



Universidad  
del Valle



# Fundamentos de Programación Orientada a Eventos

**Luis Yovany Romo Portilla, MsC.**

*Jueves 14:00 - 17:00 | Edif. B-13 -> SALA 4 -- MG -- MELENDEZ*

# Agenda

- Control de Versiones.
- Git - Introducción
- Comandos de Git
- Git Hub

# "FINAL".doc



FINAL.doc!



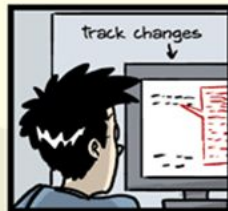
FINAL\_rev.2.doc



FINAL\_rev.6.COMMENTS.doc



FINAL\_rev.8.comments5.  
CORRECTIONS.doc



FINAL\_rev.18.comments7.  
corrections9.MORE.30.doc



FINAL\_rev.22.comments49.  
corrections.10.##\$%WHYDID  
ICOMETOGRADSCHOOL?????.doc



JORGE CHAM © 2012

WWW.PHDCOMICS.COM

# Control de Versiones

- Es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante



<https://git-scm.com/book/es/v2>

# ¿Por qué el control de versiones?

Colaboración    Versionamiento    Restaurar

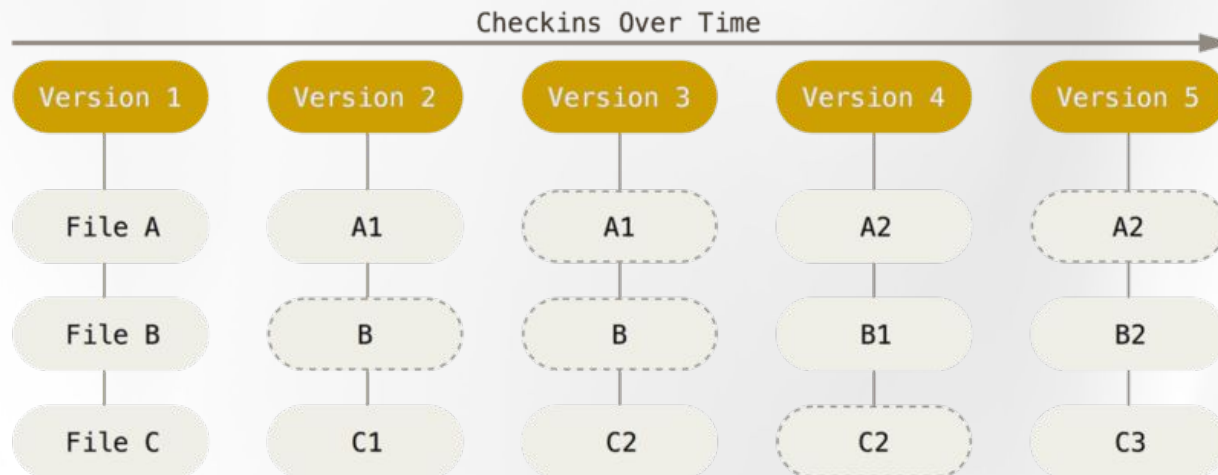
**Escenario:** Varios estudiantes están realizando un proyecto juntos

**Pregunta:** ¿Por qué no Google Drive?

Administración de Código vs Administración de Almacenamiento de Archivos

# GIT

- Software más utilizado para el seguimiento de los cambios en cualquier conjunto de archivos, normalmente utilizado para coordinar el trabajo entre los programadores que desarrollan el código fuente en colaboración durante el desarrollo de software



# Instalación de Git

## Linux

```
$ apt-get install git
```

## Windows

Descargar instalador y ejecutarlo



<https://git-scm.com/>



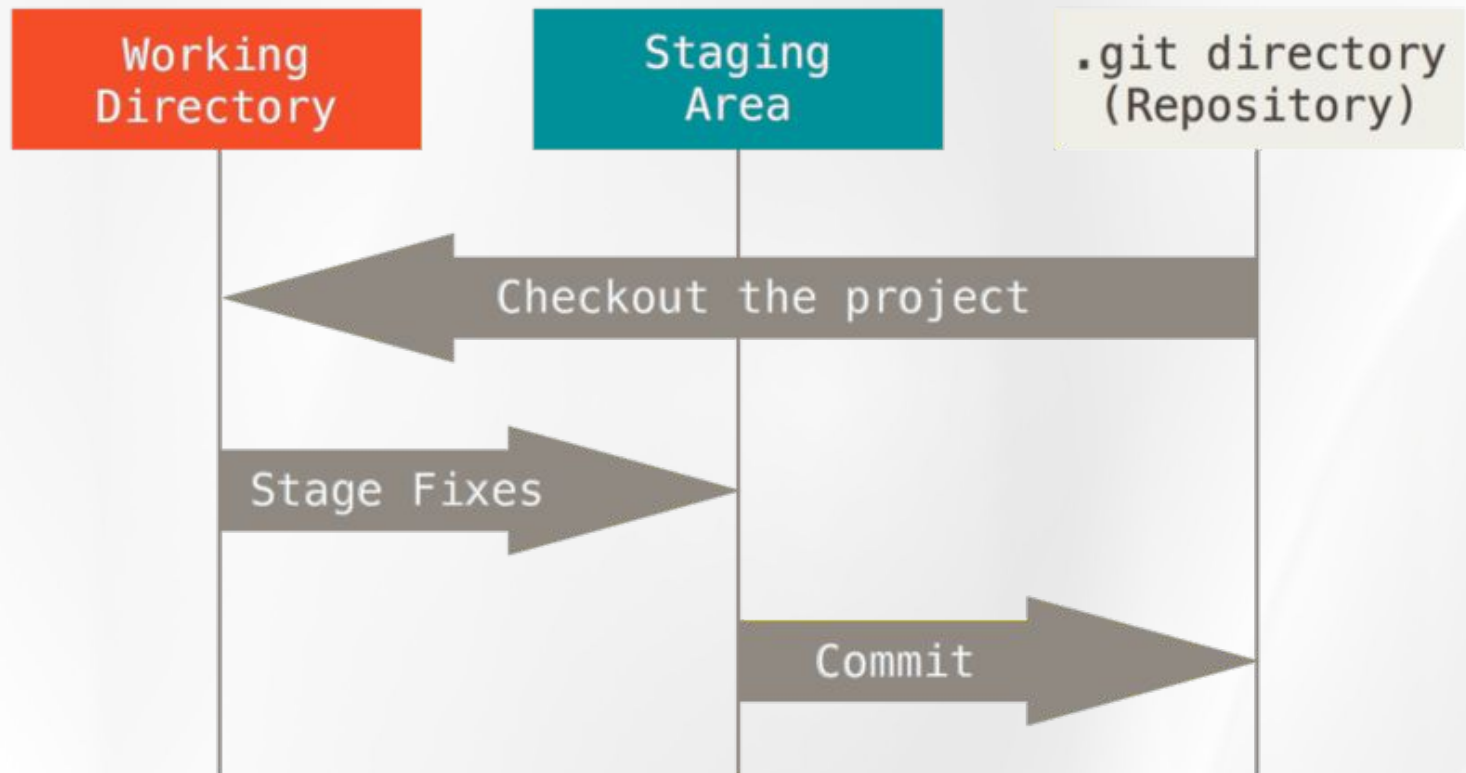
# Verificar instalación de Git

En una ventana de línea de comandos digitar git --version

```
yovaromo@Inspiron-5447: ~/l  
yovaromo@Inspiron-5447:~/Documents/dirwork$ git --version  
git version 2.34.1  
yovaromo@Inspiron-5447:~/Documents/dirwork$
```



# GIT



# Configurar identidad en Git



yovaromo@Inspiron-5447: ~/Documents/dirwork

```
yovaromo@Inspiron-5447:~/Documents/dirwork$ git config --global user.name "Yovany Romo"  
yovaromo@Inspiron-5447:~/Documents/dirwork$ git config --global user.email luis.romo@correounivalle.edu.co  
yovaromo@Inspiron-5447:~/Documents/dirwork$
```

# Verificando la configuración

```
yovaromo@Insp  
yovaromo@Inspiron-5447:~/Documents/dirwork$ git config --list  
user.name=Yovany Romo  
user.email=luis.romo@correounivalle.edu.co  
core.repositoryformatversion=0  
core.filemode=true  
core.bare=false  
core.logallrefupdates=true  
yovaromo@Inspiron-5447:~/Documents/dirwork$
```

# Ayuda sobre comandos

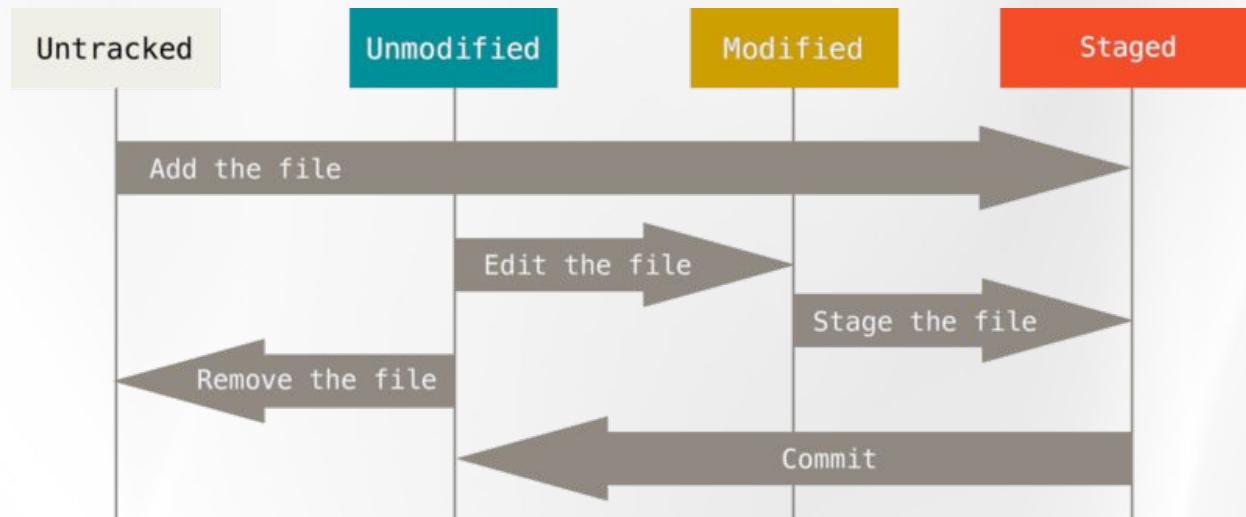
```
$ git help <verb>  
$ git <verb> --help
```

# Iniciando un Repositorio en un directorio

```
yovaromo@Inspiron-5447: ~/Documents
yovaromo@Inspiron-5447:~/Documents$ mkdir dirwork
yovaromo@Inspiron-5447:~/Documents$ cd dirwork/
yovaromo@Inspiron-5447:~/Documents/dirwork$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/yovaromo/Documents/dirwork/.git/
yovaromo@Inspiron-5447:~/Documents/dirwork$
```

# Verificando el estado del repositorio

```
yovaromo@Inspiron-5447: ~  
yovaromo@Inspiron-5447:~/Documents/dirwork$ git status  
On branch master  
  
No commits yet  
  
nothing to commit (create/copy files and use "git add" to track)  
yovaromo@Inspiron-5447:~/Documents/dirwork$
```





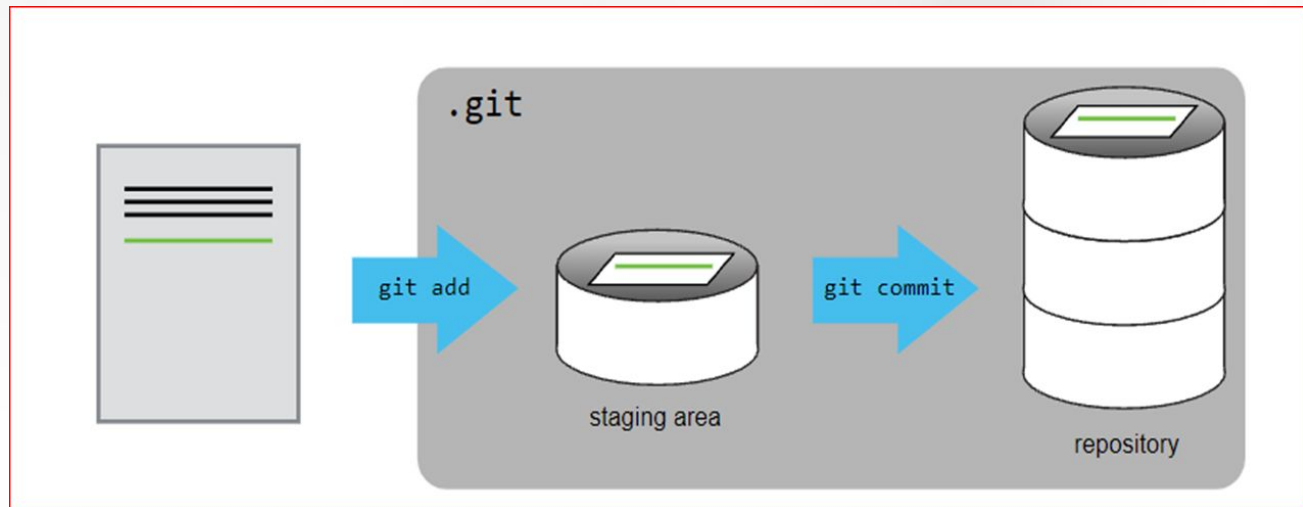
# Agregando un archivo al repositorio

```
yovaromo@
yovaromo@Inspiron-5447:~/Documents/dirwork$ touch Main.java
yovaromo@Inspiron-5447:~/Documents/dirwork$ git add .
yovaromo@Inspiron-5447:~/Documents/dirwork$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   Main.java

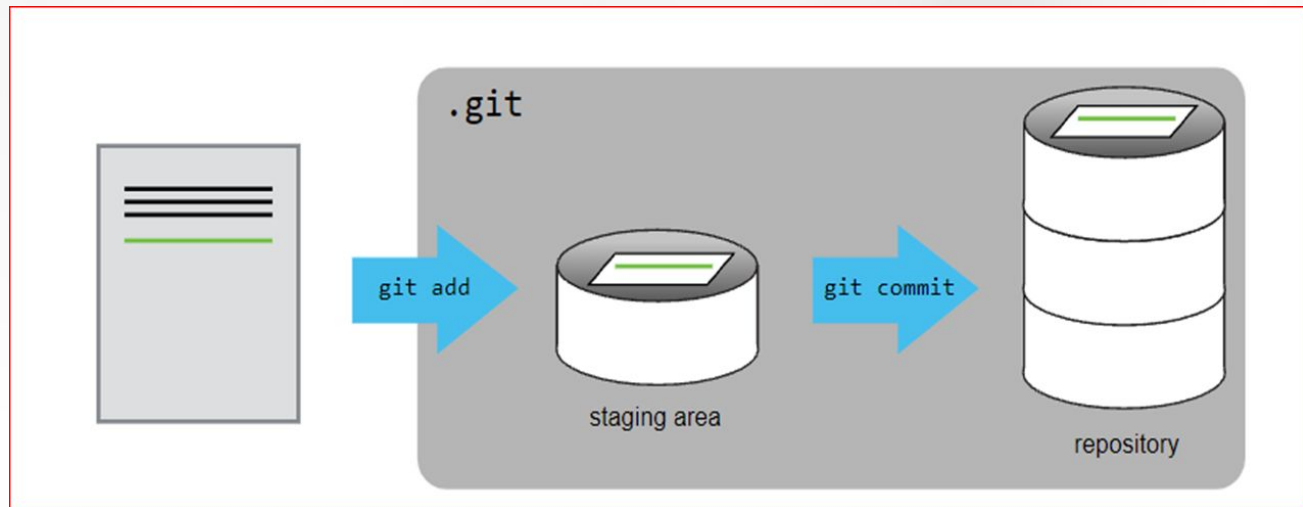
yovaromo@Inspiron-5447:~/Documents/dirwork$
```





# Haciendo Commit

```
yovaromo@Inspiron-5447: ~/Documents/dirwork$ git commit -m "Primer commit"
[master (root-commit) c6a5e97] Primer commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Main.java
yovaromo@Inspiron-5447:~/Documents/dirwork$ git status
On branch master
nothing to commit, working tree clean
yovaromo@Inspiron-5447:~/Documents/dirwork$
```

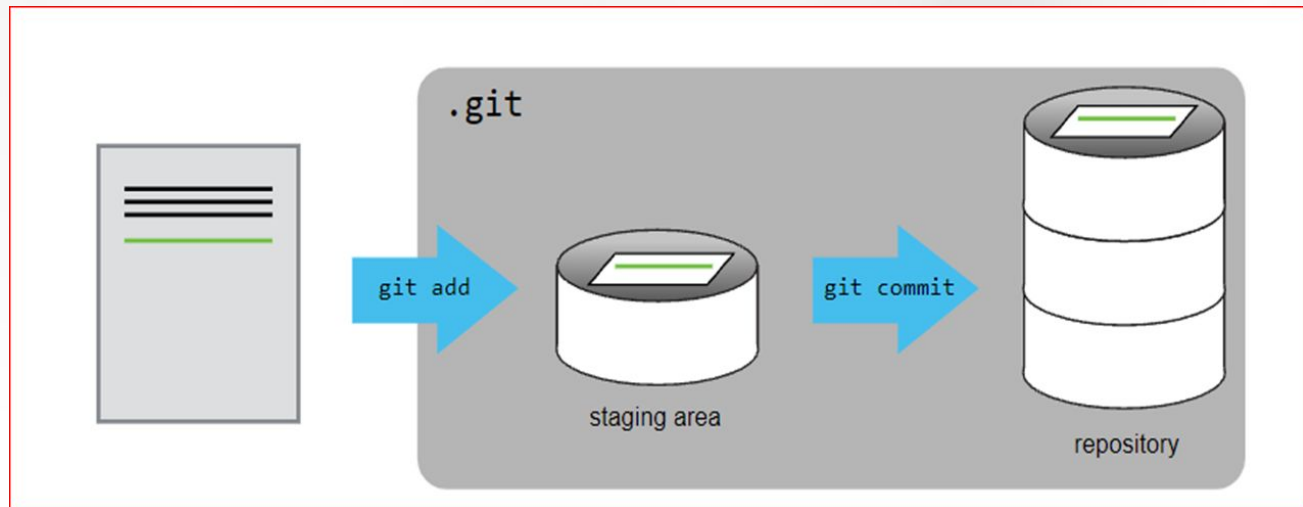


# Ver el historial de Commit

```
yovaromo@Inspiron-5447:~/Documents/dirwork$ git log
commit c6a5e97050829aa1cf199fe3eb4fb32b4e0f3b1f (HEAD -> master)
Author: Yovany Romo <luis.romo@correounivalle.edu.co>
Date: Thu May 19 06:35:18 2022 -0500

    Primer commit

yovaromo@Inspiron-5447:~/Documents/dirwork$
```

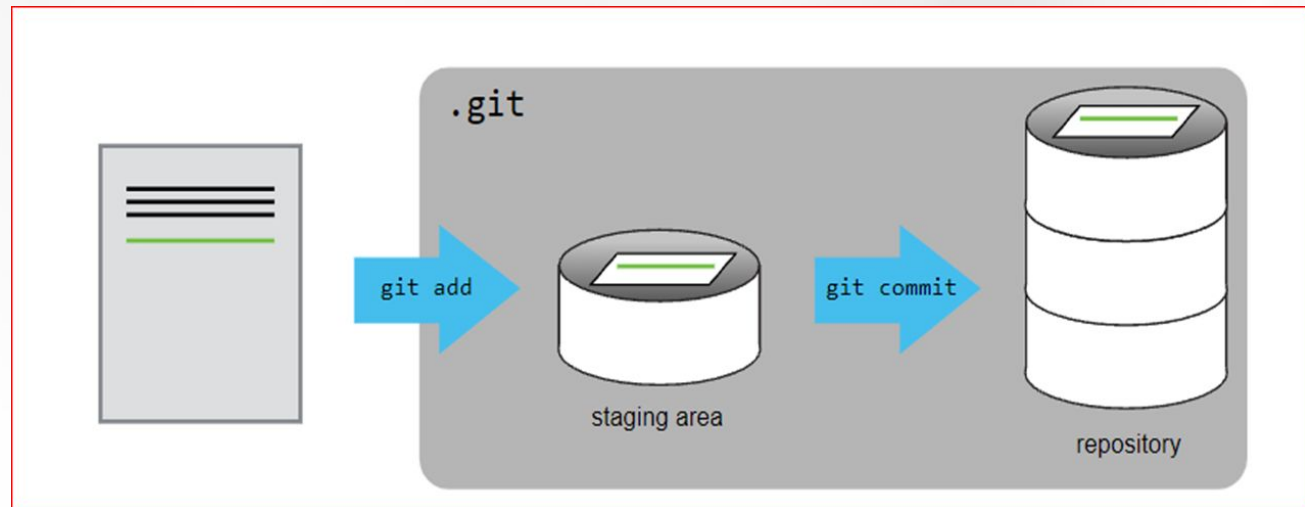


# Extraer archivo del Staging Area

```
yovaromo@Inspiron-5447: ~/Documents/dirwork$ touch Triangulo.java
yovaromo@Inspiron-5447:~/Documents/dirwork$ git add Triangulo.java
yovaromo@Inspiron-5447:~/Documents/dirwork$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   Triangulo.java

yovaromo@Inspiron-5447:~/Documents/dirwork$ git reset HEAD Triangulo.java
yovaromo@Inspiron-5447:~/Documents/dirwork$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Triangulo.java

nothing added to commit but untracked files present (use "git add" to track)
yovaromo@Inspiron-5447:~/Documents/dirwork$
```



# Github

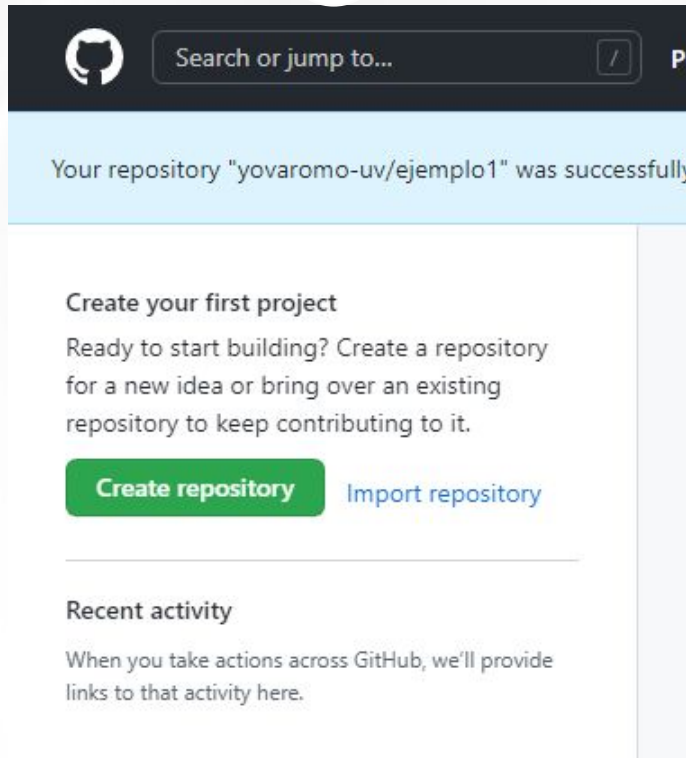
Es una plataforma de alojamiento de código para el control de versiones y la colaboración.



<https://github.com/>

# Creando Primer Repositorio en Github

1



Search or jump to...

Your repository "yovaromo-uv/ejemplo1" was successfully

Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

[Create repository](#) [Import repository](#)

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

Cuando no se tiene aún  
repositorios

2




Owner \* Repository name \*


yovaromo-uv / ejemplo1 ✓

Great repository names are short, lowercase, and contain only numbers, lowercase letters, hyphens, and underscores. **ejemplo1 is available.** Need inspiration? How about

Description (optional)

☒  **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.

 You are creating a public repository in your personal account.

[Create repository](#)

3

# Creando Primer Repositorio en Github

yovaromo-uv / ejemplo1 Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

## Quick setup — if you've done this kind of thing before

 Set up in Desktop or **HTTPS** **SSH** `https://github.com/yovaromo-uv/ejemplo1.git`

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include

## ...or create a new repository on the command line

```
echo "# ejemplo1" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/yovaromo-uv/ejemplo1.git
git push -u origin main
```

## ...or push an existing repository from the command line

```
git remote add origin https://github.com/yovaromo-uv/ejemplo1.git
git branch -M main
git push -u origin main
```

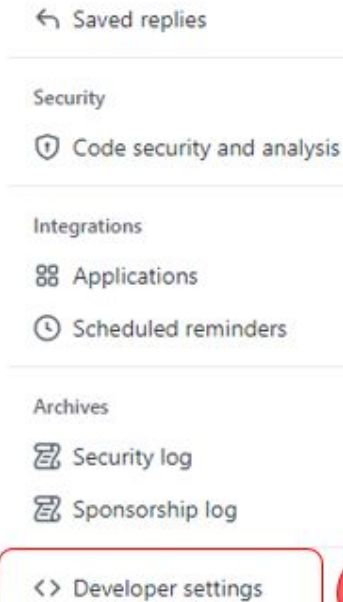
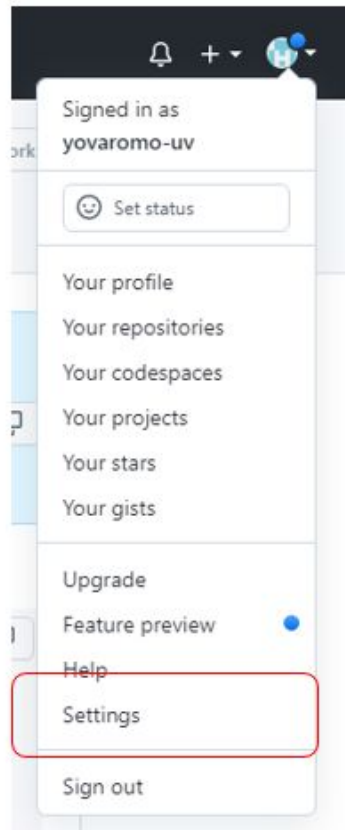


# Asociando repositorio local a remoto

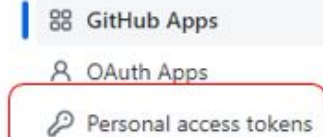
```
yovaromo@Inspiron-5447: ~/Documents/dirwork
yovaromo@Inspiron-5447:~/Documents/dirwork$ git branch
* master
yovaromo@Inspiron-5447:~/Documents/dirwork$ git branch -M main
yovaromo@Inspiron-5447:~/Documents/dirwork$ git remote add origin https://github.com/yovaromo-uv/ejemplo2.git
yovaromo@Inspiron-5447:~/Documents/dirwork$ git remote
origin
yovaromo@Inspiron-5447:~/Documents/dirwork$ git push -u origin main
Username for 'https://github.com':
```



# Creando un Personal Access Token



Settings / Developer settings



Generate new token

5

to the GitHub API.



Universidad  
del Valle

# Creando un Personal Access Token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

## Note

Personal Access Tokens

What's this token for?

## Expiration \*

No expiration The token will never expire!

GitHub strongly recommends that you set an expiration date for your token to help keep your information secure.

[Learn more](#)

## Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input checked="" type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects

7

Generate token

6

# Creando un Personal Access Token

8

## Personal access tokens

Generate new token

Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

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

✓ ghp\_NFSUN



Delete

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

# Actualizando de local a remoto

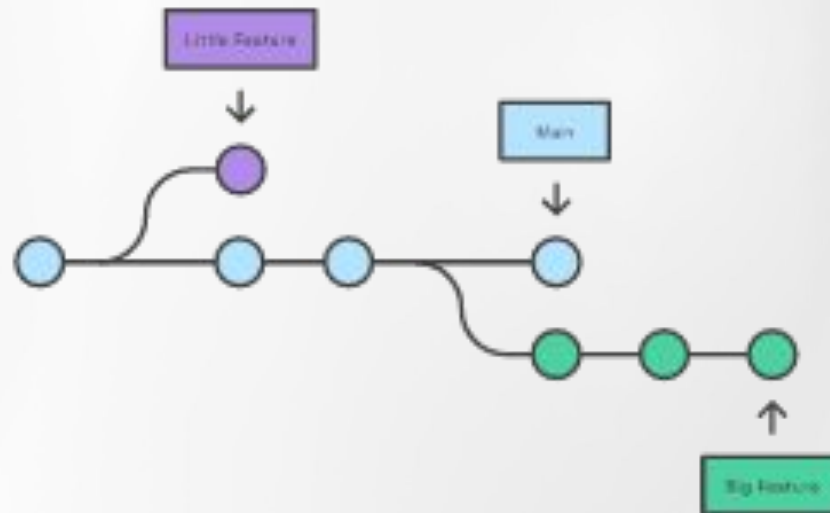
```
yovaromo@Inspiron-5447:~/Documents/dirwork$ git push origin main
Username for 'https://github.com': yovaromo-uv
Password for 'https://yovaromo-uv@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 452 bytes | 452.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/yovaromo-uv/ejemplo2.git
 * [new branch]      main -> main
yovaromo@Inspiron-5447:~/Documents/dirwork$
```

# Actualizando de remoto a local

```
yovaromo@Inspiron-5447:~/Documents/dirwork$ git pull origin main
From https://github.com/yovaromo-uv/ejemplo2
 * branch                main                -> FETCH_HEAD
Updating d51e1b7..04cace0
Fast-forward
 Ventana.java | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 Ventana.java
yovaromo@Inspiron-5447:~/Documents/dirwork$
```

# Ramas en Git

- Una rama representa una línea independiente de desarrollo. Las ramas sirven como una abstracción de los procesos de cambio, preparación y confirmación.





# Ramas en Git - Comandos

git branch o git branch -- list	Lista las ramas del repositorio
git branch <branch>	Crea una nueva rama, a partir de la actual
git branch -d <branch>	Elimina la rama especificada
git branch -m <branch>	Cambia el nombre de la rama actual
git branch -a	Lista todas la ramas remotas
git checkout -b <new-branch>	Crea una nueva rama y se cambia a ella
git checkout <branchname>	Activa la rama que se especifica
git merge <branch>	Fusiona la rama actual con la especificada



# Introducción



## Gracias!!!

**Luis Yovany Romo Portilla, MsC.**

*Fundamentos de Programación Orientada a Eventos*