

# Introducción a Yii 2

Ricardo Pérez López

IES Doñana, curso 2018-19

1. Frameworks, microframeworks y librerías
2. Patrón *Modelo-Vista-Controlador* (MVC)
3. Yii 2
4. Instalación, requisitos y puesta en marcha
5. ¡Hola, mundo!
6. Formularios
7. Bases de datos
8. Generador de código *Gii*
9. Estilo del código

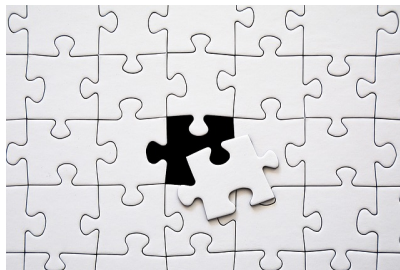
# 1. Frameworks, microframeworks y librerías

Un *framework* proporciona una forma estándar de programar y desplegar aplicaciones.

Un *framework* es un entorno de software universal y reutilizable que proporciona funcionalidades concretas como parte de una plataforma software más grande para facilitar el desarrollo de aplicaciones.

Los *frameworks* pueden incluir programas de soporte, compiladores, librerías de código, conjuntos de utilidades y APIs que, juntos, forman los diferentes componentes que permiten desarrollar un proyecto o sistema.

Un *framework* se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.





<https://symfony.com>

### Ventajas

- El más popular, junto con Laravel
- Maduro y estable
- Especialmente indicado para proyectos grandes
- Mucha documentación
- Muy demandado

### Inconvenientes

- Curva de aprendizaje algo pronunciada
- Con proyectos pequeños cuesta arrancar
- Su rendimiento no es especialmente alto





<https://laravel.com>

### **Ventajas**

- Está de moda
- Gran comunidad de usuarios
- Muy demandado

### **Inconvenientes**

- Curva de aprendizaje pronunciada
- Su rendimiento es bastante pobre
- Actualizaciones complejas



<https://cakephp.org>

### Ventajas

- Maduro y moderno al mismo tiempo
- Muchos elementos de serie
- Bien diseñado
- No es especialmente difícil de aprender
- Interesante equilibrio entre potencia, comunidad y facilidad de uso

### Inconvenientes

- No es tan utilizado como otros *frameworks* (aunque tampoco es de los menos)
- Su rendimiento no es su principal baza (aunque tampoco es de los peores)



<https://codeigniter.com>

### Ventajas

- Rendimiento espectacular
- Simple y sencillo
- Fácil de aprender
- Buena comunidad de entusiastas usuarios

### Inconvenientes

- *Demasiado* sencillo (le faltan muchos componentes que ya vienen de serie en otros *frameworks*)
- No usa Composer ni es miembro del PHP-FIG.



<https://www.yiiframework.com>

### Ventajas

- Excelente rendimiento
- Fácil de aprender
- Muy completo de serie
- Generador de código integrado incluyendo CRUD con Bootstrap
- Moderno en el desarrollo y uso de las últimas técnicas

### Inconvenientes

- No tan conocido ni usado como otros *frameworks*
- Comunidad de usuarios no muy extensa en comparación con otros

Un *microframework* carece de muchas funcionalidades propias de un *framework* web completo, como por ejemplo:

- Cuentas de usuario, autenticación, autorización, roles, etc.
- Abstracción de bases de datos mediante mapeado objeto-relacional.
- Validación y saneado de la entrada.
- Motores de plantillas web.

Normalmente, un *microframework* se encarga de:

1. Recibir una petición HTTP,
2. Encaminar dicha petición al controlador adecuado,
3. Despachar el controlador y
4. Devolver una respuesta HTTP.

A menudo se diseñan específicamente para crear las API de otro servicio o aplicación.

*By default, Flask does not include a database abstraction layer, form validation or anything else where different libraries already exist that can handle that. Instead, Flask supports extensions to add such functionality to your application as if it was implemented in Flask itself. Numerous extensions provide database integration, form validation, upload handling, various open authentication technologies, and more. Flask may be “micro”, but it’s ready for production use on a variety of needs.*

La diferencia más importante es que **el control se invierte**:

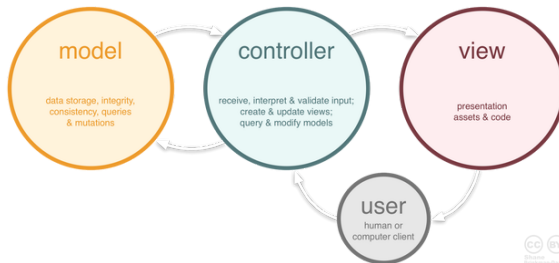
**Con una librería:** El código de la aplicación *llama* a la librería, por lo que **el control lo tiene la aplicación**.

**Con un framework:** El *framework llama* al código de la aplicación, por lo que **el control lo tiene el framework**.

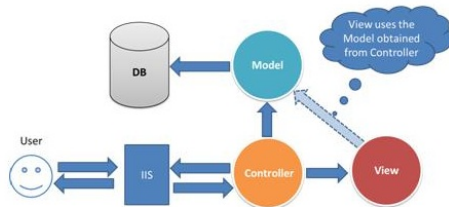


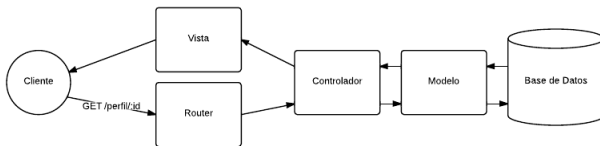
6. Investiga otros *frameworks* web para PHP.
7. Investiga otros *frameworks* web para otros lenguajes como Ruby o Python.
8. Investiga otros *microframeworks* para PHP.

## 2. Patrón *Modelo-Vista-Controlador* (MVC)



## MVC Architecture





Por ejemplo, en el caso de una red social, la capa de modelo se haría cargo de tareas tales como guardar datos del usuario, el almacenamiento de asociaciones con amigos, el almacenamiento y la recuperación de fotos de los usuarios, encontrar sugerencias de nuevos amigos, etc.

Por tanto, en este caso, los modelos podrían ser **Amigo**, **Usuario**, **Comentario** y **Foto**.

## 3. Yii 2

## 4. Instalación, requisitos y puesta en marcha



## 5. ¡Hola, mundo!

## 6. Formularios

## 7. Bases de datos

## 8. Generador de código *Gii*

## 9. Estilo del código