

Apuntes de Django

Ezequiel Remus: < *ezequielremus@gmail.com* >

Índice

1. Introducción	3
1.1. ¿Que es Django?	3
1.2. ¿Porqué usarlo?	3
2. Iniciando un proyecto	3
2.1. Entornos Virtuales	3

1. Introducción

1.1. ¿Que es Django?

Django es un framework web diseñado para realizar aplicaciones de cualquier complejidad en unos tiempos muy razonables.

Está escrito en **Python** y tiene una comunidad muy amplia, que está en continuo crecimiento

1.2. ¿Porqué usarlo?

Los motivos principales para usar Django son:

- Es muy rápido : Si tenés una startup, estas apurado por terminar un proyecto o, simplemente, querés reducir costes, con **Django** *odéis construir una aplicación muy buena en poco tiempo.*
- Viene bien cargado : Cualquier cosa que necesitéis realizar, ya estará implementada, sólo hay que adaptarla a vuestras necesidades. Ya sea porque hay módulos de la comunidad, por cualquier paquete **Python** que encontréis o las propias aplicaciones que **Django** trae, que son muy útiles.
- Es bastante seguro : Podemos estar tranquilos con **Django** ya que implementa por defecto algunas medidas de seguridad, las más clásicas, para que no haya **SQL Injection**, no haya *Cross site request forgery (CSRF)* o no haya **Clickjacking** por *JavaScript*. **Django** se encarga de manejar todo esto de una manera realmente sencilla.
- Es muy escalable : Podemos pasar desde muy poco a una aplicación enorme perfectamente, una aplicación que sea modular, que funcione rápido y sea estable.
- Es increíblemente versátil : Es cierto que en un principio **Django** comienza siendo un Framework para almacenar noticias por sitios de prensa, blogs y este estilo de webs, pero con el tiempo ha ganado tanta popularidad que se puede usar para el propósito que queráis.

*Otras bondades de **Django** que no se destacan en la web son:*

Su **ORM**, su interfaz para acceso a la base de datos , ya que hacer consultas con ella es una maravilla, es una herramienta muy buena.

Trae de serie un panel de administración, con el cual podemos dejar a personas sin ningún tipo de conocimiento técnico manejando datos importantes de una forma muy cómoda

2. Iniciando un proyecto

2.1. Entornos Virtuales

Lo primero y más importante es asegurarnos de crear un entorno para trabajar en nuestro proyecto.

Un entorno virtual es básicamente una abstracción la cual crea un conjunto vacío en **Python** , donde solo esta instalada la versión de **Python** que se utiliza junto con **pip** y las librerías básicas. En este conjunto podremos instalar todas las librerías que utilizaremos en el proyecto. Esto nos permitirá crear luego un archivo de referencia para conocer las librerías y versiones de estas utilizadas por el proyecto.

Existen varias formas de crear entornos virtuales:

■ Anaconda:

Es una distribución libre y abierta de los lenguajes **Python** y **R**, utilizada en ciencia de datos y aprendizaje automático (*Machine Learning*). Esto incluye procesamiento de grandes volúmenes de información, análisis predictivo y cómputos científicos. Esta orientado a simplificar el despliegue y administración de los paquetes de software.

Las diferentes versiones de los paquetes se administran mediante el sistema de gestión de paquetes de **conda**, el cual lo hace bastante sencillo de instalar, correr y actualizar software de ciencia de datos y machine learning como ser *Scikit-team*, *Tensorflow* y *Scipy*.

La distribución Anaconda incluye más de 250 paquetes de ciencia de datos validos para **Windows**, **Linux** y **MacOs**.

Referencias: <https://docs.anaconda.com>

- **Virtualenv:** Es una herramienta para crear entornos de [Python](#) aislados, es decir entornos donde las librerías o las versiones de [Python](#) no interfieren con las carpetas que [Python](#) tiene por defecto en la máquina. Haciendo una analogía con un edificio, un entorno vendría siendo como una planta, usa ciertos recursos como el agua o la energía eléctrica (para el caso de [Python](#) usa la misma máquina) y a su vez cada planta tiene sus propios recursos, tales como los muebles, las habitaciones y demás (para el caso de python hablamos de librerías.)
- **pyenv(Linux):**