

INDICE DEL CURSO

- **Introducción (19-oct)**
 - Instalación de symfony, creación nuevo proyecto, la consola de symfony, composer, integración con git, permisos de ficheros, servidor integrado de php
 - Estructura de un proyecto symfony.
- **Creación de la página de Home (19-oct)**
 - Introducción a Controllers
 - Introducción a Twig: etiquetas, filtros y funciones
 - Introducción al Routing
 - Internacionalización (i18n)
- **Creación de otras páginas de la zona pública (20-oct)**
 - Introducción a la consola de symfony
 - Twig: Herencia de plantillas
 - Routing avanzado
 - Twig: Generación de rutas
 - Twig: Avanzado
 - Archivos de configuración de symfony. Entornos de ejecución.
 - Caché
- **Acceso a base de datos. Doctrine. (26-oct)**
- **Formularios y validaciones. Mensajes Flash. (27-oct)**
- **Zona privada. Página de dashboard del usuario. (2-nov)**
 - Organización del código en bundles y controladores. Buenas prácticas.
 - Organización de plantillas. Buenas prácticas de organización de plantillas.
 - Configuración de la **seguridad** y de los roles, firewalls, sistema de login...
 - **FOSUserBundle**
 - Instalación de bundles con composer. Instalación de FOSUserBundle
 - Configuración de bundles. Configuración de FOSUserBundle
 - Funcionalidades de FOSUserBundle.

- **Creación de otras partes de la zona privada (3-nov)**
 - Controllers (Los objetos Request, Response, y User). Buenas prácticas en la programación de controladores
 - Identificación del usuario logeado, detección del rol(es) del usuario, detección del idioma del usuario (i18n).
 - El profiler. Entornos de ejecución (desarrollo, producción).
 - Logs
 - Envío de correos
 - Eventos
 - Creación de comandos de consola personalizados.

- **Acceso a APIs REST u otras fuentes de datos (Web Services) (7-nov)**
 - Servicios
 - Buenas prácticas en la programación de servicios.
 - Tratamiento de peticiones en formato JSON.
 - El contenedor de servicios y la inyección de dependencias
 - Parámetros de configuración personalizados. Configuración adaptada al entorno de ejecución (desarrollo, producción)

- **Creación de una API REST (7-nov)**
 - Construcción de respuestas en formato JSON.
 - Buenas prácticas.

- **Creación de la zona de admin (16-nov)**
 - SonataBundle (equivalente al plugin para admin de symfony 1)

- **Componente Workflow (16-nov)**

- **Integración con PHPUnit. Buenas prácticas de testeo. (17-nov)**

- **Integración continua con Jenkins (17-nov)**

19, 20, 26 y 27 de octubre, 2, 3, 7, 16 y 17 de noviembre, de 9:30 a 14:00 horas

Versión 3.3 de symfony:

Versión 7.X de PHP

1. INTRODUCCIÓN

Ejercicio: Instalar symfony

Ejercicio: Crear un proyecto y arrancar el server

```
> symfony new curso  
> cd curso  
> php bin/console server:run
```

Comprobar que aparece pantalla de bienvenida. Este proyecto será para ir haciendo ejercicios.

Ejercicio: Bajar proyecto symfony de github, ejecutar composer y arrancar el server

```
> git clone https://github.com/carherco/curso_symfony3  
  
> cd curso_symfony3  
> composer install  
> php bin/console server:run
```

Este proyecto tienen todos los ejercicios resueltos, servirá como referencia.

EXPLICAR COMPOSER - No hay ejercicios

EXPLICAR GIT

Ejercicios con git: Realizar estas operaciones y después de cada una, ejecutar git status

- Modificar archivos
- Crear archivos
- Borrar archivos
- Deshacer cambios de un archivo modificado
- Recuperar archivo borrado
- (Los archivos nuevos no están gestionados por git hasta que se realice git add)
- Añadir archivos al staging area
- Modificar archivos ya añadidos al staging area
- Quitar algún archivo del staging area
- Hacer commit

ESTRUCTURA DE UN PROYECTO SYMFONY

Ejercicios:

- ejecutar bin/console
- ejecutar php bin/symfony_requirements
- acceder al fichero parameters.yml a través del navegador

http://localhost:8888/carlos/curso_symfony3/app/config/parameters.yml

EXPLICAR CREACIÓN DE PÁGINAS

Ejemplo "Hola Mundo" (DefaultController::helloAction)

Ejemplo "Hola Mundo con template" (DefaultController::hello2Action)

Ejemplo "Hola Mundo con variable" (DefaultController::hello3Action)

Ejemplo "Hola Mundo con html completo" (DefaultController::hello4Action)

EXPLICAR I18N básico

Ejemplo "Hola Mundo con traducción" (DefaultController::hello5Action)

```
> bin/console translation:update --dump-messages es  
> bin/console translation:update --force es  
> bin/console translation:update --force en  
> bin/console cache:clear
```

Ejercicio Página de bienvenida con plantilla de bootstrap 4.

EXPLICAR TWIG BÁSICO (hasta funciones)

Ejercicio: Listado de usuarios

- Tabla con asignaturas (for)
- Si no hay asignaturas, mensaje de que no hay (if)
- color distinto líneas pares/impares (if + index)
- Nombre de asignatura según idioma (if)
- Cabecera de la tabla y mensaje de no hay asignaturas en doble idioma (|trans, bin/console translation:update)
- Fecha de generación del informe (fecha actual, formateada) (|date)

Ejercicio: Probar a poner {{asig.code}} en vez de {{asig.codigo}} y ver el mensaje de error.

2. PARTE PÚBLICA

Ejercicio: Generar controlador con 2 acciones: homeAction y loginAction

```
> bin/console generate:controller
```

```
created ./src/AppBundle/Resources/views/Public/  
created ./src/AppBundle/Resources/views/Public/home.html.twig  
created ./src/AppBundle/Resources/views/Public/login.html.twig  
created ./src/AppBundle/Controller/PublicController.php  
created ./src/AppBundle/Tests/Controller/  
created ./src/AppBundle/Tests/Controller/PublicControllerTest.php  
Generating the bundle code: OK
```

Ejercicio: Echar un vistazo a git

```
> git status
```

On branch master

Your branch is up-to-date with 'origin/master'.

Untracked files:

(use "git add <file>..." to include in what will be committed)

```
src/AppBundle/Controller/PublicController.php  
src/AppBundle/Resources/  
src/AppBundle/Tests/
```

nothing added to commit but untracked files present (use "git add" to track)

NOTA: Symfony genera los templates dentro del Bundle. Esto es lo que hay que hacer en caso de que el Bundle vaya a ser distribuido posteriormente, pero durante este curso nosotros pondremos los recursos en el directorio app.

Ejercicio: Crear templates de home y de login con bootstrap. Poner a cada uno.

2.1 Más Twig

EXPLICAR TWIG (Herencia de plantillas)

Ejemplo: Crear un layout para la parte pública

Ejercicio herencia de plantillas: Crear un layout común para home y para login.

Ejercicio include de plantillas: Sacar la cabecera a una template separada.

2.2 ROUTING

Ejercicio: crear controlador Routing

EXPLICAR rutas simples con nombre.

EXPLICAR rutas con parámetros

Ejercicio: crear ruta /user/edit/{id} y probar diferentes urls

<http://localhost:8000/user/edit/3>

<http://localhost:8000/user/edit/78>

<http://localhost:8000/user/edit>

<http://localhost:8000/user/edit/patata>

EXPLICAR MÁS rutas con parámetros

Ejemplo ruta con POST: Poner una ruta con POST y ver que no funciona si se pide con GET.

Ejemplo prioridad de rutas: Crear rutas, todas por GET

[/user/{id}](#)

[/user/create](#)

[/user/edit/{id}](#)

[/user/delete](#)

y comprobar que es imposible acceder al controlador asociado a /user/create.

2.3 Más Twig

EXPLICAR generación de urls de twig.

Ejercicio urls en twig: Utilizar path() para navegar entre las páginas de home y de login.

Ejercicio raw: Coger un hola mundo y cambiar \$nombre = "Carlos";

2.4 Configuración y entornos de ejecución

Ejercicio: crear nuevo entorno pre (pre-producción)

2.5 Caché

Ejercicio: Ejecutar symfony en producción, cambiar traducción y twig, comprobar que no se muestran los cambios.