

Frameworks web



¿Qué son?

La creación de aplicaciones software completas y funcionales implica el desarrollo de una serie de **funcionalidades base comunes**: enrutado, gestión de modelos/entidades de base de datos, envío de emails, serialización (xml, json), gestión de formularios y validaciones, gestión de logs, testing, etc.

En lugar de tener que programar desde cero todas estas funcionalidades base en cada proyecto, lo que se utiliza es un **framework web** que nos proporciona un marco de trabajo, un conjunto de herramientas, librerías, clases, funciones y buenas prácticas que incluyen todas las **funcionalidades base comunes** en todos los proyectos software. Esto permite:

- Crear sitios web complejos en muy **poco tiempo**, lo que se traduce en **ahorro de costes**.
- **Reutilización de código**, no reinventamos la rueda teniendo que programar funcionalidades base, el framework nos las proporciona.
- Enfoque en la **lógica de negocio**: debido a que partimos de una base ya construida podemos enfocarnos en el desarrollo de aquellos requisitos propios de nuestro proyecto.

Principios detrás de los frameworks

De esta forma, los frameworks actúan como base para el desarrollo de software, facilitando las fases de la ingeniería de software. **3 principios básicos** que reúnen los frameworks son:

- **Don't Repeat Yourself (DRY)**: si algo ya ha sido programado y es funcional en diferentes contextos siendo así un código base robusto, ¿por qué volver a programarlo una y otra vez desde cero en cada proyecto? La idea entonces es evitar duplicar código y reutilizar el trabajo ya realizado y testeado.
- **Keep it short and simple (KISS)**: en programación es común encontrar múltiples formas de resolver un mismo problema, el principio KISS consiste en elegir aquella solución que sea más simple y con menos código.
- **Convention over configuration**: este enfoque consiste en el desarrollo de software guiado por convenciones y buenas prácticas en lugar de el uso exclusivo de configuraciones creadas por el propio programador. De esta forma, al utilizar unas convenciones comunes a lo largo de todos los proyectos estos serán más fácil de mantener en el tiempo y otros desarrolladores podrán adaptarse más rápido a un nuevo proyecto.

Principales funcionalidades de los frameworks

¿Qué funcionalidades aporta un framework?

- **Sistema de plantillas:** se trata de las herramientas necesarias para poder crear documentos web a partir de plantillas y contenido dinámico procedente de diversas fuentes, principalmente la base de datos. Se componen de varios elementos:
 - El contenido dinámico: información procedente de una consulta a base de datos, de un servicio externo, de archivos json/xml/csv, generado desde código, etc...
 - Las plantillas: plantillas web en un lenguaje de dominio específico y que combinan html y elementos estáticos con sintaxis del propio lenguaje para renderizar partes dinámicas.
 - El motor de plantillas: el sistema que renderiza el documento web final a partir de las plantillas y el contenido dinámico.
- **Seguridad:** herramientas para la autenticación, autorización, sistemas de roles y permisos, protección contra diferentes tipos de ataques, validación y sanitización de datos de entrada, etc...

Principales funcionalidades de los frameworks

¿Qué funcionalidades aporta un framework?

- **Sistemas de cacheo:** mecanismo para el almacenamiento temporal de rápido acceso para aquellos documentos web, consultas costosas, peticiones que más se repiten en la aplicación con el fin de reducir los tiempos de servicio y optimizar el uso de los recursos computacionales.
- **Acceso, configuración y mapeo de bases de datos:** herramientas para la gestión de las conexiones con todo tipo de bases de datos y apis (interfaces de métodos) para trabajar con los datos. La idea principal es trabajar con cualquier base de datos sin realizar cambios en el código.
 - Destacan los **ORM (Object Relational Mapping)** y los **ODM (Object Document Mapping)** como sistemas para el mapeo entre objetos en el código y los datos de la base de datos.
 - Otra herramienta importante son los **sistemas de migración de cambios**, cada vez que se produce un cambio en nuestros modelos (clases) se ha de tener en cuenta para que coincida con los datos de la base de datos.

Principales funcionalidades de los frameworks

¿Qué funcionalidades aporta un framework?

- **Sistemas de logging:** herramientas para la gestión de logs y trazabilidad de los eventos y llamadas que ocurren a lo largo del código de la aplicación software. Niveles: debug, info, warning, error... para poder asignar grados de importancia a los eventos ocurridos.
- **Sistemas de monitorización:** herramientas para la monitorización de los recursos utilizados, hilos, memoria ram, memoria disco, peticiones servidas vs tiempos de respuesta, rendimiento, etc.
- **Enrutado** (mapeo de URLs): sistema para interpretar las peticiones HTTP que llegan a las diferentes urls de la aplicación, así como la capacidad de proporcionar una respuesta, realizar redirecciones, etc.
- **Administración de estados:** en el caso de los frameworks de frontend es de vital importancia mantener sincronizados los modelos con los elementos de la interfaz gráfica, como es el caso de los campos de los formularios y sus respectivas validaciones en los datos.

Principales funcionalidades de los frameworks

¿Qué funcionalidades aporta un framework?

- **Sistemas de notificación:** librerías para el envío de información a través de diferentes sistemas de mensajería como por ejemplo email, slack, telegram...
- **Inversión de control (IoC):** gestión del ciclo de vida de los objetos creados a lo largo del código, así como la inyección de dependencias para satisfacer las relaciones entre las distintas clases cuando se crean objetos, etc.
- **Servicios web:** herramientas para proveer de mecanismos de interoperabilidad entre diferentes máquinas a través de internet. Siguiendo el estilo arquitectural REST, los servicios web pueden ofrecer una API en la cual acceder a recursos (datos) a través del protocolo HTTP y sus diferentes métodos en formatos como json/xml. El framework proporciona los mecanismos de serialización.
- **Testing:** herramientas para facilitar las pruebas del código desarrollado a diferentes niveles, testing unitario, testing de integración, testing de interfaz gráfica, testing de rendimiento, etc.
- **Internacionalización (i18n):** soporte para la traducción multi-idioma.

Tipos de frameworks

Existen multitud de **frameworks web** para cada lenguaje de programación, por lo que antes de trabajar con uno de ellos es de vital importancia conocer el lenguaje de programación para el que aplica.

Cada año estos frameworks evolucionan, se crean nuevos y otros desaparecen. Es por tanto, importante conocer el grado de adopción de cada framework en el ecosistema software y su popularidad y apoyo por parte de la comunidad.

Los frameworks web pueden clasificarse en distintas verticales:

- **Dominio:** web, big data, data science, mobile, testing, ...
- **Lenguaje de programación:** Java, Python, C++, PHP, Golang, Ruby, Elixir...
- **Capa:** frontend (client-centric), backend (server-centric), middleware, base de datos, etc.
- **Funcionalidad:**
 - **Full stack frameworks:** aquellos que ofrecen la gran mayoría de funcionalidades en un mismo framework.
 - **Microframeworks:** aquellos pensados para desarrollos más pequeños y combinarse con otros.

Frameworks web para Python

Existen multitud de frameworks web para cada lenguaje de programación. Cada año estos frameworks evolucionan, se crean nuevos y otros desaparecen. Es por tanto, importante conocer el grado de adopción de cada framework en el ecosistema software y su popularidad y apoyo por parte de la comunidad.

Full Stack frameworks:

- **Django**
- TurboGears
- Web2py
- Zope

Microframeworks:

- **Flask**
- Bottle
- CherryPy
- Web.py

Nota: para profundizar en todos los frameworks python existentes se recomienda la lectura de [este enlace](#). Ver también tabla de rango de [popularidad frameworks Python](#).