educación IT
CODE YOUR FUTURE

Python web con Django

Módulo 1

El proyecto

Preparar el entorno

Antes de comenzar, lo primero es, claro, **¡instalar Django!**
Para eso, se debe abrir una terminal y escribir:

```
python -m pip install Django
```

El material del curso no está pensado para ningún editor de código en particular, pero **Visual Studio Code** es una buena solución gratuita para desarrollar con Django y Python.

Un primer proyecto

Django distingue *proyectos* de *aplicaciones*. **Una aplicación es un paquete de Python** (recordar que un paquete es una carpeta que contiene archivos de Python) con una estructura determinada. **Un proyecto es** —además de un paquete de Python— **un conjunto de aplicaciones con una configuración común** (por ejemplo, todas comparten la(s) mismas(s) base(s) de datos).

Si fuera necesario desarrollar un sitio web con una sección de noticias y otra de compras en línea, se podría crear un proyecto con dos aplicaciones: ***news*** y ***shop***. Esta sería una buena decisión por dos razones, veamos.

En primer lugar, estas dos aplicaciones hacen cosas bien diferentes, por lo que tiene sentido que no dependan una de otra ni que se encuentren indisociables en una misma aplicación. En segundo lugar, al tener dos paquetes independientes, pueden reutilizarse en otros proyectos.

Si más tarde, apareciera la necesidad de crear un sitio web con un foro y una sección de compras en línea, solamente se necesitaría desarrollar la primera aplicación, puesto que ***shop*** podría incluirse en el nuevo proyecto. En este curso, sin embargo, los sitios web (proyectos) no tendrán más que una aplicación.

Hechas estas aclaraciones, se creará un nuevo proyecto. Primero, se abre la terminal en una carpeta y luego se escribe:

```
django-admin startproject myproject
```

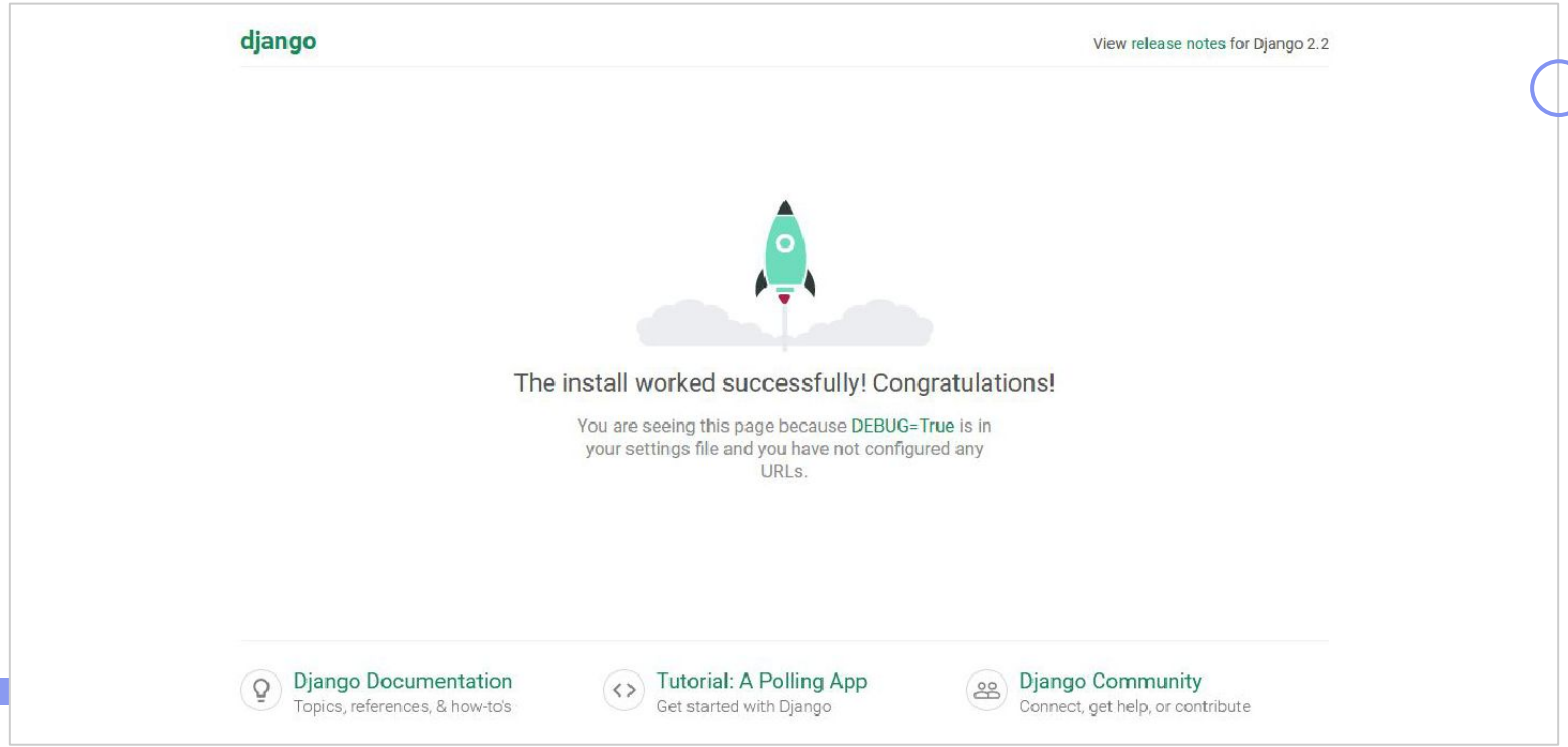
Este comando creará un nuevo proyecto (y una nueva carpeta) con el nombre ***myproject***. Si se ingresa a la carpeta recién creada, se observará que allí se encuentra otra carpeta con el mismo nombre y el archivo ***manage.py***.

A través de los archivos dentro de ***myproject/myproject*** se configurarán distintos aspectos del sitio, como las direcciones de URL y la conexión a la base de datos.

myproject/manage.py es una herramienta que permitirá ejecutar varias operaciones sobre el proyecto durante el desarrollo del sitio. Ahora bien, en la misma terminal que se abrió antes, se ingresa a la nueva carpeta **myproject** y luego se inicia el sitio de la siguiente forma:

```
cd myproject
python manage.py runserver
```

Hecho esto, se abre http://127.0.0.1:8000/ en el navegador y se debería ver algo más o menos como la imagen que vemos en la próxima pantalla.


django
View release notes for Django 2.2
The install worked successfully! Congratulations!
You are seeing this page because DEBUG=True is in your settings file and you have not configured any URLs.
Django Documentation
Topics, references, & how-to's
Tutorial: A Polling App
Get started with Django
Django Community
Connect, get help, or contribute

Si es así, ¡perfecto! Ya está corriendo el primer proyecto —vacío por el momento—de Django.

El comando **runserver** inicia un servidor web en la dirección de IP local (127.0.0.1) y el puerto 8000 para poder acceder al sitio durante la etapa de desarrollo. Este servidor web está escrito en Python y no está preparado para ser utilizado en producción (cuando se sube el sitio a un *hosting*). En la etapa de producción se suele configurar alguno de los servidores web mencionados anteriormente (Apache, NGINX, IIS, etc.) para que corra el sitio de Django.

El proyecto ya está listo. Ahora se le dará un poco de funcionalidad creando una nueva aplicación. Para eso, se deberá apagar el servidor web presionando en la terminal **CTRL + C** (Windows) o **CTRL + D** (Linux) y luego escribir:

```
python manage.py startapp myapp
```

El comando anterior creará la carpeta ***myapp*** con varios archivos en su interior. Para indicarle a Django que la aplicación ***myapp*** es parte del proyecto ***myproject***, debemos editar el archivo de configuración del proyecto ***myproject/myproject/settings.py*** y agregar nuestra aplicación así:

```
INSTALLED_APPS = [
'django.contrib.admin' ,
'django.contrib.auth' ,
'django.contrib.contenttypes' ,
'django.contrib.sessions' ,
'django.contrib.messages' ,
'django.contrib.staticfiles' ,
"myapp.apps.MyappConfig" # <-- Agregamos este elemento.
]
```

Como se puede observar, el proyecto creado vía **django-admin startproject** incluye, por defecto, algunas aplicaciones (**django.contrib.admin** , **django.contrib.auth** , etc.).

Lo importante es agregar la aplicación indicando la clase **MyappConfig** dentro del archivo **myapp/apps.py** (creada automáticamente por el comando **python manage.py startapp**).

Por último, habrá que iniciar nuevamente el servidor, que debe estar corriendo para que el sitio responda.

```
python manage.py runserver
```

Nota: El servidor web iniciado con el comando **runserver** se ocupará de recargar el código cada vez que se hacen cambios en los archivos de Python del proyecto o de la aplicación.

No obstante, en algunas ocasiones, particularmente cuando ocurren errores de sintaxis, será necesario volver a iniciarlo de manera manual escribiendo nuevamente el comando anterior.

educación IT
CODE YOUR FUTURE

¡Sigamos trabajando!