Programación en Capas

En algunos ejercicios habrás visto como en ocasiones se ha mezclado el código propio de la lógica de la aplicación, con el código necesario para crear el interface web que se presenta a los usuarios. Paginas que deberían generar sólo HTML llevan sentencias PHP y al contrario.

Existen varios métodos que permiten separar la lógica de presentación (en nuestro caso, la que genera las etiquetas HTML) de la lógica de negocio, donde se implementa la lógica propia de cada aplicación. El más extendido es el patrón de diseño Modelo – Vista – Controlador (MVC). Este patrón pretende dividir el código en tres partes, dedicando cada una a una función definida y diferenciada de las otras.

Modelo. Es el encargado de manejar los datos propios de la aplicación. Debe proveer mecanismos para obtener y modificar la información del mismo. Si la aplicación utiliza algún tipo de almacenamiento para su información (como un SGBD), tendrá que encargarse de almacenarla y recuperarla.

Vista. Es la parte del modelo que se encarga de la interacción con el usuario. En esta parte se encuentra el código necesario para generar el interface de usuario (en nuestro caso en HTML), según la información obtenida del modelo.

Controlador. En este módulo se decide qué se ha de hacer, en función de las acciones del usuario con su interface. Con esta información, interactúa con el modelo para indicarle las acciones a realizar y, según el resultado obtenido, envía a la vista las instrucciones necesarias para generar el nuevo interface.

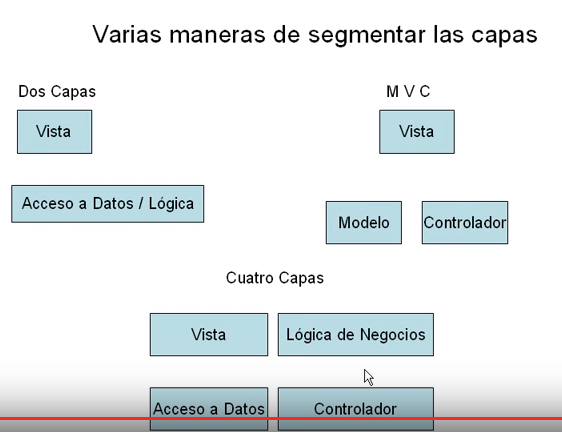
La gran ventaja de este patrón de programación es que genera código muy estructurado, fácil de comprender y de mantener.

Se llama arquitectura en 3 capas a la forma de construir una aplicación distribuyendo sus componentes en capas lógicas según su funcionalidad. La arquitectura básica es la denominada de 3 capas:



<https://www.youtube.com/watch?v=plr1H4rFg_g>

<https://www.youtube.com/watch?v=lmP_WOqUqOE>



Ejemplo:

<?php

$conexion=mysqli\_connect("localhost","root","", "estudiantes");

$resultado = mysqli\_query($conexion,'SELECT nombre, telefono FROM estudiantes');

?>

<html>

<head>

<title>**Listado de Estudiantes**</title>

</head>

<body>

<h1>**Listado de Estudiantes**</h1>

<table>

<tr><th>**Nombre**</th><th>**Telefono**</th></tr>

<?php

**while** ($fila = mysqli\_fetch\_array($resultado)){

**echo** "\t<tr>\n";

**printf**("\t\t<td> %s </td>\n", $fila['nombre']);

**printf**("\t\t<td> %s </td>\n", $fila['telefono']);

**echo** "\t</tr>\n";

}

?>

</table>

</body>

</html>

<?php

// Cerrar la conexion

mysqli\_close($conexion);

?>

El código anterior es difícil de mantener y actualizar.

Los principales problemas del código anterior son:

* No existe protección frente a errores (¿qué ocurre si falla la conexión con la base de datos?).
* El código HTML y el código PHP están mezclados en el mismo archivo e incluso en algunas partes están entrelazados.
* El código solo funciona si la base de datos es MySQL.

El código se va a dividir en 2 partes; separando la presentación:

1.- La lógica de negocio que se incluye en el script del controlador:

index.php

<?php

$conexion=mysqli\_connect("localhost","root","", "estudiantes");

$resultado = mysqli\_query($conexion,'SELECT nombre, telefono FROM estudiantes');

// crear el array de elementos para la capa de la vista

$est=**array**();

**while** ($fila = mysqli\_fetch\_array($resultado)){

$est[]=$fila;

}

mysqli\_close($conexion);

**require**('vista.php');

?>

2.- La vista

vista.php

<html>

<head>

<title>**Listado de Estudiantes**</title>

</head>

<body>

<h1>**Listado de Estudiantes**</h1>

<table>

<tr><th>**Nombre**</th><th>**Telefono**</th></tr>

<?php **foreach** ($est **as** $estu){ ?>

<tr>

<td><?php **echo** $estu['nombre'] ?></td>

<td><?php **echo** $estu['telefono'] ?></td>

</tr>

<?php } ?>

</table>

</body>

</html>

Una buena regla general para determinar si la parte de la vista está suficientemente limpia de código es que debería contener una cantidad mínima de código PHP, la suficiente como para que un diseñador HTML sin conocimientos de PHP pueda entenderla. Las instrucciones más comunes en la parte de la vista suelen ser echo, if/else, foreach y poco más. Además, no se deben incluir instrucciones PHP que generen etiquetas HTML.

Otra división; eliminar del controlador todo el código que se encarga de la manipulación de los datos y ponerlo en otro script correspondiente al nivel modelo.

Nivel modelo:

Modelo.php

<?php

**function** getTodosLosEstudiantes(){

$conexion=mysqli\_connect("localhost","root","", "estudiantes");

$resultado = mysqli\_query($conexion,'SELECT nombre, telefono FROM estudiantes');

// crear el array de elementos para la capa de la vista

$est=**array**();

**while** ($fila = mysqli\_fetch\_array($resultado)){

$est[]=$fila;

}

mysqli\_close($conexion);

**return** $est;

}

?>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Nivel controlador

Index.php

<?php

**require\_once**('modelo.php');

$est=getTodosLosEstudiantes();

**require**('vista.php');

?>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Nivel Vista

Vista.php

El mismo del ejercicio anterior.

Ejercicio:

Las capas del modelo, la vista y el controlador se pueden dividir en más capas.

La capa del modelo se puede dividir en la capa de acceso a los datos y en la capa de abstracción de la base de datos. De esta forma, las funciones que acceden a los datos no utilizan sentencias ni consultas que dependen de una base de datos, sino que utilizan otras funciones para realizar las consultas. Así, si se cambia de sistema gestor de bases de datos, solamente es necesario actualizar la capa de abstracción de la base de datos.

Dividir modelo.php del ejercicio anterior en 2: capa de acceso a los datos y capa de abstracción de la base de datos.

La capa de acceso a datos no contiene funciones dependientes de un sistema gestor de bases de datos, por lo que es independiente de la base de datos utilizada.

Además, las funciones creadas en la capa de abstracción de la base de datos se pueden reutilizar en otras funciones del modelo que necesiten acceder a la base de datos.

Ejercicio:

También podemos dividir la capa de la vista de la siguiente manera:

miplantilla.php

<h1>**Listado de Estudiantes**</h1>

<table>

<tr><th>**Nombre**</th><th>**Telefono**</th></tr>

<?php **foreach** ($est **as** $estu){ ?>

<tr>

<td><?php **echo** $estu['nombre'] ?></td>

<td><?php **echo** $estu['telefono'] ?></td>

</tr>

<?php } ?>

</table>

logica.php

<?php

$titulo = 'Listado de Estudiantes';

$contenido = 'miplantilla.php';

?>

layout.php

<html>

<head>

<title><?php **echo** $titulo ?></title>

</head>

<body>

<?php **include($contenido)** ?>

</body>

</html>

Comprobar el ejercicio entero modificando vista.php por las 3 partes anteriores.