



# ED1.Frameworks

Ingeniería Tecnologías de la Información

8°A

INTEGRANTES:

- Mendieta Chimal Sony Luis	MCS0220598
- Ascencio Onofre Carlos Gerardo	AOC0220155
- Nava Sanchez Axel	NSA0220388

Docente: LORENZO ANTONIO CARDOSO CONTRERAS

Materia: Tecnologías y aplicaciones en Internet

25/02/2025

# Índice

Índice	1
Introducción	2
Requerimientos de Software para su Instalación	2
Pasos para el Desarrollo de una Aplicación Web	3
Funcionamiento del Framework	4
Características del framework	5
Bibliografía	6

# Introducción

Este documento tiene como objetivo proporcionar una guía sobre el framework Symfony, utilizado para el desarrollo de aplicaciones web robustas y escalables. Se detallarán los requisitos de instalación, pasos para crear una aplicación, funcionamiento interno, características clave, bibliotecas y beneficios. Symfony, escrito en PHP, sigue el patrón MVC (Modelo-Vista-Controlador) y se destaca por su modularidad, rendimiento y amplia comunidad de soporte.

## Requerimientos de Software para su Instalación

Para instalar Symfony 6.x (última versión estable), se necesita:

- PHP 8.1 o superior (con extensiones: PHP-XML, PHP-MBSTRING, PHP-CURL, PHP-Tokenizer, PHP-Intl).
- Composer (gestor de dependencias).
- Base de datos: MySQL, PostgreSQL, SQLite, o MariaDB.
- Servidor web: Apache, Nginx, o el servidor integrado de Symfony (symfony server:start).

# Pasos para el Desarrollo de una Aplicación Web

1. Crear en xampp/htdocs la carpeta donde se instalará symfony
  - a. **C:\xampp\htdocs\symfony\my\_proyecto**
2. Instalación de Symfony:
  - a. **composer create-project symfony/skeleton my\_proyecto**
3. Acceder al directorio del proyecto:
  - a. **cd my\_proyecto**
4. Instalar composers
  - a. **composer require symfony/orm-pack**
  - b. **composer require symfony/maker-bundle --dev**
  - c. **composer require form validator twig-bundle security-csrf**
5. Configuración de la Base de Datos:
  - a. Abre el archivo **.env**
  - b. Modifica la variable **DATABASE\_URL** por ejemplo:
    - i. **DATABASE\_URL="mysql://root:@127.0.0.1:3306/prueba?serverVersion=8.0"**
6. Crear la base de datos:
  - a. **php bin/console doctrine:database:create**
7. Crear una Entidad:
  - a. **php bin/console make:entity Producto**
  - b. Agregar New property name:
    - i. nombre
  - c. Field type
    - i. string
  - d. Field length
    - i. 255, no
8. Generar la migración:
  - a. **php bin/console make:migration**
9. Ejecutar la migración:
  - a. **php bin/console doctrine:migrations:migrate**
  - b. Después poner "yes"
10. Generar un CRUD:
  - a. **php bin/console make:crud Producto**
  - b. Choose a name for your controller class:
    - i. ProductoController, no
11. Abrir en el navegador
  - a. **http://localhost/symfony/my\_proyecto/public/**

# Funcionamiento del Framework

Symfony opera bajo el patrón MVC:

- **Modelo:** Gestiona datos con Doctrine ORM.
- **Vista:** Renderiza contenido dinámico usando Twig.
- **Controlador:** Maneja la lógica de negocio y respuestas HTTP.

Características Clave:

- **Sistema de Rutas:** Asocia URLs a controladores.
- **Doctrine ORM:** Abstracción de la base de datos.
- **Servicios e inyección de Dependencias:** Modularidad y reutilización de código.
- **Consola de Comandos:** Automatiza tareas (ej: php bin/console).

Librerías Principales:

- **HttpFoundation:** Manejo de peticiones/respuestas HTTP.
- **Form:** Creación y validación de formularios.
- **Security:** Autenticación y autorización.
- **Mailer:** Envío de correos electrónicos.

Beneficios:

- **Escalabilidad:** Ideal para proyectos pequeños y enterprise.
- **Flexibilidad:** Componentes desacoplados (usables individualmente).
- **Comunidad Activa:** Más de 3,000 paquetes en Symfony Packagist.
- **Documentación Completa:** Tutoriales, best practices y ejemplos.

Métodos y Recursos:

- **MakerBundle:** Genera código automáticamente (entidades, CRUD).
- **Webpack Encore:** Integración con JavaScript y CSS modernos.
- **SymfonyCasts:** Plataforma de aprendizaje con videos y ejercicios.

# Características del framework

## 1. Productividad y Desarrollo Ágil

- **Uso del ORM Doctrine:** Facilita la interacción con bases de datos mediante un sistema de mapeo objeto-relacional (ORM).
- **Twig como Motor de Plantillas:** Proporciona una sintaxis clara y eficiente para la generación de vistas en las aplicaciones.
- **Symfony Console:** Una potente herramienta de línea de comandos que permite ejecutar tareas automatizadas como la generación de código y la gestión de base de datos.
- **Symfony Flex:** Un sistema que facilita la instalación y configuración de paquetes dentro del framework.

## 2. Seguridad y Gestión de Usuarios

- **Sistema de Autenticación y Autorización:** Proporciona control de accesos seguro mediante múltiples métodos como OAuth, JWT y autenticación tradicional.
- **Protección contra ataques comunes:** Incluye medidas de seguridad contra ataques CSRF, XSS e inyecciones SQL.
- **Gestión de Roles y Permisos:** Permite definir niveles de acceso en la aplicación de forma flexible.

## 3. Flexibilidad y Escalabilidad

- **Compatible con Microservicios:** Se puede utilizar tanto para aplicaciones monolíticas como para arquitecturas basadas en microservicios.
- **Integración con APIs:** Facilita la creación de APIs REST y GraphQL de manera eficiente.
- **Alta Escalabilidad:** Soporta caché, optimización de rendimiento y balanceo de carga para manejar aplicaciones de gran volumen de tráfico.

## 4. Soporte y Comunidad

- **Documentación Completa:** Proporciona guías detalladas y ejemplos prácticos para facilitar el desarrollo.
- **Gran Comunidad Activa:** Cuenta con foros, contribuyentes y soporte constante para la evolución del framework.
- **Soporte a Largo Plazo (LTS):** Algunas versiones reciben soporte extendido, lo que garantiza estabilidad en proyectos empresariales.

# Bibliografía

Symfony. (s/f). Symfony documentation. Symfony.com. Recuperado el 26 de febrero de 2025, de <https://symfony.com/doc>

PHP: The right way. (s/f). PHP: The Right Way. Recuperado el 26 de febrero de 2025, de <https://phptherightway.com/>

Composer. (s/f). Getcomposer.org. Recuperado el 26 de febrero de 2025, de <https://getcomposer.org/>

Symfony. (s/f-b). Twig - The flexible, fast, and secure PHP template engine. Symfony. Recuperado el 26 de febrero de 2025, de <https://twig.symfony.com/>

Doctrine. (s/f). Doctrine-project.org. Recuperado el 26 de febrero de 2025, de <https://www.doctrine-project.org/>