

El patrón MVC en la práctica

(UD 4)

Nuestro ejemplo

Mostrar lista de artículos de un blog

Código monolítico

```
<?
$db = new mysqli('localhost', 'usuario', 'clave', 'dbname');
$resultado=$db->query('SELECT fecha, titulo FROM articulo');
?>
<h1>Listado de Artículos</h1>
<table>
    <tr> <th>Fecha</th> <th>Titulo</th> </tr>
<?php
while ($fila = $resultado->fetch_array()) {
    echo "<tr>";
    echo "<td> ".$fila['fecha']." </td>";
    echo "<td> ".$fila['titulo']." </td>";
    echo "</tr>";
}
echo "</table>";
$db->close();
?>
```

Primera mejora

Dividir el código en:

- Un controlador.
- Una vista.

Primera mejora: controlador + vista

controlador.php

```
<?
// Este es el controlador (esta aplicación solo realiza una acción)

$db = new mysqli('localhost', 'usuario', 'clave', 'dbname');
$resultado=$db->query('SELECT fecha, titulo FROM articulo');

$articulos = array();
while ($fila = $resultado->fetch_array()) {
    $articulos[] = $fila;
}

$db->close();
?>

include('vista.php');
?>
```

Primera mejora: controlador + vista

vista.php

```
<!-- Esta es la vista -->
```

```
<h1>Listado de Articulos</h1>
```

```
<table>
```

```
  <tr> <th>Fecha</th> <th>Titulo</th> </tr>
```

```
<?php foreach($articulos as $articulo): ?>
```

```
  <tr>
```

```
    <td><?php echo $articulo['fecha']?></td>
```

```
    <td><?php echo $articulo['titulo']?></td>
```

```
  </tr>
```

```
<?php endforeach;?>
```

```
</table>
```

Segunda mejora

Dividir el código en:

- Un controlador.
- Una vista.
- Un modelo.

Segunda mejora: controlador + vista + modelo

controlador.php

// Este es el controlador (solo realiza una acción)

include('articulos.php'); // En este archivo estará el modelo
\$articulos = Articulos::getAllArticulos();

require('vista.php'); // En este archivo estará la vista

Segunda mejora: controlador + vista + modelo

articulos.php

```
<?php
class Articulos {
    public function getArticulos()
    {
        $db = new mysqli('localhost', 'usuario', 'clave', 'dbname');
        $resultado=$db->query('SELECT fecha, titulo FROM articulo');

        $articulos = array();
        while ($fila = $resultado->fetch_array()) {
            $articulos[] = $fila;
        }

        $db->close();

        return $articulos;
    }
}
?>
```

Segunda mejora: controlador + vista + modelo

vista.php

```
<h1>Listado de Articulos</h1>
<table>
  <tr> <th>Fecha</th> <th>Titulo</th> </tr>
  <?php foreach($articulos as $articulo): ?>
    <tr>
      <td><?php echo $articulo['fecha']?></td>
      <td><?php echo $articulo['titulo']?></td>
    </tr>
  <?php endforeach;?>
</table>
```

Tercera mejora

Dividir el código en:

- Un controlador.
- Una vista.
- Un modelo en dos capas:
 - Capa de abstracción de datos.
 - Capa de acceso a datos.

Tercera mejora: controlador + vista + modelo en 2 capas

DbAbstract.php

```
class DbAbstract {
    private $db;

    function crearConexion($servidor, $usuario, $clave, $dbname) {
        $db = new mysqli($servidor, $usuario, $clave, $dbname);
    }

    function cerrarConexion() {
        if ($db) $db->close();
    }

    function consulta($consulta) {
        $res = $db->query($consulta);
        $resArray = array();
        if ($res) {
            $resArray = $res->fetch_all();
        }
        return $resArray;
    }
    ...etc... (añadimos cualquier función que acceda directamente a MySQL)
}
```

Tercera mejora: controlador + vista + modelo en 2 capas

Articulos.php

```
include "DbAbstract.php";
```

```
class Articulos {
```

```
    public function getAllArticulos() {
```

```
        $db = new dbAbstract();
```

```
        $db->crearConexion('localhost', 'usuario', 'clave','dbName');
```

```
        $articulos = $db->consulta('SELECT fecha, titulo FROM articulo');
```

```
        $db->cerrarConexion();
```

```
        return $articulos;
```

```
    }
```

```
}
```

Cuarta mejora

Implementar el código siguiendo el paradigma de orientación a objetos.

- Crear clase Controlador.
- Crear clase Vista.

Cuarta mejora: orientación a objetos

index.php

```
$controlador = new Controlador();  
$controlador->main();
```

Cuarta mejora: orientación a objetos

Controlador.php

```
include ("Vista.php");
include ("Articulos.php");

class Controlador {

    public function main() {
        $estado = (isset($_REQUEST['do']) ? $_REQUEST['do'] ; "formLogin");

        switch ($estado) {
            case "formLogin": ...
            case "checkLogin": ...
            case "showAllArticles":
                $articulos = Articulos::getAllArticulos();
                Vista::show("showAllArticles");
                break;
            case "...etc...": ...
        }
    }
}
```


Cuarta mejora: orientación a objetos

Vista.php

```
<?php
```

```
class Vista
```

```
{
```

```
    public function show($nombreVista) {
```

```
        include("header.php");
```

```
        include("$nombreVista.php");
```

```
        include("footer.php");
```

```
    }
```

```
}
```

```
?>
```

Cuarta mejora: orientación a objetos

showAllArticles.php

```
<h1>Listado de Articulos</h1>
<table>
  <tr> <th>Fecha</th> <th>Titulo</th> </tr>
  <?php foreach($articulos as $articulo): ?>
    <tr>
      <td><?php echo $articulo['fecha']?></td>
      <td><?php echo $articulo['titulo']?></td>
    </tr>
  <?php endforeach;?>
</table>
<?php
}
}
```

Cuarta mejora: orientación a objetos

Articulos.php

```
include "DbAbstract.php";
```

```
class Articulos {
```

```
    public function getAllArticulos() {
```

```
        $db = new dbAbstract();
```

```
        $db->crearConexion('localhost', 'usuario', 'clave','dbName');
```

```
        $articulos = $db->consulta('SELECT fecha, titulo FROM articulo');
```

```
        $db->cerrarConexion();
```

```
        return $articulos;
```

```
    }
```

```
}
```

Cuarta mejora: orientación a objetos

DbAbstract.php

```
class DbAbstract {
    private $db;
    function crearConexion($servidor, $usuario, $clave, $dbname) {
        $db = new mysqli($servidor, $usuario, $clave, $dbname);
    }

    function cerrarConexion() {
        if ($db) $db->close();
    }

    function consulta($consulta) {
        $res = $db->query($consulta);
        $resArray = array();
        if ($res) {
            $resArray = $res->fetch_all();
        }
        return $resArray;
    }
    ...etc... (añadimos cualquier función que acceda directamente a MySQL)
}
```