



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

RMF v.1.0

# RMF01: instal·lació

---

## Instal·lando y configurando el RMF

# Índice



- Antes de nada
- ¿Qué debemos saber?
- Características
- Primeros pasos
- Descargando / clonando desde el repositorio
- Creando un *virtualhost*
- Preparando la *BDD*
- Configurando
- Ejecutando `setup.php`
- Probando
- Carpetas y ficheros

# Antes de nada



- Esta presentación forma parte de los apuntes del certificado de profesionalidad **IFCD0210 – Desarrollo de aplicaciones con tecnologías web** que imparte Robert Sallent.



- También se incluye como documentación del *framework* RMF.

# Antes de nada



- Este documento describe cómo descargar, instalar y preparar el *RMF* para desarrollar aplicaciones web.
- En las próximas presentaciones se muestran con más detalles sus características y se explica cómo desarrollar una aplicación web completa.
- El entorno de desarrollo utilizado es *Eclipse* para *PHP*.



# Antes de nada



- *RMF* es un marco de trabajo pensado para desarrollar aplicaciones web con **PHP** (7+) de forma sencilla.
- Incorpora algunas de las funcionalidades explicadas y puestas en práctica en nuestros proyectos de clase.
- No es una herramienta 100% segura, está pensada para **uso docente**.
- La usamos para trabajar en clase los conceptos de: arquitectura *MVC*, *frameworks PHP* y desarrollo de aplicaciones web.

# Antes de nada



- Una de las principales ventajas radica en el hecho de que dispone de una interfaz gráfica básica.
- Esto permite centrarse en el desarrollo de la aplicación con *PHP* y despreocuparse en gran medida del *HTML* y *CSS*.
- La *GUI* dispone de las vistas de portada, operaciones con usuarios, panel de control, error y éxito...

# Ejemplo interfaz gráfica (aspecto inicial)





## RMF - RobS Micro Framework

Para el desarrollo de aplicaciones web

Inicio

### Presentación

Este framework PHP, ha sido desarrollado con fines docentes por Robert Sallent para los cursos de desarrollo de aplicaciones que imparte en diferentes centros formativos, entre los que se encuentran: CIFO Vallès, CIFO la Violeta, ActiuHumà e IES Jaume Viladoms.

Se trata de un Framework de trabajo con arquitectura Modelo-Vista-Controlador para entender los conceptos y trabajarlos en clase, siendo paso previo a trabajar con otros frameworks como CodeIgniter, Symfony o Laravel.

A lo largo del curso se desarrollan varios proyectos usando este pequeño framework para conocer las características comunes a este tipo de herramientas de trabajo, así como comprender cómo funcionan internamente.

Este framework implementa:

- Arquitectura MVC.
- Fichero de configuración independiente.
- Autoloader de clases.
- Router/dispatcher para la gestión de peticiones.
- Control de sesiones y gestión básica de usuarios.
- URLs amigables con `mod_rewrite`.
- Librerías para la conexión con BDD (`mysql`), subida de ficheros, generación de XML...
- Un sencillo motor de templates basado en una clase con métodos estáticos

**NO ES 100% SEGURO**, así que no se debe usar para desarrollos en entornos de producción. P contactad conmigo mediante twitter.

RobS micro Framework para fines docentes.  
Creado por Robert Sallent en CIFO del Vallès.





## RMF - RobS Micro Framework

Para el desarrollo de aplicaciones web

Hola Administrador (admin@rmf), eres administrador

Inicio ADMIN

### Panel de control

Operaciones para el administrador

A continuación se muestran las operaciones que puede hacer el administrador.



- Listar
- Nuevo



- Operación 1
- Operación 2



- Operación 1
- Operación 2



- 2 usuarios registrados

RobS Micro Framework

### Formulario de modificación de datos

Usuario para el que se van a modificar los datos: robert (robert@jugayestudia.com)

User:

Nombre:

Email:

Nivel de privilegio:

Administrador: ☐

Nueva imagen:  Ningún archivo seleccionado max 1000KB



Atrás Inicio

RobS micro Framework para fines docentes.  
Creado por Robert Sallent en CIFO del Vallès.



# ¿Qué debemos saber?



- Para comenzar a trabajar con *RMF*, desarrollando proyectos completos **debemos conocer**:
  - Unos mínimos de maquetación (*HTML* y algo de *CSS*).
  - Desarrollo en el lado del servidor con *PHP*.
  - Programación orientada a objetos.
  - Bases de datos relacionales y lenguaje *SQL*.
  - Arquitectura Modelo Vista Controlador (*MVC*).



# ¿Qué debemos saber?



- También debemos tener conocimientos de:
  - *Linux* (si trabajamos en este sistema operativo, cosa que recomiendo).
  - Instalación y configuración de un servidor web y de base de datos (o al menos una instalación básica con *Xampp* o similar).
  - Uso de un entorno de desarrollo para proyectos en *PHP* (yo usaré *Eclipse PHP*).

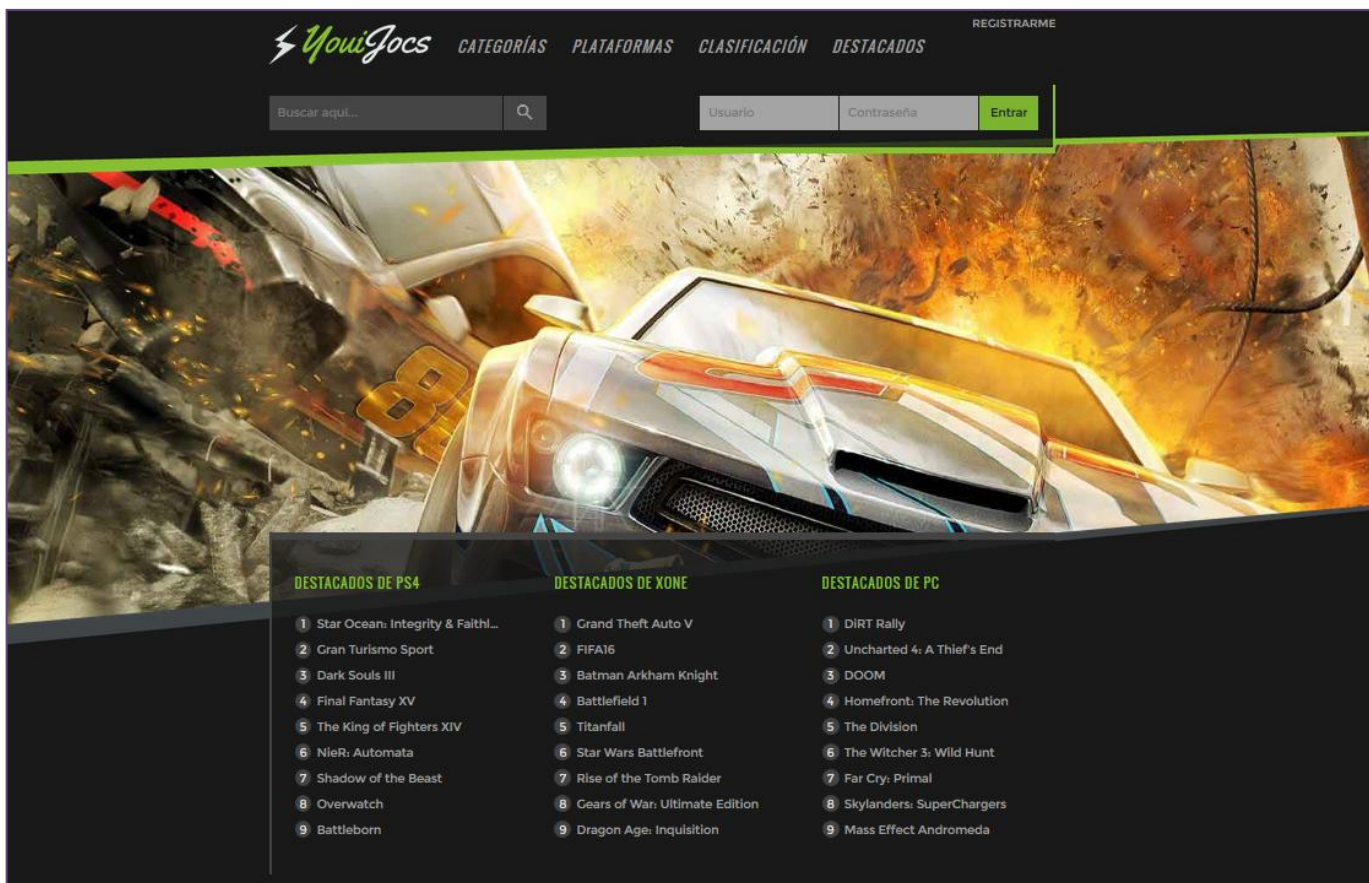
*NOTA: conocimientos no obligatorios pero recomendados: expresiones regulares, URLs amigables, configuración del servidor web avanzada, sistema de ficheros en Linux, GIT...*

# 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 ficheros.
  - Librería que permite exportación simple de objetos a *XML*.
  - Posibilidad de utilizar *templates*.

# Ejemplo Aplicación sobre el framework RMF




# Ejemplo Aplicación sobre el framework RMF




## HOMO LUDENS

[Inicio](#) [Lista de Juegos](#)



### HYPER LIGHT DRIFTER

#### El día 12 de junio se presenta oficialmente la Temporada 3 de The Walking Dead Artículo



El domingo día 12 de junio tienes muchas citas con los videojuegos. La primera y más obvia es la de las primeras conferencias de la previa del E3 2016, a las que puedes echar un vistazo siguiendo este enlace, pero la segunda se ha anunciado esta madrugada.

Se trata de la presentación y puesta de largo de la The Walking Dead - Season Three, es decir la tercera temporada de la serie de aventuras gráficas de TellTale Games que ofrecerá un primera vistazo a su propuesta a lo largo de ese día.

[Leer más...](#)

## 2.0

### Homo Ludens 2.0

Pronto tendremos nuevos detalles de la nueva versión de homo ludens

[Ir a noticia](#)

# Primeros pasos

---

Descargando o clonando y preparando el nuevo proyecto



# Primeros pasos



- A continuación se describe cómo descargar y configurar el *framework* para comenzar la fase de implementación de un proyecto.
- *RMF* se puede descargar desde *GitHub*, en:  
<https://github.com/robertsallent/rmf>



# Primeros pasos



- Los pasos que seguiremos en esta presentación son:
  1. Descarga o clonación del repositorio desde *GitHub*.
  2. Añadir el proyecto a *Eclipse*.
  3. Realizar la configuración (fichero *Config.php*).
  4. Crear la base de datos y la tabla para los usuarios
  5. Comprobar el correcto funcionamiento.
- A lo largo del proceso se detallará también como crear un *virtualhost* para el proyecto, cómo habilitar las *URLs* amigables y como solventar algunas pequeñas dificultades que podamos encontrar, no relacionadas con *RMF*.



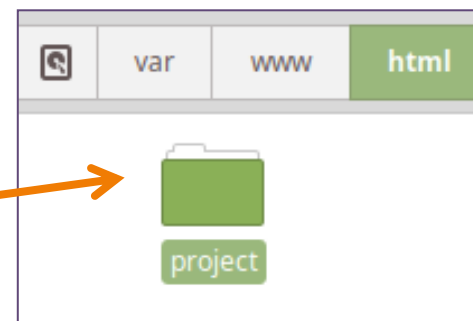
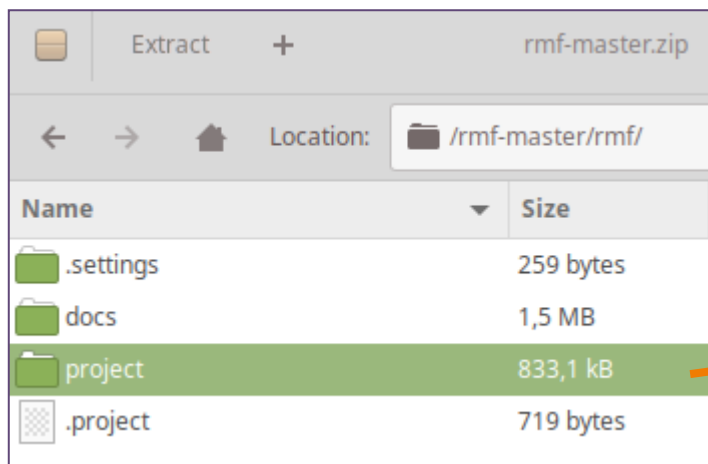
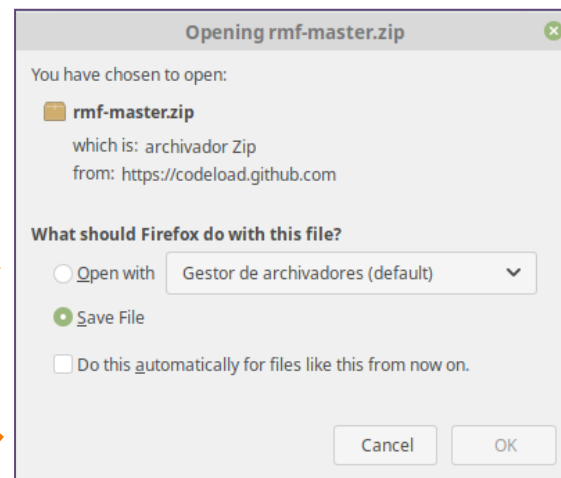
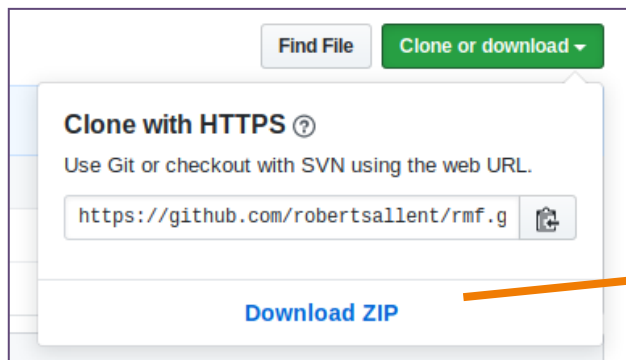
# Descarga y descompresión



- Si optáis por descargar el fichero comprimido con el *RMF*, solamente tenéis que descomprimir la carpeta *project* en vuestro *DocumentRoot* y crear el nuevo proyecto en *Eclipse* con el mismo nombre que la carpeta (se puede cambiar).
- Esta es la opción **más sencilla**.
- Posteriormente, podéis crear un *virtualhost* (como se describe en esta presentación) o configurar una ruta respecto al *DocumentRoot* como raíz de vuestro proyecto (también se describe más adelante).



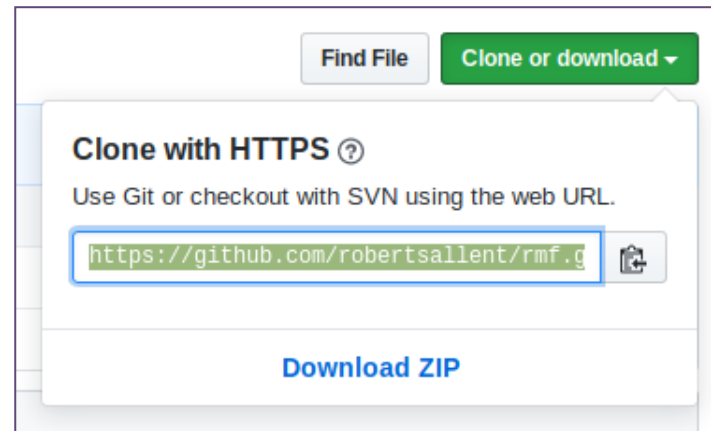
# Ejemplo descarga y descompresión



# Clonado del repositorio

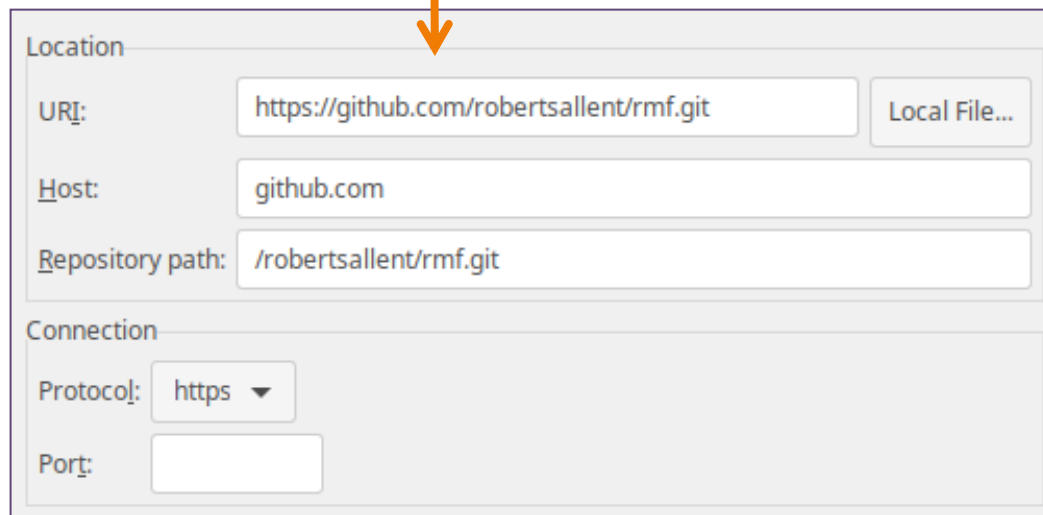
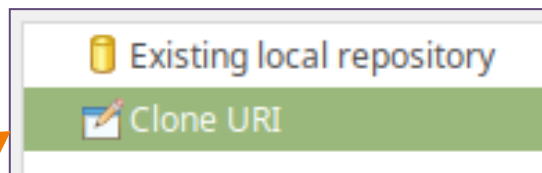
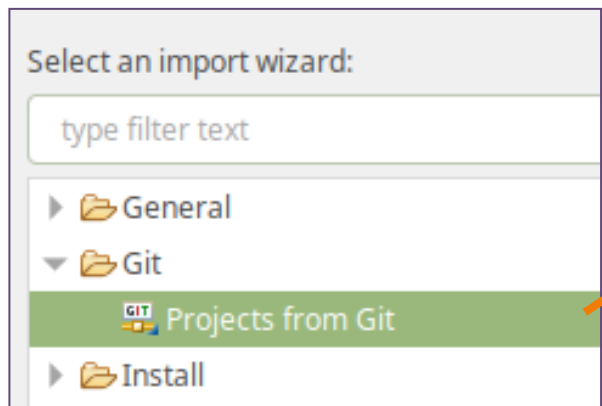


- La otra opción es clonar el repositorio directamente desde *GitHub*.

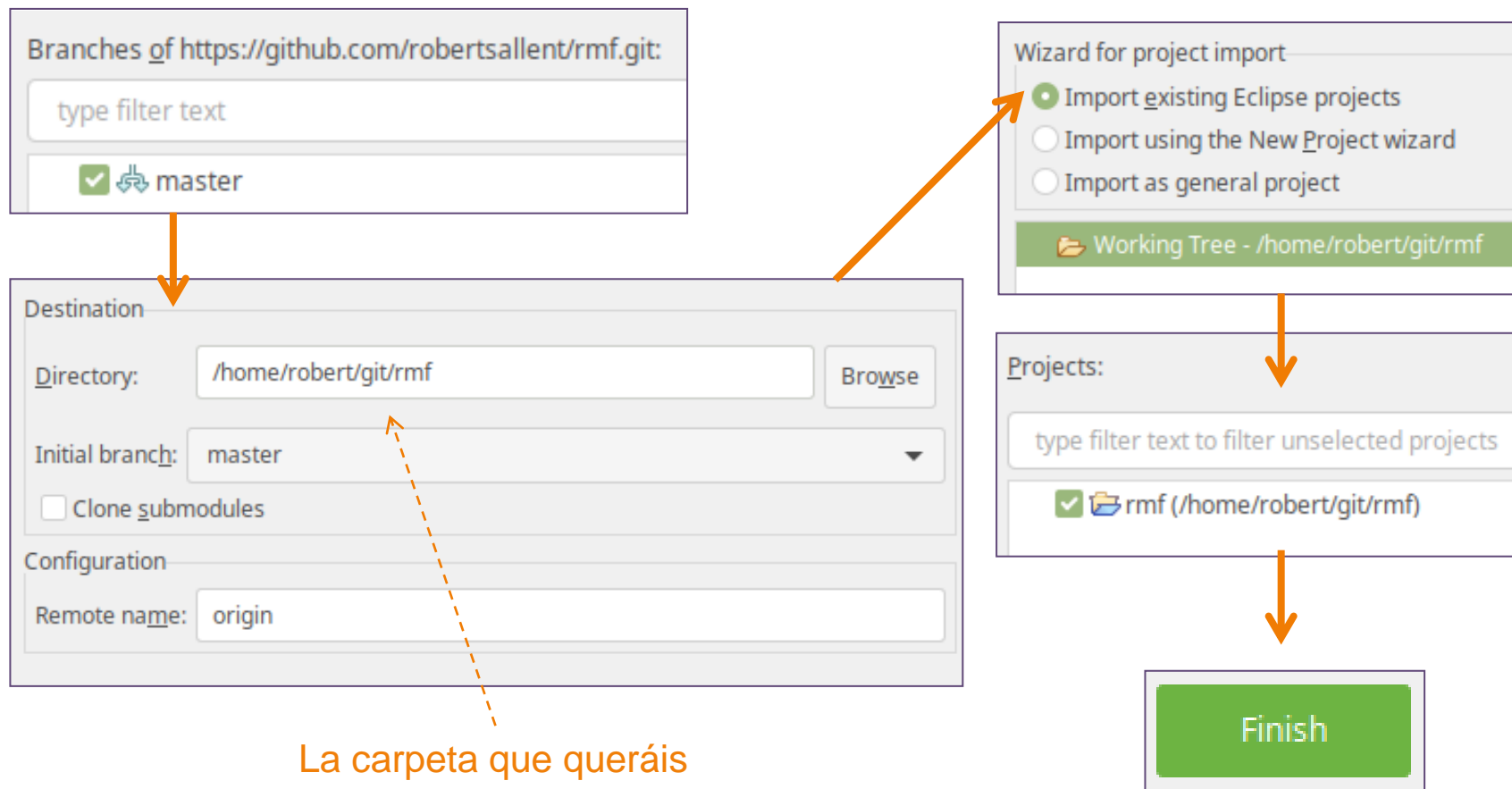


- Para ello, copiad la ruta del repositorio y, en *Eclipse*, seleccionad la opción “*file – import*”, siguiendo los pasos que se muestran a continuación.

# Ejemplo clonado del repositorio (1/2)

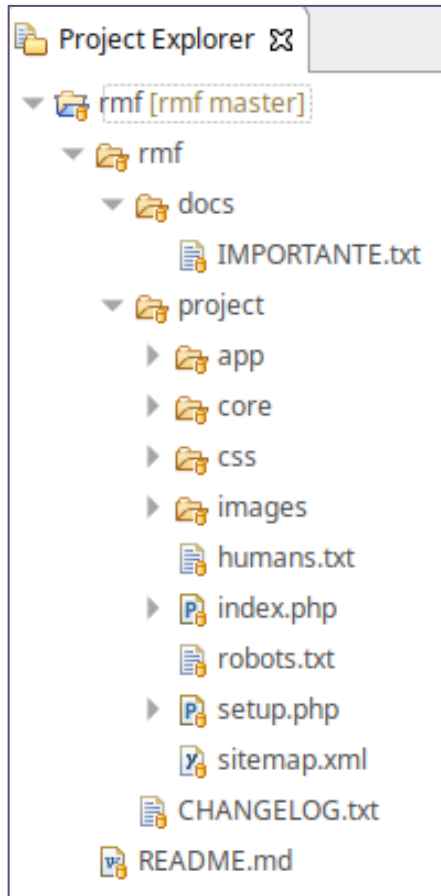


# Ejemplo clonado del repositorio (2/2)



La carpeta que queráis

# Clonando el repositorio



- Una vez finalizado el proceso, comprobaremos que el proyecto se ha importado correctamente en *Eclipse*.
- La estructura de carpetas del proyecto se comenta un poco más adelante.
- Sobre la carpeta raíz, haremos clic con el botón derecho y seleccionaremos “*team – disconnect*”.

# Creando virtualhost



- Aunque no es obligatorio, os recomiendo crear un *virtualhost* para el proyecto.
- En el ejemplo he creado uno llamado `example.local`.
- A continuación se muestra el proceso de creación del *virtualhost*, tanto para un servidor *Apache 2.4* en *Linux* como para un *Apache* en *Windows* con *XAMPP*.
- Para más información sobre cómo crear *virtualhosts* en *Apache*, consultad los apuntes de clase.

# VirtualHost

---

Creando un virtualhost con *Xampp*



# Ejemplo creando virtualhost (1/2)



- Para crear *virtualhosts* en Windows con *XAMPP*, buscaremos, abriremos y editaremos el fichero:

C:\xampp\apache\conf\extra\httpd-vhosts.conf

```
NameVirtualHost *:80

<VirtualHost *:80>
    DocumentRoot "C:/Users/Robert/GIT/rmf/rmf/project"
    ServerName example.local
    <Directory "C:/Users/Robert/GIT/rmf/rmf/project/">
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

← Vuestra ruta  
←



# Ejemplo creando virtualhost (2/2)



- Después editaremos también el fichero de hosts del sistema, ubicado en `c:\windows\system32\drivers\etc\hosts` y escribiremos en él la siguiente línea:

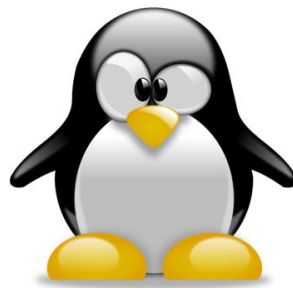
```
127.0.0.1    example.local
```

- Se debe abrir con permisos de administrador.
- Finalmente, detenemos y arrancamos de nuevo el servicio de Apache desde el panel de control de *XAMPP*.

# VirtualHost

---

Creando un virtualhost en Apache 2.4 para Linux y configurando varias cosas más



# Virtualhost en Linux



- En este caso, para configurarlo todo correctamente, haremos:
  1. Crear la configuración del nuevo sitio en la carpeta `sites-available` (para ello copiaremos y editaremos la configuración por defecto).
  2. Crear un enlace desde la carpeta `sites-enabled`.
  3. Editar el fichero de configuración de Apache `apache2.conf`.
  4. Editar el fichero de `hosts` del sistema.
  5. Habilitar el `mod_rewrite` (por si no está activado ya).
  6. Reiniciar el servidor de apache.
  7. Dar permiso para el acceso a los ficheros.

# Ejemplo creando virtualhost (1/3)



```
cd /etc/apache2/sites-available  
sudo cp 000-default.conf 001-example.conf  
sudo gedit 001-example.conf
```

```
<VirtualHost *:80>  
    ServerName example.local  
    ServerAdmin webmaster@localhost  
    DocumentRoot /home/robert/git/rmf/rmf/project  
</VirtualHost>
```

← Vuestra ruta



# Ejemplo creando virtualhost (2/3)

```
cd /etc/apache2  
sudo ln sites-available/001-example.conf  
    sites-enabled/001-example.conf
```

```
sudo gedit apache2.conf
```

```
<Directory /home/robert/git/rmf/rmf/project/>  
    Options Indexes FollowSymLinks  
    AllowOverride All  
    Require all granted  
</Directory>
```

←----- Vuestra ruta



# Ejemplo creando virtualhost (3/3)

- De paso, habilitamos el *mod\_rewrite* para las *URLs* amigables.

```
sudo a2enmod rewrite
```

- Reiniciamos el *Apache*.

```
sudo service apache2 restart
```

- Y finalmente incluimos *example.local* al fichero de hosts.

```
sudo gedit /etc/hosts
```

127.0.0.1	localhost
127.0.1.1	PHENOMII
127.0.1.1	example.local

# Permisos



- En *Linux*, podemos tener problemas de permisos en diversas operaciones (cambiar la imagen de perfil, subir una imagen al registrar un nuevo usuario...).
- Hay que dar permisos en las carpetas, de forma que el usuario *www-data* pueda trabajar con ellas.
- En este ejemplo simplemente daremos todos los permisos para la carpeta en la que hayamos colocado el proyecto.

```
sudo chmod -R 777 /home/robert/git/rmf/rmf/project
```

# Primera prueba

---

Comprobando que funciona

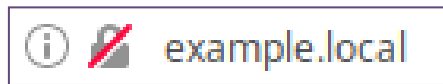




# Primera prueba



- Comprobemos que lo que hemos hecho hasta ahora funciona. Para ello, escribiremos en la barra de direcciones del navegador `example.local`.



# Sin virtualhost...



- Si has omitido la creación del *virtualhost*, tendrás que escribir en la barra de direcciones la ruta desde el *DocumentRoot*, por ejemplo: `localhost/project/`.
- Debes editar el fichero de configuración ubicado en `app/config`:

```
//Ruta donde se encuentre el proyecto, relativa al DOCUMENT_ROOT  
private $url_base = '/project/'; //EJEMPLO: '/miweb/project/';
```

- Y el fichero `.htaccess`:

```
# Base para todas las peticiones (DocumentRoot)  
RewriteBase "/project/"
```

# Ejemplo primera prueba sin virtualhost



localhost/project/





## RMF - RobS Micro Framework

Para el desarrollo de aplicaciones web

Inicio



### Presentación

Este framework PHP, ha sido desarrollado con fines docentes por **Robert Sallent** para los cursos de desarrollo de aplicaciones que imparte en diferentes centros formativos, entre los que se encuentran: **CIFO Vallès, CIFO la Violeta, ActiuHumà e IES Jaume Viladoms**.

Se trata de un Framework de trabajo con **arquitectura Modelo-Vista-Controlador** para entender los conceptos y trabajarlos en clase, siendo paso previo a trabajar con otros frameworks como **CodeIgniter, Symfony o Laravel**.

A lo largo del curso se desarrollan varios proyectos usando este pequeño framework para conocer las características comunes a este tipo de herramientas de trabajo, así como comprender cómo funcionan internamente.

Este framework implementa:

- Arquitectura MVC.
- Fichero de configuración independiente.
- Autoloader de clases.
- Router/dispatcher para la gestión de peticiones.
- Control de sesiones y gestión básica de usuarios.
- URLs amigables con *mod\_rewrite*.
- Librerías para la conexión con BDD (*mysql*), subida de ficheros, generación de XML...
- Un sencillo motor de templates basado en una clase con métodos estáticos

**NO ES 100% SEGURO**, así que no se debe usar para desarrollos en entornos de producción. Para cualquier duda o consulta, contactad conmigo mediante twitter.

RobS micro Framework para fines docentes.  
Creado por Robert Sallent en CIFO del Vallès.

 **RobS Micro Framework**

Robert Sallent  
[www.robertsallent.com](http://www.robertsallent.com)

**SOC** Servei d'Ocupació de Catalunya  
CIFO Sabadell- Mancomunitat (Vallès)

 **Generalitat de Catalunya**

 **Unió Europea**  
Fons social europeu  
L'FSE inverteix en el teu futur

35

# La base de datos

---

Creando y configurando la base de datos  
para nuestro proyecto



# Creando la base de datos



- *RMF* necesita de una base de datos.
- Esta pensado para trabajar con *MySQL/MariaDB* y usa el conector *mysqli* de *PHP*.
- Crearemos una *BDD* en nuestro servidor mediante algún cliente (*phpMyAdmin*, *MySQL Workbench*...). En el ejemplo lo hago con *phpMyAdmin*.

Crear base de datos ?

example utf8\_spanish\_ci Crear

# Configurando el framework



- Tras crear la base de datos, procederemos a configurar el *RMF*.
  - Para ello, disponemos del fichero `app/config/Config.php`.
  - El fichero contiene la clase *Config*, donde las propiedades se corresponden con los parámetros configurables,
  - De entrada, tendremos que configurar al menos la base de datos.



# Ejemplo el fichero Config.php

```
//CONFIGURACIÓN GENERAL
private $app_name='RMF - RobS Micro Framework'; //nombre de la aplicación
private $company_name=''; //empresa para la que se desarrolla la aplicación

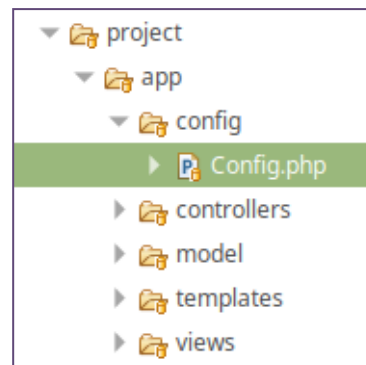
//Ruta donde se encuentre el proyecto, relativa al DOCUMENT_ROOT
private $url_base = '/'; //EJEMPLO: '/miweb/project/';

//ENTORNO (development | production)
//algunos mensajes de error solamente aparecerán en development
private $environment='development';

private $locale='es'; //localización de idioma

//CONFIGURACIÓN DE LA BASE DE DATOS
private $db_host = 'localhost'; //host donde se encuentra la BDD (normalmente localhost)
private $db_user = 'robert'; //usuario
private $db_pass = '1234'; //password
private $db_name = 'example'; //nombre de la BDD a utilizar
private $db_charset = 'utf8'; //codificación de caracteres
private $db_user_table = 'usuarios'; //nombre de la tabla para los usuarios

//CONTROLADOR Y OPERACION POR DEFECTO
private $default_controller = 'Welcome'; //controlador por defecto
private $default_method = 'index'; //método por defecto
```



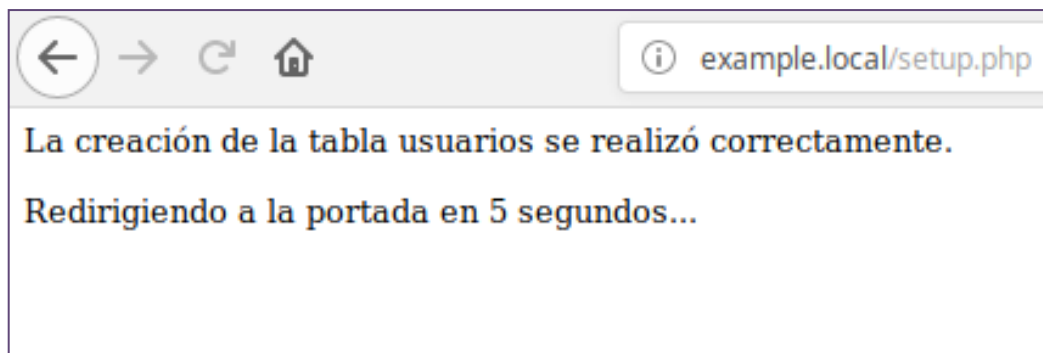
# Ejecutando setup.php



- Una vez creada la base de datos y configurado el proyecto, ejecutaremos el script de nombre `setup.php`, escribiendo en la barra de direcciones del navegador `example.local/setup.php`.
- Esto creará la tabla para los usuarios y un usuario *admin* con *password* '1234'.
- Si el *script* se ha ejecutado correctamente, se te redirigirá a la portada y se borrará automáticamente el fichero `setup.php`.
- Después deberás identificarte como *admin* y cambiar el *password* (en clase no será necesario).



# Ejemplo ejecutando setup.php



# Realizando pruebas

---

Comprobando que todo funciona  
correctamente



# Realizando pruebas




- En el *RMF*, ya se encuentran implementadas las operaciones básicas con usuarios: registro, *login* y *logout*, modificación de datos. y baja.
- Además existe un panel para el administrador, desde el que puede gestionar las diferentes operaciones (*CRUD*) con usuarios:
  - Crear usuario.
  - Listar usuarios.
  - Ver usuario.
  - Modificar usuario.
  - Eliminar usuario.

# Ejemplo registrando un nuevo usuario



## Formulario de registro

 Rellena todos los campos para completar el proceso de registro.

User:

Password:

Nombre:

Email:

Imagen:  student.png max 1000KB



## Éxito

La operación solicitada se ha completado satisfactoriamente, con el siguiente mensaje:

Operación de registro completada con éxito

# Ejemplo login y visualización de perfil




[Registro](#)

Hola CIFO DEL VALLES (cifo@gencat.cat)



## Perfil de usuario

 Estos son tus datos de usuario.

Mis datos	
Usuario:	cifo
Nombre:	CIFO DEL VALLES
Email:	cifo@gencat.cat
Fecha de alta:	05/04/2019
Hora de alta:	05:56:56



# Ejemplo modificación de datos



User: cifo

Password actual: ....

Nuevo password: En blanco para no cambiarlo


Nombre: CIFO VIOLETA

Email: cifovioleta@gencat.cat

Nueva imagen:  student2.jpg

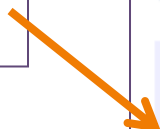


## Formulario de modificación de datos


 Modifica tus datos mediante el formulario.

Modificación realizada correctamente.

# Ejemplo baja de usuario



## Formulario de baja de usuario

 Por favor, confirma tu solicitud de baja introduciendo el password asociado a tu cuenta.

User:

cifo

Password:

....

Confirmar

# Ejemplo cambiar el password de admin




admin [password field] Login → Hola Administrador (admin@rmf)

**Perfil de usuario**

*i* Estos son tus datos de usuario.

Mis datos	
Usuario:	admin
Nombre:	Administrador
Email:	admin@rmf
Fecha de alta:	19/03/2019
Hora de alta:	08:21:41



[Editar mis datos](#)

**Formulario de modificación de datos**

*i* Modifica tus datos mediante el formulario.

User:

Password actual:

Nuevo password:  En blanco para no modificar el actual

Nombre:

Email:

Nueva imagen: [Seleccionar archivo](#) Ningún archivo seleccionado max 1000KB

[modificar](#)



[Darse de baja](#)



# Ejemplo panel del administrador



Hola **Administrador** (*admin@rmf*) , eres administrador [Logout](#)

[Inicio](#) **ADMIN**

## Panel de control

### Operaciones para el administrador

A continuación se muestran las operaciones que puede hacer el administrador.

- Listar
- Nuevo

Usuarios

- Operación 1
- Operación 2

Entidad 1

- Operación 1
- Operación 2

Entidad 2

- 2 usuarios registrados

Estadísticas

[Atrás](#) [Inicio](#)



# Ejemplo administrador visualiza un usuario



## Lista de usuarios registrados












 Lista de usuarios registrados en la aplicación RMF - RobS Micro Framework

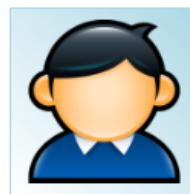
Imagen	Usuario	Nombre	Email	Operaciones
	admin	Administrador	<a href="mailto:admin@rmf">admin@rmf</a>	  
	robert	Robert	<a href="mailto:robert@juegayestudia.com">robert@juegayestudia.com</a>	  

 Crear un nuevo usuario

## Perfil de usuario

 Visualizando los datos del usuario: robert (Robert)

Datos del usuario robert	
Nombre de usuario:	robert
Nombre:	Robert
Email:	<a href="mailto:robert@juegayestudia.com">robert@juegayestudia.com</a>
Nivel de privilegio:	500
Administrador:	NO
Fecha de alta:	04/04/2019
Hora de alta:	08:07:32





# Ejemplo administrador modifica un usuario

## Lista de usuarios registrados

Lista de usuarios registrados en la aplicación RMF - RobS Micro Framework

Imagen	Usuario	Nombre	Email	Operaciones
	admin	Administrador	<a href="mailto:admin@rmf">admin@rmf</a>	
	robert	Robert	<a href="mailto:robert@juegayestudia.com">robert@juegayestudia.com</a>	

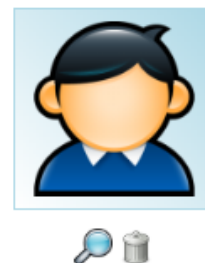
[Crear un nuevo usuario](#)



## Formulario de modificación de datos

Usuario para el que se van a modificar los datos: robert (robert@juegayestudia.com)

User:   
Nombre:   
Email:   
Nivel de privilegio:   
Administrador: ☐  
Nueva imagen:  Ningún archivo seleccionado max 1000KB





# Ejemplo administrador elimina un usuario

## Lista de usuarios registrados

Lista de usuarios registrados en la aplicación RMF - RobS Micro Framework

Imagen	Usuario	Nombre	Email	Operaciones
	admin	Administrador	<a href="mailto:admin@rmf">admin@rmf</a>	
	robert	Robert	<a href="mailto:robert@juegayestudia.com">robert@juegayestudia.com</a>	

Crear un nuevo usuario

## Eliminación de usuario

Por favor, confirma la eliminación del usuario: **robert** (Robert)

Usuario a eliminar: **robert**

Nombre real: **Robert**

Email: **robert@juegayestudia.com**



# Carpetas y ficheros

---

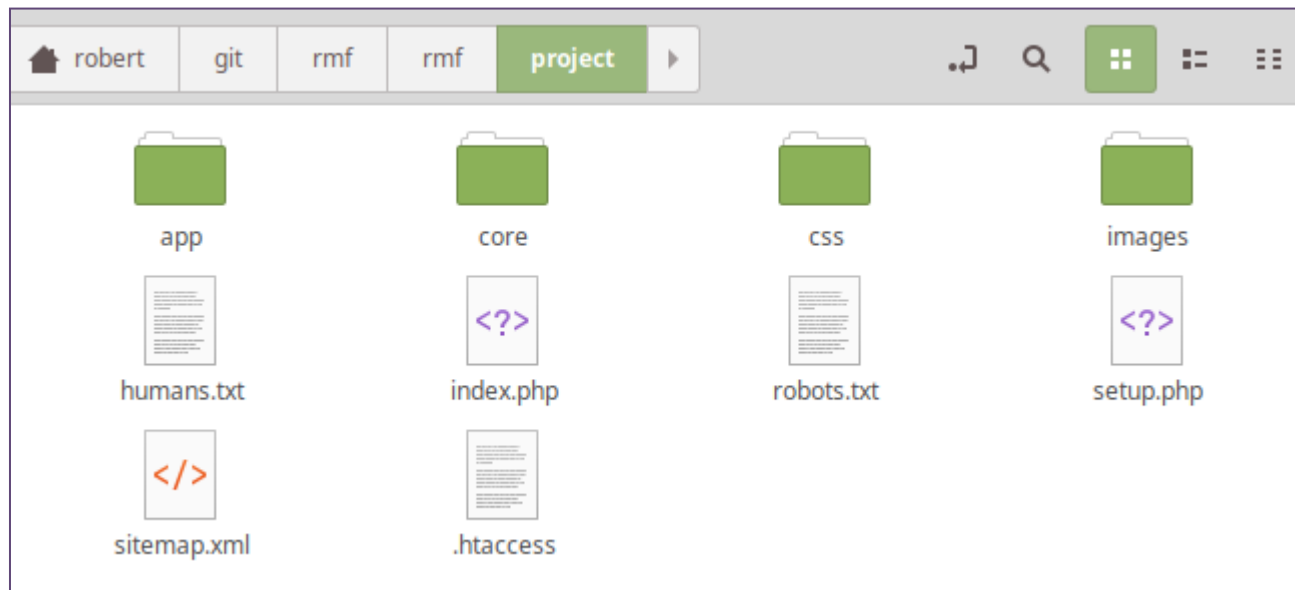
## Carpetas del RMF y ficheros importantes



# Carpetas y ficheros



- Dentro de la carpeta `project`, se encuentra todo el *framework*.



# Carpetas y ficheros



- Fichero `index.php`: punto de entrada para todas las peticiones.
- Fichero `setup.php`: *script* para la creación de la tabla usuarios (se debe eliminar tras ser ejecutado satisfactoriamente).
- Fichero `.htaccess`: contiene la configuración para el *Apache*, a destacar la configuración para las *URLs* amigables.
- Ficheros `robots.txt`, `humans.txt` y `sitemaps.xml`: permiten definir las rutas de exclusión para robots, la información acerca de los autores del sitio y el mapa del sitio. Útiles para *SEO*.

# Carpetas y ficheros



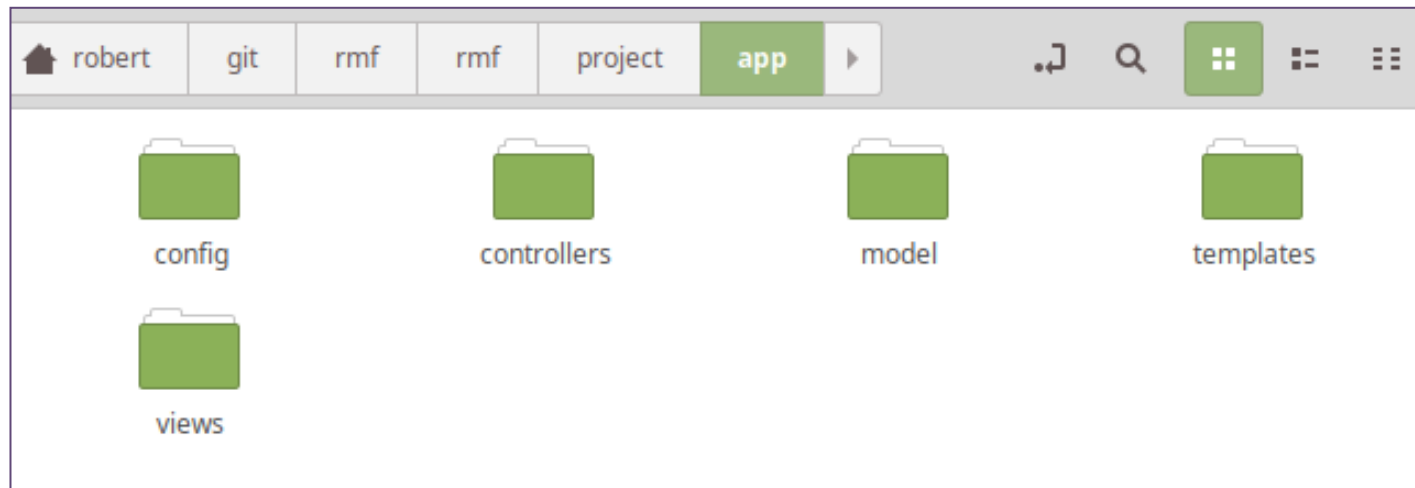
- Carpeta `app`: ficheros que forman parte de la aplicación de usuario.
- Carpeta `core`: ficheros que forman parte de la estructura del *framework* y que no es necesario modificar para proyectos sencillos.
- Carpeta `css`: ficheros CSS.
- Carpeta `images`: carpeta para las imágenes.



# Carpetas y ficheros



- Dentro de la carpeta `app`, encontramos:



# Carpetas y ficheros

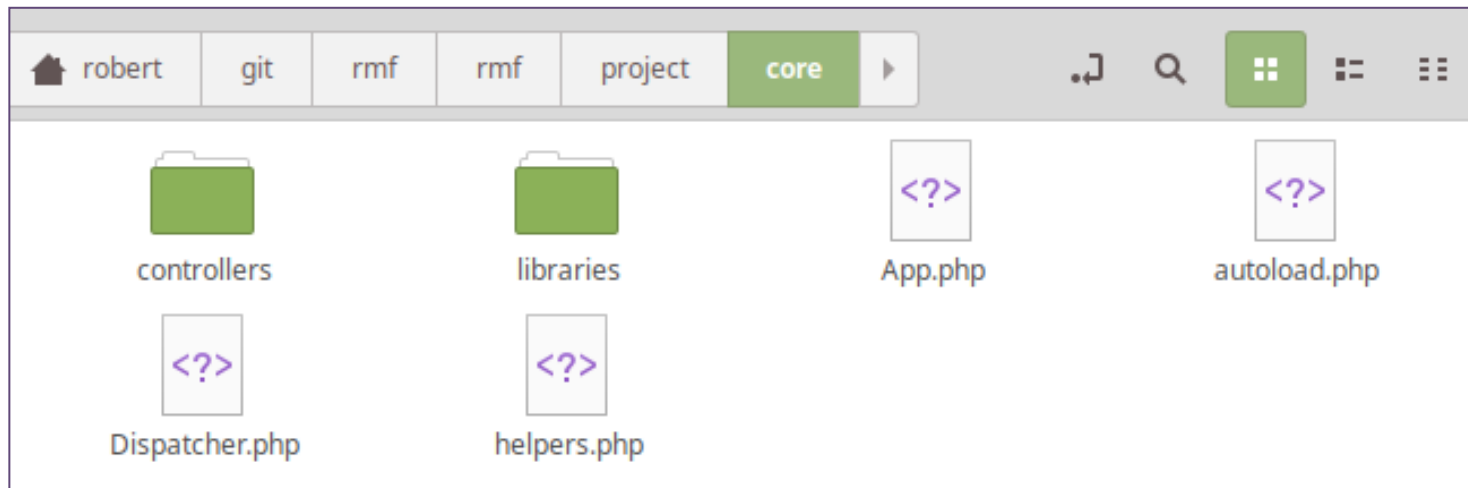


- Carpeta `app/config`: contiene el fichero *Config.php*.
- Carpeta `app/controllers`: controladores del *MVC*. Inicialmente, solamente contiene el controlador *Usuario*.
- Carpeta `app/models`: carpeta para los modelos del *MVC*. Inicialmente contiene el modelo *UsuarioModel*.
- Carpeta `app/templates`: *templates* para el *HTML* de las vistas.
- Carpeta `app/views`: vistas.

# Carpetas y ficheros



- Dentro de la carpeta `core` encontramos:



# Carpetas y ficheros



- Carpeta `core/controllers`: contiene la clase *Controller*, de la que deben heredar todos los controladores que implementemos.
- Carpeta `core/libraries`: contiene las librerías para gestionar el *Login*, exportar e importar a *XML*, conectar con la *BDD* ...
- Ficheros `core/App.php` y `core/Dispatcher.php`: actúan como controlador frontal: analizan y gestionan las peticiones además de controlar los errores que se puedan producir durante la ejecución.
- Fichero `core/helpers.php`: funciones globales útiles.
- Fichero `core/autoload.php`: *autoload* de clases.