

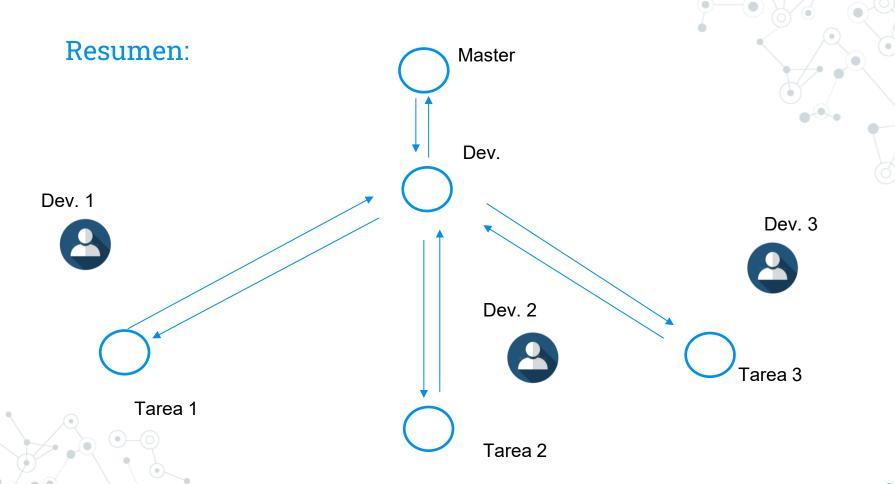


Definición:

Git es una herramienta fundamental que realiza la función del control de versiones de código de forma distribuida, de la que podemos resaltar varias características:

- Es muy potente.
- No depende de un repositorio central
- Es software libre (Open Source)
- Con ella podemos mantener un historial completo de versiones
- Podemos movernos, como si tuviéramos un puntero en el tiempo, por todas las revisiones de código y desplazarnos una manera muy ágil.
- Tiene un sistema de trabajo con <u>ramas</u> que lo hace especialmente potente
- En cuanto a la funcionalidad de las ramas, las mismas están destinadas a provocar proyectos divergentes de un proyecto principal, para probar nuevas funcionalidades.

• Las ramas pueden tener una línea de progreso diferente de la rama principal donde está el core de nuestro desarrollo. En algún momento podemos llegar a probar algunas de esas mejoras o cambios en el código y hacer una fusión a nuestro proyecto principal, ya que todo esto lo maneja Git de una forma muy eficiente



Saber la versión de git instalada:

git --version

1. Ejecutar git init dentro de mi proyecto mediante la terminal. git init

* Para saber el estado de mis archivos (modificados o no): git status

2. Agregar todos los archivos de mi proyecto: git add .

3. Crear un comentario de mis archivos a subir (importante). git commit -am"creación de archivos en mi repositorio"

* Para apuntar o asociar nuestro proyecto local a nuestro repositorio creado (<u>Una sola vez</u>):

git remote add origin https://github.com/user001/test-python.git (url de tu repositorio)

Esta línea de comando también la puedes obtener de tu repositorio al momento que terminas de crearlo o La URL se puede copiar directamente desde la interfaz web del repositorio (En GitHub: botón "Code" → elegir HTTPS o SSH → copiar la URL).

4. Subir mis archivos a mi repositorio:

git push origin master

```
hp@LAPTOP-3NKORD70 MINGW64 ~/SampleProject (master)
$ git push origin master
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 916 bytes | 36.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/yskumar007/SampleProject.git
d9382ad..672da20 master -> master
```

5. Para clonar nuestro proyecto del repositorio (<u>Una sola vez</u>):

git clone "enlace http de tu repositorio"

En caso pidan las credenciales aplicar los siguientes comandos (<u>Una sola vez</u>):

```
joanh@DESKTOP-178HFCN MINGW64 ~/PycharmProjects (master)
$ git commit -am"creación de archivos en mi primer repositorio"
Author identity unknown

*** Please tell me who you are.

Run

git config --global user.email "you@example.com"
git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'joanh@DESKTOP-178HFCN.(none)')
```

git config --global user.email "correo@gmail.com" git config --global user.name "Nombre Apellido"



* Para saber qué ramas hay en el proyecto y en cual nos ubicamos:

git branch

```
Anthony@DESKTOP-GDJJ6B3 MINGW64 /d/Developer/Python/Clase Python/G07 Básico/Sesiones (conexion-base-datos)

$ git branch

* conexion-base-datos
filters
master
```

* Crear ramas:

git checkout -b nombre-rama

* Eliminar ramas:

git branch -d nombre-rama

*Cambiar rama:

git checkout nombre-rama

*Verificar en a qué repositorio estás enlazado:

git remote -v

En caso solo hayas realizado cambios en ciertos archivos del proyecto aplicas los siguientes comandos:

git add . git commit –am"comentario descriptivo" git push origin rama

Resumen:







Repositorios



Definición:

Es un tipo de almacenamiento digital centralizado que los desarrolladores utilizan para realizar y administrar cambios en el código fuente de una aplicación.

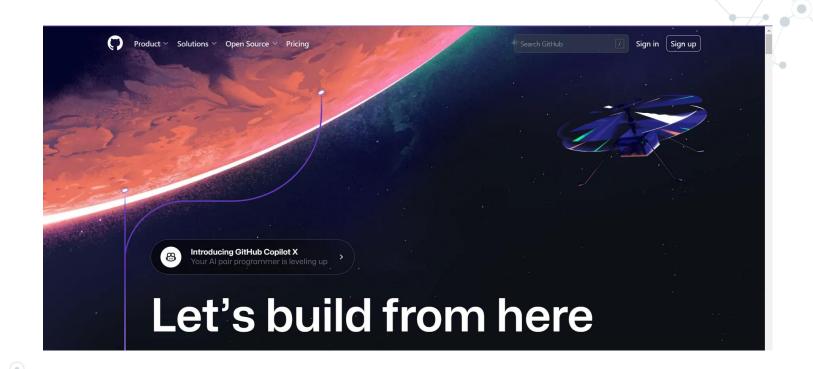
A lo largo del trabajo de un proyecto los desarrolladores tienen que almacenar y compartir carpetas, archivos de texto y otros tipos de documentos al desarrollar software.

Actualmente existen varios repositorios muy usados entre la comunidad de developers como: Github, Gitlub, bitbucket, etc.



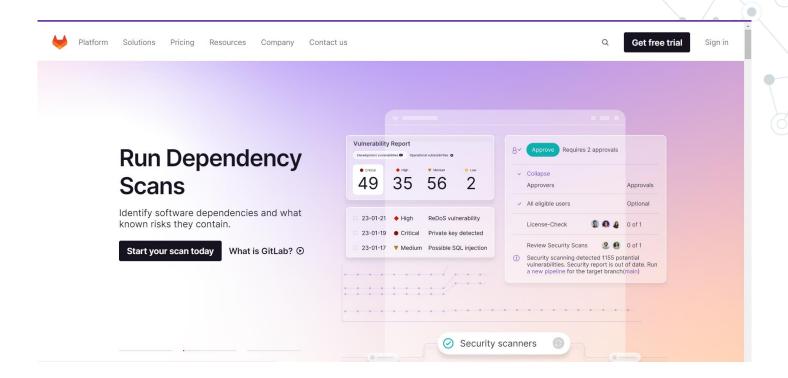
GitHub













Universidad Nacional Mayor de San Marcos

