



Tecnicatura Universitaria  
en Programación

## PROGRAMACIÓN II

GitHub. Primeros pasos.

Material Complementario  
1° Año – 2° Cuatrimestre

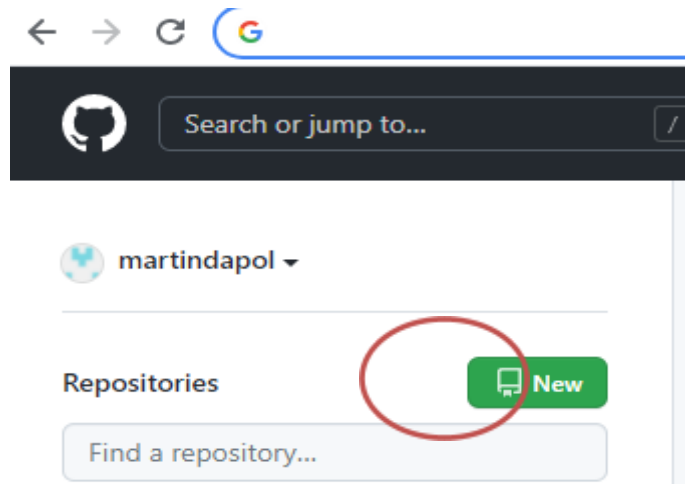


## GitHub. Primeros pasos.

El objetivo de este material es introducir los conceptos básicos para comenzar a trabajar con un repositorio de versionado de código mediante la herramienta **Git** utilizando la plataforma **GitHub** como soporte en la nube.

<https://github.com/>

1. Con las credenciales habilitadas entonces vamos a crear un repositorio vacío en la nube. Para ello nos autenticamos y una vez dentro del sitio seleccionamos la opción **New**.



Para completar la creación de nuestro repositorio ingresados los datos: nombre, public/private y opciones para generar archivos de configuración, tal como se muestra en la siguiente imagen:

Owner \* martindapol / Repository name \* ProglI ✓

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

Description (optional)  
Repositorio primeros pasos con GitHub

☒ **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.

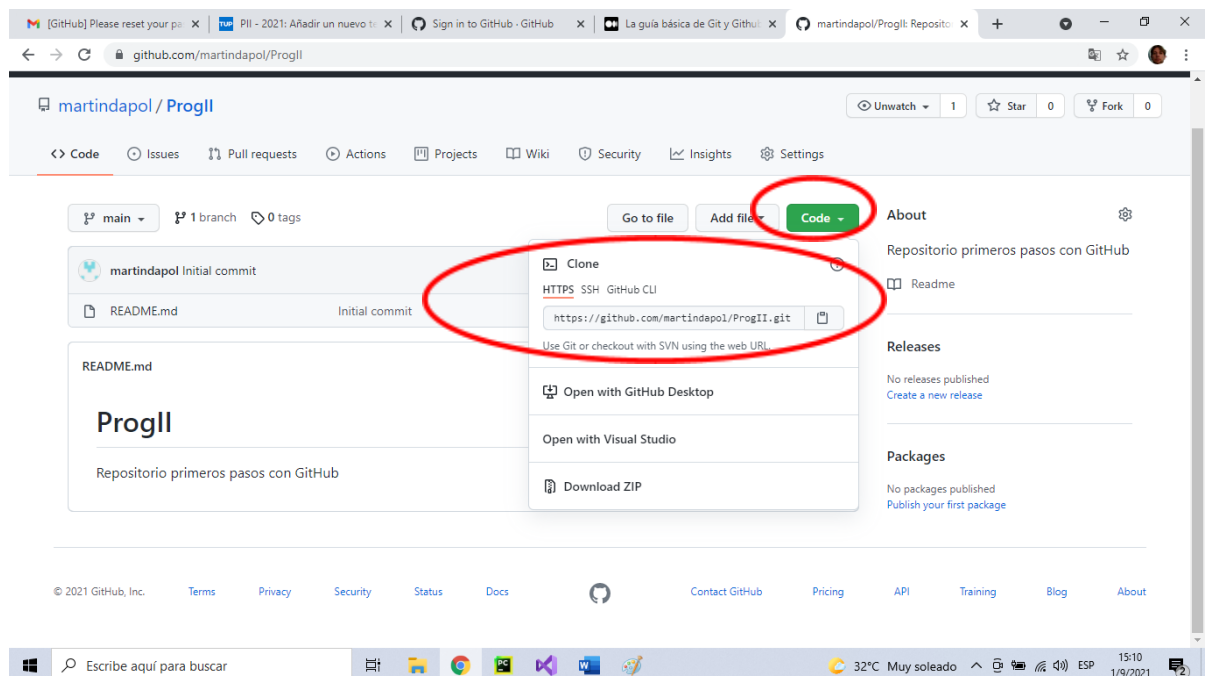
☒ **Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)

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

Permite indicar que cualquier persona a través de Internet puede acceder al repositorio

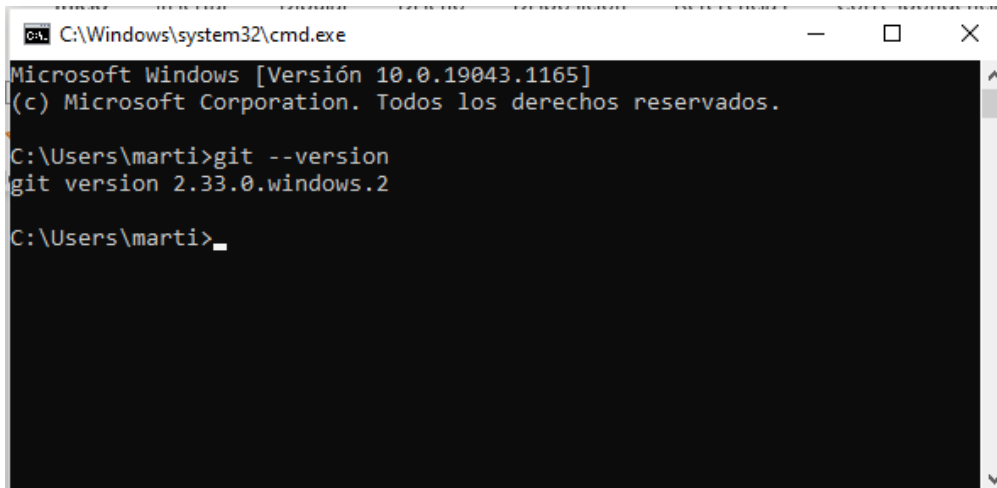
Permite crear un archivo índice con una breve descripción del repositorio.

2. Una vez creado el repositorio vacío vamos a realizar un **clone** de dicho repositorio a nuestro equipo local. Para ello seleccionamos la opción **Code** y copiamos la url de nuestro repositorio desde el botón situado a la derecha de la dirección:



3. Ahora para crear una copia local en nuestro equipo necesitamos tener instalado la herramienta **Git**, que la descargamos de: <https://git-scm.com/downloads>. Una vez descargado e instalado vamos a

corroborar que se instaló correctamente. Para ello abrimos una ventana de comando o cmd (para el caso de los usuarios Windows) y escribimos el siguiente comando:



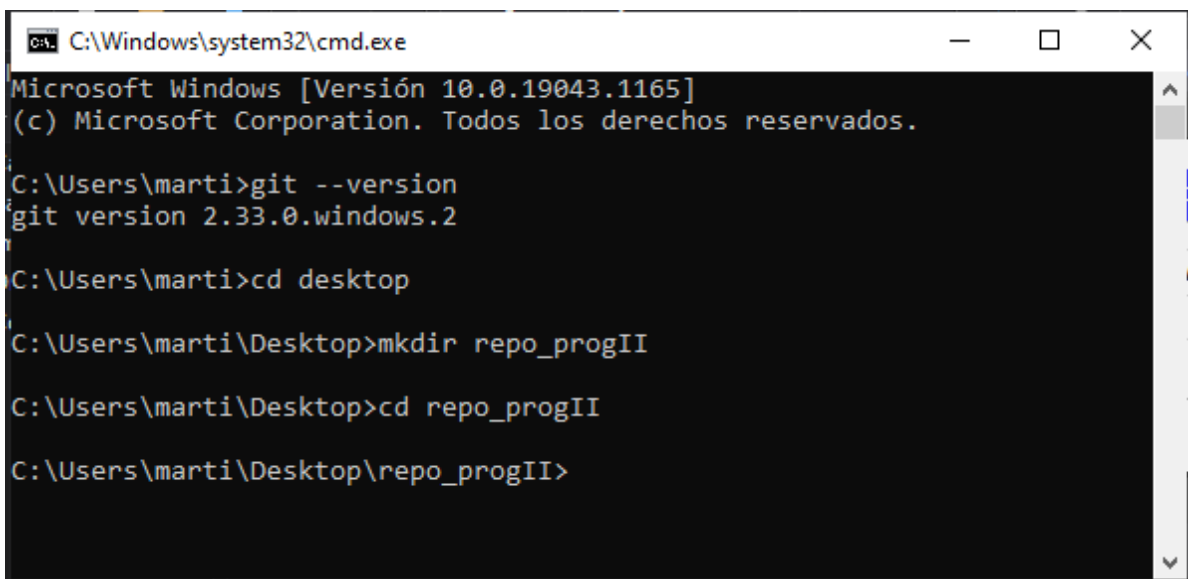
```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.19043.1165]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\marti>git --version
git version 2.33.0.windows.2

C:\Users\marti>
```

Con el comando **git --version** obtenemos la versión de Git instalada localmente. Esto nos garantiza que se instaló correctamente en nuestro equipo.

Ahora creamos una carpeta en nuestro sistema de archivos local y nos posicionamos en dicha carpeta:



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.19043.1165]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\marti>git --version
git version 2.33.0.windows.2

C:\Users\marti>cd desktop

C:\Users\marti\Desktop>mkdir repo_progII

C:\Users\marti\Desktop>cd repo_progII

C:\Users\marti\Desktop\repo_progII>
```

Puede crear la carpeta desde el entorno de ventanas o bien mediante los comandos **mkdir** y **cd**.

Parados en dicha carpeta ejecutamos el comando:

```
C:\Windows\system32\cmd.exe

C:\Users\marti>cd desktop

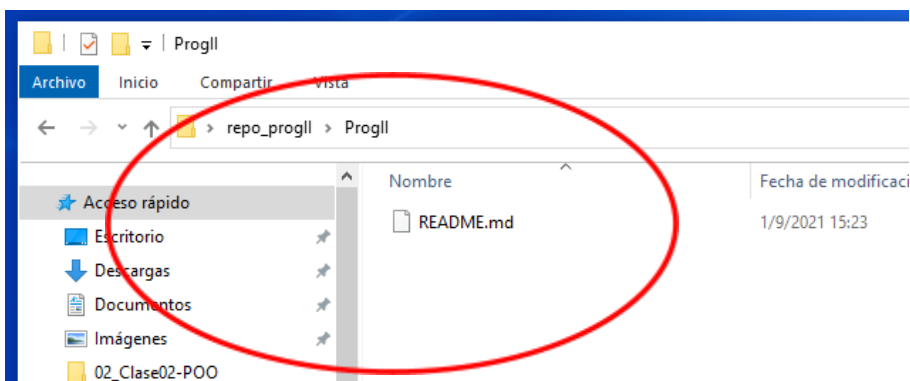
C:\Users\marti\Desktop>mkdir repo_progII

C:\Users\marti\Desktop>cd repo_progII

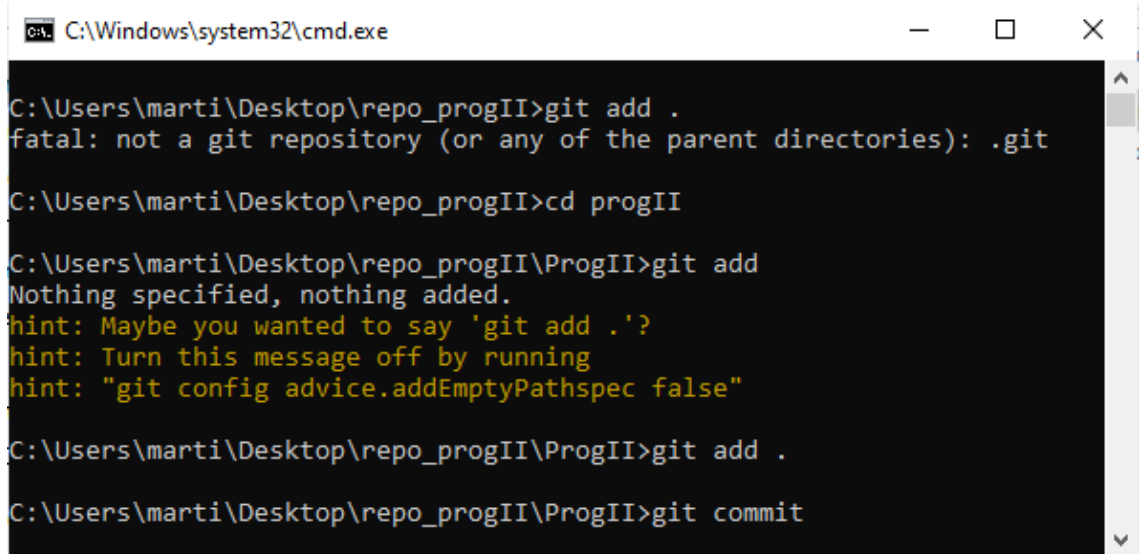
C:\Users\marti\Desktop\repo_progII>git clone https://github.com/martindapol/ProgII.git
Cloning into 'ProgII'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

C:\Users\marti\Desktop\repo_progII>
```

El comando **git clone + URL** del repositorio permite generar una copia local generando los siguientes archivos: **carpeta con el mismo nombre que el repositorio + archivo readme.**



4. Ahora vamos a copiar dentro del repo local la carpeta con el código de nuestro proyecto **Carpintería** y lo vamos a confirmar localmente mediante los comandos: **add y commit.**



```
C:\Windows\system32\cmd.exe

C:\Users\marti\Desktop\repo_progII>git add .
fatal: not a git repository (or any of the parent directories): .git

C:\Users\marti\Desktop\repo_progII>cd progII

C:\Users\marti\Desktop\repo_progII\ProgII>git add
Nothing specified, nothing added.
hint: Maybe you wanted to say 'git add .'
hint: Turn this message off by running
hint: "git config advice.addEmptyPaths false"

C:\Users\marti\Desktop\repo_progII\ProgII>git add .

C:\Users\marti\Desktop\repo_progII\ProgII>git commit
```

El comando **git add** permite agregar a nuestra carpeta de versionado uno o más archivos que serán incluidos en los próximos commits.

La sintaxis es **git add <path>** o bien **git add .** El punto (.) permite indicar que se agreguen al versionado todas las carpetas y archivos nuevos.

Luego vamos a generar nuestro primer commit mediante el siguiente comando:

**git commit -m "primer commit"**

Esto permite confirmar todos los cambios realizados en el repositorio local, que luego serán enviados al repositorio remoto (github) mediante el comando **git push**. Para ver los cambios y pendientes de confirmación podemos utilizar el comando **git status**.

5. Los últimos 2 comandos que necesitamos son **git push** y **git pull**.

Estos nos permiten sincronizar nuestro repositorio local con el remoto:

- a. **git push** (enviar cambios hacia la rama principal del repositorio remoto)
- b. **git pull** (recibir en nuestra rama principal local los cambios desde gitHub)

Estos 5 pasos constituyen la gestión básica de un versionado de archivos con Git. Todos estos comandos pueden ejecutarse de manera visual mediante un cliente GUI. Uno de los más usados es <https://desktop.github.com/>. Instalando esta herramienta luego de instalar Git, podemos ejecutar todos los comandos anteriores desde una interfaz Windows.

Para completar esta introducción se sugiere ver el siguiente video:

<https://www.youtube.com/watch?v=hWglK8nWh60>



#### Atribución-No Comercial-Sin Derivadas

Se permite descargar esta obra y compartirla, siempre y cuando no sea modificado y/o alterado su contenido, ni se comercialice. Referenciarlo de la siguiente manera:

Universidad Tecnológica Nacional Facultad Regional Córdoba (S/D). Material para la Tecnicatura Universitaria en Programación, modalidad virtual, Córdoba, Argentina.