### Utilizar Git para rastrear los archivos del proyecto

Como se comentó en el Capítulo 17, Git es un programa de control de versiones que te permite tomar una instantánea del código de tu proyecto cada vez que implementas con éxito una nueva característica. Si algo va mal, puedes volver fácilmente a a la última instantánea funcional de tu proyecto; por ejemplo, si accidentalmente introduces un error mientras trabajas en una nueva función. Cada instantánea se denomina *commit*.

Utilizando Git, puedes probar a implementar nuevas funciones sin preocuparte de romper tu proyecto. Cuando despliegues en un servidor activo, debes asegurarte de que estás desplegando una versión funcional de tu proyecto. Para leer más sobre Git y el control de versiones, consulta el Apéndice D.

#### Instalar Git

Puede que Git ya esté instalado en tu sistema. Para averiguarlo, abre una nueva ventana de terminal y emite el comando git --version:

(ll\_env)learning\_log$ git --version  
git version 2.30.1 (Apple Git-130)

Si aparece un mensaje indicando que Git no está instalado, consulta las instrucciones de instalación en el Apéndice D.

#### Configurar Git

Git hace un seguimiento de quién realiza cambios en un proyecto, incluso cuando sólo una persona está trabajando en el proyecto. Para ello, Git necesita saber tu nombre de usuario y tu correo electrónico. Debes proporcionar un nombre de usuario, pero puedes inventarte un correo electrónico para tus proyectos de prácticas:

(ll\_env)learning\_log$ git config --global user.name "eric"  
(ll\_env)learning\_log$ git config --global user.email "eric@example.com"

Si olvidas este paso, Git te pedirá esta información cuando hagas tu primera confirmación.

#### Ignorar archivos

No necesitamos que Git rastree todos los archivos del proyecto, así que le diremos que ignore algunos archivos. Crea un archivo llamado *.gitignore* en la carpeta que contiene *manage.py*. Observa que este nombre de archivo comienza con un punto y no tiene extensión. Este es el código que va en *.gitignore*:

**.gitignore**

ll\_env/  
\_\_pycache\_\_/  
\*.sqlite3

Le decimos a Git que ignore todo el directorio *ll\_env*, porque podemos volver a crearlo automáticamente en cualquier momento. Tampoco rastreamos el directorio *\_\_pycache\_\_*, que contiene los archivos *.pyc* que se crean automáticamente cuando se ejecutan los archivos *.py*. No hacemos un seguimiento de los cambios en la base de datos local, porque es una mala costumbre: si alguna vez utilizas SQLite en un servidor, podrías sobrescribir accidentalmente la base de datos activa con tu base de datos de prueba local cuando envíes el proyecto al servidor. El asterisco en \*.sqlite3 indica a Git que ignore cualquier archivo que termine con la extensión *.sqlite3*.

[anterior](c20_22.html)[Subtema 23 de 37: (Ver todo)](c20.html)[siguiente](c20_24.html)