

# ¿Qué es Django?

Django es un framework web de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles. Desarrollado por programadores experimentados, Django se encarga de gran parte de las complicaciones del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto, tiene una comunidad próspera y activa, una gran documentación y muchas opciones de soporte gratuito y de pago.

## Completo

Django sigue la filosofía "Baterías incluidas" y provee casi todo lo que los desarrolladores quisieran que tenga "de fábrica". Porque todo lo que necesitas es parte de un único "producto", todo funciona a la perfección, sigue principios de diseño consistentes y tiene una amplia y actualizada documentación.

## Versátil

Django puede ser (y ha sido) usado para construir casi cualquier tipo de sitio web — desde sistemas manejadores de contenidos y wikis, hasta redes sociales y sitios de noticias. Puede funcionar con cualquier framework en el lado del cliente, y puede devolver contenido en casi cualquier formato (incluyendo HTML, RSS feeds, JSON, XML, etc).

Internamente, mientras ofrece opciones para casi cualquier funcionalidad que desees (distintos motores de base de datos, motores de plantillas, etc.), también puede ser extendido para usar otros componentes si es necesario.

## Seguro

Django ayuda a los desarrolladores evitar varios errores comunes de seguridad al proveer un framework que ha sido diseñado para "hacer lo correcto" para proteger el sitio web automáticamente. Por ejemplo, Django, proporciona una manera segura de administrar cuentas de usuario y contraseñas, evitando así errores comunes como colocar informaciones de sesión en cookies donde es vulnerable (en lugar de eso las cookies solo contienen una clave y los datos se almacenan en la base de datos) o se almacenan directamente las contraseñas en un hash de contraseñas.

Un hash de contraseña es un valor de longitud fija creado al enviar la contraseña a una cryptographic hash function. Django puede validar si la contraseña ingresada es correcta enviándola a través de una función hash y comparando la salida con el valor hash almacenado. Sin embargo debido a la naturaleza

"unidireccional" de la función, incluso si un valor hash almacenado se ve comprometido es difícil para un atacante resolver la contraseña original.

Django permite protección contra algunas vulnerabilidades de forma predeterminada, incluida la inyección SQL, scripts entre sitios, falsificación de solicitudes entre sitios y clickjacking (consulte Seguridad de sitios web para obtener más detalles sobre dichos ataques).

### Escalable

Django usa un componente basado en la arquitectura "shared-nothing" (cada parte de la arquitectura es independiente de las otras, y por lo tanto puede ser reemplazado o cambiado si es necesario). Teniendo en cuenta una clara separación entre las diferentes partes significa que puede escalar para aumentar el tráfico al agregar hardware en cualquier nivel: servidores de cache, servidores de bases de datos o servidores de aplicación. Algunos de los sitios más concurridos han escalado a Django para satisfacer sus demandas (por ejemplo, Instagram y Disqus, por nombrar solo dos).

### Mantenible

El código de Django está escrito usando principios y patrones de diseño para fomentar la creación de código mantenible y reutilizable. En particular, utiliza el principio No te repitas "Don't Repeat Yourself" (DRY) para que no exista una duplicación innecesaria, reduciendo la cantidad de código. Django también promueve la agrupación de la funcionalidad relacionada en "aplicaciones" reutilizables y en un nivel más bajo, agrupa código relacionado en módulos (siguiendo el patrón Model View Controller (MVC)).

### Portable

Django está escrito en Python, el cual se ejecuta en muchas plataformas. Lo que significa que no está sujeto a ninguna plataforma en particular, y puede ejecutar sus aplicaciones en muchas distribuciones de Linux, Windows y Mac OS X. Además, Django cuenta con el respaldo de muchos proveedores de alojamiento web, y que a menudo proporcionan una infraestructura específica y documentación para el alojamiento de sitios de Django.

## ¿Qué es la máquina virtual de Django?

El entorno de desarrollo es una instalación de Django en tu computadora local que puedes usar para desarrollar y probar apps Django antes de desplegarlas al entorno de producción.

Las principales herramientas que el mismo Django proporcionan son un conjunto de scripts de Python para crear y trabajar con proyectos Django, junto con un simple *servidor web de desarrollo* que puedes usar para probar de forma local (es decir en tu computadora, no en un servidor web externo) aplicaciones web Django con el explorador web de tu computadora.

Hay otras herramientas periféricas, que forman parte del entorno de desarrollo, que no cubriremos aquí. Estas incluyen cosas como un editor de textos o IDE para editar código, una herramienta de gestión del control de fuentes como Git para gestionar con seguridad las diferentes versiones de tu código. Asumimos que tienes ya un editor de textos instalado.

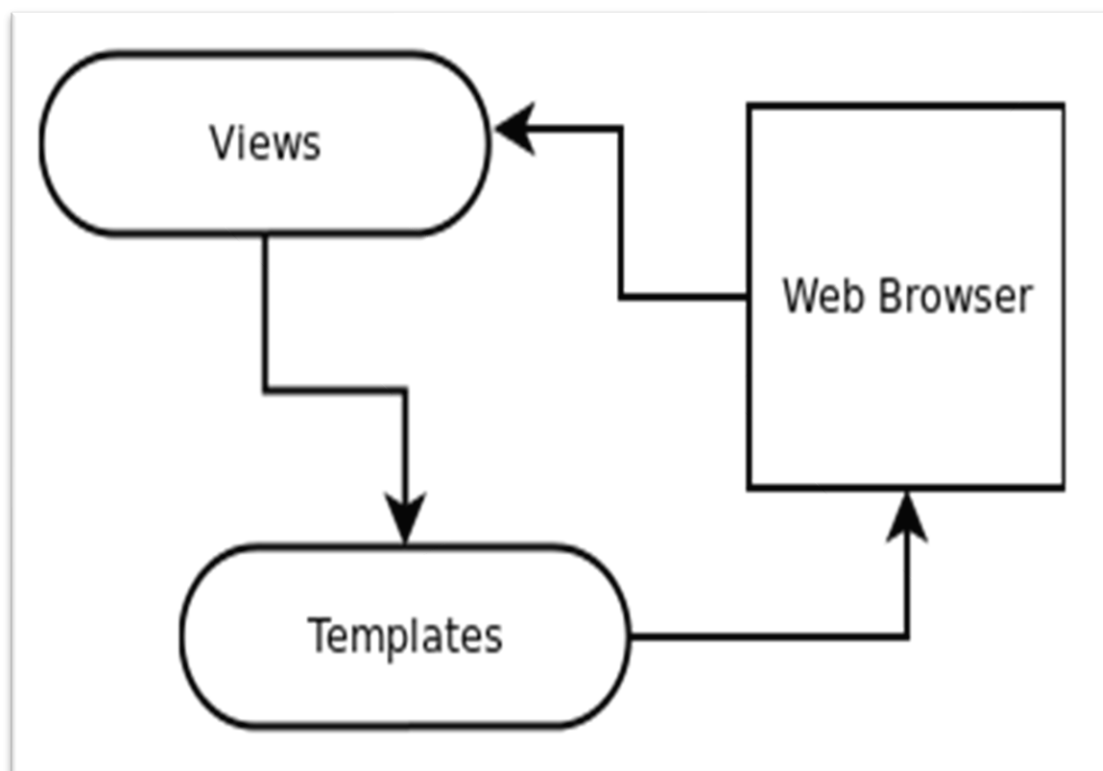
```
Directorio de C:\Users\Andrey\Documents\proyecto\Scripts
06/03/2016  04:08 p. m.      <DIR>      .
06/03/2016  04:08 p. m.      <DIR>      ..
06/03/2016  04:08 p. m.           2.203 activate
06/03/2016  04:08 p. m.           774 activate.bat
06/03/2016  04:08 p. m.          8.325 activate.ps1
06/03/2016  04:08 p. m.          1.137 activate_this.py
06/03/2016  04:08 p. m.           508 deactivate.bat
06/03/2016  04:08 p. m.         89.479 easy_install-2.7.exe
06/03/2016  04:08 p. m.         89.479 easy_install.exe
06/03/2016  04:08 p. m.         89.451 pip.exe
06/03/2016  04:08 p. m.         89.451 pip2.7.exe
06/03/2016  04:08 p. m.         89.451 pip2.exe
06/03/2016  04:08 p. m.         26.624 python.exe
06/03/2016  04:08 p. m.         27.136 pythonw.exe
06/03/2016  04:08 p. m.         89.458 wheel.exe
                13 archivos          603.476 bytes
                2 dirs  177.184.448.512 bytes libres

C:\Users\Andrey\Documents\proyecto\Scripts>activate
(proyecto) C:\Users\Andrey\Documents\proyecto\Scripts>cd..
(proyecto) C:\Users\Andrey\Documents\proyecto>
```

## ¿Qué es el modelo MVT en Django?

Si tenéis experiencia en el mundo de la programación seguro que habéis oído hablar del famoso patrón **MVC: Modelo-Vista-Controlador**. Django redefine este modelo como **MVT: Modelo-Vista-Téplate**.

Hasta ahora lo que hemos hecho no requería de interactuar con la base de datos. Podríamos decir que simplemente se recibe una petición del navegador, se ejecuta la vista correspondiente y se renderiza el Téplate para que el navegador muestre el HTML resultante:



## ¿Como Crear un proyecto con la máquina virtual?

Usando el comando `django-admin startproject` como se muestra, y luego navegaremos dentro de la carpeta con el comando.

```
django-admin startproject locallibrary.  
cd locallibrary.
```

## ¿Qué es la Carpeta Templates?

Los templates son los archivos que contienen el código de las páginas de su web site. Se trata, entre otras cosas, de determinar cómo las informaciones van a aparecer en la pantalla (incluso la posición de los placeholders).

## ¿Qué es la Carpeta Static?

Esta carpeta contiene todos los archivos de estilos: css y scss, archivos de comportamientos, imágenes incluidas dentro del theme y el archivo checkout.

## ¿Qué es un modelo Django?

Un modelo en Django es un tipo especial de objeto que se guarda en la base de datos. Una base de datos es una colección de datos. Es un lugar en el cual almacenarás la información sobre usuarios, tus entradas de blog, etc.