Clase **DBConexion**

<?php

Class DBConexion {

private $host = "localhost";

private $base = "proyecto";

private $user = "root";

private $pwd = "";

private $charset = "utf8";

protected $pdo;

public $status;

protected function start(){

try {

$this->pdo = new PDO(

'mysql:host='.$this->host.';dbname='.$this->base.';charset='.$this->charset,

$this->user,

$this->pwd

);

$this->pdo->setAttribute(PDO::ATTR\_DEFAULT\_FETCH\_MODE, PDO::FETCH\_OBJ);

$this->status = 202;

return true;

} catch (PDOException $ex) {

$this->status = 503;

return false;

}

}

protected function stop(){

$this->pdo = null;

}

}

?>

Clase **AmigosModelo**

Dentro de esta clase se declaran las variables con el mismo nombre de las columnas dentro de la tabla Amigo. Se utiliza para manejar los resultados de las consultas sobre la tabla de Amigos como un objeto. La función **set** se utiliza para establecer los valores dentro de las instancias del objeto.

<?php

Class AmigosModelo{

public $id\_amigos;

public $estado;

public $id\_usuario1;

public $id\_usuario2;

public function \_\_construct(){

}

public function set( $id\_amigos, $estado, $id\_usuario1, $id\_usuario2){

$this->id\_amigos = $id\_amigos;

$this->estado = $estado;

$this->id\_usuario1 = $id\_usuario1;

$this->id\_usuario2 = $id\_usuario2;

}

}

?>

Dentro de la clase se declaran las variables *id\_amigos, estado*, *id\_usuario1* y *id\_usuario2* como variables públicas, se tiene un constructor vacío y el método **set**.

El método **set** recibe como parámetros las variables *id\_amigos, estado, id\_usuario1* y *id\_usuario2* y dentro le asigna al objeto el valor de las variables por parámetro a las instancias correspondientes de la clase.

Clase **AmigosControlador**

Esta clase es utilizada para manejar toda la información que se relaciona con la tabla de Amigos. La clase se hereda de la clase **DBConexion** y contiene las funciones **ListaAmigos**, **Amigo**, **Estado**, **Solicitudes**, **Agregar**, **Eliminar** y **Aceptar**.

Función **ListaAmigos**

La función **ListaAmigos** utilizada para retornar una lista de objetos **AmigosModelo** de los amigos (solicitudes de amistad aceptadas) que tiene el usuario actual.

Primero se llama la función de **start** que inicia la conexión con la base de datos, a la variable *id\_usuario* se le asigna el valor de la variable de sesión *id\_usuario* Para saber el id del usuario actual*.* Con objeto stmt se llama a la función **prepare** que prepara la sentencia de la consulta a la tabla de amigosdonde el estado sea igual a 1 y el id\_usuario2 o id\_usuario1 sea igual a la variable *id\_usuario*, se llama a la función **execute** que ejecuta la sentencia preparada, se declara la variable *lista* como un array donde se guardara la lista de amigos del resultado de la consulta. Se abre un ciclo while que asigna la siguiente fila de stmt mediante la función **fetch** a la variable *Amigo*, dentro del ciclo se declara un objeto Amigos de la clase **AmigosModelo**, mediante la función **set** se le asignan los valores de *Amigo a* Amigos y este se agrega a la siguiente fila de *lista.*Se termina la conexión llamando a la función **stop** y la función retorna *lista.*

public function ListaAmigos(){

$this->start();

$id\_usuario = $\_SESSION['id\_usuario'];

$stmt = $this->pdo->prepare(

"SELECT \* FROM amigos where estado = 1 and (id\_usuario2 = $id\_usuario or id\_usuario1= $id\_usuario)"

);

$stmt->execute();

$lista = array();

while($Amigo = $stmt->fetch(PDO::FETCH\_ASSOC)):

$Amigos = new AmigosModelo();

$Amigos->set(

$Amigo["id\_amigos"],

$Amigo["estado"],

$Amigo["id\_usuario1"],

$Amigo["id\_usuario2"]

);

$lista[] = $Amigos;

endwhile;

$this->stop();

return $lista;

}

Función **Amigo**

La función **Amigo** es utilizada para comprobar si el usuario actual y otro usuario tienen una interacción de amistad registrada en la tabla de Amigos (son amigos o existe una solicitud de amistad pendiente).

public function Amigo($id\_usuario2){

$this->start();

$id\_usuario1 = $\_SESSION['id\_usuario'];

$stmt = $this->pdo->prepare(

"SELECT \* FROM amigos where ((id\_usuario1 = $id\_usuario1 and id\_usuario2 = $id\_usuario2) or (id\_usuario1 = $id\_usuario2 and id\_usuario2 = $id\_usuario1))"

);

$stmt->execute();

$Amigo = $stmt->fetch(PDO::FETCH\_ASSOC);

$Amigos = new AmigosModelo();

$Amigos->set(

$Amigo["id\_amigos"],

$Amigo["estado"],

$Amigo["id\_usuario1"],

$Amigo["id\_usuario2"]

);

return $Amigos;

}

Función **Estado**

La función **Estado** sirve para que cuando el usuario actual entre al perfil de otra persona, compruebe la relación de amistad que tienen. Si no hay alguna, puede enviar una solicitud para que pueda enviar una, si hay una solicitud pendiente, la marca como pendiente, o si ya son amigos, también muestra esto.

public function Estado($id\_usuario2){

$this->start();

$id\_usuario1 = $\_SESSION['id\_usuario'];

//$id\_usuario2 = $\_GET['id\_usuario'];

$stmt = $this->pdo->prepare(

"SELECT \* FROM amigos where estado = 1 and ((id\_usuario1 = $id\_usuario1 and id\_usuario2 = $id\_usuario2) or (id\_usuario1 = $id\_usuario2 and id\_usuario2 = $id\_usuario1))"

);

$stmt->execute();

if($stmt->rowCount() > 0){

$this->stop();

return 1;

}else{

$stmt = $this->pdo->prepare(

"SELECT \* FROM amigos where estado = 0 and id\_usuario1 = $id\_usuario2 and id\_usuario2 = $id\_usuario1"

);

$stmt->execute();

if($stmt->rowCount() > 0){

$this->stop();

return 2;

}else{

$stmt = $this->pdo->prepare(

"SELECT \* FROM amigos where estado = 0 and id\_usuario1 = $id\_usuario1 and id\_usuario2 = $id\_usuario2"

);

$stmt->execute();

if($stmt->rowCount() > 0){

$this->stop();

return 3;

}else{

$this->stop();

return 0;

}

}

}

}

Función **Solicitudes**

La función de **Solicitudes** se utiliza para que cuando el usuario actual dé click en Solicitudes en su perfil, abra la lista de solicitudes de amistad de otros usuarios.

public function Solicitudes(){

$this->start();

$id\_usuario = $\_SESSION['id\_usuario'];

$stmt = $this->pdo->prepare(

"SELECT \* FROM amigos where estado = 0 and id\_usuario2 = $id\_usuario"

);

$stmt->execute();

$lista = array();

while($Amigo = $stmt->fetch(PDO::FETCH\_ASSOC)):

$Amigos = new AmigosModelo();

$Amigos->set(

$Amigo["id\_amigos"],

$Amigo["estado"],

$Amigo["id\_usuario1"],

$Amigo["id\_usuario2"]

);

$lista[] = $Amigos;

endwhile;

$this->stop();

return $lista;

}

Función **Agregar**

La función **Agregar** inserta una nueva relación de amistad en la tabla de Amigos cuando el usuario actual acepta una solicitud de amistad.

public function Agregar(){

$this->start();

$id\_usuario1 = $\_SESSION['id\_usuario'];

$id\_usuario2 = $\_GET['id\_usuario'];

$stmt = $this->pdo->prepare(

"INSERT INTO amigos VALUES( NULL,0,$id\_usuario1,$id\_usuario2)"

);

$stmt->execute();

$this->stop();

header("Location: Control.php?c=Perfiles&a=Perfiles&id=$id\_usuario2");

}

Función **Eliminar**

La función **Eliminar** se utiliza para que cuando el usuario actual se encuentre en el perfil de uno de sus amigos y dé click en Eliminar amigo, se borre la relación de amistad.

public function Eliminar(){

$this->start();

$id\_usuario = $\_GET['id\_usuario'];

$id\_amigos = $\_GET['id'];

$stmt = $this->pdo->prepare(

"DELETE FROM amigos where id\_amigos= $id\_amigos"

);

$stmt->execute();

$this->stop();

header("Location: Control.php?c=Perfiles&a=Perfiles&id=$id\_usuario");

}

Función **Aceptar**

La función **Aceptar** se utiliza para que cuando un usuario mande una solicitud de amistad al usuario actual, este pueda aceptar esa Solicitud con el botón Aceptar.

public function Aceptar(){

$this->start();

$id\_usuario = $\_GET['id\_usuario'];

$id\_amigos = $\_GET['id'];

$stmt = $this->pdo->prepare(

"UPDATE amigos SET estado = 1 where id\_amigos= $id\_amigos"

);

$stmt->execute();

$this->stop();

header("Location: Control.php?c=Perfiles&a=Perfiles&id=$id\_usuario");

}

}

?>

Clase **ArtistaModelo**

<?php

Class ArtistaModelo {

public $id\_artista;

public $fn;

public $imagen\_perfil;

public $informacion\_contacto;

public $tecnica\_interes;

public $id\_pais;

public $id\_usuario;

public $id\_diseno;

public $id\_portafolio;

public $id\_perfil;

public function \_\_construct(){

}

public function set( $id\_artista, $fn, $imagen\_perfil, $informacion\_contacto, $tecnica\_interes, $id\_pais, $id\_usuario, $id\_diseno, $id\_portafolio, $id\_perfil){

$this->id\_artista = $id\_artista;

$this->fn = $fn;

$this->imagen\_perfil = $imagen\_perfil;

$this->informacion\_contacto = $informacion\_contacto;

$this->tecnica\_interes = $tecnica\_interes;

$this->id\_pais = $id\_pais;

$this->id\_usuario = $id\_usuario;

$this->id\_diseno = $id\_diseno;

$this->id\_portafolio = $id\_portafolio;

$this->id\_perfil = $id\_perfil;

}

}

?>

Clase **ArtistaControlador**