UT3-2 Guiones de servidor PHP

Desarrollo en entorno servidor

Patrones de diseño

El Objetivo de este proyecto es construir un conjunto de carpetas y archivos que cumplan los objetivos de los Patrones de diseño:

- Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Formalizar un vocabulario común entre diseñadores.
- Estandarizar el modo en que se realiza el diseño.
- Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.

Asimismo, no pretenden:

- Imponer ciertas alternativas de diseño frente a otras.
- Eliminar la creatividad inherente al proceso de diseño.

(wikipedia)

Patrones de diseño

Una vez trabajadas las siguientes presentaciones:

Fase 1: Modelo-Vista-Controlador. Patrón MVC.

Fase 2: Saneamiento y Validación de formularios.

Fase 3: Conexión a BD con PDO. Patrón DAO.

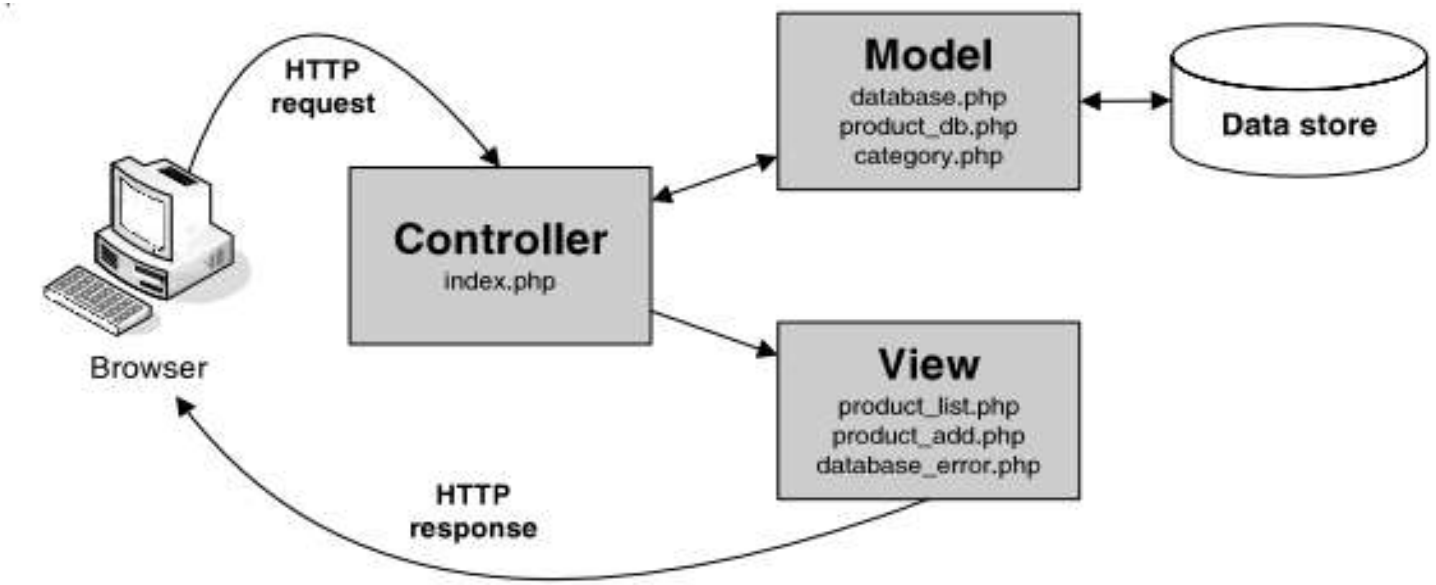
Se obtendrá un conjunto de carpetas, archivos, es decir una estructura/armazón para construir una aplicación web en PHP de 0 utilizando los patrones de Diseño MVC, DAO y la extensión PDO.

Organización

- **Organizar** adecuadamente el código de nuestra aplicación web para que sea más fácil de leer y mantener:
  - A través de sentencias include separamos el código HTML del código PHP (cabecera.php, pie.php) e introdujimos el concepto de plantilla (template) que nos ayudó a organizar inicialmente nuestro código.
- Intentaremos, a partir de ahora, realizar un diseño de nuestras aplicaciones que se acerque lo más posible al patrón MVC (modelo-vista-controlador)

MVC – Modelo Vista Controlador

Patrón MVC



Patrón de diseño utilizado para estructurar aplicaciones web y hacer más fácil su codificación y mantenimiento

MVC – Modelo Vista Controlador

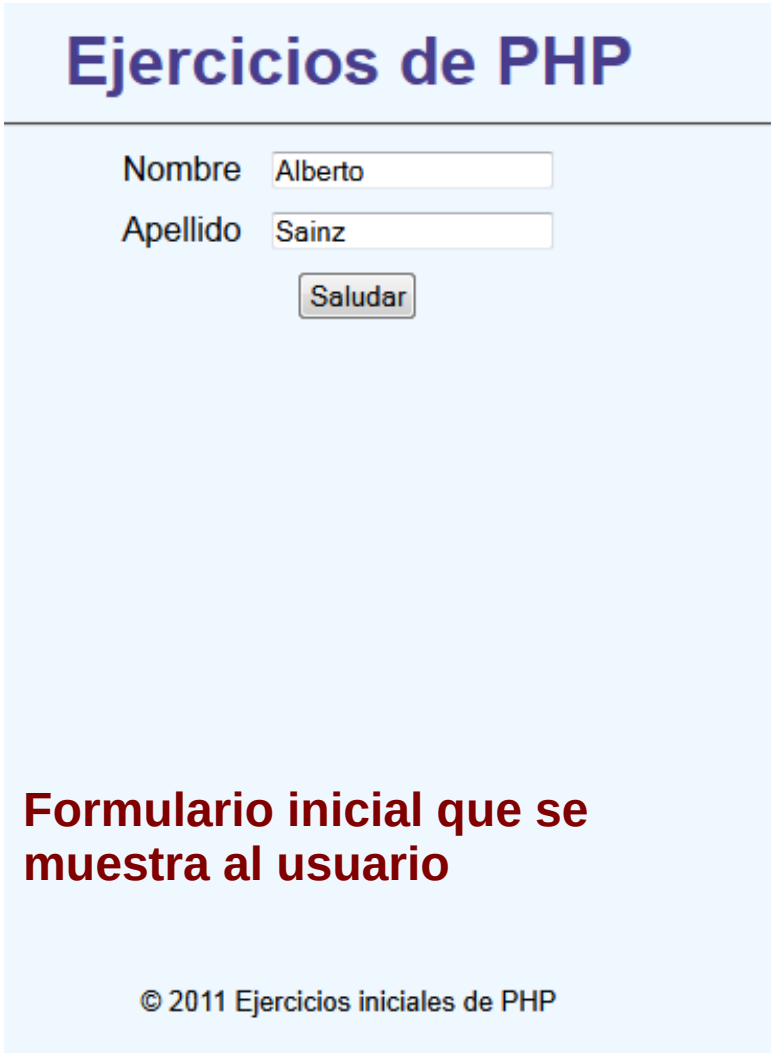
- Controlador
  - recibe todas las peticiones del usuario (del cliente web) y responde a esas peticiones seleccionado del modelo los datos apropiados y mostrando al usuario la vista adecuada, es decir, eligiendo qué plantilla mostrar (un formulario, una vista de resultados, una página de error)
  - controla toda la lógica de la aplicación
  - contiene sólo código PHP
- Modelo
  - ficheros PHP que obtienen los datos para la aplicación (acceso a la base de datos, o funciones que procesan algún valor, ...)

MVC – Modelo Vista Controlador

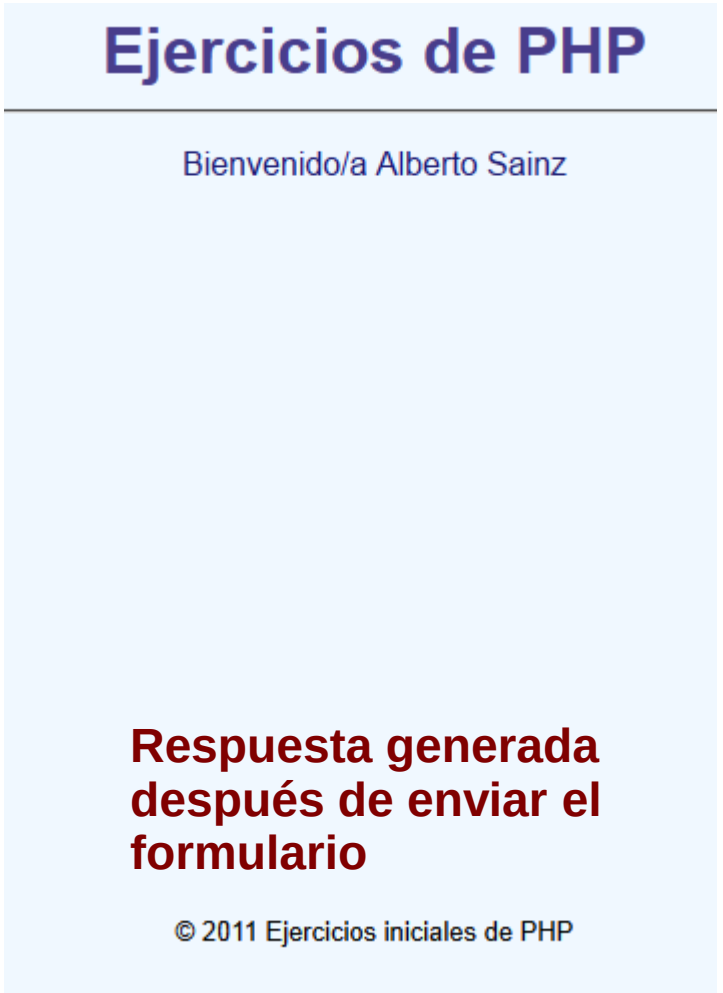
- Vista
  - ficheros HTML y PHP que representan el interfaz de usuario, las diferentes plantillas a mostrar por el controlador

- Puesto que todavía no hemos visto nada de acceso a BD nos centraremos en el controlador y las vistas

Un controlador, varias plantillas



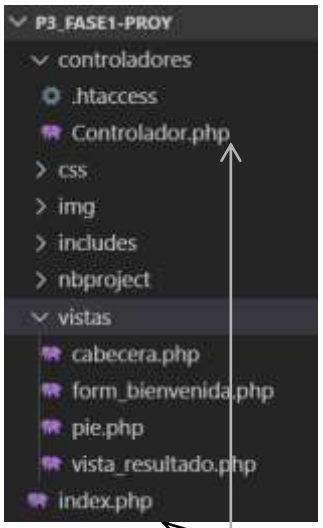
Formulario inicial que se muestra al usuario



Respuesta generada después de enviar el formulario

Un controlador, varias plantillas

Estructura del proyecto



http://localhost/P3_Fase1-Proy

Al ejecutar el proyecto se llama al script index.php, no al formulario inicial

En la carpeta controladores creamos un archivo **Controlador.php** para la clase Controlador.

Esta clase será instanciada en el archivo **index.php.** Este archivo gestionará todas las peticiones.

Un controlador, varias plantillas

En el archivo **Controlador.php** la clase Controlador

```php
<?php
class Controlador
{
    public function run()
    {
        if (!isset($_POST['saludar']))//no se ha enviado el formulario
        { // primera petición
            //se llama al método para mostrar el formulario inicial
            $this->mostrarFormulario();
            exit();
        } else
        {
            //el formulario ya se ha enviado
            //se recogen y procesan los datos
            //se llama al método para mostrar el resultado
            $nombre=$_POST['nombre'];
            $apellido=$_POST['apellido'];
            $resultado ="Bienvenido/a $nombre $apellido ";
            $this->mostrarResultado($resultado);
            exit();
        }
    }
    private function mostrarFormulario()
    {...4 lines }
    private function mostrarResultado($resultado)
    {...4 lines }
}
?>
```

El controlador recibe las peticiones, decide en el método **run()** el bloque de código a ejecutar dependiendo de la opción elegida, y da respuesta.

En el controlador implementaremos los métodos necesarios.

Un controlador, varias plantillas

```
public function run()     // El método run
    {
        if (!isset($_POST['saludar']))//no se ha enviado el formulario
        { // primera petición
            //se llama al método para mostrar el formulario inicial
            $this->mostrarFormulario();
            exit();
        } else
        {
            //el formulario ya se ha enviado
            //se recogen y procesan los datos
            //se llama al método para mostrar el resultado
            $nombre=$_POST['nombre'];
            $apellido=$_POST['apellido'];
            $resultado ="Bienvenido/a $nombre $apellido ";
            $this->mostrarResultado($resultado);
            exit();
        }
    }
```

Un controlador, varias plantillas

//Los métodos privados (en este caso) pueden ser públicos *mostrarFormulario()* y *mostrarResultado()* de la clase Controlador

```
private function mostrarFormulario()
    {
     //se muestra la vista del formulario (la plantilla
       form_bienvenida.php)
        include 'vistas/form_bienvenida.php';
    }
    private function mostrarResultado($resultado)
    {
    // y se muestra la vista del resultado (la plantilla resultado.,php)
        include 'vistas/resultado.php';
    }
```

Un controlador, varias plantillas

1. El controlador necesita saber si la petición actual corresponde a un envío del formulario u a otra petición

   ```
   if (!isset($_POST['saludar']))
   ```

2. Si la petición no es un envío del formulario el controlador llama al método mostrarFormulario() para el formulario inicial

   ```
   include "vistas/form_bienvenida.php";
   ```

3. Si la petición es un envío del formulario se recogen los datos enviados, se validan (añadiremos al final)y procesan y

4. Se llama al método mostrarResultado() y muestra la vista del resultado include "vistas/vista_resultado.php";

Un controlador, varias plantillas

```
<?php
    require_once 'controladores/Controlador.php';
    $controlador = new Controlador();
    $controlador->run();
    ?>
```

El archivo index.php. Gestionará todas las peticiones creando una instancia de la clase Controlador nada más iniciar el proyecto.

En el archivo .htacces si incluimos el siguiente código:

```
deny from all
ErrorDocument 403 "<body bgcolor=#ff6633><h1>Acceso prohibido a esta carpeta</h1>"
```

Al acceder con el navegador a la carpeta controladores, se obtiene:

Un controlador, varias plantillas

Plantilla (vista) del formulario form_bienvenida.php

```
<?php
   include "cabecera.php";
 ?>
<form id="form"  action="index.php" method="post">
   <div id="datos">
      <label>Nombre</label>
      <input type="text" name="nombre" /><br />
      <label>Apellido</label>
      <input type="text" name="apellido" /><br />
      <label> </label>
      <input type="submit" name="saludar"  value="Saludar"/>
   </div>

</form>
<?php
   include "pie.php";
?>
```

Un controlador, varias plantillas

```php
<?php
   include "cabecera.php";

   echo '<div class=“texto">';
   echo $resultado;
   echo "</div>"  ;

   include "pie.php";
?>
```

Plantilla (vista) del resultado vista_resultado.php

cabecera.php

pie.php

pueden ser los archivos utilizados a lo largo del curso u otros hechos por vosotros.

Un controlador, varias plantillas

- <form id="form" **action="" method="post"**>
  - si se omite el nombre del script en action, **action="",** se asume que procesará el formulario el mismo script que lo ha generado, es decir, **index.php** (enviar los datos del formulario a la misma URL desde la que se recibió el formulario)
  - get / post dependiendo del método de envío que elijamos

Un controlador, varias plantillas

Se ha utilizado la arquitectura MVC
(todavía el modelo NO)
Se busca SPA (Single-Page Application/Página única)

Vista formulario y resultados en la misma página

El formulario y el resultado en la misma página vistas/form_bienvenida.php

Controlador.php

```
private function mostrarFormulario()
{
 //se muestra la vista del formulario (la plantilla form_bienvenida.php)
    include 'vistas/form_bienvenida.php';
}
private function mostrarResultado($resultado)
{
// y se muestra la vista del resultado (la plantilla resultado.,php)
    include 'vistas/form_bienvenida.php';
}
```

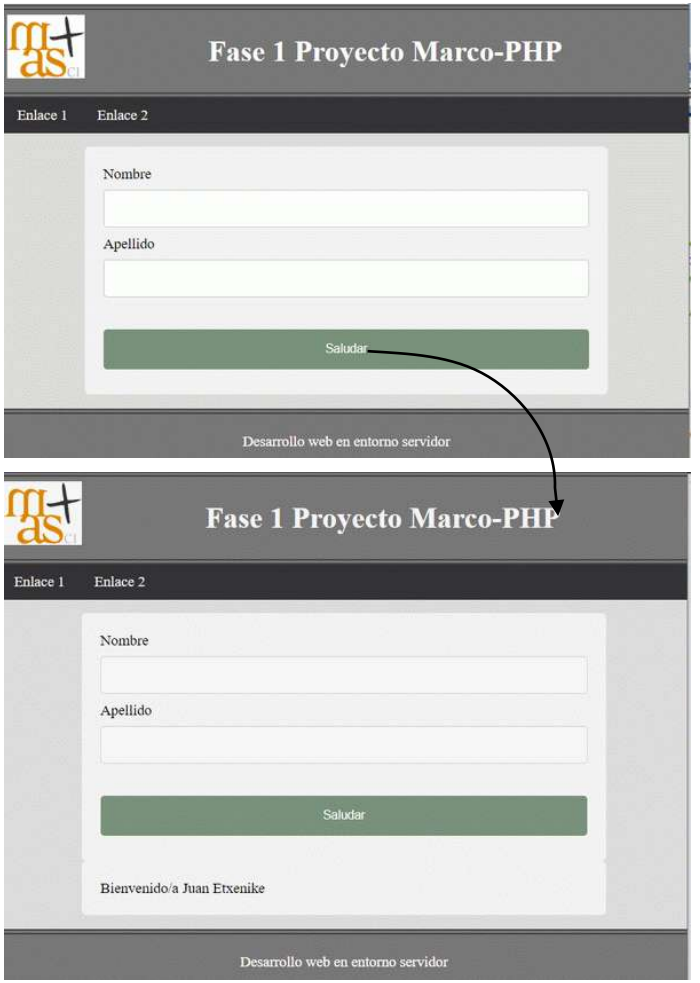
Vista formulario y resultados en la misma página

```php
<?php
include "cabecera.php";
?>
<form id="form"  action="index.php" method="post">
    <div id="datos">
        <label>Nombre</label>
        <input type="text" name="nombre" /><br />
        <label>Apellido</label>
        <input type="text" name="apellido" /><br />
        <label> </label>
        <input type="submit" name="saludar"  value="Saludar"/>
    </div>

</form>
<?php
if (isset($resultado))
{
    echo "<div class='resultado'>";
    echo $resultado;
    echo "</div>";
}

include "pie.php";
?>
```

Si hay resultado, se incluye vista_resultado.php



Vista formulario y resultados en la misma página

Una vez entendido,
Se añaden otro/otros
tipo/s de control en el formulario.
Todavía sin validaciones.

Vista formulario y resultados en la misma página

Vista formulario y resultados en la misma página

```php
<form id="form" action="index.php" method="post">
    <div id="datos">
        <label>Nombre</label>
        <input type="text" name="nombre" /><br />
        <label>Apellido</label>
        <input type="text" name="apellido" /><br />
        <label>Curso</label>
        <div class='uno'>
            <input type="radio" name="curso" value="DAM" />DAM <br />
            <input type="radio" name="curso" value="DAW" />DAW
        </div>
        <br />
        <label>Módulo</label>
        <select name="modulo">
            <?php
            $modulos = array("Bases de datos", "PHP", "Lenguajes de marcas", "Programación");
            foreach ($modulos as $m) {
                echo "<option value='$m'>$m</option>";
            }
            ?>
        </select><br />
        <input type="submit" name="saludar" value="Saludar" />
    </div>
</form>
```

En el formulario de entrada y sigue siendo SPA, manteniendo todo lo anterior.

Vista formulario y resultados en la misma página

```php
} else {
    //el formulario ya se ha enviado
    //se recogen y procesan los datos
    //se llama al método para mostrar el resultado
    $nombre = $_POST['nombre'];
    $apellido = $_POST['apellido'];
    $modulo = $_POST['modulo'];
    if (isset($_POST['curso'])) {
        $curso = $_POST['curso'];
    }
    $resultado = "Bienvenido/a $nombre $apellido ";
    if (isset($curso)) {
        $resultado .= " estás en el curso $curso.<br />";
    }
    $resultado .= "Matriculado/a en el módulo: $modulo";
    $this->mostrarResultado($resultado);
    exit();
}
```

Una posibilidad en la recogida de datos del formulario

Un controlador, varias plantillas

Arquitectura MVC
(todavía el modelo NO)
SPA (Single-Page Application/Página única)
Aplicar al tema elegido.