

# Symfony

---

## Tutorial de instalación y uso en Ubuntu

**Lucas Matwiejczuk . Cristian Bouix, Héctor Maidana**

**13/12/2011**

En el siguiente documento se explicará como instalar todo lo necesario para usar el framework Symfony y como hacer aplicaciones con él de una forma fácil y rápida.

# Introducción

---

## Que es Symfony?

Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja.

Ademas, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

## Características

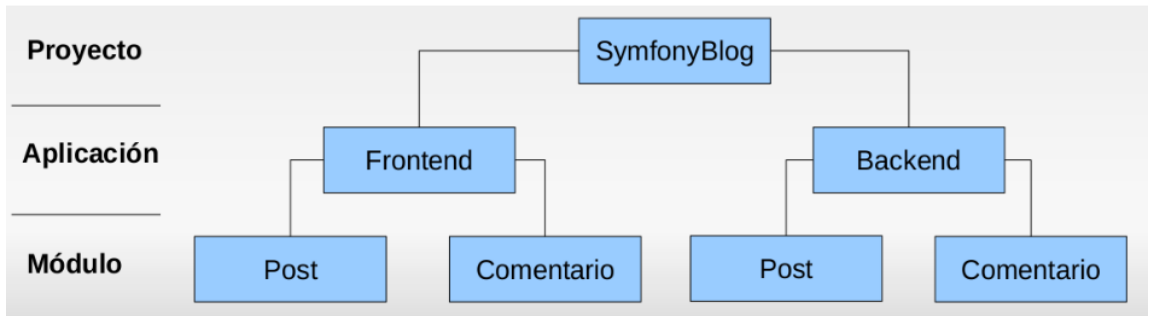
- Symfony está desarrollado completamente con PHP 5.
- Es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft.
- Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC.
- El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista

## Estructura

El código está dividido en muchos directorios y muchos scripts y los archivos son un conjunto de clases PHP, código HTML e incluso una mezcla de los dos. Además, existen referencias a clases que no se pueden encontrar dentro del directorio del proyecto y la anidación de directorios puede llegar hasta los seis niveles.

## Estructura del proyecto: Aplicaciones, Módulos y Acciones

Dentro de un proyecto, las operaciones se agrupan de forma lógica en aplicaciones. Normalmente, una aplicación se ejecuta de forma independiente respecto de otras aplicaciones del mismo proyecto. Lo habitual es que un proyecto contenga dos aplicaciones: una para la parte pública y otra para la parte de gestión, compartiendo ambas la misma base de datos. Cada aplicación está formada por uno o más módulos. Un módulo normalmente representa a una página web o a un grupo de páginas con un propósito relacionado.



Los módulos almacenan las acciones, que representan cada una de las operaciones que se puede realizar en un módulo.

# Instalación

---

## Preinstalación:

Antes que nada, instalamos php5 (debe ser como mínimo php 5.2.4):

➤ `sudo apt-get install php5`

Si ya lo tienes instalado y quieres saber la versión ejecutar:

➤ `php -v`

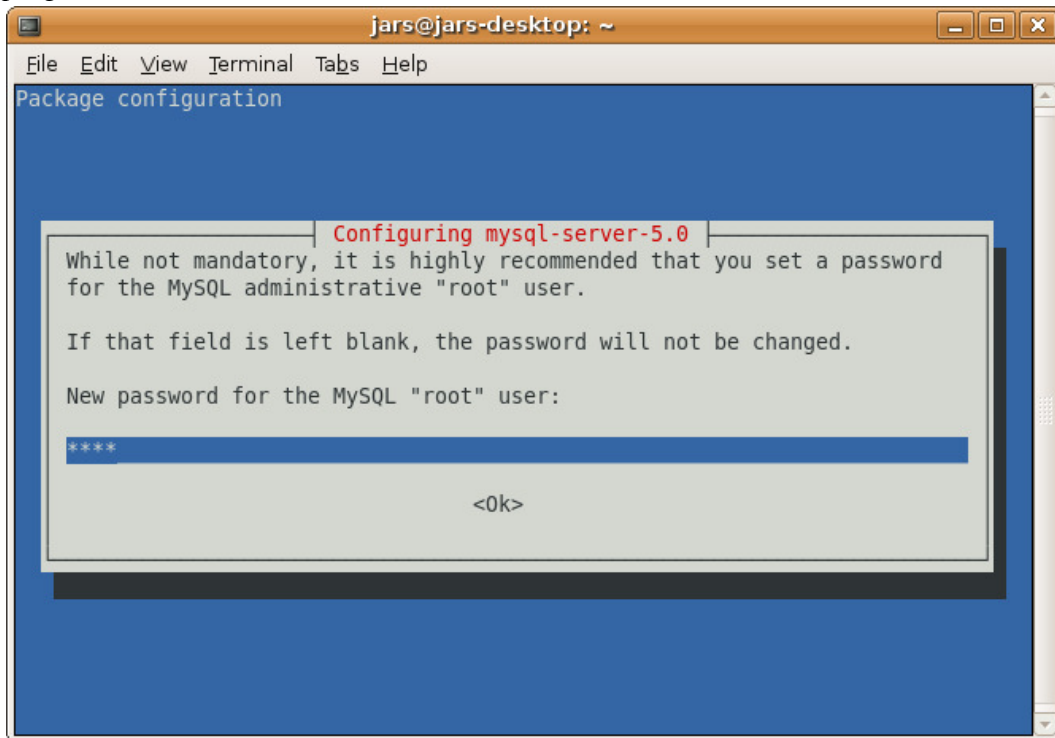
Instalamos apache el servidor HTTP y configuramos:

➤ `sudo apt-get install apache2`

Instalamos mysql:

➤ `sudo apt-get install mysql-server`

Aparecerá una pantalla como esta para configurar el mysql-server que solo te pedirá que le pongas una contraseña:



Instalamos la extensión para MySQL:

- `sudo apt-get install php5-mysql`

Necesitamos instalar un acelerador php:

- `sudo apt-get install php5-xsl php-apc`

*\*Un Acelerador PHP toma uno o más script ya compilados, los guarda en una memoria compartida y lo usa cada vez que se solicite, mejorando así el rendimiento de un servidor.*

Y puesto que todo esto lo estamos pensando para utilizar Symfony será muy recomendable instalar el paquete de línea de comandos, CLI:

- `sudo apt-get install php5-cli`

*\* PHP CLI (Interfaz de Línea de Comandos PHP) Como su nombre lo indica, esta es una forma de usar PHP en la línea de comandos del sistema. O por otras palabras, es una forma de ejecutar scripts PHP que no están en un servidor web (por ejemplo, el servidor web Apache o Microsoft IIS).*

Para asegurarnos de que los cambios surgen efecto, reiniciamos el servidor apache:

- `sudo /etc/init.d/apache2 restart`

## Instalación de Symfony

Descargar symfony de la página oficial:

➤ <http://symfony.com/download>

[Home](#) » [Get Started](#) » [Download](#)

# Download



**Symfony Standard Edition**

The Symfony Standard Edition is the best distribution to use when starting a new project. It contains the most common bundles, and comes with a simple configuration system.

Symfony Standard 2.0.7 (.tgz)

[http://symfony.com/download?v=Symfony\\_Standard\\_Vendors\\_2.0.7.tgz](http://symfony.com/download?v=Symfony_Standard_Vendors_2.0.7.tgz)

**DOWNLOAD NOW**

**INSTALLATION INSTRUCTIONS**

1. Download one of the archives somewhere under your local web server's root directory and unpack it. If you have Git installed, you should download Symfony2 "without vendors", as it adds a bit more flexibility.
2. Follow the instructions in the README.md file (which is just a text file) that's packaged with the download.
3. Enjoy!

Mover el archivo al directorio raíz del servidor web:

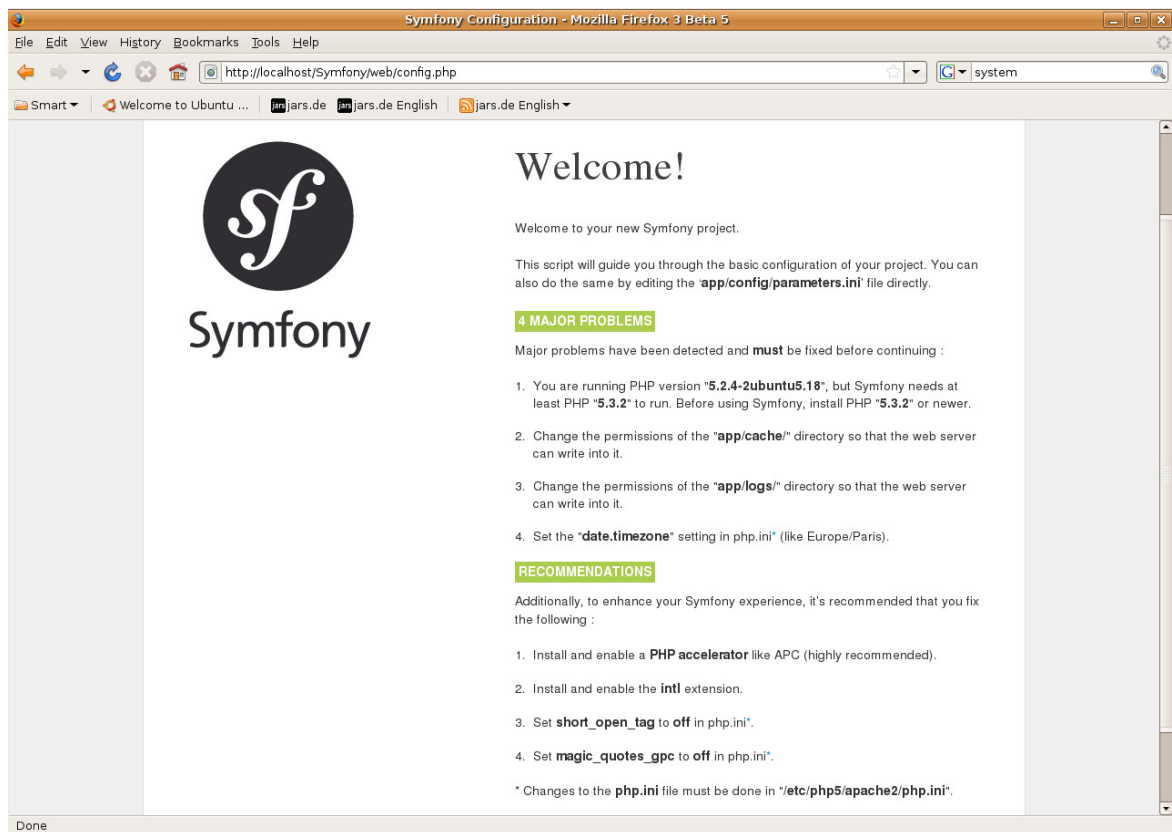
➤ `mv Symfony_Standard_Vendors_2.0.7.tgz /var/www`

Ir al directorio /var/www y descomprimir el archivo:

➤ `sudo tar xzf Symfony_Standard_Vendors_2.0.7.tgz`

Podemos ingresar a la URL:

➤ <http://localhost/Symfony/web/config.php>



## Problemas y sus soluciones:

Instalar SQLite:

➤ `apt-get install php5-sqlite`

Instalar la extension intl:

➤ `apt-get install php5-intl`

Cambiar permisos a los directorios "app/cache" y "app/logs":

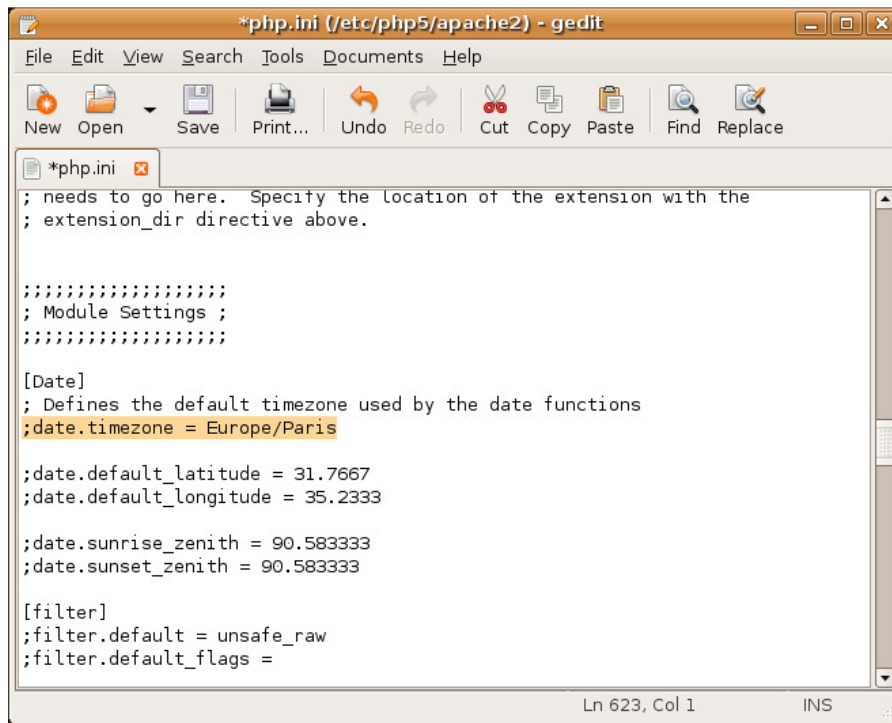
➤ `sudo chmod 777 /var/www/Symfony/app/cache/`  
➤ `sudo chmod 777 /var/www/Symfony/app/logs/`

## Configurar el archivo /etc/php5/apache2/php.ini :

Ir al directorio /etc/php5/apache2/ y editar el php.ini con gedit:

➤ `sudo gedit php.ini`

Establecer el "date.timezone" con por ejemplo Europe/Paris:



The screenshot shows a gedit window titled "\*php.ini (/etc/php5/apache2) - gedit". The window contains the following configuration for the [Date] section:

```
; needs to go here. Specify the location of the extension with the
; extension_dir directive above.

;;;;;;;;;;;;;;;;;;;;;;;;;
; Module Settings ;
;;;;;;;;;;;;;;;;;;;;;;;;;

[Date]
; Defines the default timezone used by the date functions
;date.timezone = Europe/Paris

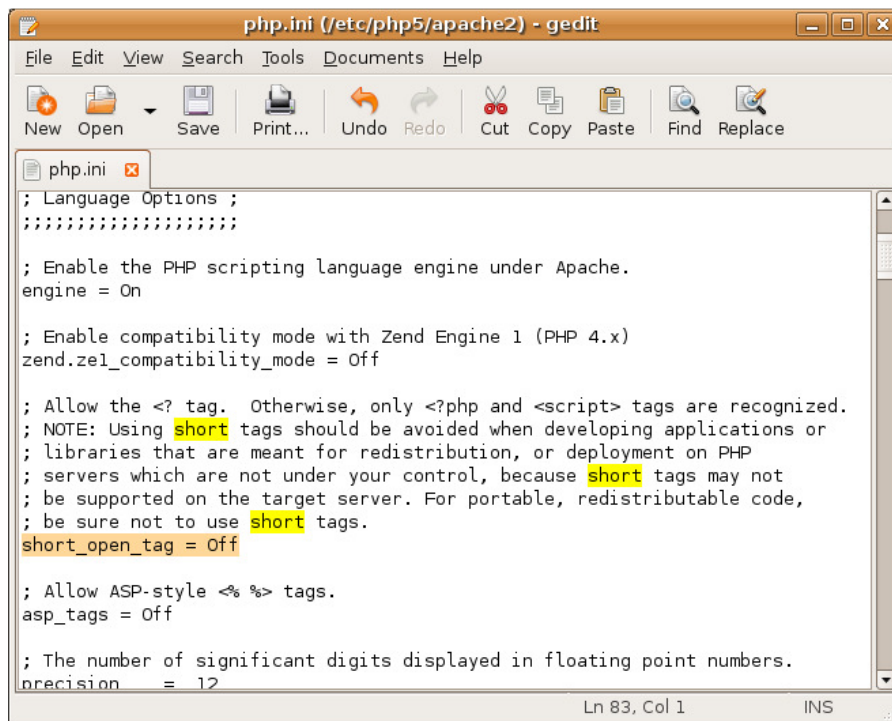
;date.default_latitude = 31.7667
;date.default_longitude = 35.2333

;date.sunrise_zenith = 90.583333
;date.sunset_zenith = 90.583333

[filter]
;filter.default = unsafe_raw
;filter.default_flags =
```

The status bar at the bottom indicates "Ln 623, Col 1" and "INS".

Establecer "short\_open\_tag" en Off:



The screenshot shows a gedit window titled "php.ini (/etc/php5/apache2) - gedit". The window contains the following configuration for the [Language Options] section:

```
; Language Options ;
;;;;;;;;;;;;;;;;;;;;;;;;;

; Enable the PHP scripting language engine under Apache.
engine = On

; Enable compatibility mode with Zend Engine 1 (PHP 4.x)
zend.z1_compatibility_mode = Off

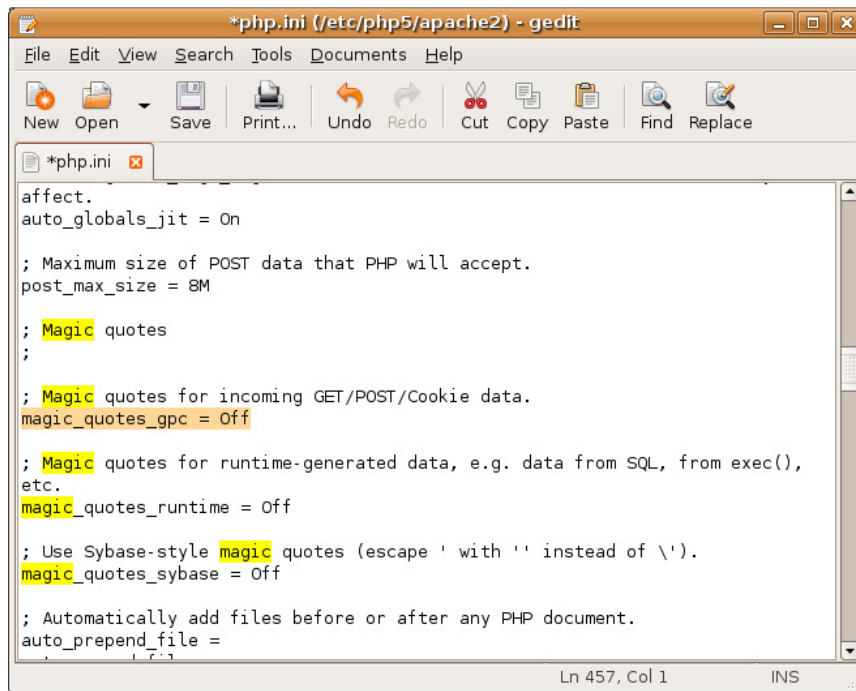
; Allow the <? tag. Otherwise, only <?php and <script> tags are recognized.
; NOTE: Using short tags should be avoided when developing applications or
; libraries that are meant for redistribution, or deployment on PHP
; servers which are not under your control, because short tags may not
; be supported on the target server. For portable, redistributable code,
; be sure not to use short tags.
short_open_tag = Off

; Allow ASP-style <% %> tags.
asp_tags = Off

; The number of significant digits displayed in floating point numbers.
precision = 12
```

The status bar at the bottom indicates "Ln 83, Col 1" and "INS".

Establecer "magic\_quotes\_gpc" en Off:



Reiniciar el servidor apache

- `sudo apache2ctl restart`

## Usando Symfony

---

### Bundle

Un bundle no es más que un directorio que tiene una estructura definida, puede alojar cualquier cosa, desde clases para los controladores y recursos web. Symfony está compuesto de bundles en su totalidad, el mismo núcleo de Symfony2 es un bundle.

### Crear un Bundle:

En la carpeta `/var/www/Symfony`

- `php app/console generate:bundle`

### Crear las Entidades:

En la carpeta `/var/www/Symfony`

- `sudo php app/console generate:doctrine:entity`



## Configurar la Base de Datos:

En la carpeta /var/www/Symfony

➤ `sudo gedit app/config/parameters.ini`

## Crear la Base de Datos:

➤ `sudo mysql -h localhost -u root -p`  
➤ `mysql> CREATE DATABASE nombre_de_la_base_de_datos_igual_que_el_.ini;`

## Crear la tabla de la entidad con:

➤ `php app/console doctrine:schema:update --force`

## Crear el Crud en /var/www/Symfony\$:

➤ `sudo php app/console generate:doctrine:crud`

## Agregar rutas de las entidades creadas al app/config/routing\_dev.yml porque no se agregan solas:

```
"entidad":
  pattern:  /"entidad"
  defaults: { _controller: MyBundle:"entidad":index }
"entidad"_new:
  pattern:  /"entidad"_new
  defaults: { _controller: MyBundle:"entidad":new }
"entidad"_show:
  pattern:  /"entidad"_show/{id}
  defaults: { _controller: MyBundle:"entidad":show }
"entidad"_edit:
  pattern:  /"entidad"_edit/{id}
  defaults: { _controller: MyBundle:"entidad":edit }
"entidad"_create:
  pattern:  /"entidad"_create
  defaults: { _controller: MyBundle:"entidad":create }
"entidad"_delete:
  pattern:  /"entidad"_delete/{id}
  defaults: { _controller: MyBundle:"entidad":delete }
"entidad"_update:
  pattern:  /"entidad"_update/{id}
  defaults: { _controller: MyBundle:"entidad":update }
```

# Instalación y configuración de Bundle

---

## FOSUserBundle

### 1. Agregar dependencias

Ir al archivo `deps` y agregar:

```
[FOSUserBundle]
    git=git://github.com/FriendsOfSymfony/FOSUserBundle.git
    target=bundles/FOS/UserBundle
```

Y luego instalarlos:

```
➤ php bin/vendors install
```

O de esta manera:

```
➤ git submodule add
    git://github.com/FriendsOfSymfony/FOSUserBundle.git
    vendor/bundles/FOS/UserBundle
➤ git submodule update -init
```

O bajar el `bundle.zip`, descomprimirlo, y copiar el contenido en `/vendor/bundle/nombre_organizacion/nombre_bundle/`.

### 2. Configurar el Autoloader

Ir al archivo `app/autoload.php`

```
➤ loader->registerNamespaces(array(
    // ...
    'FOS' => __DIR__.'/../vendor/bundles',
));
```

### 3. Agregar la ruta del Bundle creado

Ir al archivo `app/AppKernel.php` y agregar:

```
public function registerBundles()
{
    $bundles = array(
        // ...
        new FOS\UserBundle\FOSUserBundle(),
    );
}
```

## 4. Clase Usuario

Si tienes una clase Usuario extenderla de BaseUser o sino tienes que crear una y el id debe ser protected.

```
➤ use FOS\UserBundle\Entity\User as BaseUser;
➤ use Doctrine\ORM\Mapping as ORM;

/**
 * @ORM\Entity
 * @ORM\Table(name="fos_user")
 */
class User extends BaseUser
{
    /**
     * @ORM\Id
     * @ORM\Column(type="integer")
     * @ORM\GeneratedValue(strategy="AUTO")
     */
    protected $id;

    public function __construct()
    {
        parent::__construct();
        // your own logic
    }
}
```

## 5. Configurar el security.yml

Ir al archivo app/config/security.yml y agregar:

```
security:
    providers:
        fos_userbundle:
            id: fos_user.user_manager

    firewalls:
        main:
            pattern: ^/
            form_login:
                provider: fos_userbundle
            logout: true
            anonymous: true

    access_control:
        - { path: ^/login$, role: IS_AUTHENTICATED_ANONYMOUSLY }
        - { path: ^/register, role: IS_AUTHENTICATED_ANONYMOUSLY }
        - { path: ^/resetting, role: IS_AUTHENTICATED_ANONYMOUSLY }
        - { path: ^/admin/, role: ROLE_ADMIN }

    role_hierarchy:
        ROLE_ADMIN: ROLE_USER
        ROLE_SUPER_ADMIN: ROLE_ADMIN
```

## 6. Configurar el FOSUserBundle

Agregar la ruta de la clase User:

```
fos_user:
  db_driver: orm # other valid values are 'mongodb', 'couchdb'
  and 'propel'
  firewall_name: main
  user_class: Acme\UserBundle\Entity\User
```

## 7. Agregar las rutas del Bundle

En el archivo app/config/routing.yml agregar:

```
fos_user_security:
  resource:
"@FOSUserBundle/Resources/config/routing/security.xml"

fos_user_profile:
  resource:
"@FOSUserBundle/Resources/config/routing/profile.xml"
  prefix: /profile

fos_user_register:
  resource:
"@FOSUserBundle/Resources/config/routing/registration.xml"
  prefix: /register

fos_user_resetting:
  resource:
"@FOSUserBundle/Resources/config/routing/resetting.xml"
  prefix: /resetting

fos_user_change_password:
  resource:
"@FOSUserBundle/Resources/config/routing/change_password.xml"
  prefix: /profile
```

## 8. Actualizar la Base de Datos

Por último actualizamos la base de datos con:

➤ `php app/console doctrine:schema:update --force`