**Symfony**

Symfony es un framework diseñado para desarrollar aplicaciones web basado en el patrón Modelo Vista Controlador.

Proporciona herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Está desarrollado en PHP. Es compatible con la mayoría de gestores de bases de datos (MySQL, PostgreSQL, Oracle y Microsoft SQL Server). Se puede ejecutar tanto en plataformas \*nix (Unix, Linux, etc.) como en plataformas Windows.

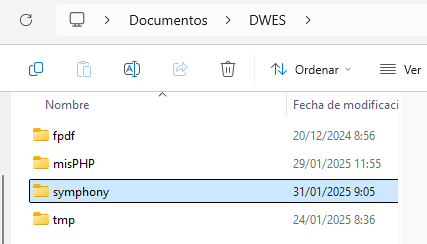
**Características de Symfony:**

* Fácil de instalar y configurar en Windows y Linux.
* Independiente del SGBD. Su capa de abstracción y el uso de ORM (Doctrine), permiten cambiar con facilidad de SGBD en cualquier fase del proyecto.
* Utiliza programación orientada a objetos y los espacios de nombres: <https://desarrolloweb.com/articulos/uso-namespaces-php.html>
* Sencillo de usar en la mayoría de los casos, aunque es preferible para el desarrollo de grandes aplicaciones web que para pequeños proyectos.
* Aunque utiliza Modelo Vista Controlador (MVC), tiene su propia forma de trabajo en este punto, con variantes del MVC clásico como la capa de abstracción de base de datos.
* Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.
* Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
* Preparado para aplicaciones empresariales y adaptable, además de estable para desarrollar aplicaciones a largo plazo.
* Código fácil de leer que incluye comentarios que permite un mantenimiento sencillo.
* Fácil de extender, lo que permite su integración con las bibliotecas de otros fabricantes.
* Una línea de comandos que facilita la generación de código.

**Página Symfony:**

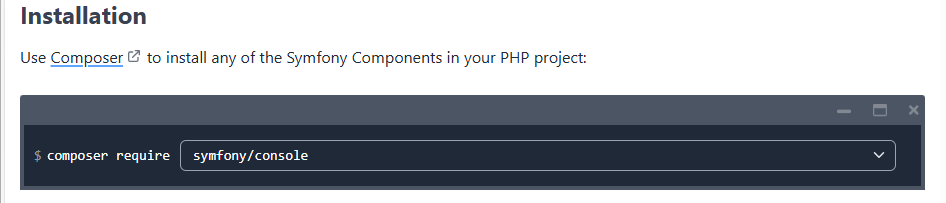
<https://symfony.com/>

**Documentación: qué podemos encontrar**

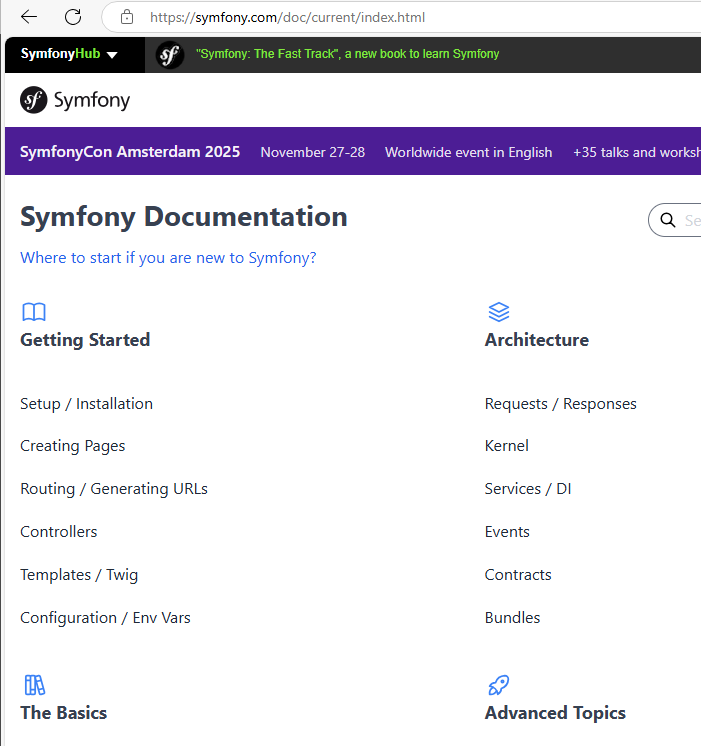


Creamos una carpeta llamada symphony

Lista de comandos para instalar componentes



Documentación general



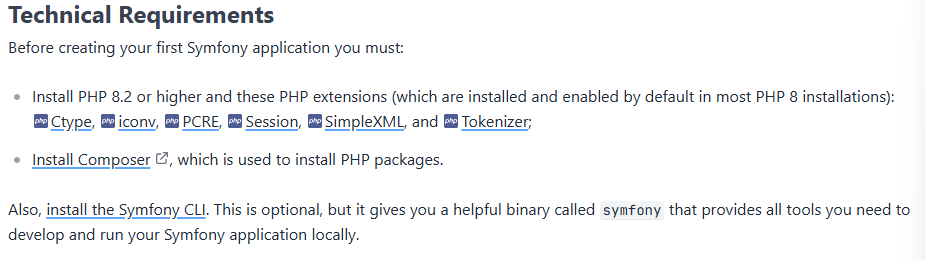
**Explica para qué se usa un namespaces e incluye un ejemplo**

Básicamente sirven de contenedores para el código de PHP, de modo que cuando creemos elementos del lenguaje como constantes, funciones o clases, se queden en un ámbito más restringido, evitando colisiones o conflictos de nombres con otros elementos que puedas crear tú más adelante, otras personas de tu equipo o incluso otros desarrolladores.

En resumen: Organizar mejor el código y poder repetir el nombre de clases/funciones.

**Comprobar requisitos en el equipo**

symfony check:requirements



php -v

composer -v

**Instalación: pasos y explicación de la instalación**

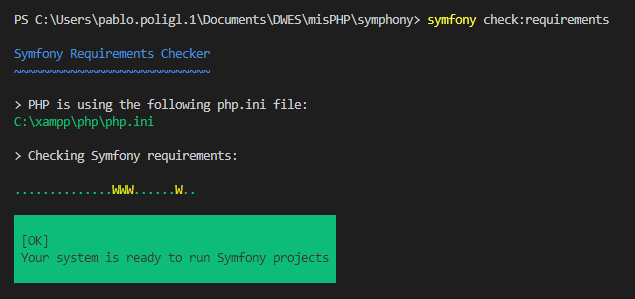


(Si no hay scoop instalarlo):

**Set-ExecutionPolicy** -ExecutionPolicy RemoteSigned -Scope CurrentUser

**Invoke-RestMethod** -Uri https://get.scoop.sh | **Invoke-Expression**

**Comprobar instalación**



**Crear un proyecto (4 formas): symfony/composer skeleton/web**

1. Symfony
   1. API (skeleton) - Proyecto 1
   2. Webapp – Proyecto 2
2. Composer
   1. API (skeleton) - Proyecto 3
   2. Webapp – Proyecto 4

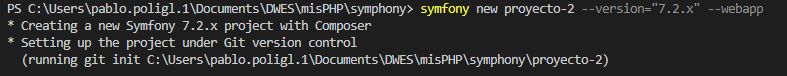
1A

symfony new my\_proyecto-1 --version="7.2.x"



1B

symfony new my\_project\_directory --version="7.2.x" --webapp



2A

composer create-project symfony/skeleton:"7.2.x" my\_project\_directory



2B

composer create-project symfony/skeleton:"7.2.x" my\_project\_directory

$ cd my\_project\_directory

$ composer require webapp



**Comandos disponibles**

php bin/console list

...

Available commands:

about Display information about the current project

completion Dump the shell completion script

help Display help for a command

list List commands

assets

assets:install Install bundle's web assets under a public directory

cache

cache:clear Clear the cache

**Estructura de directorios de un Proyecto Symfony:**

