

Asier F

```
<?php
ob_start();
session_start();
require_once './classes/libro.php';
require_once './classes/autor.php';
require_once './classes/genero.php';
$autores = Autor::obtenerTodos();
$generos = Genero::obtenerTodos();

$_SESSION['usuario'] = $usuario;
if (isset($_POST['registrar']))
isset($_GET['id']);

//$_SERVER['REQUEST_METHOD'] === 'POST')
//$nombre = $_POST['nombre'];

$sentencia = $pdo->prepare($sql);
$sentencia->setFetchMode(PDO::FETCH_ASSOC);
$sentencia->execute();

$libros = [];
while($t = $sentencia->fetch())
?>
```

Xabi:

Funcion Obtener Todos()

```
// esta funcion estatica obtenerTodos nos permite obtener todos los
autores de la base de datos
// es estatica porque no necesitamos crear una instancia de la
clase para usarla
// podemos llamarla directamente como Autor::obtenerTodos()
public static function obtenerTodos()
{
    // obtenemos la instancia de la conexion a la base de datos
    $pdo = Database::getInstance();
    // preparamos la consulta sql para seleccionar todos los
autores
    $stmt = $pdo->prepare("SELECT * FROM autores");
    // ejecutamos la consulta
    $stmt->execute();
```

```

        // creamos un array vacío para guardar los autores, SI NO LO
        HACES TE LOS CARGAS CADA VEZ
        $autores = [];
        // recorremos cada fila del resultado, aunque la parte de
        PDO::FETCH_ASSOC es opcional
        while ($row = $stmt->fetch(PDO::FETCH_ASSOC /*Esto es
        opcional*/)) {
            // por cada fila creamos un nuevo objeto Autor y lo
            guardamos en el array
            // usando los datos de la fila id y nombre
            $autores[] = new Autor($row['id'], $row['nombre']);
        }
        // devolvemos el array con todos los objetos Autor
        return $autores;
    }
}

```

function buscar()

```

//buscamos los libros que coincidan con el autor y genero que nos pasan
desde el formulario BUSCAR_LIBRO.PHP
$libros = Libro::buscar($_POST['id_autor'],$_POST['id_genero']);
//si no hay libros que coincidan con la busqueda ponemos un array
vacío
if(!isset($libros)){
    $libros = [];
}
//si no es post significa que es la primera vez que entramos y ponemos
un array vacío
}else{
    $libros = [];
}

public static function buscar($id_autor = null, $id_genero = null){

    //obtenemos la instancia de la conexion a la base de datos

    //preparamos la consulta sql para buscar libros por id de autor e id de
    genero

    $sentencia -> bindParam(":id_autor",$id_autor);

```

```

        $sentencia -> bindParam(":id_genero",$id_genero);
// ejecutamos la consulta

        o

        $sentencia ->
execute([$arrayEnOrden['primerValor'],'segundoValor']);
    }

```

function Insertar()

```

//agregar el libro que se pasa desde el formulario_agregar_libro.php

Libro::agregar($_POST['titulo'], $_POST['id_autor'],
$_POST['id_genero'], $_POST['fecha_publicacion']);

public static function agregar(

    $titulo,

    $id_autor,

    $id_genero,

    $fecha_publicacion

) {

// funcion que sirve para meter un libro nuevo en la base de datos
//obtenemos la instancia de la conexion a la base de datos
$password" => password_hash($password,PASSWORD_DEFAULT)

```

Métodos CRUD

Gene Iñigo

ObtenerTodos()

1. conectar a la BBDD
2. sentencia sql
3. crear array[]
4. con un while recorrer las diferentes líneas de la query
5. devolver el array

```
$pdo = Database::getInstance();
$sql = "SELECT * FROM [tabla]";
$sentencia = $pdo->query($sql);
$array = [];
while ($row = $sentencia->fetch(PDO::FETCH_ASSOC)) {
    $obj1 = new Objeto1($row['parametro1'], $row['parametro2']);
    $obj2 = new Objeto2($row['parametro1'], $row['parametro2']);
    $array[] = new Objeto($row['parametro1'], $row['parametro2'],
    $obj1, $obj2,);
}
return $array;
```

Insert()

1. la función recibe los atributos
2. conexión a la BBDD
3. sentencia sql
4. Rellenar los parámetros
5. devolver la ejecución de la sentencia sql

Ejemplo:

```
public static function insert($nombre,$apellido) {
    $pdo = Database::getInstance();
    $sql = "INSERT INTO tabla (nombre, apellido) VALUES (:nombre,
:apellido)";
    $sentencia = $pdo->prepare($sql);
    $sentencia->bindParam(':nombre', $nombre);
    $sentencia->bindParam(':apellido', $apellido);
    return $sentencia->execute();
}
```


ObtenerUno()

1. recibir los parametros de busqueda
2. sentencia sql
3. conexión a BBDD
4. rellenar los parámetros
5. ejecutar la sentencia
6. rellenar el array con los resultados de la búsqueda

Ejemplo:

```
public static function buscar($nombre = null, $apellido = null) {  
    $sql = "Select WHERE ...";  
    $pdo = Database::getInstance();  
    $sentencia = $pdo->prepare($sql);  
    $sentencia->bindParam(':nombre', $nombre);  
    $sentencia->bindParam(':apellido', $apellido);  
    $sentencia->execute();  
    $objetos = [];  
    while ($row = $sentencia->fetch(PDO::FETCH_ASSOC)) {  
$objetos[] = new Objeto($row['id'], $row['nombre'], $row['apellido']);  
    }  
    return $objetos;  
}
```

Consultas

PDO::FETCH_ASSOC: array indexado cuyos keys son el nombre de las columnas.

PDO::FETCH_NUM: array indexado cuyos keys son números.

PDO::FETCH_BOTH: valor por defecto. Devuelve un array indexado cuyos keys son tanto el nombre de las columnas como números.

Login y password

`password_hash()` para almacenar la contraseña en la base de datos a la hora de hacer el INSERT

PASSWORD_DEFAULT almacenamos la contraseña usando el método de encriptación `bcrypt`

PASSWORD_BCRYPT almacenamos la contraseña usando el algoritmo `CRYPT_BLOWFISH` compatible con `crypt()`

`password_verify()` para verificar el usuario y la contraseña

Sesiones

```
session_start(); // carga la sesión
session_id() // devuelve el id
$_SESSION[clave] = valor; // inserción
session_destroy(); // destruye la sesión
unset($_SESSION[clave]); // borrado
```

Comprobación doble

```
<?php
if (isset($_GET["parametro"])) {
    $par = $_GET["parametro"];
    // comprobar si $par tiene el formato adecuado, su valor, etc...
}
```

Operaciones con Bases de Datos usando PDO en PHP

Conexión a la Base de Datos usando PDO

La conexión a la base de datos es esencial para realizar operaciones como inserción, eliminación, actualización y selección de registros. A continuación, se detalla cómo establecer una conexión segura y eficiente:

1. Inserción de Datos en la Base de Datos

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "GET") {
    $nombre = $_GET["nombre"] ;
    // SQL para insertar datos

    $sentencia = $pdo->prepare($sql);
    $sentencia->bindParam(":nombre", $nombre);

    // Ejecutar la sentencia
    $isOk = $sentencia->execute();

    // Obtener el último ID insertado
    $idGenerado = $pdo->lastInsertId();
?>
```

2. Eliminación de Datos de la Base de Datos

```
$sentencia->bindParam(":name", $nombre);

// Ejecutar la eliminación
```

```
$isOk = $sentencia->execute();  
// Obtener el número de filas afectadas  
$cantidadAfectada = $sentencia->rowCount();
```

3. Actualización de Datos en la Base de Datos

```
$sentencia = $pdo->prepare($sql); // hay que usar bind : para los atributos  
// Ejecutar la actualización  
$isOk = $sentencia->execute();  
}  
?>
```

4. Selección de Datos de la Base de Datos CLASS

objetos de la clase Cliente
\$sentencia->execute();

5. Manejo de Contraseñas

Hashing de Contraseñas:

```
password => password_hash($password, PASSWORD_DEFAULT) // Contraseña hasheada
```

Verificación de Contraseñas:

```
// Verificar si la contraseña es correcta  
password_verify($password, $usuario['password'])
```

6. Manejo de Sesiones

```
<?php  
session_start(); // Inicia la sesión  
session_destroy();  
// Comprobar si el usuario está logueado  
if (!isset($_SESSION['usuario'])) {  
    header('Location: login.php'); // Redirigir al login si no está logueado  
    exit();  
}  
?>
```

7. Manejo de Sesiones Fechas

```
$fecha_nueva = Date("Y-m-d", strtotime($fecha_publicacion));
```

Apuntes de Iker Lourido (CHIVATO EL QUE LO BORRE)

para llamar a un metodo publico de otra clase:

```
$libros = Libro::buscar($id_autor,$id_genero);
```

ejemplo de llamada a base de datos:

```
$pdo = Database::getInstance();  
$query = "SELECT * FROM autores";  
$stmt = $pdo->prepare($query);  
$stmt->execute();  
  
$resultado = $stmt->fetchAll();
```

ejemplo de llamada a base de datos usando prepare:

```
$pdo = Database::getInstance();  
$query = "INSERT INTO libros(titulo) VALUES (:titulo)";  
$stmt = $pdo->prepare($query);  
$stmt->bindParam(":titulo", $titulo);  
  
$stmt->execute();
```


ejemplo codificación:

```
$contraseña = contraseña;
```

```
$cCodificado = password_hash($contraseña,PASSWORD_DEFAULT);
```

```
password_verify($contraseña,$cCodificado)
```

Gorka

Ejemplo Obtener() :

```
$pdo = Database::getInstance();
```

```
$sql = "";
```

```
$sentencia = $pdo->prepare($sql);
```

```
$sentencia->execute();
```

```
$libros = [];
```

```
while ($fila = $sentencia->fetch()) {
```

```
//se llena el array
```

```
}
```

```
return $libros;
```

```
}
```

Ejemplo Agregar ()

```
public static function
```

```
agregar($titulo,$id_autor,$id_genero,$fecha_publicacion) {
```

```
$pdo = Database::getInstance();
```

```
$sql = "//insert into tabla (,,) values (:, :, :, :)";
```

```
$sentencia = $pdo->prepare($sql);
```

```
$sentencia->execute([]);
```

```
}
```

Ejemplo Buscar()

```
public static function buscar(//valor que sirve para buscar)
```

```
{
```

```
$pdo = Database::getInstance();
```

```
$sql = "";
```

```
$sentencia = $pdo->prepare($sql);
```

```
$sentencia->execute(//valores que sirven para buscar(se
```

```
escribe como el resto de executes]]);
```

```
while ($fila = $sentencia->fetch()) {
```

```
//array = fila["valor que corresponde"]
```

```
}
```

```
return $libros;
```

```
}
```

Ejemplo Eliminar()

```
public static function eliminar(//identificante)
```

```
{
```

```
$pdo = Database::getInstance();
```

```
$sql = "";
```

```
$sentencia = $pdo->prepare($sql);
```

```
$sentencia->execute([':id_libro' => $id_libro]);
```

```
}
```

Ejemplo Modificar()

```
public static function modificar(//todos los valores a modificar y un
```

```
identificante)
{
    $pdo = Database::getInstance();
    $sql = "UPDATE //tabla
    SET //valores
    WHERE //identificante ;
    $sentencia = $pdo->prepare($sql);
    $sentencia->execute([
    ':titulo' => $titulo,]);
```