



<http://www.robertsallent.com> 
[@robertsallent](https://twitter.com/robertsallent) 

Característiques del RMF

Característiques del framework RMF (v1.0)





Índice

- Antes de nada
- Características
- Cambiando el aspecto
- Navegación
- Controlador por defecto
- *URLs* amigables
- Cargando modelos
- Acceso a la *BDD*
- Cargando vistas
- ANEXOS



Antes de nada

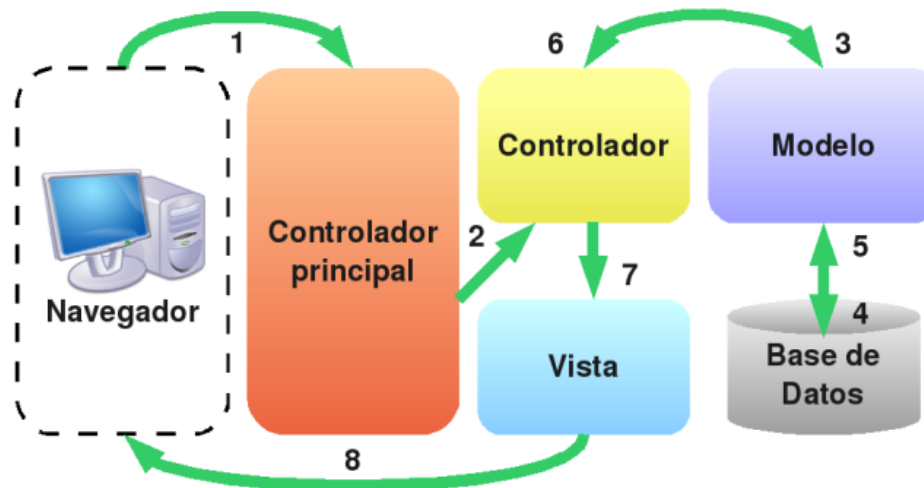
- En la presentación anterior se explicó cómo configurar y probar el *RobS Micro Framework (RMF)* que usamos en clase .
- **En esta presentación se describen las características del *framework* y se dan ejemplos cortos de uso.**
- En la próxima presentación (tutorial de uso y creación de aplicaciones) se verá cómo utilizar dichas características en el proceso de creación de una aplicación completa.





Antes de nada

- Recordemos que el *RMF* trabaja usando una arquitectura **modelo-vista-controlador** con **controlador frontal**.





Característiques

- En la versión 1.0 del *framework*, las principales características son:
 - *PHP* usando arquitectura *MVC* con controlador frontal.
 - Fichero de configuración independiente.
 - Gestión de usuarios (registro, modificación y baja).
 - Tratamiento de sesiones (*login* y *logout*).
 - Librería que simplifica y optimiza la conexión con la *BDD*.
 - Librería que simplifica la carga de imágenes.
 - Librería que permite exportación simple de objetos a *XML*.
 - Posibilidad de gestionar plantillas (*templates*).

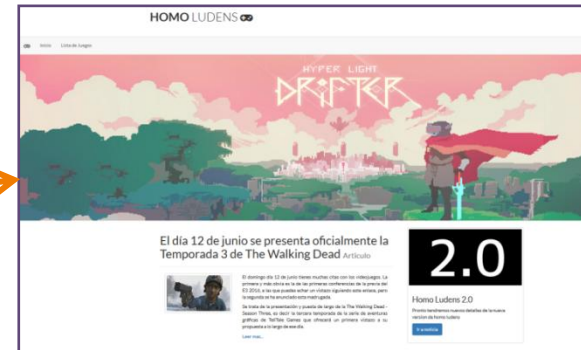


Cambiando el aspecto

- Si queremos cambiar ya el aspecto de la aplicación, podemos editar directamente los siguientes ficheros:
 - El fichero `css/estilo.css` modifica el estilo de la aplicación.
 - El fichero `templates/Template.php` contiene patrones para crear de forma rápida las áreas comunes de todas las vistas (*header, footer, menú, login, logout...*).
 - Si modificamos las vistas directamente en la carpeta `views` sólo cambiaremos la vista concreta que editemos.



Ejemplos cambiando el aspecto





Navegación

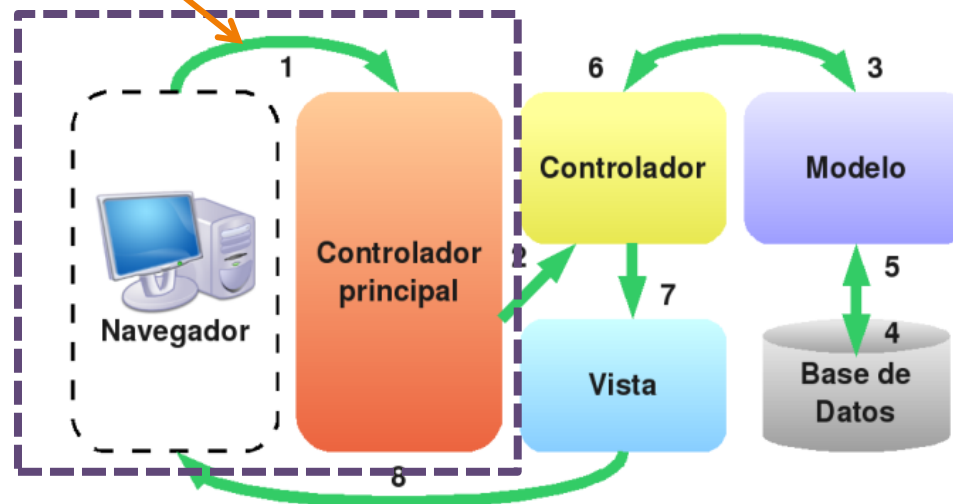
forma de la petición HTTP

- Al crear las vistas, **necesitaremos poner enlaces** a cada una de las operaciones de que dispone nuestra aplicación.
- Debemos tener en cuenta que **todas las operaciones pasan por el fichero** `index.php` (que invoca al controlador frontal).
- Las peticiones van en parámetros *HTTP GET*, con lo que **tienen la forma**: `index.php?controlador=capitulo&operacion=ver¶metro=3`
- En este ejemplo se mostrarían los datos del capítulo con código 3..



Ejemplo forma de la petición HTTP

index.php?controlador=capitulo&operacion=ver¶metro=3

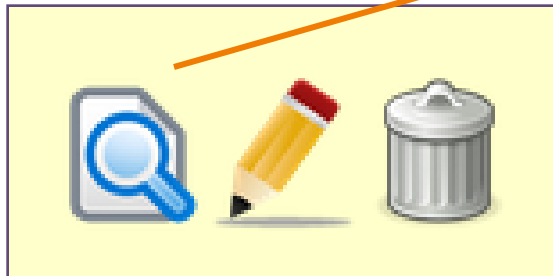




Ejemplo resultado de la petición

index.php?controlador=capitulo&operacion=ver¶metro=3

<a>



The screenshot shows a web application interface for 'THE SIMPSONS'. The header features the title 'THE SIMPSONS' and a subtitle 'LISTADO DE CAPITULOS DE LOS SIMPSONS'. Below the header, there is a navigation bar with links: 'INICIO', 'PERSONAJES', 'CAPITULOS', 'REGISTRO', 'ADMINISTRADOR', 'NUEVO CAPITULO', and 'NUEVO PERSONAJE'. The main content area displays 'DETALLES DEL CAPITULO: SIN BLANCA NAVIDAD'. It includes fields for 'Num del capitulo' (1), 'Temporada' (1), and 'Descripción'. The description text reads: 'Homer y Marge van al Festival de Navidad de la Escuela Primaria, cuando entran encuentran a un grupo de estudiantes cantando. Después viene el segundo grado, aquí Lisa representa a Tawanga, el Santa Claus de los mares del Sur y Ralph representa al Japonés. Después sigue el curso de cuarto, en donde Bart canta junto con el resto del curso, pero lo sabotea todo y lo sacan del escenario. Más tarde se encuentran todos en la casa preparando todo para Navidad, Marge está escribiendo una carta a los amigos de la familia en la que dice que: Murió Snowball I atropellado por un coche, pero que compraron otro, que el abuelo está bien, que Maggie ya camina sola, que Lisa solo saca dieces y que a Bart lo quieren mucho, dice que Homer envía su afecto aunque está renegando porque Marge no ha acabado la carta, luego le pregunta a los niños qué quieren para la Navidad. Lisa pide un pony por tercer año consecutivo, Bart quiere un tatuaje y Homer dice que si quiere uno tendrá que pagarlo con sus domingos. En ese momento Iana Patty y dice que ella y Selma estarán muy contentas de ir en Navidad a donde su hermanita.' Below the description, there are fields for 'Duración' (19 min.) and 'Fecha de emisión' (1989-12-17). A 'Volver al listado' button is located at the bottom right of the content area. The footer includes the text 'Powered By Robs Micro Framework' and 'Robert Sallent'16' with a Twitter icon.



Navegación relación con la estructura del fw

- En la URL:
`index.php?controlador=capitulo&operacion=ver¶metro=3`
- `controlador`: indica el controlador a cargar. En este ejemplo, **debe existir un fichero `Capitulo.php` en la carpeta `controller` y contener la definición de la clase `Capitulo`.**
- `operación`: método del controlador a invocar. En este ejemplo, **debe existir un método llamado `ver()` dentro de la clase `Capitulo`.**
- `parametro`: parámetro que se le pasará al método del controlador.



Ejemplo relación con la estructura del fw

index.php?controlador=capitulo&operacion=ver¶metro=3

```
<?php
//CONTROLADOR CAPITULO: operaciones con los capítulos
class Capitulo extends Controller{

    //método para ver un capítulo concreto
    public function ver($id=0){
        //pedirle al modelo que me pase el capítulo deseado
        $this->load('model/CapituloModel.php');
        $capitulo = CapituloModel::getCapitulo($id);

        if(!$capitulo)
            throw new Exception('No se encuentra el capítulo');

        //pasarle el capítulo a la vista
        $datos = array();
        $datos['usuario'] = Login::getUsuario();
        $datos['capitulo'] = $capitulo;
        $this->load_view('view/capitulos/detalles.php', $datos);
    }
}
```



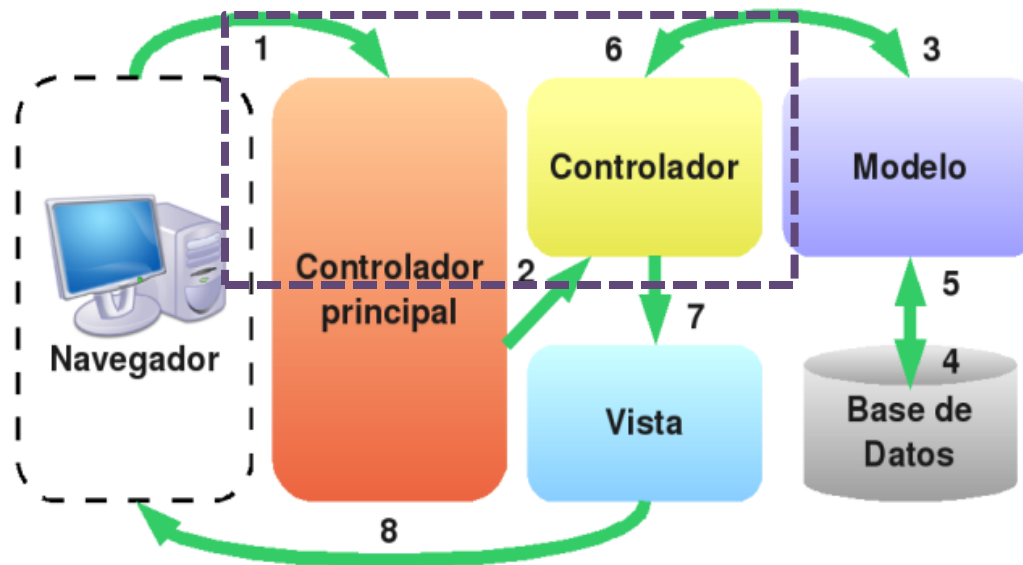


Arquitectura MVC con CF

- Ante una petición con la forma:
`index.php?controlador=capitulo&operacion=ver¶metro=3`
- 1. El fichero `index.php` carga el **controlador frontal** (dispatcher).
- 2. El controlador frontal llama al **controlador** `Capitulo`, ejecutando el **método** `ver()` con el **parámetro** 3.
- 3. El método `ver()` usa el **modelo** para recuperar los datos del capítulo 3 **de la base de datos** y los retorna al controlador.
- 7. El controlador carga la vista y le pasa los datos.



Navegación relación con la estructura del fw



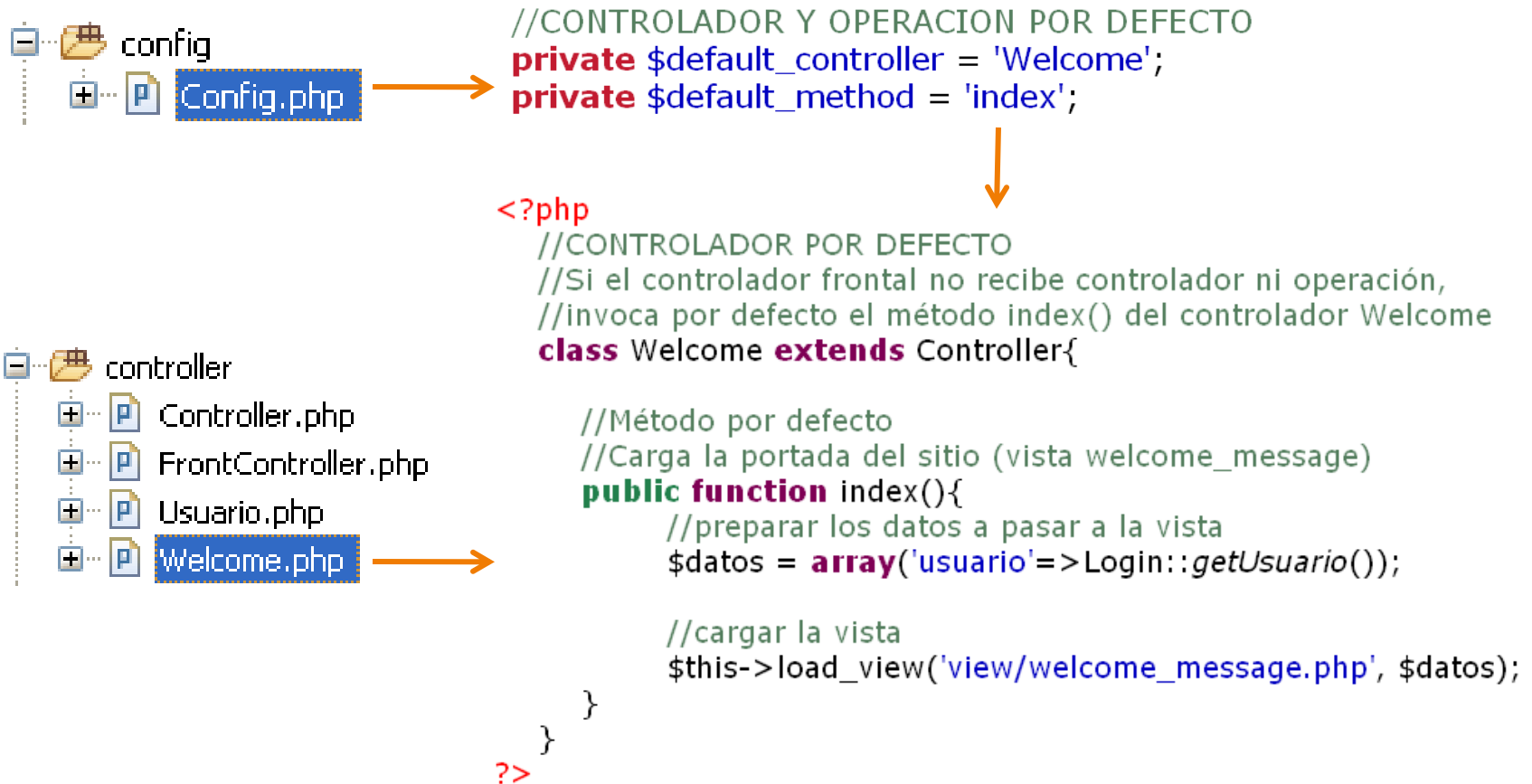


Controlador por defecto

- Una particularidad que habréis notado al probar el *framework* es que, de entrada, ya se muestra un resultado (la vista de portada).
- Esto se debe a que **si en la petición *HTTP* no se adjuntan los parámetros controlador y operación, se usa por defecto el controlador `welcome` y el método `index()`**, que carga la vista `welcome_message.php`.
- Esto se puede cambiar en el fichero `Config.php`.
- **Si editáis la vista `welcome_message.php`, estaréis cambiando la portada por defecto de la aplicación.**



Ejemplo controlador por defecto





URLs amigables

- Las *URL* comentadas anteriormente son largas, complicadas, difíciles de recordar y dan detalles del funcionamiento de la aplicación que es posible que no queramos dar.
- *RMF* incorpora la posibilidad de utilizar **el mecanismo de URLs amigables**, con el que la ruta:

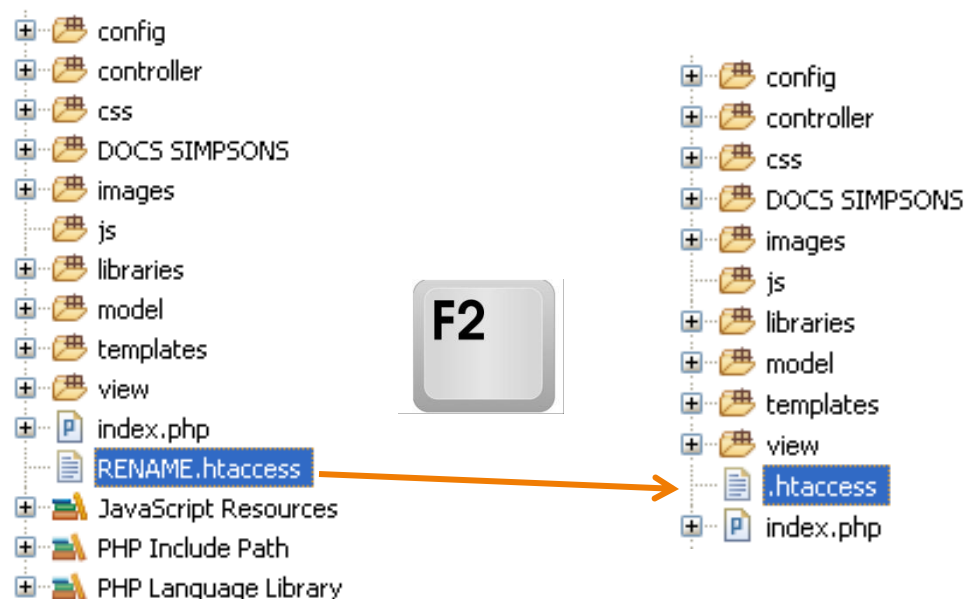
`index.php?controlador=capitulo&operacion=ver¶metro=3`

Se convierte en: `index.php/capitulo/ver/3`



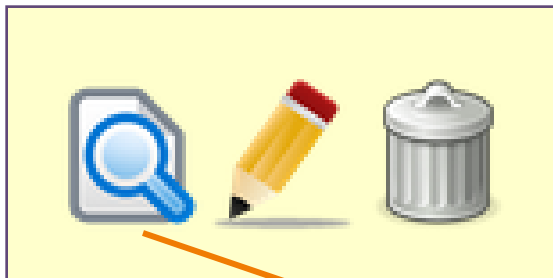
URLs amigables

- Para usarlo, tan solo hay que renombrar el fichero `RENAME.htaccess` a `.htaccess`.





Ejemplo ejemplo de URLs amigables (resultado)



<a>

index.php/capitulo/ver/3





URLs amigables

- Para que las URL amigables funcionen, **debe estar habilitado el módulo mod rewrite en el servidor web de Apache 2.**
- Si estáis trabajando con *Xampp* ya estará habilitado y también debería estarlo en vuestro hosting.
- Sin embargo, es posible que si estáis usando *Linux* tengáis que habilitarlo manualmente. En ese caso seguid las instrucciones de las siguientes transparencias.



Habilitar URLs amigables

- En caso que no tengáis habilitado ***mod_rewrite***, abrid una terminal en *Linux* y escribid:
 - Para habilitar el módulo:
`sudo a2enmod rewrite`
 - Para reiniciar el apache:
`sudo service apache2 restart`
 - Para comprobar si está habilitado:
`apache2ctl -M`



Habilitar URLs amigables

- Ahora, debéis editar la configuración de vuestro sitio en uno de estos ficheros (depende de la versión de Apache):

`/etc/apache2/sites-available/000-default.conf`

`/etc/apache2/sites-available/default`

`/etc/apache2/apache2.conf`

- Buscad estas líneas, donde `/var/www` es la ruta a vuestro *Document Root*, y aseguráros que `AllowOverride` está en `All`.

```
<Directory /var/www/>
```

```
    AllowOverride All
```

```
</Directory>
```



Cargar modelos

- Pasemos ahora a detallar algunos de los conceptos necesarios para la implementación de nuestras aplicaciones.
- Los controladores trabajan con el modelo, solicitando datos y operaciones.
- **Para que un controlador pueda usar los métodos del modelo, antes debe cargarlo.** Para ello existe el método `load()` de la clase `Controller` (que heredan todos los controladores).



Cargar modelos

- La sintaxis es: `load(url);`
- **url: ruta del fichero que contiene el modelo.** Generalmente indicaremos una ruta relativa respecto a la ubicación del fichero `index.php`.
- Lo más habitual será: `model/NombreModelo.php`.
- Este método también permite cargar controladores.



Ejemplo cargar modelos

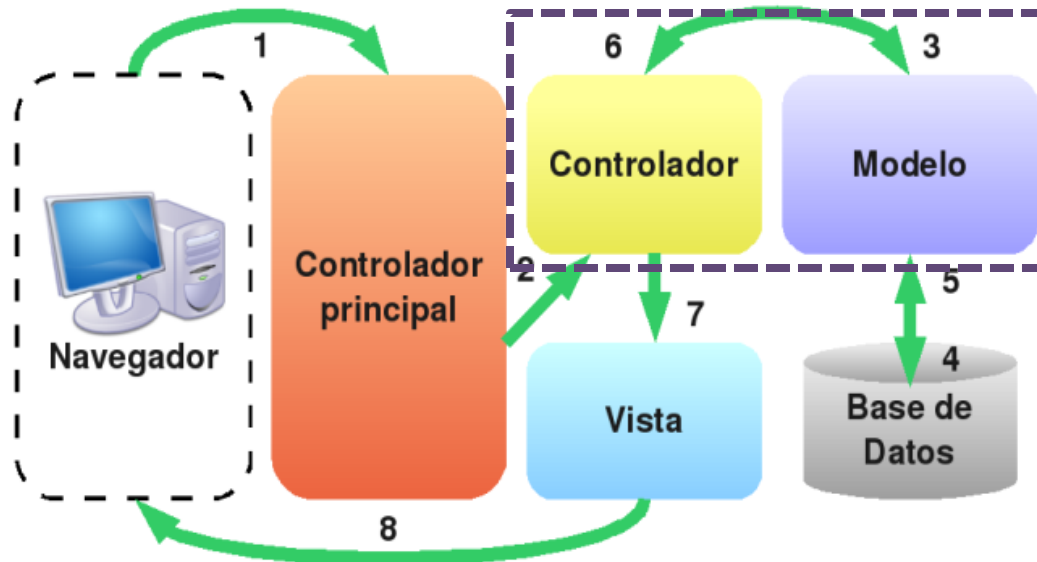
```
//método para ver un capítulo concreto
public function ver($id=0){
    //pedirle al modelo que me pase el capítulo deseado
    $this->load('model/CapituloModel.php'); ←
    $capitulo = CapituloModel::getCapitulo($id);

    if(!$capitulo)
        throw new Exception('No se encuentra el capítulo');

    //pasarle el capítulo a la vista
    $datos = array();
    $datos['usuario'] = Login::getUsuario();
    $datos['capitulo'] = $capitulo;
    $this->load_view('view/capitulos/detalles.php', $datos);
}
```



Cargar modelos





Acceso a la BDD

- **Los modelos interactúan con la base de datos** para guardar, recuperar, modificar o borrar información.
- Para trabajar con la BDD, los modelos conectarán, ejecutarán sus operaciones y desconectarán.
- Para simplificar la tarea y evitar errores (como la apertura de excesivas conexiones), el *RMF* incorpora una herramienta sencilla de conexión con la *BDD*.



Database_library

- La librería `libraries/database_library.php` contiene un método estático `get()` **que establece o recupera la conexión con la base de datos** a partir de los parámetros indicados en el fichero `Config.php`.
- Esto simplifica la conexión y además controla que no se creen múltiples conexiones a la *BDD* desde un mismo *script*.
- El prototipo es: `public static mysqli get();`
- Para usarlo, podemos hacer: `Database::get();`



Ejemplo database_library

```
//Método que recupera un capítulo concreto de la BDD
//(devuelve null si no lo encuentra)
public static function getCapitulo($id=0){
    //preparar la consulta
    $consulta = "SELECT * FROM capitulos WHERE id=$id;";

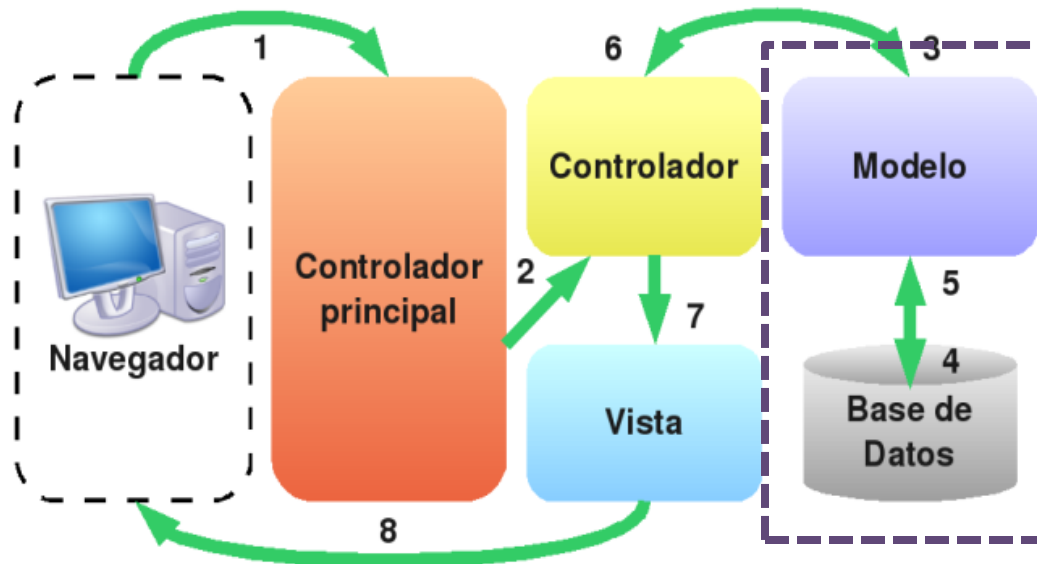
    //ejecutar la consulta (devuelve un mysqli_result)
    $resultado = Database::get()->query($consulta); ←
    if(!$resultado) return null;

    $capitulo = $resultado->fetch_object('CapituloModel');
    $resultado->free();

    return $capitulo;
}
```



Acceso a la BDD





Cargar las vistas

- Los controladores cargan vistas para que las vea el usuario que está utilizando la aplicación.
- Cuando un controlador carga una vista, debe pasarle la información que ésta debe mostrar.
- Para cargar la vista, hay un **método llamado `load_view()`** en la clase `Controller` (que heredan todos los controladores), que además **permite pasar un *array* asociativo o un objeto con los datos a la vista.**



Cargar las vistas

- La sintaxis es: `load_view(url [, datos]);`
- `url`: **ruta del fichero que contiene la vista**. Generalmente indicaremos una ruta relativa respecto a la ubicación del fichero `index.php`. Lo más habitual será: `view/nombrevista.php`.
- `datos`: **array asociativo u objeto** que contendrá la información que debe mostrar la vista, en forma de pares de nombre/valor.



Ejemplo pasar info a las vistas

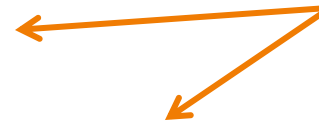
```
//método para ver un capítulo concreto
public function ver($id=0){
    //pedirle al modelo que me pase el capítulo deseado
    $this->load('model/CapituloModel.php');
    $capitulo = CapituloModel::getCapitulo($id);

    if(!$capitulo)
        throw new Exception('No se encuentra el capítulo');

    //pasarle el capítulo a la vista
    $datos = array();
    $datos['usuario'] = Login::getUsuario();
    $datos['capitulo'] = $capitulo;
    $this->load_view('view/capitulos/detalles.php', $datos);
}
```



Array de datos que se le pasa a la vista





Pasar info a las vistas

- En la vista, las claves del *array* asociativo pasado (o las propiedades del objeto pasado), **se convierten en variables globales de *PHP***.

```
$datos = array();  
$datos['usuario'] = Login::getUsuario();  
$datos['capitulo'] = $capitulo;
```

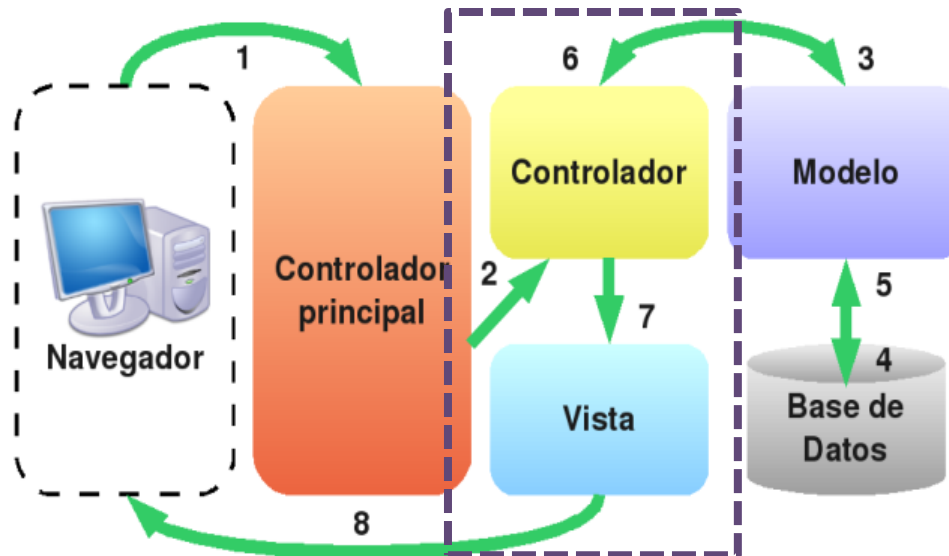
controlador

vista

```
<h3>Num del capítulo</h3>  
<p><?php echo $capitulo->capitulo;?></p>  
<h3>Temporada</h3>  
<p><?php echo $capitulo->temporada;?></p>  
<h3>Descripción</h3>  
<p><?php echo preg_replace("/\n/i", '</p><p>', $capitulo->descripcion);?></p>  
<h3>Duración</h3>  
<p><?php echo $capitulo->duracion;?> min.</p>
```



Pasar info a las vistas



Anexos

Librerías y herramientas





Login_library

- En la librería `libraries/login_library.php` encontramos los siguientes métodos útiles:
 - `Login::getUsuario()`; recupera un objeto `UsuarioModel` con los datos del usuario que ha hecho *login* o `null` si no hay ningún usuario identificado.
 - `Login::isAdmin()`; retorna `true` si hay un usuario identificado en el sistema y tiene permiso de administrador. Retorna `false` en caso contrario.



Ejemplo de uso login_library

```
//método del controlador para el listado de capitulos
public function listar(){
    //pedirle al modelo que recupere los capítulos
    $this->load('model/CapituloModel.php');
    $capitulos = CapituloModel::getCapitulos();

    //pasarle los capítulos a la vista
    $datos = array();
    $datos['usuario'] = Login::getUsuario(); ←
    $datos['capitulos'] = $capitulos;

    if(!Login::isAdmin()) //el admin verá una lista diferente ←
        $this->load_view('view/capitulos/lista.php', $datos);
    else
        $this->load_view('view/capitulos/admin/lista_admin.php', $datos);
}
```



Upload_library

- Para subir un fichero de imagen, podemos usar la librería `upload_library`, que se corresponde con el fichero `libraries/upload_library.php`.

- Ejemplo de subida de un fichero de imagen:

```
//recuperar y guardar la imagen (solamente si ha sido enviada)
if($_FILES['imagen']['error']!=4){
    $upload = new Upload($_FILES['imagen'], 'images/capitulos/', 1000000);
    $capitulo->imagen = $upload->upload_image();
}
```

- Primero se construye un objeto de tipo `Upload` y posteriormente se llama al método `upload_image()`;



Librerías del framework

- La sintaxis para construir un nuevo Upload es:
`new Upload(array_fichero, ruta_final [, tamaño_max]);`
- `array_fichero`: es la posición del array `$_FILES` que se corresponde con el fichero que deseamos subir.
- `ruta_final`: es la ruta donde se ubicará el fichero en el servidor. El usuario *www-data* debe tener permiso de escritura.
- `tamaño_max`: es el tamaño máximo del fichero en bytes. Si no se indica, el tamaño del fichero vendrá limitado por las directivas de configuración en el fichero de configuración `php.ini`.

Ejemplos

Ejemplo de aplicación hecha con RMF





Ejemplo del tutorial

- El ejemplo que se explicará en la siguiente presentación lo podéis probar en: <http://ejemplos.robertsallent.com/simpsons>

THE SIMPSONS
LISTADO DE CAPÍTULOS DE LOS SIMPSONS

Hola Administrallet admin (admin@rapp.com), eres administrador [Logout]

Inicio PERSONAJES CAPÍTULOS REGISTRO

ADMINISTRADOR NUEVO CAPÍTULO NUEVO PERSONAJE

BIENVENIDOS

Guía de capítulos de Los Simpsons.
En realidad, esto no es una aplicación real, solamente se trata de un ejemplo del curso FCO0210 - Desarrollo de aplicaciones con tecnologías web que imparte Robert Sallent.

La aplicación ha sido desarrollada en PHP con un framework MVC llamado **Robo Micro Framework** de desarrollo propio y que utilizamos en clase.

Si estás interesado en aprender a realizar aplicaciones web como ésta, no dudes en apuntarte a los cursos subvencionados (Certificados de Profesionalidad) que imparte.

Para estar al día, consulta este Twitter...

Powered By **Robo Micro Framework** Robert Sallent '16

THE SIMPSONS
LISTADO DE CAPÍTULOS DE LOS SIMPSONS

Hola Administrallet admin (admin@rapp.com), eres administrador [Logout]

Inicio PERSONAJES CAPÍTULOS REGISTRO

ADMINISTRADOR NUEVO CAPÍTULO NUEVO PERSONAJE

DETALLES DE BART SIMPSON

Nombre
Bart Simpson

Edad
10 min.

Biografía
Bartholomew JoJo «Bart» Simpson 1 2 (Bartolomeo J. Simpson en Hispanoamérica), es uno de los personajes ficticios protagonistas de la serie de televisión de dibujos animados Los Simpson. Bart tiene 10 años y es el primogénito, y único hijo varón de Homer y Marge Simpson. Es el hermano mayor de Lisa y Maggie.1

Este personaje fue creado improvisadamente por Matt Groening y debutó en el corto Good Night de El show de Tracey Ullman el 19 de abril de 1987, originalmente, con la voz de Nancy Cartwright. En América Latina, Bart ha sido siempre doblado por Marina Huerta, salvo por un lapsus entre la décima y decimoquinta temporada, que fue interpretado por Claudia Mota, mientras que en España, Sara Vivas le ha puesto regularmente su voz.3

Bart es uno de los personajes principales de la serie y ha resultado ser uno de los más conocidos en la historia de la animación en la televisión estadounidense.4 La revista Time consideró a Bart como el cuadragésimo sexto personaje (de cien) más influyente del siglo XX; la revista norteamericana TV Guide le otorgó el undécimo lugar (junto a Lisa) en la lista de los «50 mejores personajes de dibujos animados de todos los tiempos» y el Entertainment Weekly nombró a Bart el «artista del año» en 1990. Cartwright ha recibido varios premios por dar su voz a Bart, incluyendo el premio Emmy por la voz de doblaje más destacada en horario de máxima audiencia en 1992 y el premio Annie por su «interpretación en el campo de la animación» en 1995.

Volver al listado

Powered By **Robo Micro Framework** Robert Sallent '16

Foto	Nombre	Edad	Operaciones
	Homer Simpson	39	
	Marge Simpson	34	
	Bart Simpson	10	