

## Introducción

### ¿Qué es Git?

*“Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código.” Fuente Wikipedia*

En resumen, es una herramienta con la cual se puede mantener un registro de los cambios y evoluciones en el desarrollo de un software. Pudiendo así trabajar de forma colaborativa con otros desarrolladores. Todo esto alojado en uno o más servidores.

### ¿En qué vamos a utilizar GIT?

La propuesta es crear un repositorio personal para la clase y así los alumnos pueden subir los trabajos realizados en cada clase, de esta manera aprenden de forma practica el uso de la herramienta y permiten a los docentes disponer de los trabajos en una plataforma común para su evaluación y seguimiento.

## Primeros Pasos

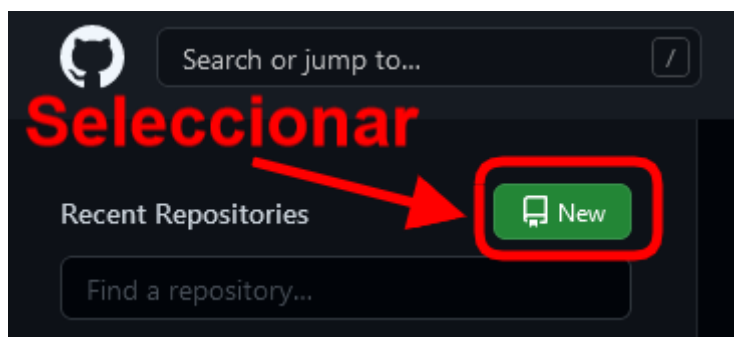
### Crear Cuenta

Existen múltiples proveedores de servidores GIT, uno de los más conocidos es GITHUB, que además permite su uso gratuito, por lo que utilizaremos este para el repositorio personal de la clase. Se debe crear una cuenta ingresado al sitio web: <https://github.com/signup>

### Crear un Repositorio

Se debe crear un nuevo repositorio en GITHUB, en las capturas que se encuentran a continuación se pueden visualizar los pasos necesarios.

- Paso N°:1



- Paso N°:2

**Create a new repository** **Nombre del Repositorio**

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

Owner \*

Repository name \* **Descripción**

UTN-TUP-SPD-Comision-Alumno ✓

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

Description (optional)

Repositorio para Sistemas de Procesamiento de Datos de la Carrera Tecnicatura Universitaria en Programación

☒ **Public** **Publico**  
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: **Seleccionar**

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](#).

**Completar**

Create repository

- Paso N°:3

**Seleccionar**

main 1 branch 0 tags

Go to file Add file **Code**

**Seleccionar**

Clone

HTTPS SSH GitHub CLI

Use a password-protected SSH key.

Open with GitHub Desktop

Download ZIP

Initial commit

README.md

UTN-TUP-SPD-Comision-Alumno

Repositorio para Sistemas de Procesamiento de Datos de la Carrera Tecnicatura Universitaria en Programación de la UTN Facultad Regional Rosario

## Instalar Herramienta

Como en el caso de los servidores también existen múltiples clientes GIT, según el sistema operativo que se esté utilizando se debe instalar de diferentes maneras:

- **Windows:** (Descargar el Instalador según si el sistema operativo es de 32 o 64 bits)
  - **GIT-SCM:** <https://git-scm.com/download/win>
- **Linux** (Se requieren Permisos de Administrador):
  - **Debian/Ubuntu:** `apt-get install git`
  - **RedHat/Centos:** `yum install git`
- **Mac** (Se requieren Permisos de Administrador):
  - **Homebrew:** `brew install git`
  - **MacPorts:** `port install git`

## Comandos Básicos

A continuación, se encuentran los comandos básicos más comunes. Todos estos comandos se deben realizar desde la terminal, según sea el sistema operativo que se esté utilizando, las opciones son:

- **Windows:**
  - CMD (No Recomendado)
  - Windows Shell
  - Git Client
- **Linux y Mac:**
  - Cualquier Terminal
  - Git Client

## Configuración Inicial

Para un uso más ordenado de los repositorios y la herramienta se debe configurar datos del usuario para identificar al mismo. Son simplemente el Nombre y el Correo Electrónico.

```
git config --global user.name "NOMBRE"
```

```
git config --global user.email "CORREO"
```

## Inicializar un Repositorio Local

Este comando permite iniciar un nuevo repositorio local, ósea sin repositorio remoto GIT.

```
git init
```

## Agregar Repositorio Remoto

El comando en cuestión permite agregar un repositorio remoto a un repositorio local.

```
git remote add origin URL-REPO-GIT
```

## Clonar Repositorio Remoto

Este comando permite crear una copia local de un repositorio remoto, el mismo evita utilizar los comandos anteriores.

```
git clone URL-REPO-GIT .
```

### Actualizar Repositorio Local

El comando en cuestión permite descargar los cambios de un repositorio remoto, que no estén aplicados en el repositorio local.

```
git pull
```

### Actualizar Repositorio Remoto

Lo siguientes comandos permite subir los cambios locales a un servidor remoto.

```
git add .
```

```
git commit -m "COMENTARIO"
```

```
git push -u origin main
```