

# M1B1T3. Herramientas de Gestión del Dato

## Actividad guiada. 1

### Primeros Pasos con Git y GitHub

#### Exposición de la tarea

Has sido contratado por una multinacional como Data Engineer. Van a comenzar un nuevo proyecto y van a utilizar GitHub como sistema de control de versiones del código.

Debes crear un nuevo repositorio en GitHub, realizar el clonado a tu equipo local y realizar los pasos necesarios para trabajar con este sistema de control de versiones.

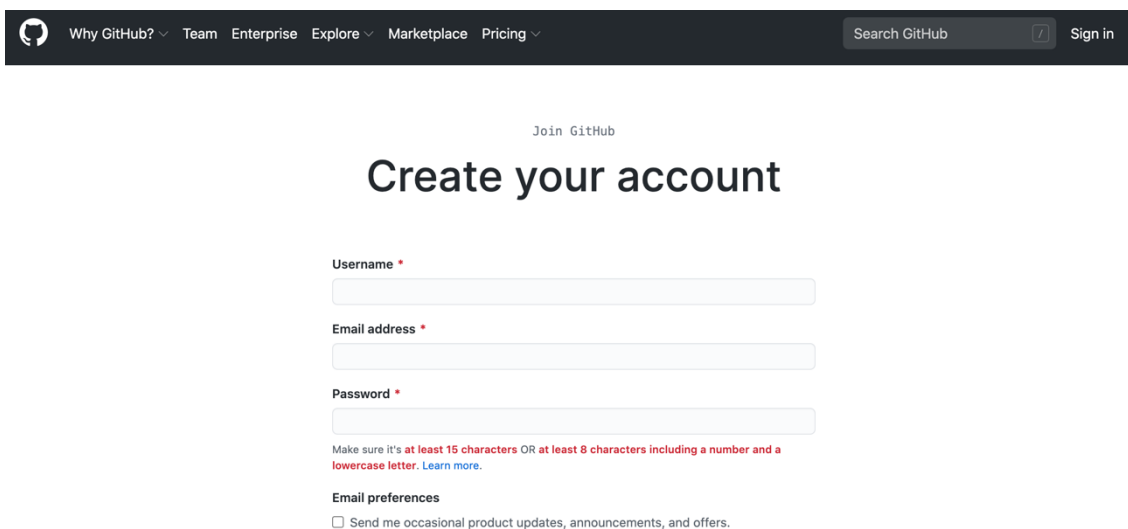
#### Objetivo

Dar los primeros pasos con Git y GitHub.

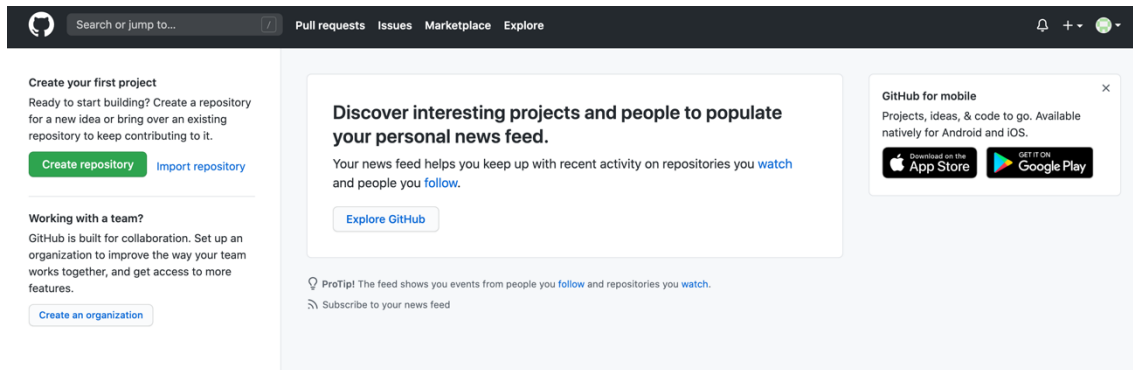
#### Pasos para la realización de la actividad

1. Lo primero que necesitas es tener una cuenta en GitHub.

Para ello, accede a [GitHub](https://github.com) y realiza el registro (botón “Sign Up”), introduciendo nombre de usuario, dirección de correo y contraseña:



2. Una vez que has accedido a GitHub con tu usuario, pulsa el botón “Create Repository” para crear tu repositorio remoto:




3. Aparecerá la siguiente ventana en la que debes introducir los siguientes datos:
- Nombre de tu repositorio (nosotros hemos elegido “test”).
  - Descripción (es opcional).
  - Si va a ser público o privado (nosotros hemos elegido privado).
  - Si quieres inicializar tu repositorio con algún fichero. Elige añadir un fichero README.


## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner \*

Repository name \*


 ignaciopereztorres ▾

/ test 


Great repository names are short and memorable. Need inspiration? How about [verbose-waffle](#)?

Description (optional)

Mi primer repositorio

☐  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☒  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☒ **Add a README file**


This is where you can write a long description for your project. [Learn more](#).

☐ **Add .gitignore**

Choose which files not to track from a list of templates. [Learn more](#).

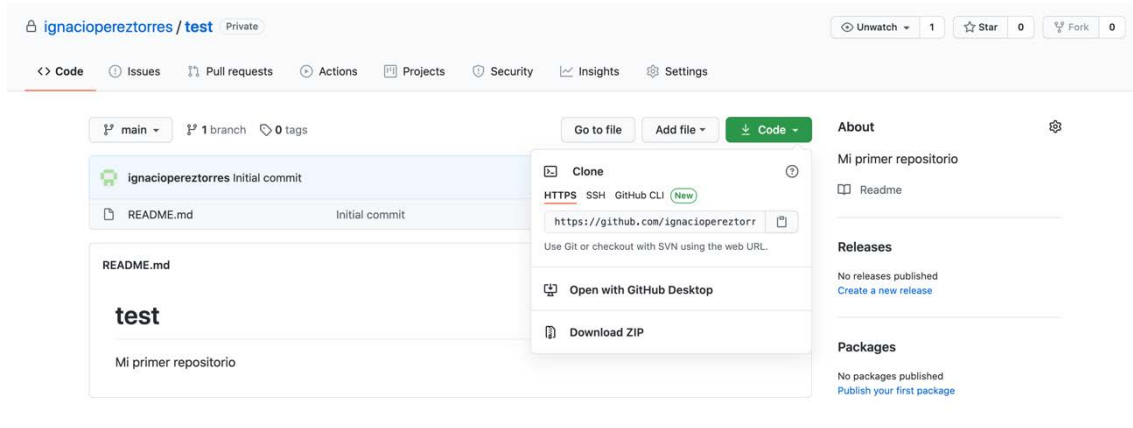
☐ **Choose a license**

A license tells others what they can and can't do with your code. [Learn more](#).

This will set  `main` as the default branch. Change the default name in your [settings](#).

Create repository

4. Una vez creado el repositorio remoto, debes clonarlo a tu equipo local. Para ello, pulsa el botón “Code” y copia la URL que aparece bajo la opción HTTPS:



5. Para trabajar con Git en tu equipo local, necesitas un cliente Git.

Existen varias opciones:

- Descargar Git y utilizar la línea de comandos.

Se puede descargar del siguiente enlace:

<https://git-scm.com/downloads>.

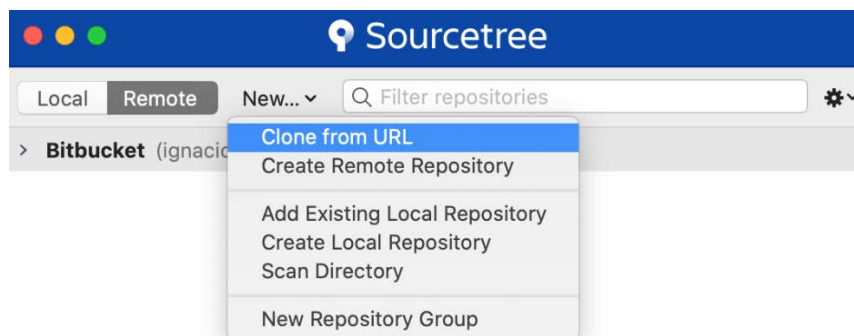
Esta opción suele ser utilizada por los desarrolladores que ya tienen experiencia en el uso de Git, ya que les resulta más rápido el trabajo usando la línea de comandos.

- Utilizar herramientas gráficas. En este caso, utiliza la herramienta Atlassian SourceTree que puedes descargar del siguiente enlace:

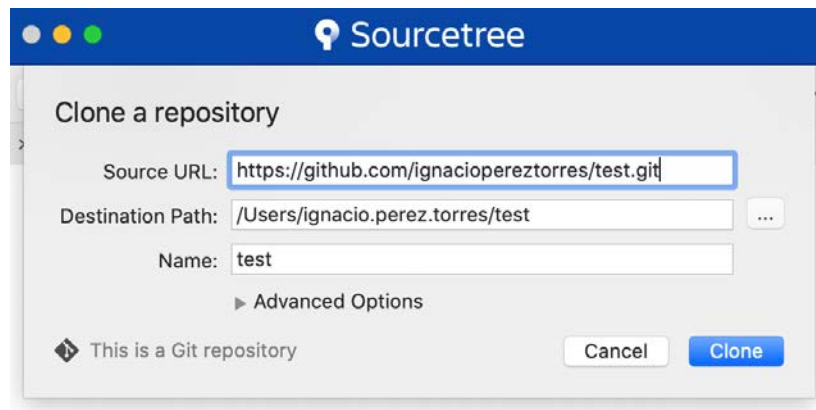
<https://www.sourcetreeapp.com/>

6. Una vez descargada e instalada, ábrela y pulsa la siguiente opción:

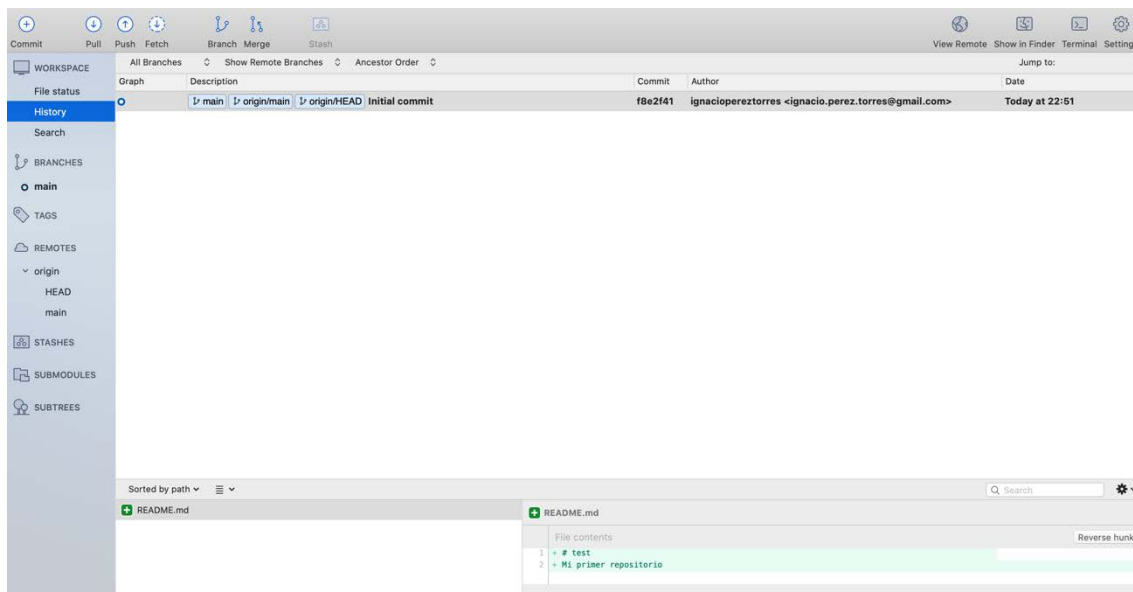
New... -> Clone from URL



- Introduce la URL obtenida en el paso 4 y elige una ruta en tu equipo local. Pulsa el botón “Clone” y, si te pide la contraseña, introduce la que utilizaste en el alta de la cuenta de GitHub.

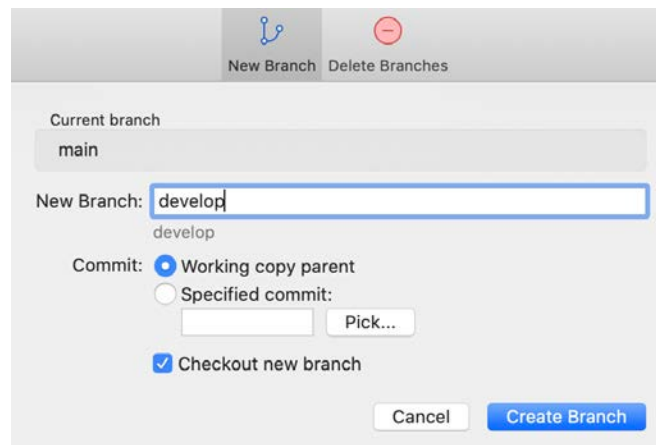


- Una vez clonado, aparecerá una ventana mostrando la rama “main” (tradicionalmente esta rama se ha llamado “master”) y tu primer “commit” que has hecho en GitHub.



- No es buena práctica trabajar sobre la rama “main”, por lo que, siguiendo las recomendaciones de una buena estrategia de ramas, debes crear una nueva rama llamada “develop”.

Para ello, pulsa botón “Branch”, escribe un nombre para la rama y pulsa el botón “Create Branch”.

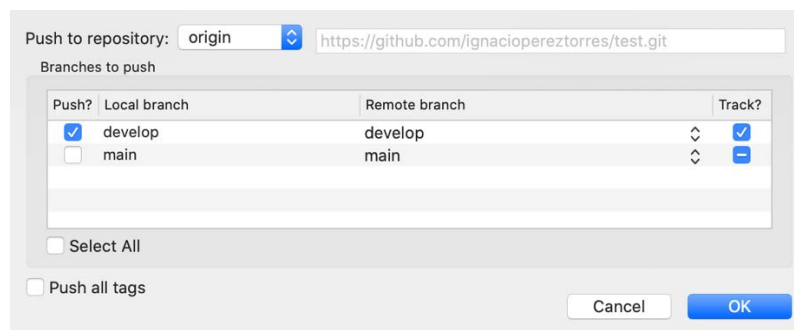
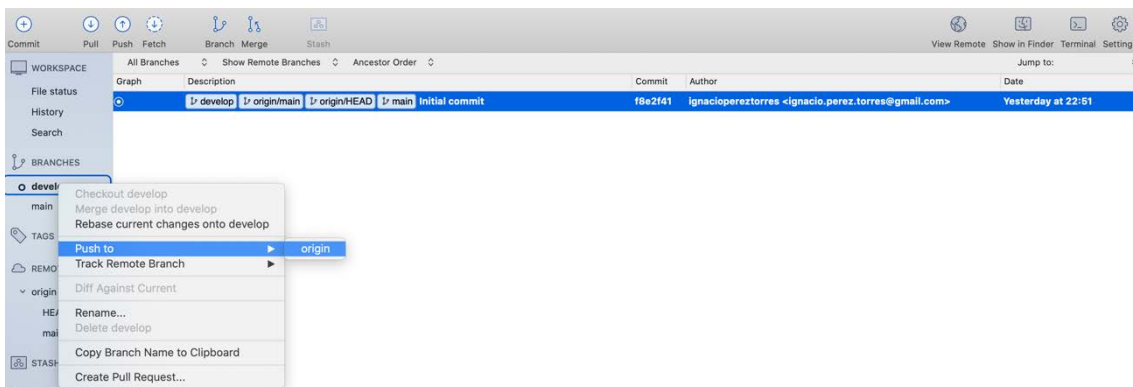


10. Esta nueva rama sólo está creada en el repositorio local, por lo que debes hacer un “Push” al repositorio remoto.

Para ello, pulsa, con el botón derecho del ratón, sobre la nueva rama:

Push to -> origin

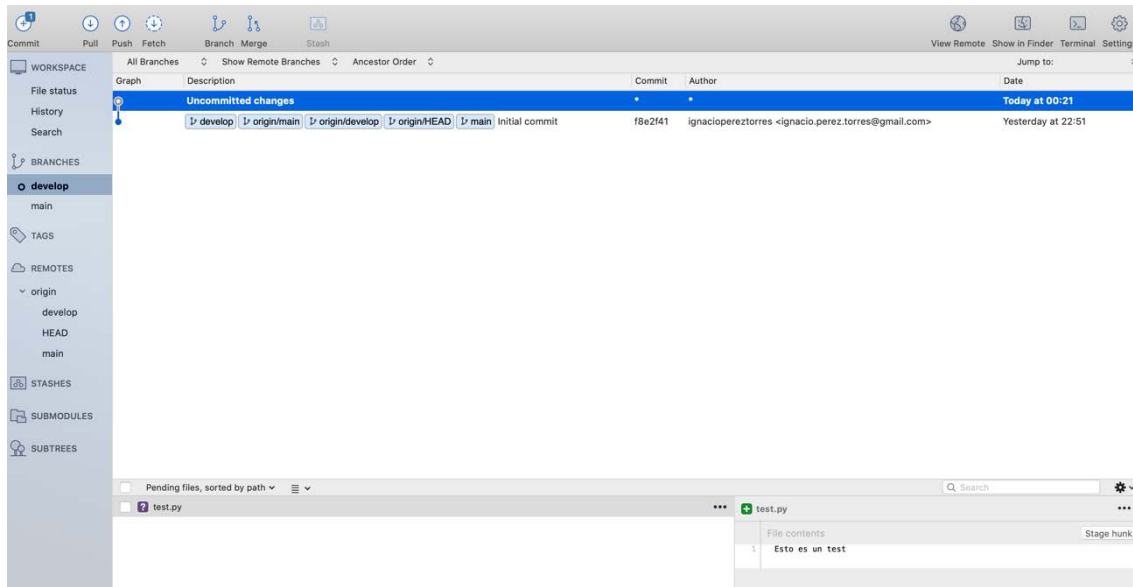
Se abrirá una nueva ventana y, con las opciones por defecto, pulsa el botón “OK”.



11. Ahora ya tienes preparado tu equipo local para trabajar con Git y GitHub, por lo que vas a empezar a crear los ficheros necesarios para codificar tu aplicación.

Vas a crear un nuevo fichero.

Para ello, ve a tu sistema de ficheros local donde has clonado tu repositorio (directorio del paso 7) y crea un nuevo fichero Python, por ejemplo test.py. Automáticamente SourceTree lo detectará y aparecerá como “Uncommitted changes”.

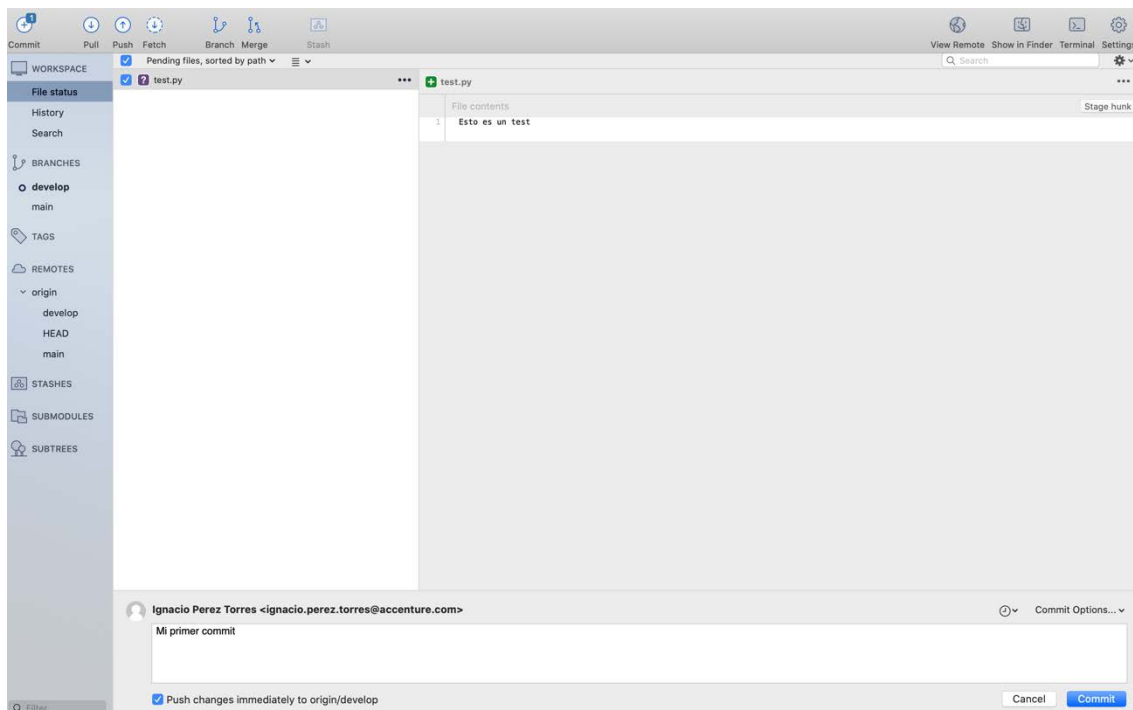


12. Cuando tengas todos los cambios realizados y quieras hacer un “commit”, selecciona los ficheros (En el apartado Pending Files, en nuestro caso sólo hay un fichero) y pulsas el botón “Commit” (Arriba a la izquierda).

Aparecerá una caja de texto donde introduciras la descripción de tu “commit” y harás check sobre “Push changes immediately to origin/develop”.

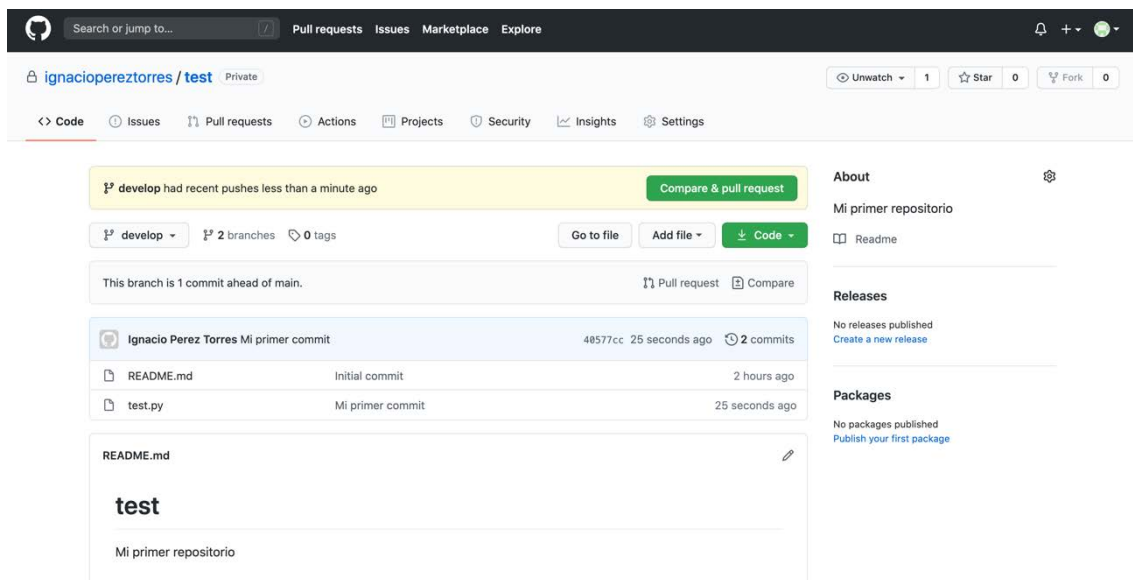
Con esta última opción, se hará Push del “commit” al repositorio remoto en GitHub.

Pulsa el botón “Commit” abajo a la derecha.



## Resultado

Si ahora vuelves a GitHub podrás ver que se ha creado la nueva rama “develop” y se ha añadido el archivo test.py.



A partir de este momento, seguiras codificando (creando y modificando ficheros) y realizando “commits” para subir tus cambios al repositorio remoto.

Podrías invitar a otros colaboradores/as para que se clonen el repositorio y ya les aparecerían las dos ramas de trabajo “main” y “develop” con los archivos que hayas subido hasta ese momento.