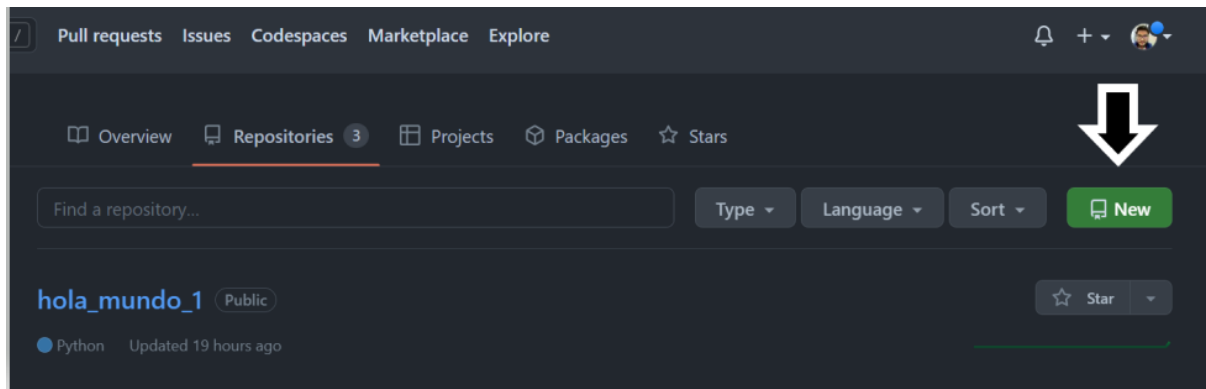


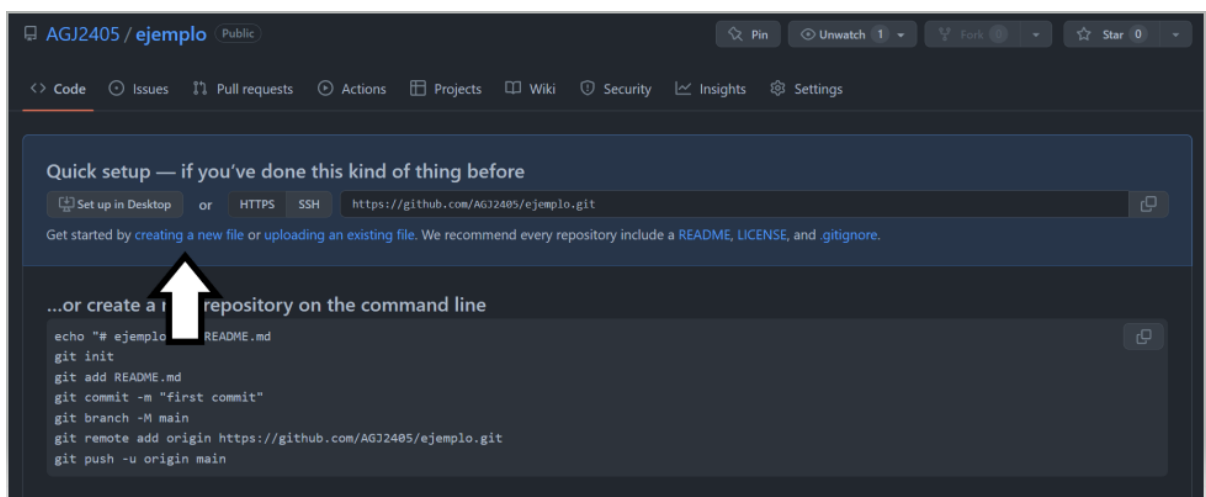
# Ejercicio

Antes de empezar el ejercicio, lean todo el documento, ya que hay configuraciones que tendrán que hacer dado los errores que salieron ayer.

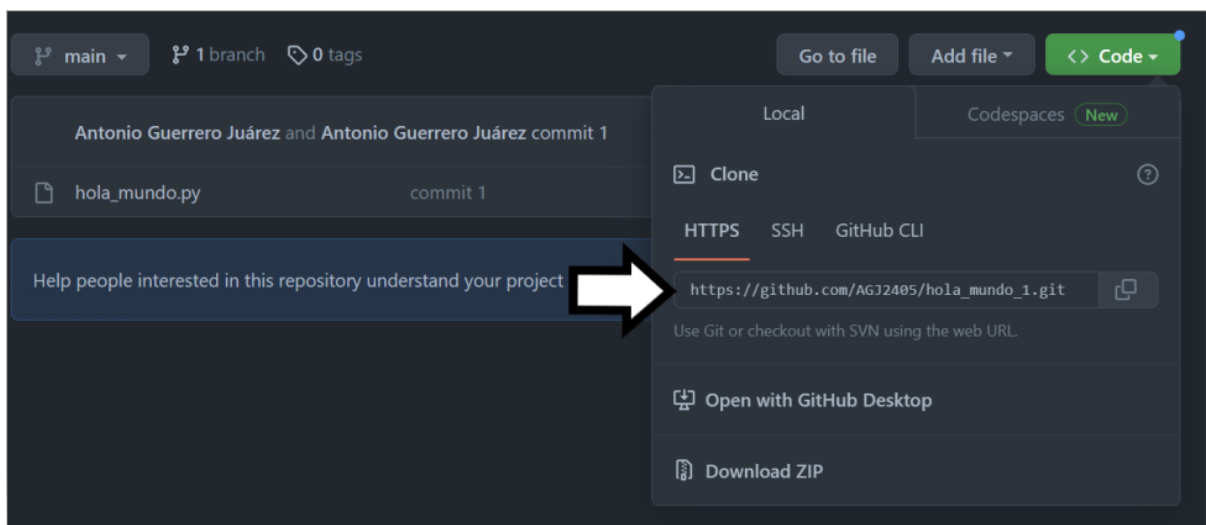
- Crear un repositorio en su cuenta de GitHub.



- Crear un archivo vacío llamado hola\_mundo.py (Esto crea la rama principal).



- Clonar el repositorio a su máquina personal por medio de la **Terminal** (en MAC) o **Git Bash** (en microsoft), para este punto debieron **crear una carpeta** llamada Git dentro de la carpeta documentos, recuerden que el comando a ejecutar (git clone) debe ser dentro de la carpeta que acaban de crear, para esto pueden usar los comandos **mkdir** (crea una carpeta), **cd** (moverse entre carpetas) y **ls** (ver el contenido de la carpeta).



**git clone** <pega la URL de tu repositorio>

- Realizar un programa que muestre un “hola mundo” en **Python** con un editor de texto, el día de ayer usamos SublimeText.
- Agrega los cambios al área de preparación  
**git add** <nombre de tu archivo>
- Validar los cambios:  
**git status**

- Haz un commit con un mensaje descriptivo:  
**git commit -m** “Se agregaron cambios”
- Validar el commit:  
**git status**
- Envía los cambios de tu rama local main a la rama principal de tu repositorio remoto, recuerden que el nombre **main** es el nombre de la rama en tu equipo personal, también puede ser **master**:

```
MINGW64 ~/Documents/Git2/hola_mundo_1 (main)
```

**git push origin** main

- Corrobores sus cambios en Github.

## Notas importantes:

En la clase de ayer nos faltó una configuración inicial, debemos establecer el nombre de usuario y dirección de correo electrónico. Esto es importante porque los "commits" de Git usan esta información, y es introducida de manera inmutable en los commits que envías, para ello ejecuta los siguientes comandos:

```
git config --global user.name "ingresa tu nombre"
git config --global user.email "correo@example.com"
```

**SOLO PARA MAC:** Para solucionar el error que se genera en MAC, sigan las siguientes instrucciones:

### Paso 1: Crear token de acceso en GitHub

En primer lugar, debe crear un token de acceso personal en GitHub. Para hacerlo, siga los pasos a continuación:

- Haga clic en **el ícono de su perfil de GitHub** en la esquina superior derecha.
- Haga clic en **Settings**.
- En el menú que se muestra a la izquierda, haga clic en **Developer Settings**.
- Haga clic en **Personal access tokens** y después seleccione **Tokens (classic)**.
- Haga clic en **Generate new token(classic)**.
- Agrega una nota que te ayude a identificar el alcance del token de acceso a generar, por ejemplo "Token para primera práctica".
- **Elija el período de caducidad** del menú desplegable, pueden elegir 7 días o sin expiración.
- Finalmente, **seleccione los ámbitos a los que desea otorgar el acceso correspondiente al token de acceso generado** . Asegúrese de seleccionar los ámbitos mínimos requeridos; de lo contrario, seguirá teniendo problemas para realizar ciertas

operaciones de Git, para esta práctica seleccionen todos los recuadros.

- Finalmente haga clic en **Generate Token**

A estas alturas, debería haber generado su token de acceso personal con éxito y el siguiente mensaje debería estar visible en su pantalla.



Make sure to copy your personal access token now. You won't be able to see it again!

Debajo de ese mensaje, también debería poder ver su token de acceso personal. **Asegúrese de copiarlo**, ya que lo necesitaremos en los siguientes pasos.

## Paso 2: cambia la URL remota

Ahora que hemos creado nuestro token de acceso personal, debemos ejecutar el siguiente comando:

```
git remote set-url origin https://<githubtoken>@github.com/<username>/<repositoryname>.git
```

En el comando anterior, asegúrese de reemplazar:

- `<githubtoken>` con el token de acceso personal que has copiado en el paso anterior.
- `<username>` con tu nombre de usuario de GitHub.
- `<repositoryname>` con el nombre de tu repositorio de GitHub.

## Paso 3: Prueba que funciona

```
git push origin master
```