

## **Pasos para realizar el fork del repositorio y luego trabajar en el mismo**

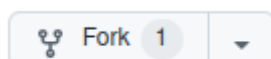
A continuación, veremos un instructivo para utilizar un repositorio de GitHub como “plantilla” en el desarrollo de nuestros trabajos durante la cursada. Con este instructivo paso a paso, crearemos un nuevo repositorio que hereda la estructura y los archivos de un repositorio existente (la plantilla), facilitando así la organización de las carpetas y archivos trabajados en cada unidad.

### **Acceso al Repositorio Plantilla**

Ingresa a GitHub y abra el repositorio que servirá como plantilla para su proyecto usando la siguiente url: <https://github.com/sbruselario/UTN-TU-PaD-P1>

### **Creación de un Nuevo Repositorio a partir del repositorio plantilla**

En la página principal del “repositorio plantilla”, haga clic en el botón "Fork" situado en la parte superior derecha de la pantalla.



### **Configuración del Nuevo Repositorio:**


- Seleccione su cuenta personal que será propietaria del nuevo repositorio.
- Ingrese un nombre representativo al nuevo repositorio (puede dejar el original [UTN-TUPaD-P1](#)).
- Puede cambiar la breve descripción que indique el propósito del repositorio.
- Solo copiaremos la rama predeterminada (main).
- Una vez configurados los parámetros anteriores, haga clic en "Crear fork" para finalizar el proceso.

## Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (\*).

Owner \*

 Guillermo Puertas ▾

Repository name \*

UTN-TUPaD-P1

✔ UTN-TUPaD-P1 is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

Mi repositorio para Programación 1.

☒ Copy the `main` branch only

Contribute back to sbruselario/UTN-TUPaD-P1 by adding your own branch. [Learn more.](#)

Create fork

¡Ya tenemos nuestro fork! ¡Ya tenemos un repositorio propio con todas las carpetas creadas!

## Clonación del Nuevo Repositorio en el Entorno Local

En la página del nuevo repositorio en GitHub (el nuevo que se encuentra en tu cuenta de GitHub), copiar la URL proporcionada para la clonación. Ahora abrir una terminal en su sistema (cmd, gnome-terminal o similar) y ejecute el siguiente comando reemplazando URL\_DEL\_REPOSITORIO por la URL copiada (la URL de su repositorio, ¡NO la plantilla!):

```
git clone URL_DEL_REPOSITORIO
```

Navegue al directorio recién clonado:

```
cd nombre_del_repositorio
```

## Desarrollo y Registro de Cambios

Realice las modificaciones y adiciones necesarias en los archivos del proyecto utilizando un editor de texto o entorno de desarrollo integrado (IDE) como Visual Studio Code.

Una vez realizadas las ediciones, agregamos los archivos modificados al área de preparación:

```
git add .
```

Documente las modificaciones efectuadas mediante un commit, incluyendo un mensaje descriptivo:

```
git commit -m "Descripción breve y representativa de los  
cambios realizados"
```

## Sincronización con GitHub

Transfiera los commits locales a GitHub ejecutando:

```
git push origin nombre_de_la_rama
```

Este paso actualizará el repositorio en línea con los cambios efectuados localmente.

## ¡Listo! Ahora, ¡sigamos programando!

Repita los pasos de *Desarrollo y Registro de Cambios* y *Sincronización con GitHub* según avance en las actividades. Es recomendable realizar commits frecuentes, con mensajes breves y descriptivos para mantener un historial claro de las modificaciones.