td>'.$fila["apellidos_alumna"].'</td>
                <td>'.$grupo["nombre_grupo"].'</td>
                <td>'.$fila["fecha_nac"].'</td>
                <td><a href="index.php?action=modificaralumna&id='.$fila["id_alumna"].'" class="btn btn-info"><i class="fa fa-edit"></i></a></td>
                <td><a href="index.php?action=eliminaralumna&idBorrar='.$fila["id_alumna"].'" class="btn btn-danger"><i class="fa fa-times"></i></a></td>
            </tr>';
        }
    }
}
```

En la función **vistaAlumnasController()** se cargan los registros existentes lógicamente desde un modelo a una variable y esa variable será iterada insertando cada registro como una fila de una tabla html.

```

//Funcion que registra un alumna a la base de datos
public function registrarAlumnaController(){
    //Se comprueba que el boton se haya seleccionado
    if(isset($_POST["aceptar"])){
        //Se asigna los valores ingresados a variables
        $nombre = $_POST["n"];
        $apellido = $_POST["a"];
        $grupo = $_POST["grupo"];
        $fechaN = $_POST["fechaNac"];
        //Se ejecuta el modelo que agrega la alumna a la base de datos
        $respuesta=Datos::registrarAlumnaModel($nombre,$apellido,$grupo,$fechaN);
        //Si se ejecuta el modelo se dirige a la pagina principal de alumnas
        if($respuesta){
            header("location:index.php?action=alumnas");
        }
    }
}

```

La función **registrarAlumnaController()** contiene una condición que valida la acción y que carga los datos ingresados en su variable respectiva, esas variables son enviadas como parámetros a un modelo que registra en la base de datos y si se ejecuta correctamente se dirige a la página de alumnas.

```

//Funcion que elimina una alumna logicamente
public function eliminarAlumnaController(){
    //Se comprueba que las variables id y contra esten en la url
    if(isset($_GET["id"]) && isset($_GET["contra"])){
        //Se almacena el id en una variable
        $id = $_GET["id"];
        //Se ejecuta el modelo que elimina una alumna de la base de datos
        $respuesta = Datos::eliminarAlumnaModel($id);
        //Si el modelo se ejecuta correctamente se dirige a l pagina principal de alumnas
        if($respuesta){
            echo"<script language='javascript'>window.location='index.php?action=alumnas';</script>";
        }
    }
}

```

La función **eliminarAlumnaController()** contiene la validación de dos variables en la URL que si se encuentran activas se tomará y se ejecutara el modelo que elimina un registro de alumna lógicamente de la base de datos, si esto se ejecuta correctamente se dirigirá a la página de alumnas.

```
//Funcion que elimina un pago de la base de datos
public function eliminarPagoController(){
    //Se comprueba que la variable id y contra esten el url
    if(isset($_GET["id"]) && isset($_GET["contra"])){
        //Se asigna el id a una variable
        $id = $_GET["id"];
        //Se ejecuta el modelo que elimina logicamente un pago
        $respuesta = Datos::eliminarPagoModel($id);
        //Si se ejecuta el modelo se direcciona a la pagina principal de pagos
        if($respuesta){
            echo"<script language='javascript'>window.location='index.php?action=pagos';</script>";
        }
    }
}
```

La función **eliminarPagoController()** contiene la validación de dos variables en la URL que si se encuentran activas se tomará y se ejecutara el modelo que elimina un registro un pago lógicamente de la base de datos, si esto se ejecuta correctamente se dirigirá a la página de pagos.

```
//Funcion que modifica un pago de la base de datos
public function modificarPagoController(){
    //Se confirma que se haya seleccionado el boton
    if(isset($_POST["modificar"])){
        //Se almacenan los datos en variables
        $id = $_GET["id"];
        $grupo = $_POST["grupo"];
        $alumna = $_POST["alumna"];
        $mNom = $_POST["nombreM"];
        $mApe = $_POST["apellidoM"];
        $fecha_pago = $_POST["fechaPago"];
        $folio = $_POST["folio"];
        //Se ejecuta el modelo que actualiza los datos actuales con los que se acaban de ingresar e
        $respuesta = Datos::modificarPagoModel($id,$grupo,$alumna,$mNom,$mApe,$fecha_pago,$folio);
        //Si la consulta se realiza correctamente se dirige a la pagina principal de pagos
        if($respuesta){
            echo"<script language='javascript'>window.location='index.php?action=pagos';</script>";
        }
    }
}
```

En la función **modificarPagoController()** se comprueba que se haya seleccionado el botón, toma el valor de la variable id de la URL, además del valor del texto del input. Después se ejecuta el modelo que actualiza esos campos y si se ejecuta correctamente se direcciona a la página de pagos.

```

public function editarAlumnaController(){
    //Se comprueba que el id de la alumna este en el url
    if(isset($_GET["id"])){
        //Se asigna el id a una variable
        $id = $_GET["id"];
        //Se ejecuta la consulta que busca una alumna en específico
        $respuesta = Datos::buscarAlumnaModel($id);
        //Si la consulta se ejecuta se imprime el formulario de una alumna
        if($respuesta){
            //Se ejecuta la consulta los grupos existentes
            $grupos = Datos::vistaGrupoModel();
            //Se imprimen los campos con su valor
            echo '<input type="text" name="n" id="n" placeholder="Nombre(s)" value="'. $respuesta["nombre_alumna"].' ">';
            echo '<input type="text" name="a" id="a" placeholder="Apellidos" value="'. $respuesta["apellidos_alumna"].' ">';
            echo '<select class="js-example-basic-single" name="grupo" id="grupo" style="width: 100%">';
            foreach ($grupos as $fila) {
                if($fila["id_grupo"]==$respuesta["grupo_alumna"]){
                    echo '<option value="'. $fila["id_grupo"].' "selected>'. $fila["nombre_grupo"].' </option>';
                }else{
                    echo '<option value="'. $fila["id_grupo"].' ">'. $fila["nombre_grupo"].' </option>';
                }
            }
            echo '</select>';
            echo '<br><br>';
            echo '<label>Fecha de Nacimiento</label>';
            echo '<input type="date" name="fechaNac" id="fechaNac" placeholder="Fecha de nacimiento" value="'. $respuesta["fecha_nac"].' ">';
            echo '<input type="submit" class="button tiny" name="modificar" id="modificar" value="Modificar" style="width: 100%; background-color: #8538BE" onclick="modA();">';
        }
    }
}

```

En la función **editarAlumnasController()** se cargan los datos a la interfaz con el formulario y los datos de la alumna que se haya seleccionado.

```

//Funcion que imprime los grupos en un select2
public function verGruposController(){
    $respuesta = Datos::vistaGrupoModel();
    if($respuesta){
        foreach ($respuesta as $fila) {
            echo '<option value="'. $fila["id_grupo"].' ">'. $fila["nombre_grupo"].' </option>';
        }
    }
}

```

En la función **verGruposController()** contiene una llamada a un modelo que consulta todos los registros en la tabla grupos y los guarda en una variable, si esto se ejecuta correctamente, la variable se recorre en un foreach y se imprime cada uno como una opción de un select2.

```

//Funcion que modifica los datos de una alumna en la base de datos
public function modificarAlumnaController(){
    //Se comprueba que el boton se haya seleccionado
    if(isset($_POST["modificar"])){
        $id = $_GET["id"];
        $nombre = $_POST["n"];
        $apellido = $_POST["a"];
        $grupo = $_POST["grupo"];
        $fechaN = $_POST["fechaNac"];
        //Se ejecuta el modelo que modifica los datos de una alumna
        $respuesta=Datos::modificarAlumnaModel($id,$nombre,$apellido,$grupo,$fechaN);
        //Si el modelo se ejecuta correctamente se dirige a la pagina principal de alumnos
        if($respuesta){
            header("location:index.php?action=alumnas");
        }
    }
}

```

En la función **modificarAlumnaController()** se comprueba que se haya seleccionado el botón, toma el valor de la variable id de la URL, además del valor del texto del input. Después se ejecuta el modelo que actualiza esos campos y si se ejecuta correctamente se direcciona a la página de alumnas.

```

//Funcion que registra un pago a la base de datos
public function registrarPagoController(){
    //Se comprueba que el boton acepta se haya seleccionado
    if(isset($_POST["aceptar"])){
        //Se ejecuta las condiciones que comprueban la imagen que se cargo
        $b=0;
        $ruta="";
        $target_dir = "uploads/";
        $target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
        $uploadOk = 1;
        $imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));
        // Check if image file is a actual image or fake image
        if(isset($_POST["aceptar"])) {
            $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
            if($check !== false) {
                echo "File is an image - " . $check["mime"] . ".";
                $uploadOk = 1;
            } else {
                echo "File is not an image.";
                $uploadOk = 0;
            }
        }
        // Check if file already exists
        if (file_exists($target_file)) {
            echo "Sorry, file already exists.";
            $uploadOk = 0;
        }
        // Check file size
        if ($_FILES["fileToUpload"]["size"] > 500000) {
            echo "Sorry, your file is too large.";
            $uploadOk = 0;
        }
    }
}

```

```

        echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
        $uploadOk = 0;
    }
    // Check if $uploadOk is set to 0 by an error
    if ($uploadOk == 0) {
        echo "Sorry, your file was not uploaded.";
        // if everything is ok, try to upload file
    } else {
        if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $target_file)) {
            $b=1;
            $ruta = basename($_FILES["fileToUpload"]["name"]);
            //echo "The file ". basename( $_FILES["fileToUpload"]["name"]). " has been uploaded.";
        } else {
            echo "Sorry, there was an error uploading your file.";
        }
    }
}
//Si la imagen se cargo se almacenan los datos que se ingresaron
if($b==1){
    $grupo = $_POST["grupo"];
    $alumna = $_POST["alumna"];
    $nMa = $_POST["nombreM"];
    $aMa = $_POST["apellidoM"];
    $fecha_pago = $_POST["fechaPago"];
    $img = $ruta;
    $folio = $_POST["folio"];
    $fecha_envio = date("d-m-Y H:i:s");
    //Se ejecuta el modelo que registra el pago a la base de datos
    $respuesta = Datos::registrarPagoModel($grupo,$alumna,$nMa,$aMa,$fecha_pago,$img,$folio,$fecha_envio);
    //Si se registra el modelo se dirige a la pagina principal de lugares
    if($respuesta){
        echo "<script language='javascript'>window.location='index.php?action=lugares';</script>";
    }
}
}

```

La función **registrarPagoController()** contiene una condición que valida la acción y que intenta cargar la imagen que el usuario está intentando subir, si la imagen es correcta se carga los datos ingresados en su variable respectiva, esas variables son enviadas como parámetros a un modelo que registra en la base de datos y si se ejecuta correctamente se dirige a la página de pagos.

```

//Funcion que muestra los datos del pago a modificar
public function editarPagoController(){
    //Se comprueba que la variable id se encuentre en la url
    if(isset($_GET["id"])){
        //Se almacena el id en una variable
        $id = $_GET["id"];
        //Se ejecuta el modelo que busca los datos de un pago en base al id
        $respuesta = Datos::buscarPagoModel($id);
        //Se comprueba que la consulta se haya realizado correctamente
        if($respuesta){
            //Se ejecuta el modelo que obtiene los datos de todos los grupos
            $grupos = Datos::vistaGrupoModel();
            //Se ejecuta el modelo que obtiene los datos de todas las alumnas
            $alumnas = Datos::vistaAlumnasModel();
            //Se imprimen todos los grupos existentes en un select 2
            echo '<label>Grupo</label>';
            <select class="js-example-basic-single" name="grupo" id="grupo" style="width: 100%" onchange="act();">';
            //Ciclo que recorre el modelo con todas las filas para agregarlo al select2
            foreach ($grupos as $fila) {
                //Si la fila en la que esta es a la que pertenece la alumna se imprime seleccionado
                if($fila["id_grupo"]==$respuesta["id_grupo"]){
                    echo '<option value="'. $fila["id_grupo"].'" selected>'. $fila["nombre_grupo"].'</option>';
                }else{
                    echo '<option value="'. $fila["id_grupo"].'">'. $fila["nombre_grupo"].'</option>';
                }
            }
            echo '</select><br><br>';
            //Se imprimen todos los grupos existentes en un select 2
            echo '<label>Alumna</label>';
            <select class="js-example-basic-single" name="alumna" id="alumna" style="width: 100%">';
            //Ciclo que recorre el modelo con todas las filas para agregarlo al select2
            foreach ($alumnas as $fila) {

```

En la función **editarPagosController()** se cargan los datos a la interfaz con el formulario y los datos del pago que se haya seleccionado.

```

//Funcion que muestra las filas de la tabla de todos los pagos realizados
public function vistaLugaresController(){
    //Se ejecuta un modelo que obtiene todos los pagos almacenados en la base de datos
    $respuesta = Datos::vistaPagosModel();
    //Si el modelo se ejecuta correctamente se imprime cada registro en forma de fila de un
    if($respuesta){
        //Ciclo que recorre cada registro e imprime su informacion
        foreach ($respuesta as $fila) {
            //Modelo que busca la informacion de un grupo en particular
            $grupo = Datos::buscarGrupoModel($fila["id_grupo"]);
            //Modelo que busca la informacion de una alumna en particular
            $alumna = Datos::buscarAlumnaModel($fila["id_alumna"]);
            //Se imprimen los datos en forma de fila
            echo '<tr>
                <td>'.$fila["id_pago"].'</td>
                <td>'.$fila["folio"].'</td>
                <td>'.$alumna["nombre_alumna"].' ' . $alumna["apellidos_alumna"].'</td>
                <td>'.$grupo["nombre_grupo"].'</td>
                <td>'.$fila["nom_mama"].' ' . $fila["ape_mama"].'</td>
                <td>'.$fila["fecha_pago"].'</td>
                <td>'.$fila["fecha_envio"].'</td>
            </tr>';
        }
    }
}

//Funcion que imprime los grupos en un select2

```

En la función **vistaLugaresController()** se cargan los registros existentes lógicamente de todos los pagos que se han realizado desde un modelo a una variable y esa variable será iterada insertando cada registro como una fila de una tabla html.

```

//Funcion que carga todas las alumnas
/**ESTA FUNCION SOLO SE REQUIERE PARA PRECARGAR LOS DATOS DE LAS ALUMNAS PARA QUE SEAN AGREGADOS DINAMICANTE EN EL SELECT2
public function cargarAluController(){
    //Modelo que carga los registros de alumnas existentes logicamente
    $a = Datos::vistaAlumnasModel();
    //Variable que almacena la informacion de cada alumna en cadena
    $al = "";
    foreach ($a as $fila) {
        $al = $al . $fila["id_alumna"] . "," . $fila["nombre_alumna"] . " " . $fila["apellidos_alumna"] . ',' . $fila["
        grupo_alumna"] . "$";
    }
    return $al;
}

```

En la función **cargarAluController()** carga los registros de todas las alumnas existentes lógicamente en la base de datos mediante un modelo que carga sus resultados en una variable y se itera con un foreach para crear una cadena con todas las alumnas concatenadas.

```

//Funcion que muestra todos los pagos existentes
public function vistaPagosController(){
    //Se ejecuta el modelo que trae todos los registros existentes logicamente de pagos en la base de datos
    $respuesta = Datos::vistaPagosModel();
    //Si la consulta se ejecuta se agregan los registros en una tabla
    if($respuesta){
        //Ciclo que agrega los pagos como filas de una tabla
        foreach ($respuesta as $fila) {
            //Se ejecuta el modelo que obtiene los datos de un grupo en particular
            $grupo = Datos::buscarGrupoModel($fila["id_grupo"]);
            //Se ejecuta el modelo que obtiene los datos de una alumna en particular
            $alumna = Datos::buscarAlumnaModel($fila["id_alumna"]);
            //Se imprime la fila con los datos de la alumna
            echo '<tr>
                <td>'.$fila["id_pago"].'</td>
                <td>'.$alumna["nombre_alumna"].' ' . $alumna["apellidos_alumna"].'</td>
                <td>'.$grupo["nombre_grupo"].'</td>
                <td>'.$fila["nom_mama"].' ' . $fila["ape_mama"].'</td>
                <td>'.$fila["fecha_pago"].'</td>
                <td>'.$fila["fecha_envio"].'</td>
                <td><a href="uploads/'.$fila["url"].'"class="btn btn-warning"><i class="fa fa-photo"></a></td>
                <td>'.$fila["folio"].'</td>
                <td><a href="index.php?action=modificarpago&id='.$fila["id_pago"].'"class="btn btn-info"><i class="fa
                    fa-edit"></i></a></td>
                <td><a href="index.php?action=eliminarpago&idBorrar='.$fila["id_pago"].'"class="btn btn-danger"><i
                    class="fa fa-times"></i></a></td>
            </tr>';
        }
    }
}

```

En la función **vistaPagosController()** se cargan los registros existentes lógicamente de todos los pagos que se han realizado desde un modelo a una variable y esa variable será iterada insertando cada registro como una fila de una tabla html. Dentro del ciclo foreach se consultan otros modelos para cargar los datos de grupos y alumnas en base al id del modelo arrojado en el registro del pago.

CRUD

En el archivo crud contiene un `require_once("conexion.php")` que manda a llamar otro archivo que contiene la conexión a la base de datos. Después contiene toda una clase que contiene las funciones necesarias para el controller.

```

k?php
//Se manda a llamar el archivo que contiene la conexión a la base de datos
require_once("conexion.php");

//Clase general que contiene todos los modelos que consume el controlador principal y que existe la clase
encuentra en el modelo conexión
Class Datos extends Conexion{

    //Modelo que consulta los datos que se mandaron desde el controlador en la base de datos
    public function ingresoUsuarioModel($datosModel, $tabla){
        //Se almacenan los datos del array datosModel en variables separadas
        $u = $datosModel["usuario"];
        $c = $datosModel["contra"];
        //Se prepara la consulta
        $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE usuario = '$u' AND pass = '$c'");
        //Se ejecuta la consulta
        $stmt->execute();
        //Se retorna la fila si es que existe
        return $stmt->fetch();
        //se cierra la consulta
        $stmt->close();
    }
}

```

La clase Datos extiende la clase Conexión que contiene la conexión a la base de datos que es necesaria para la ejecución de todos los modelos.

```

//Registra un grupo en la base de datos
public function registroGrupoModel($grupo){
    $stmt = Conexion::conectar()->prepare("INSERT INTO grupos (nombre_grupo) VALUES('$grupo')");
    return $stmt->execute();
    $stmt->close();
}

```

El modelo registroGrupoModel que recibe como parámetro el nombre del grupo nuevo y que realiza una inserción en la base de datos.

```

//Obtiene los grupos de la base de datos
public function vistaGrupoModel(){
    //Se prepara la consulta
    $stmt = Conexion::conectar()->prepare("SELECT * FROM grupos WHERE eliminado=0");
    //Se ejecuta la consulta
    $stmt->execute();
    //Se retorna la fila si es que existe
    return $stmt->fetchAll();
    //se cierra la consulta
    $stmt->close();
}

//Busca un grupo por su id
public function buscarGrupoModel($id){
    $stmt = Conexion::conectar()->prepare("SELECT * FROM grupos WHERE id_grupo = '$id'");
    //Se ejecuta la consulta
    $stmt->execute();
    //Se retorna la fila si es que existe
    return $stmt->fetch();
    //se cierra la consulta
    $stmt->close();
}

//Modifica un grupo
public function modificarGrupoModel($id, $grupo){
    $stmt = Conexion::conectar()->prepare("UPDATE grupos SET nombre_grupo='$grupo' WHERE id_grupo='$id'");
    //Se ejecuta la consulta
    return $stmt->execute();
    //se cierra la consulta
    $stmt->close();
}

```

El modelo vistaGrupoModel realiza una selección de todos los grupos registrados lógicamente en la base de datos y los retorna.

El modelo buscarGrupoModel recibe como parámetro el id de un grupo y ejecuta una selección de ese grupo y los datos que contiene.

El modelo modificarGrupoModel recibe como parámetro el id del grupo y el nombre para poder realizar una actualización de ese registro.

```

//Elimina un grupo logicamente
public function eliminarGrupoModel($id){

    $stmt = Conexion::conectar()->prepare("UPDATE grupos SET eliminado=1 WHERE id_grupo='$id'");
    //Se ejecuta la consulta
    $r = $stmt->execute();
    if($r){
        //Se prepara la segunda consulta
        $stmt2 = Conexion::conectar()->prepare("UPDATE alumnas SET eliminado=1 WHERE grupo_alumna='$id'");
        //Se ejecuta la consulta
        return $stmt2->execute();
        //Se cierra la consulta
        $stmt2->close();
    }
    //se cierra la consulta
    $stmt->close();
}

//Regitra una alumna a la base de datos
public function registrarAlumnaModel($nombre,$apellido,$grupo,$fecha){
    //Se prepara la consulta
    $stmt = Conexion::conectar()->prepare("INSERT INTO alumnas (nombre_alumna,grupo_alumna,apellidos_alumna,fecha_nac) VALUES('$nombre','$grupo','$apellido','$fecha')");
    //Se ejecuta la consulta
    return $stmt->execute();
    //Se cierra la consulta
    $stmt->close();
}

```

El modelo eliminarGrupoModel recibe como parámetro el id de un grupo y actualiza el campo eliminado de esta tabla a 1 para que no se muestre en las interfaces además de actualizar el campo para alumnas también.

El modelo registrarAlumnaModel recibe como parámetro los campos necesarios para realizar una inserción en la base de datos.

```

//Obtiene los datos de las alumnas
public function vistaAlumnasModel(){
    //Se prepara la consulta
    $stmt = Conexion::conectar()->prepare("SELECT * FROM alumnas WHERE eliminado=0");
    //Se ejecuta la consulta
    $stmt->execute();
    //Se retorna la fila si es que existe
    return $stmt->fetchAll();
    //se cierra la consulta
    $stmt->close();
}

//Busca una alumna en la base de datos por su id
public function buscarAlumnaModel($id){

    $stmt = Conexion::conectar()->prepare("SELECT * FROM alumnas WHERE id_alumna = '$id'");
    //Se ejecuta la consulta
    $stmt->execute();
    //Se retorna la fila si es que existe
    return $stmt->fetch();
    //se cierra la consulta
    $stmt->close();
}

//Modifica los datos de una alumna
public function modificarAlumnaModel($id,$nombre,$apellido,$grupo,$fecha){
    $stmt = Conexion::conectar()->prepare("UPDATE alumnas SET nombre_alumna='$nombre', grupo_alumna='$grupo', apellidos_alumna='$apellido', fecha_nac='$fecha' WHERE id_alumna='$id'");
    //Se ejecuta la consulta
    return $stmt->execute();
    //se cierra la consulta
    $stmt->close();
}

```

El modelo vistaAlumnaModel que realiza una selección de todas las alumnas registradas lógicamente.

El modelo buscarAlumnaModel recibe como parámetro el id de una alumna y selecciona sus datos.

El modelo modificarAlumnaModel recibe como parámetro los campos necesarios para realizar una actualización de estos.

```

//Elimina una alumna logicamente de la base de datos
public function eliminarAlumnaModel($id){
    $stmt = Conexion::conectar()->prepare("UPDATE alumnas SET eliminado=1 WHERE id_alumna='$id'");
    //Se ejecuta la consulta
    return $stmt->execute();
    //Se cierra la consulta
    $stmt->close();
}

//Muestra los pagos existentes
public function vistaPagosModel(){
    //Se prepara la consulta
    $stmt = Conexion::conectar()->prepare("SELECT * FROM pagos WHERE eliminado=0");
    //Se ejecuta la consulta
    $stmt->execute();
    //Se retorna la fila si es que existe
    return $stmt->fetchAll();
    //se cierra la consulta
    $stmt->close();
}

//Busca un pago en la base de datos por su id
public function buscarPagoModel($id){
    //Se prepara la consulta
    $stmt = Conexion::conectar()->prepare("SELECT * FROM pagos WHERE id_pago='$id' AND eliminado=0");
    //Se ejecuta la consulta
    $stmt->execute();
    //Se retorna la fila si es que existe
    return $stmt->fetch();
    //se cierra la consulta
    $stmt->close();
}

```

El modelo eliminarAlumnaModel recibe como parámetro el id de una alumna y actualiza sus datos en base a él.

El modelo vistaPagosModel realiza una selección de todos los pagos registrados lógicamente en la base de datos.

El modelo buscarPagoModel recibe como parámetro el id de un pago que selecciona sus datos en base a él.

```

//Elimina un pago logicamente de la base de datos
public function eliminarPagoModel($id){
    $stmt = Conexion::conectar()->prepare("UPDATE pagos SET eliminado=1 WHERE id_pago='$id'");
    //Se ejecuta la consulta
    return $stmt->execute();
    //Se cierra la consulta
    $stmt->close();
}

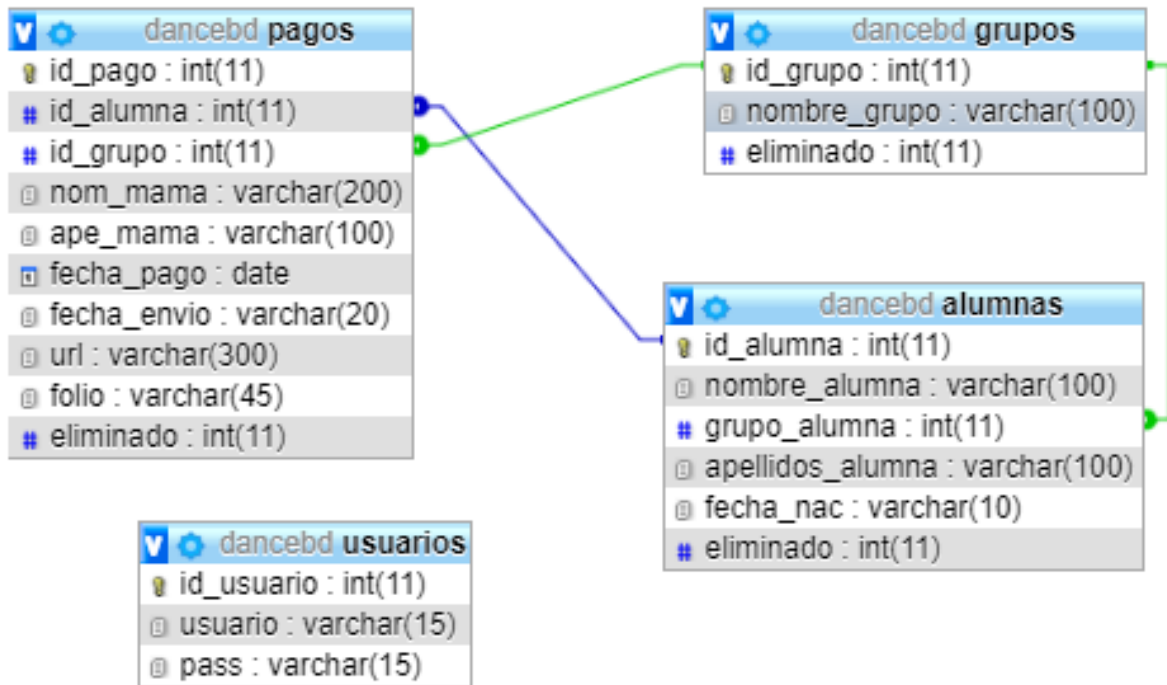
//Registra un pago en la base de datos
public function registrarPagoModel($grupo,$alumna,$nMa,$aMa,$fecha_pago,$img,$folio,$fecha_envio){
    //Se prepara la consulta
    $stmt = Conexion::conectar()->prepare("INSERT INTO pagos (id_alumna, id_grupo, nom_mama, ape_mama, fecha_pago, fecha_envio, url, folio) VALUES ('$alumna', '$grupo', '$nMa', '$aMa', '$fecha_pago', '$fecha_envio', '$img', '$folio')");
    //Se ejecuta la consulta
    return $stmt->execute();
    //Se cierra la consulta
    $stmt->close();
}

```

El modelo eliminarPagoModel recibe como parámetro el id de un pago y actualiza el estado del campo eliminado a 1.

El modelo registrarPagoModel recibe como parámetro todos los campos necesarios para realizar la inserción a la base de datos.

BASE DE DATOS



La base de datos consta de cuatro tablas, tres de ellas están relacionadas (pagos, grupos, alumnas) y la otra es independiente (usuarios).