Summer School 2025

Repositorios de git

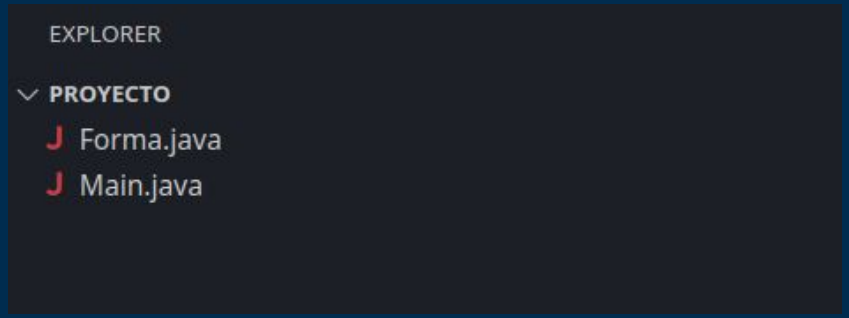
Qué son, cómo usarlos y buenas prácticas

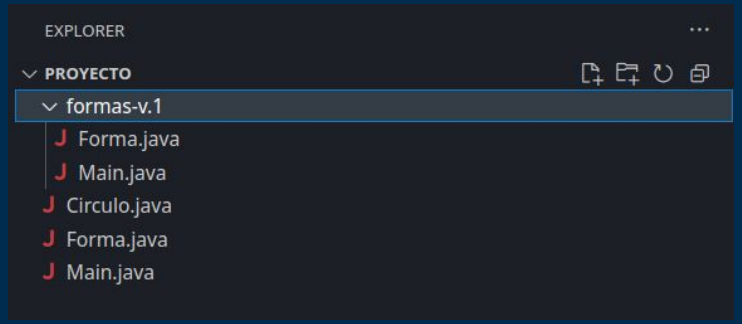
Markel Mencía

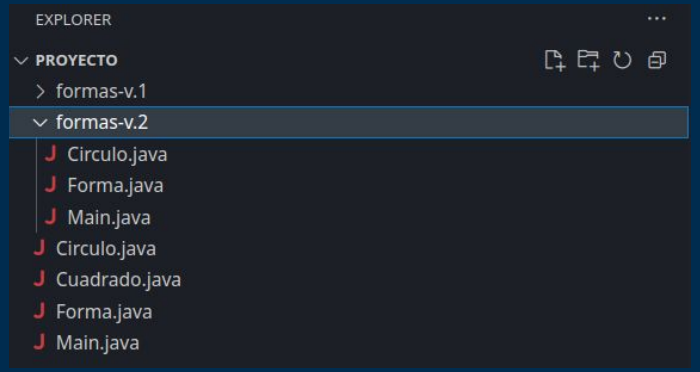
¿Qué es git?


BitKeeper

¿Cómo funciona?

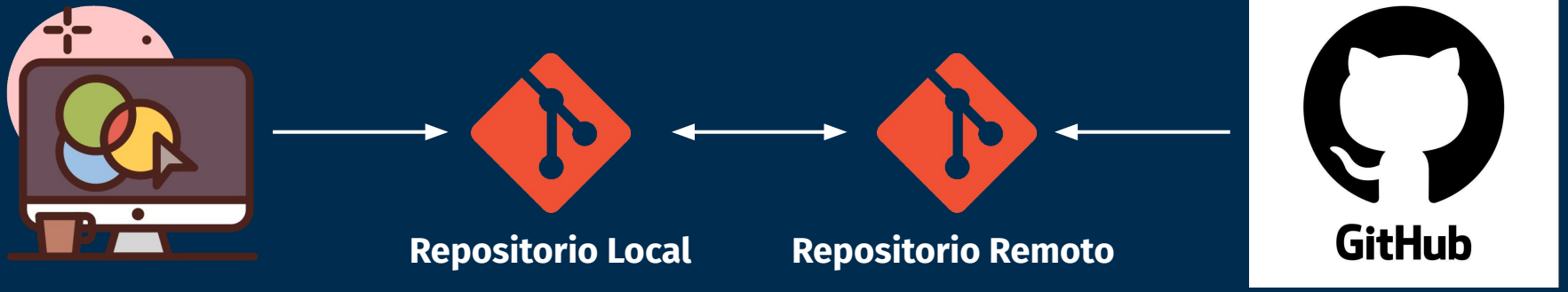

EXPLORER
PROYECTO
Forma.java
Main.java

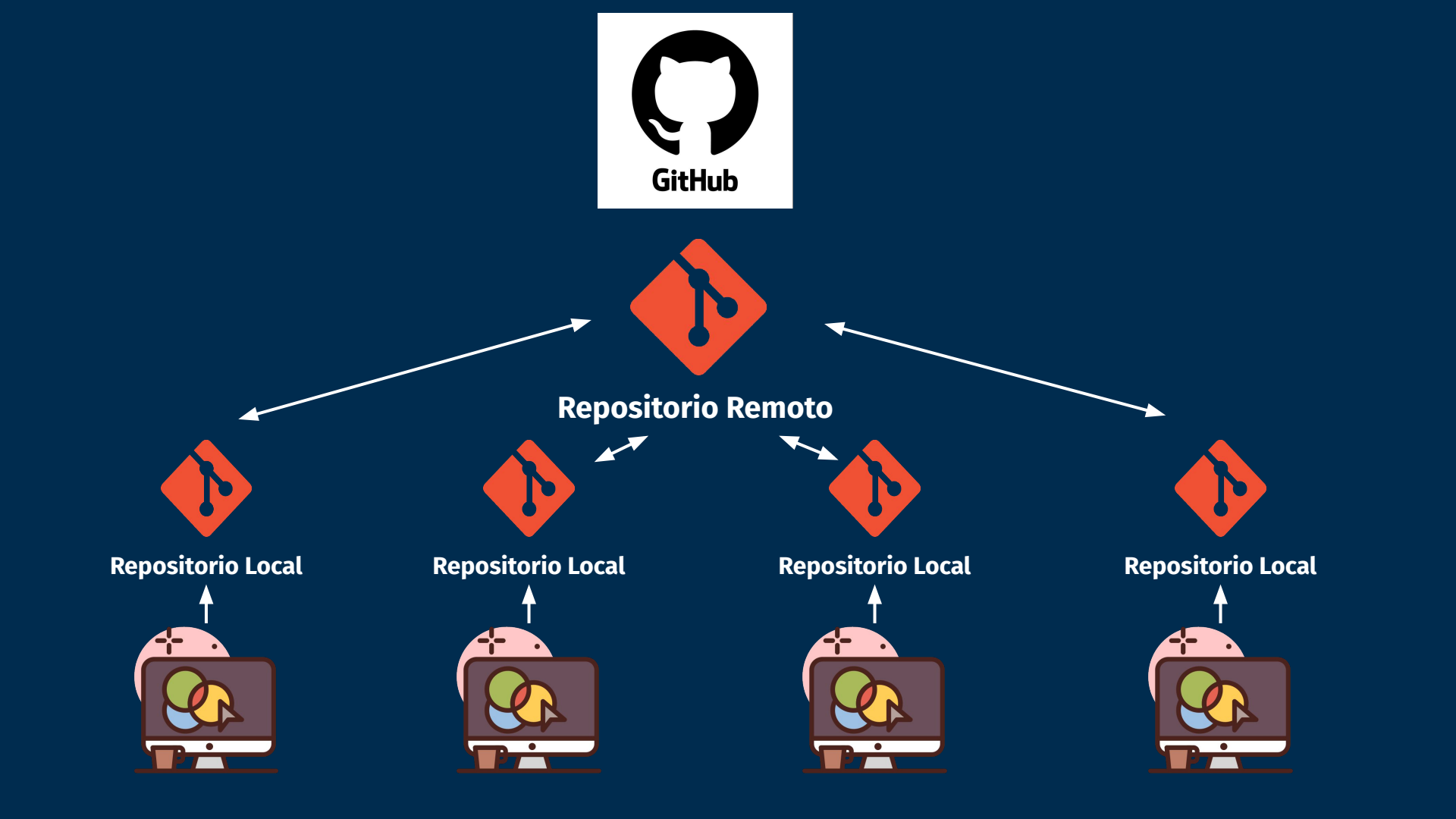

EXPLORER
PROYECTO
formas-v.1
Forma.java
Main.java
Circulo.java
Forma.java
Main.java


EXPLORER
PROYECTO
formas-v.1
formas-v.2
Circulo.java
Forma.java
Main.java
Circulo.java
Cuadrado.java
Forma.java
Main.java

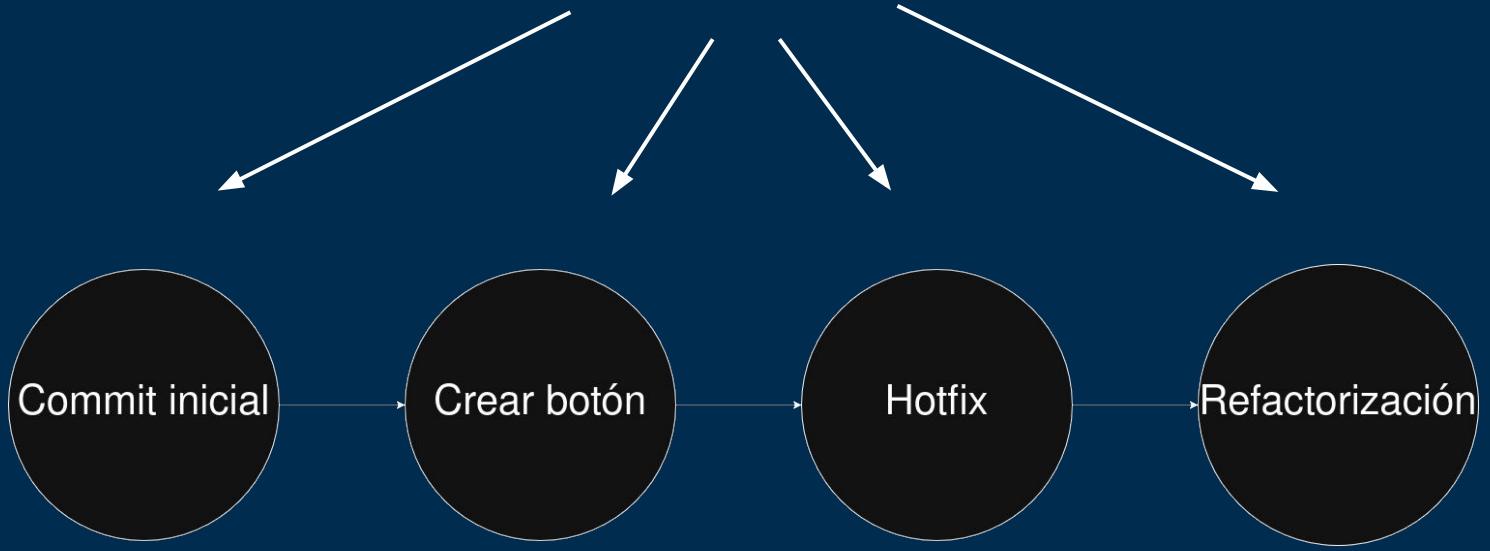

Commit inicial
Añadir Circulo
Añadir Cuadrado
Añadir Triángulo
v.0
v.1
v.2
v.3

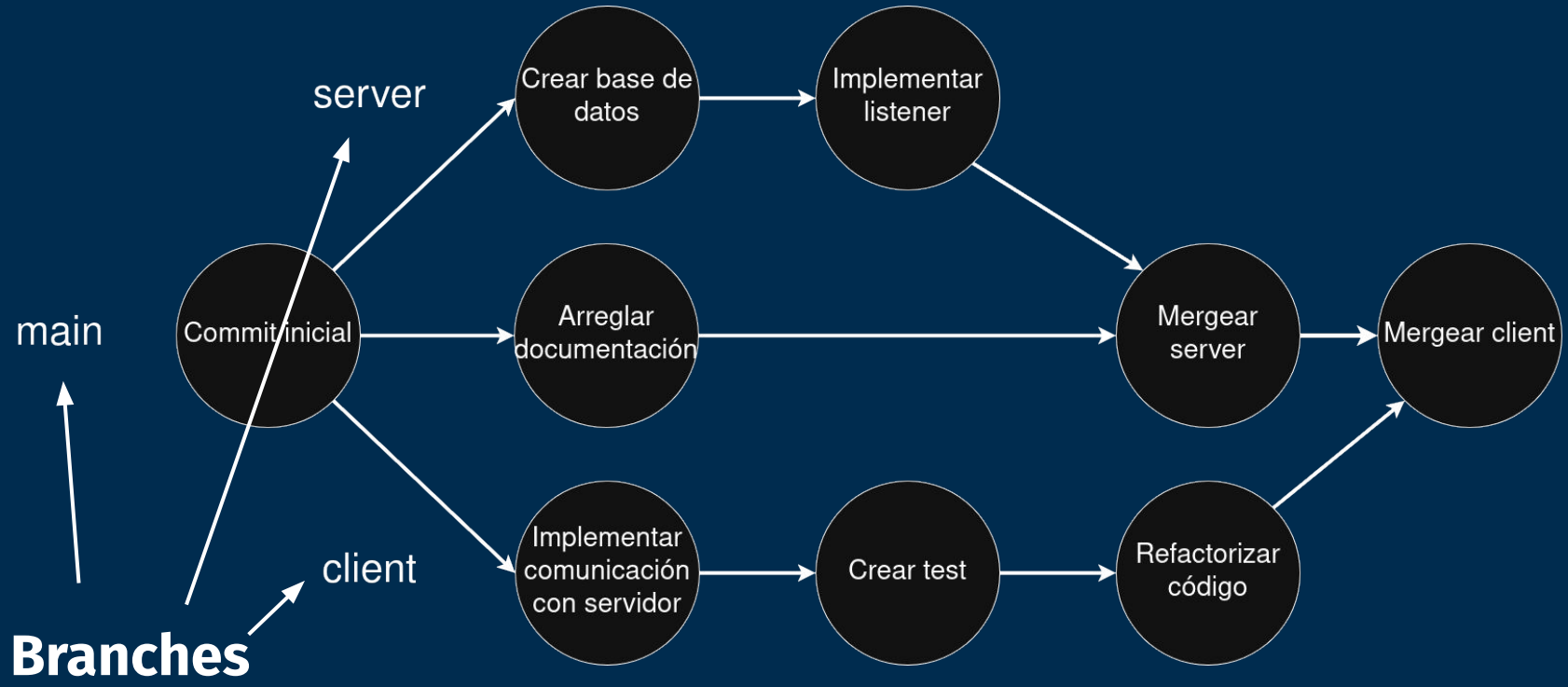

GitHub


Repositorio Local
Repositorio Remoto
GitHub


GitHub
Repositorio Remoto
Repositorio Local
Repositorio Local
Repositorio Local
Repositorio Local

Commits




server
Crear base de datos
Implementar listener
main
Commit inicial
Arreglar documentación
Mergear server
Mergear client
Implementar comunicación con servidor
Crear test
Refactorizar código
client
Branches

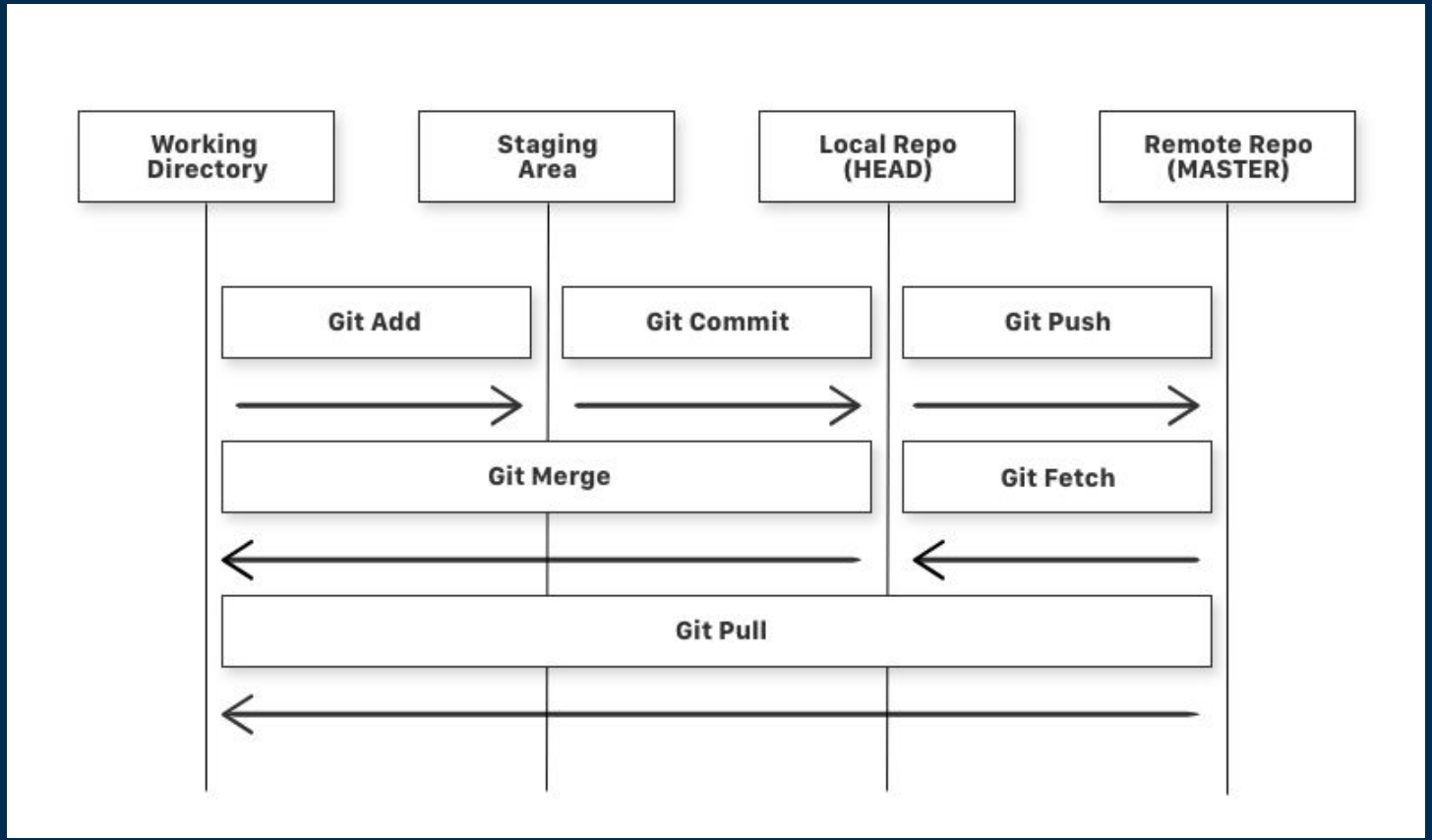
Cómo usar git

Instalación

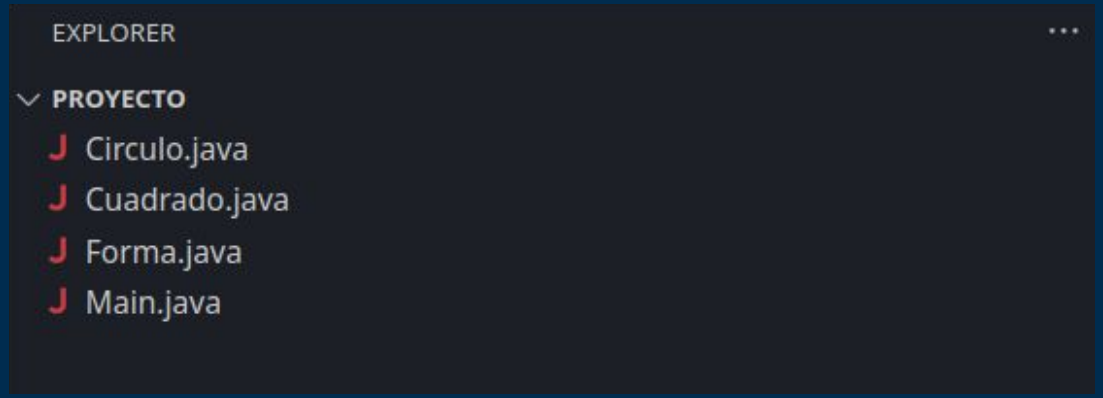
$ sudo apt install git

$ sudo pacman -S git

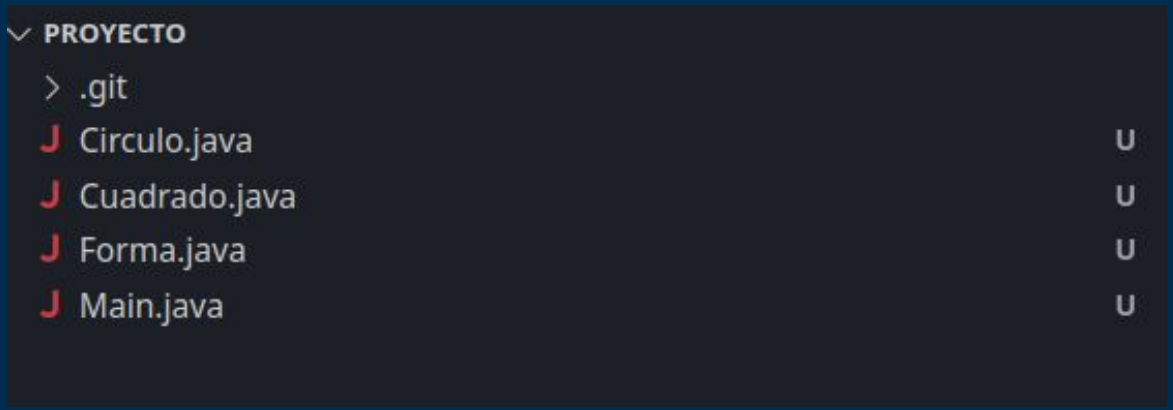
$ sudo dnf install git


Working Directory
Staging Area
Local Repo (HEAD)
Remote Repo (MASTER)
Git Add
Git Commit
Git Push
Git Merge
Git Fetch
Git Pull

Espacio de trabajo

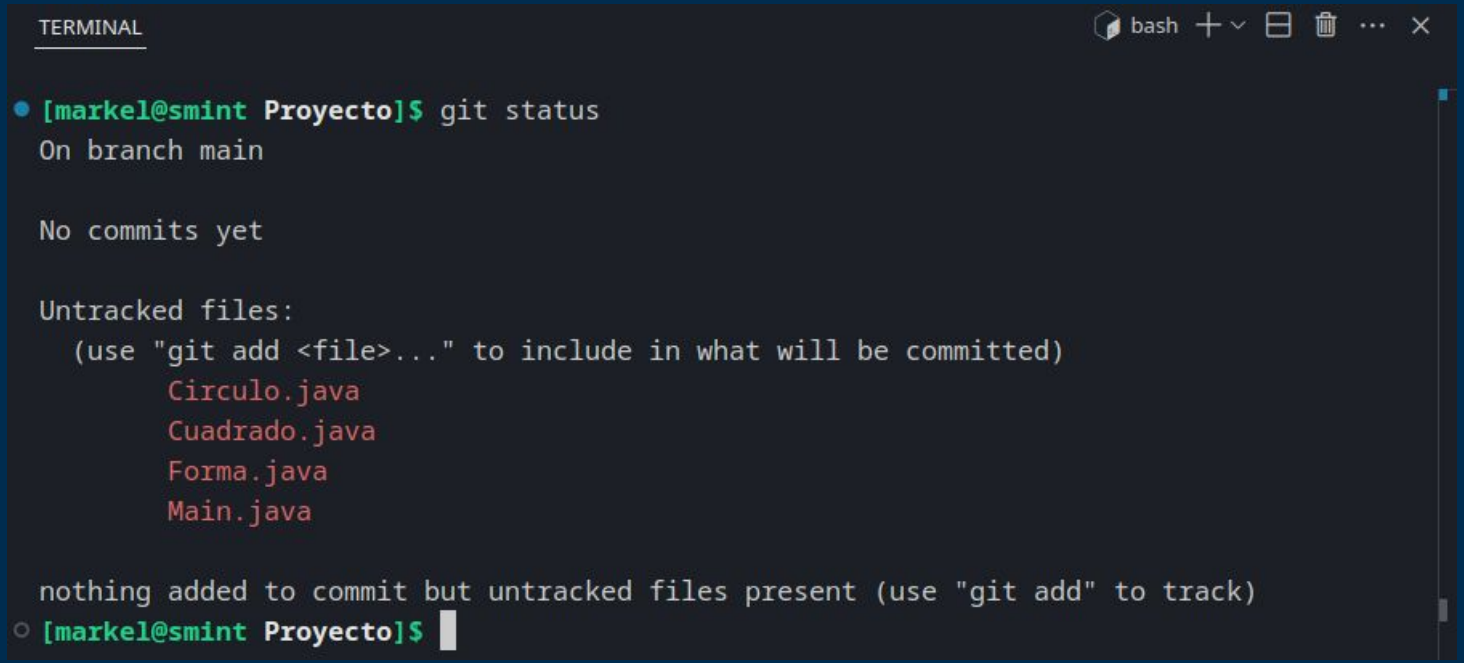


Espacio de trabajo

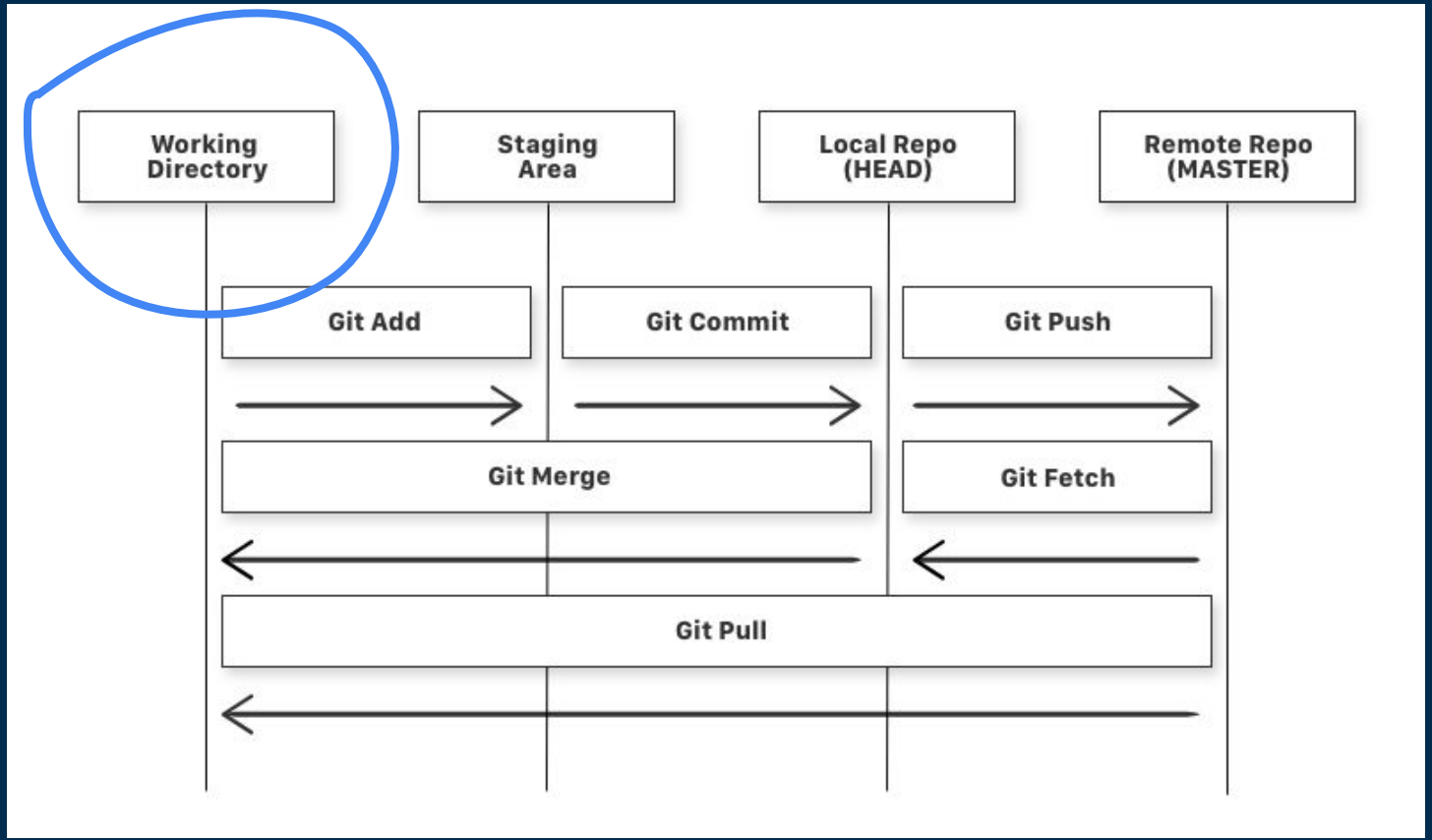


$ git init

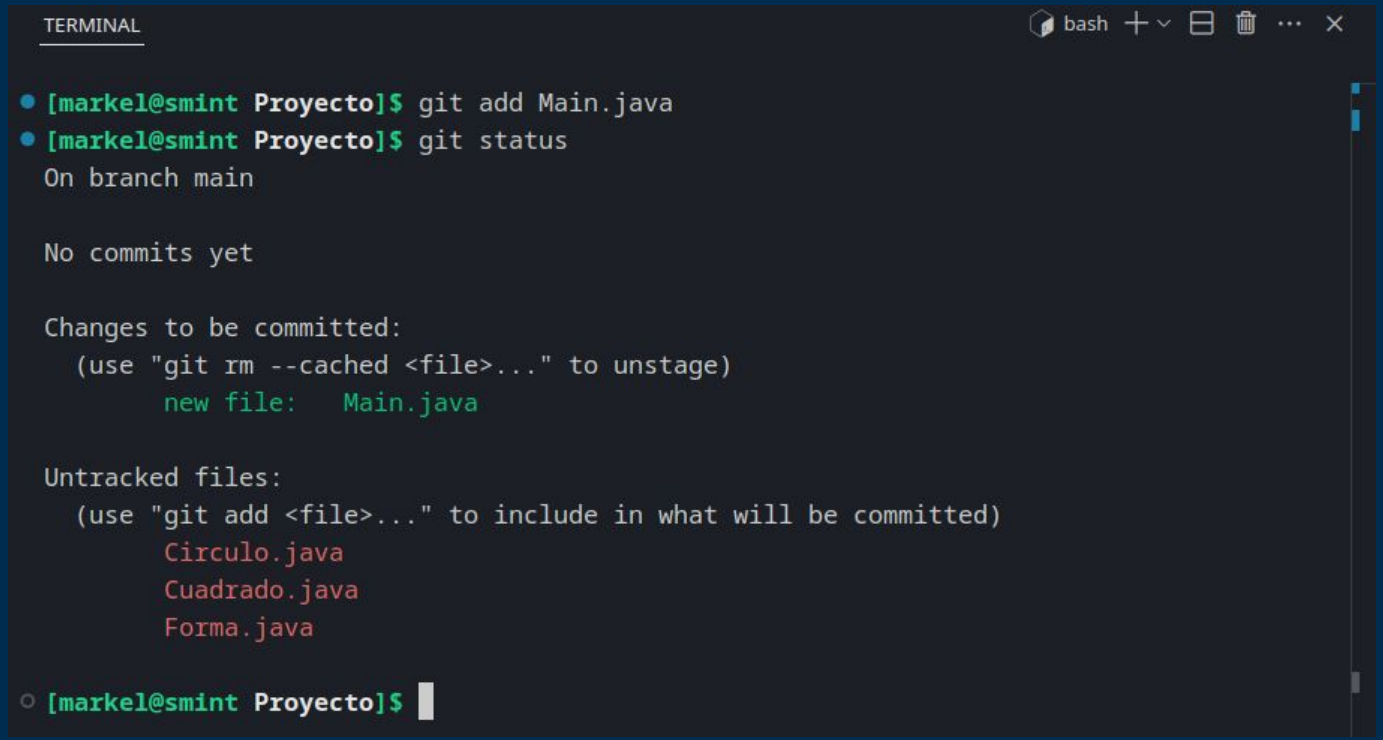
Espacio de trabajo



$ git status

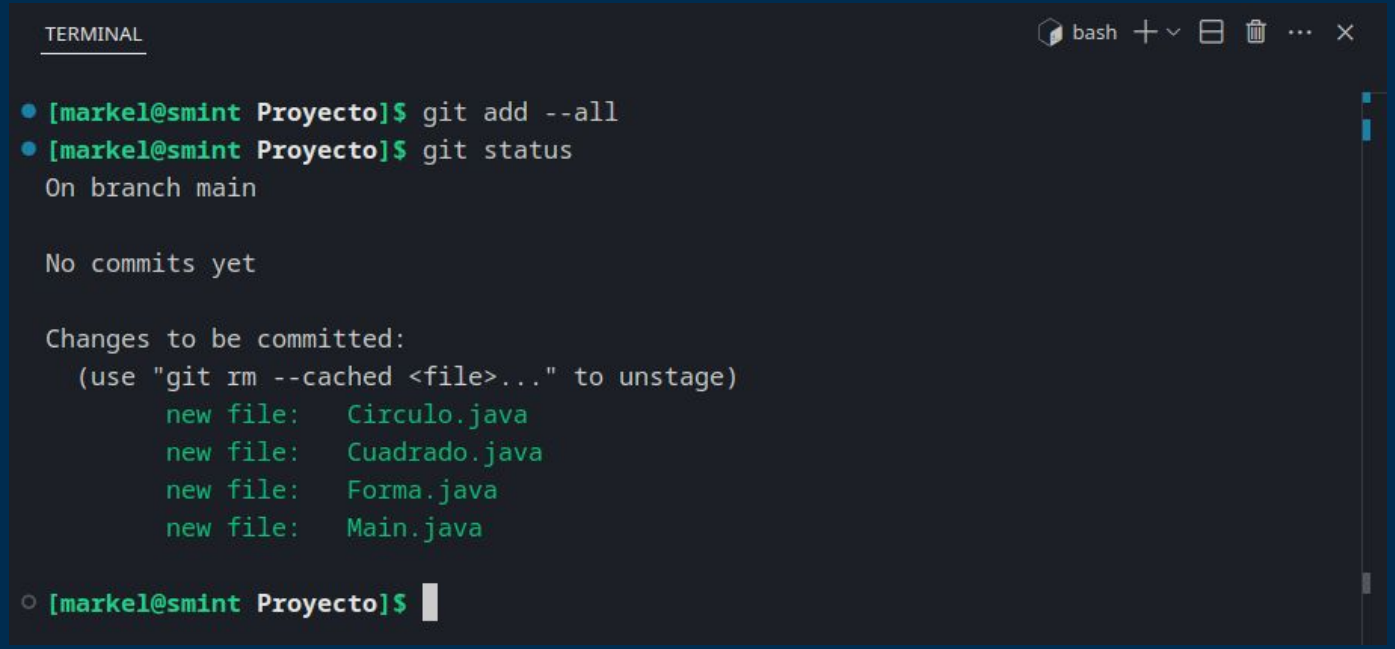

Working Directory
Staging Area
Local Repo (HEAD)
Remote Repo (MASTER)
Git Add
Git Commit
Git Push
Git Merge
Git Fetch
Git Pull

Área de staging



$ git add <fichero>

Área de staging



$ git add --all

Área de staging

Main.java M ×

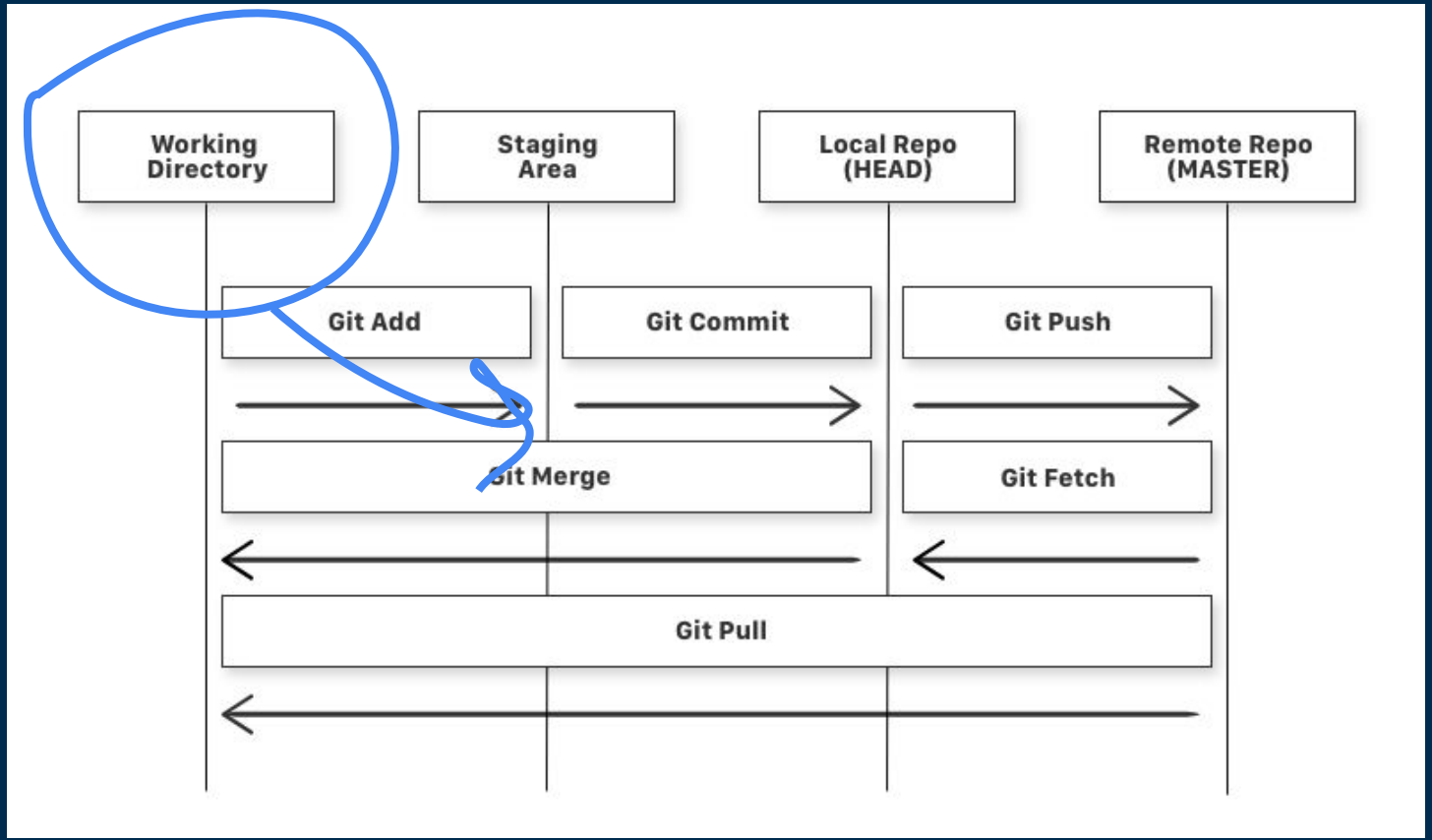
Main.java

```java
public class Main {
    public static void main() {
        System.out.println("Estoy modificando un fichero stageado")
    }
}
```

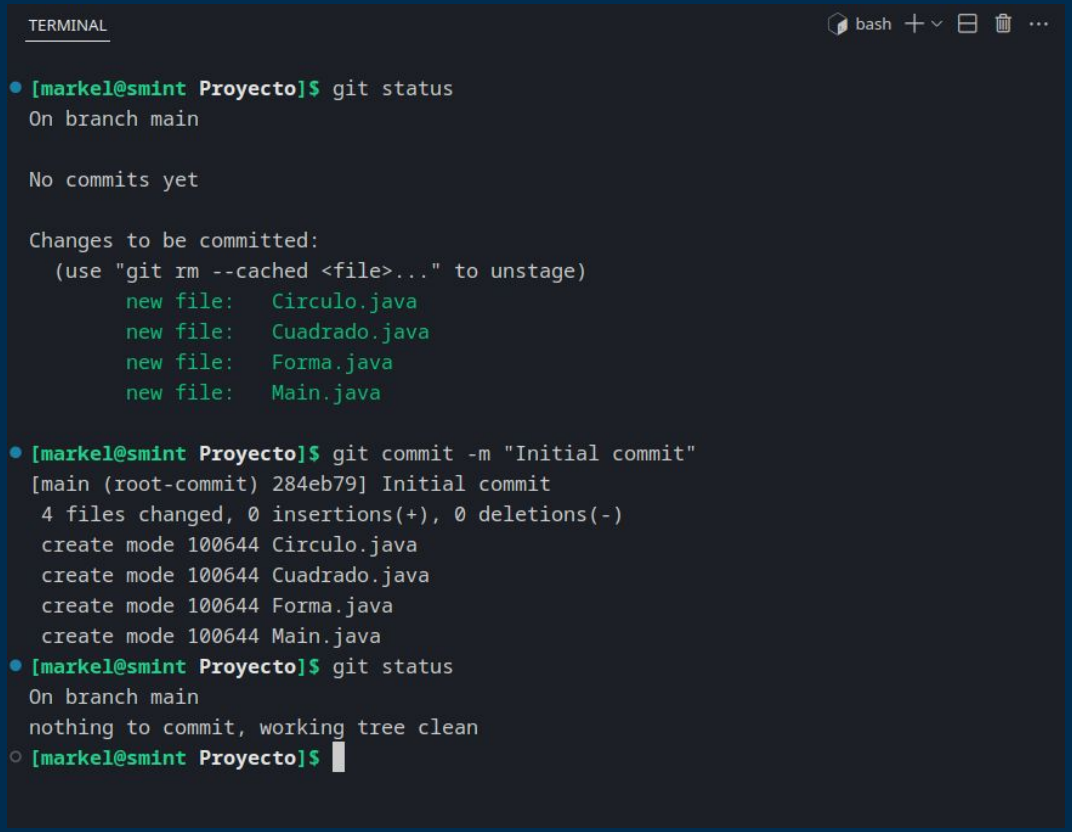
TERMINAL

bash

```
[markel@smint Proyecto]$ git status
On branch main

No commits yet

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

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Main.java

[markel@smint Proyecto]$
```


Working Directory
Staging Area
Local Repo (HEAD)
Remote Repo (MASTER)
Git Add
Git Commit
Git Push
Git Merge
Git Fetch
Git Pull

Commits



$ git commit -m <nombre del commit>

Commits

Main.java M

Main.java

```java
public Class Main {
    public static int main() {
        System.out.println("hello world");
    }
}
```

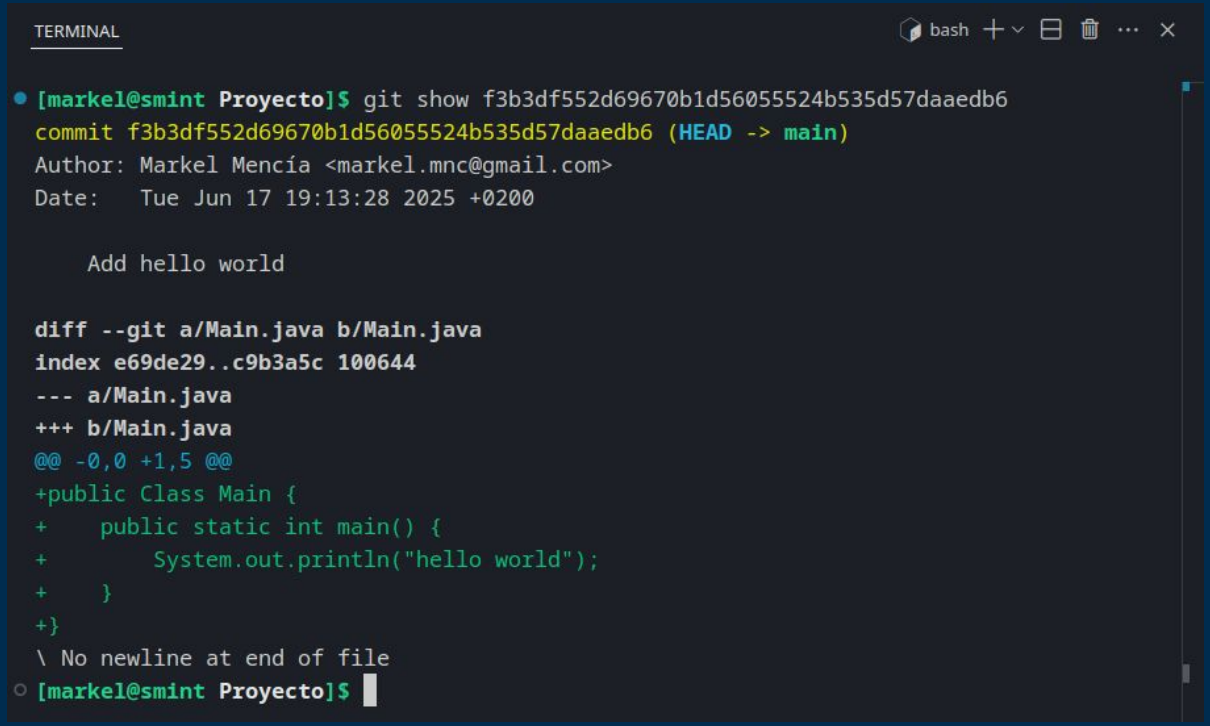
TERMINAL

```
[markel@smint Proyecto]$ git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Main.java

no changes added to commit (use "git add" and/or "git commit -a")
[markel@smint Proyecto]$ git add Main.java
[markel@smint Proyecto]$ git commit -m "Add hello world"
[main f3b3df5] Add hello world
 1 file changed, 5 insertions(+)
[markel@smint Proyecto]$
```

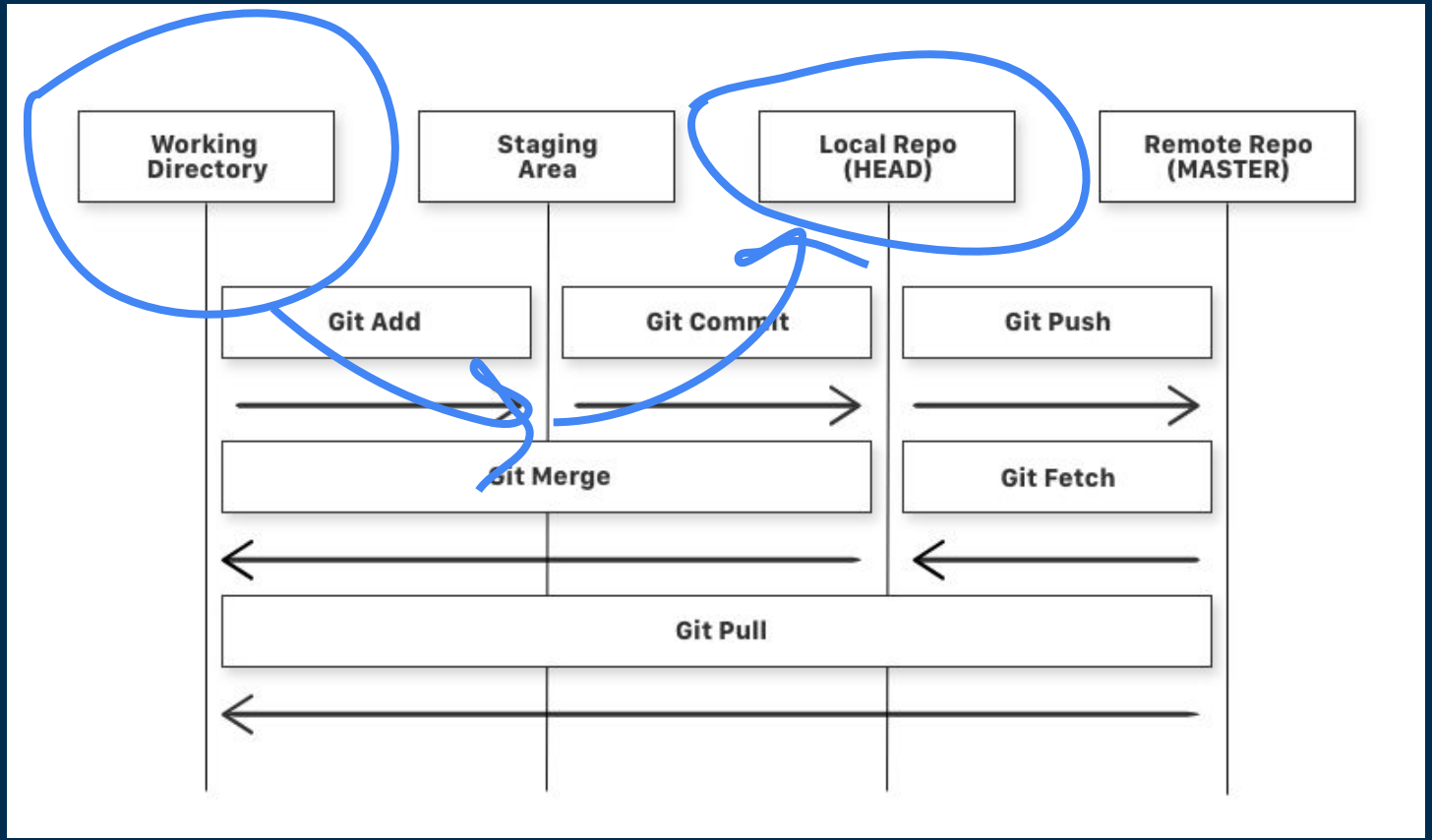
Commits

$ git log

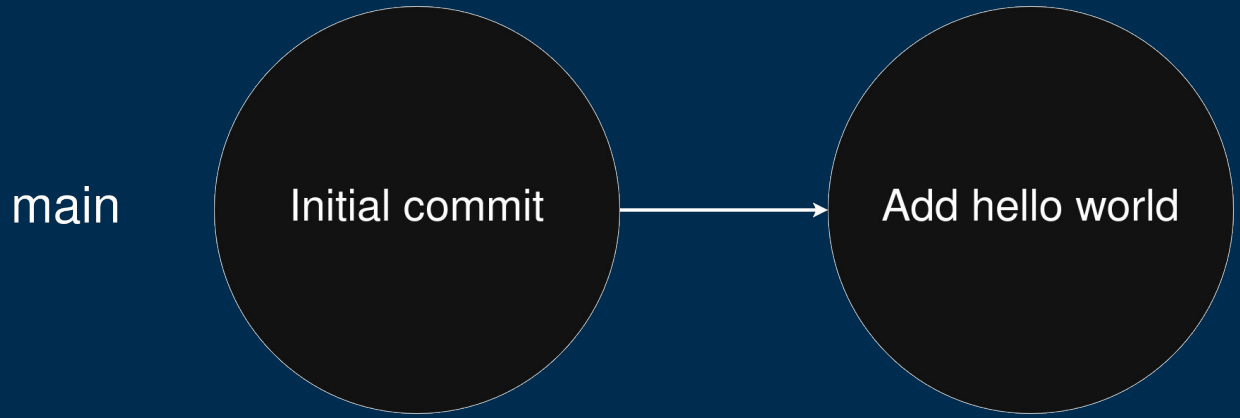
Commits



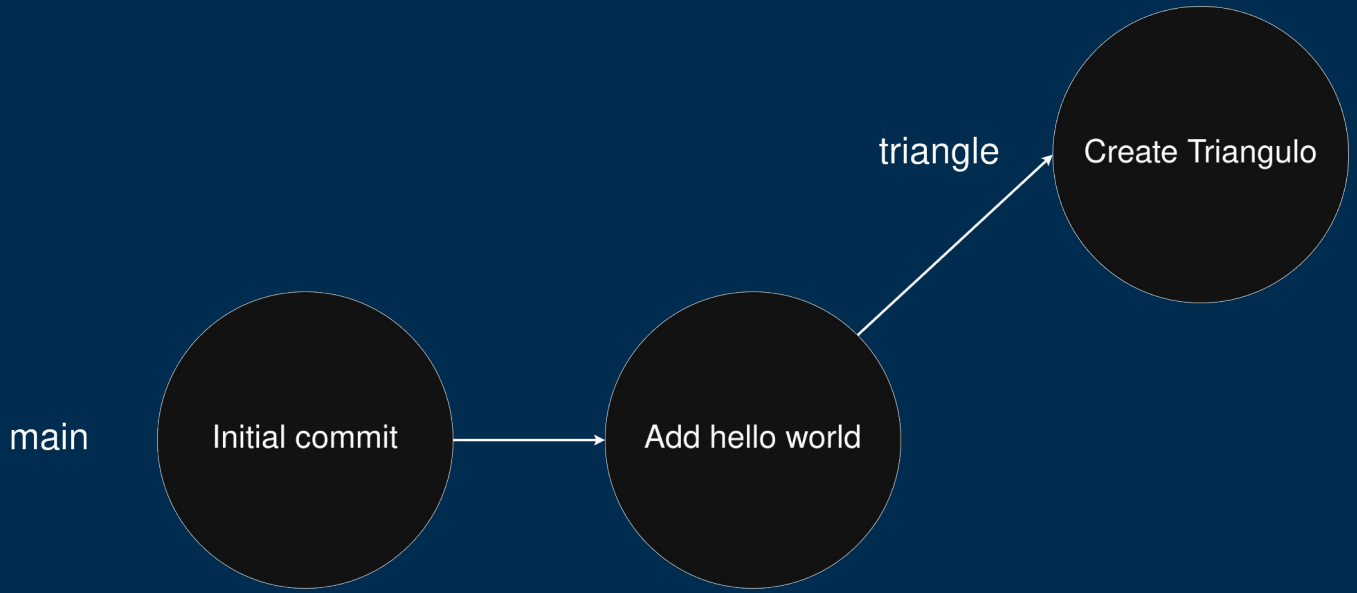
$ git show <hash del commit>


Working Directory
Staging Area
Local Repo (HEAD)
Remote Repo (MASTER)
Git Add
Git Commit
Git Push
Git Merge
Git Fetch
Git Pull

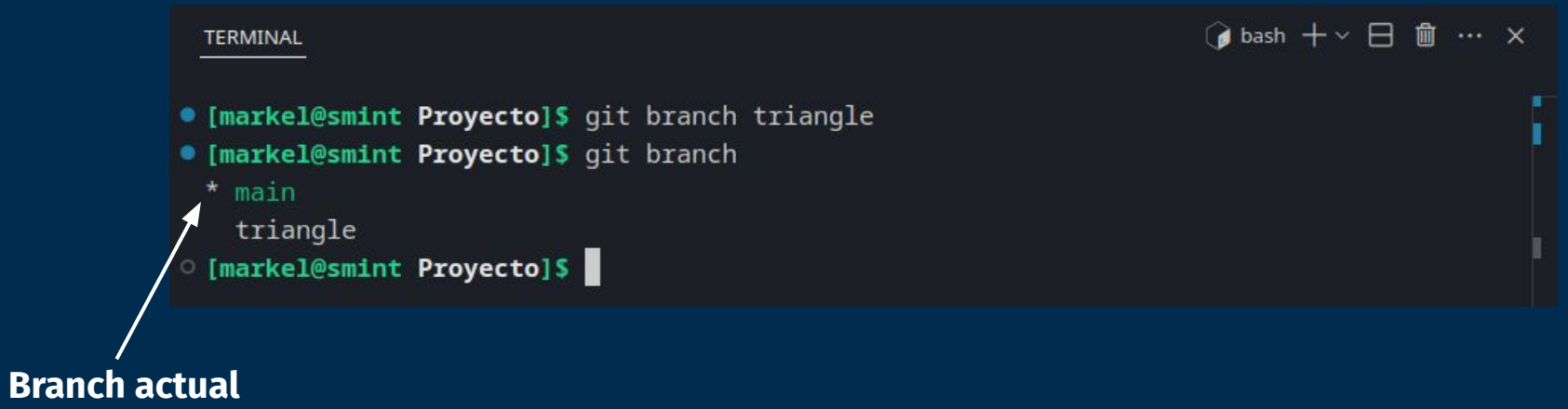
Branches



Branches

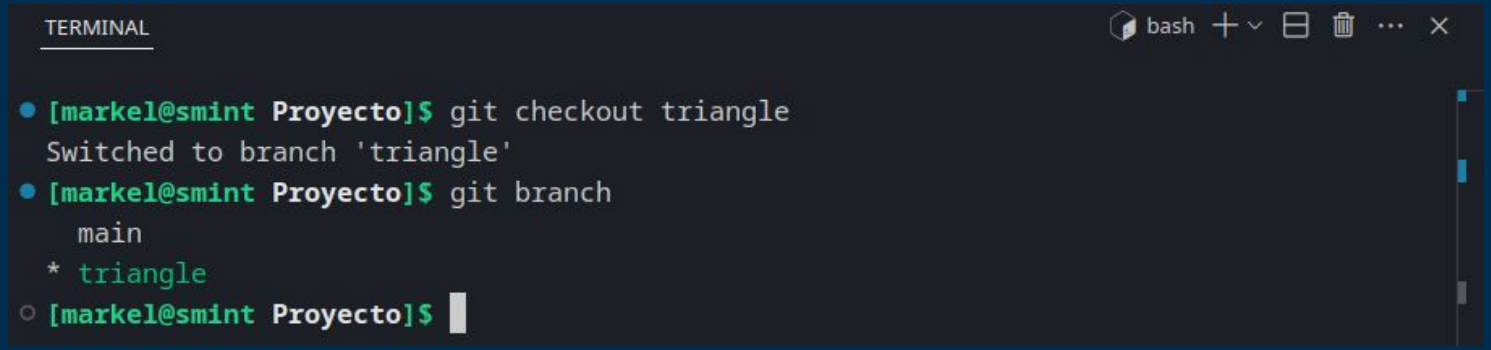

triangle
Create Triangulo
main
Initial commit
Add hello world

Branches



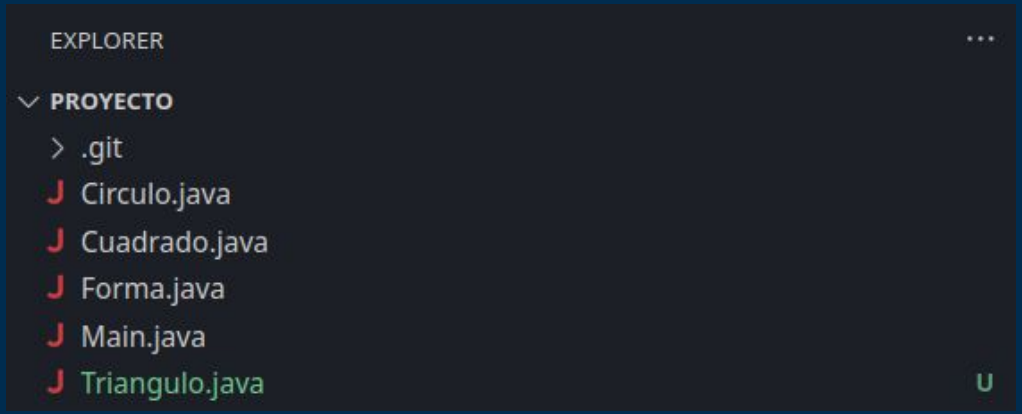
$ git branch <nueva branch>

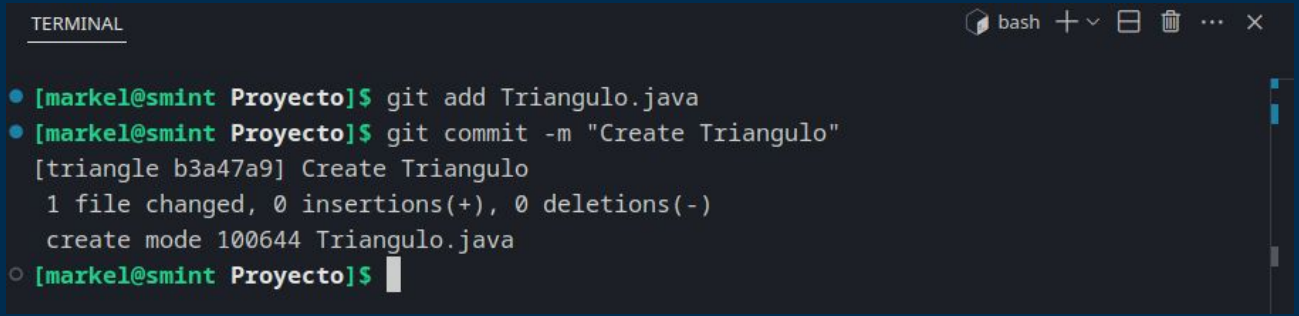
Branches



$ git checkout <branch>

Branches


EXPLORER
PROYECTO
.git
Circulo.java
Cuadrado.java
Forma.java
Main.java
Triangulo.java
U


TERMINAL
bash
[markel@smint Proyecto]$ git add Triangulo.java
[markel@smint Proyecto]$ git commit -m "Create Triangulo"
[triangle b3a47a9] Create Triangulo
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Triangulo.java
[markel@smint Proyecto]$

Branches

Triangulo.java M

Triangulo.java

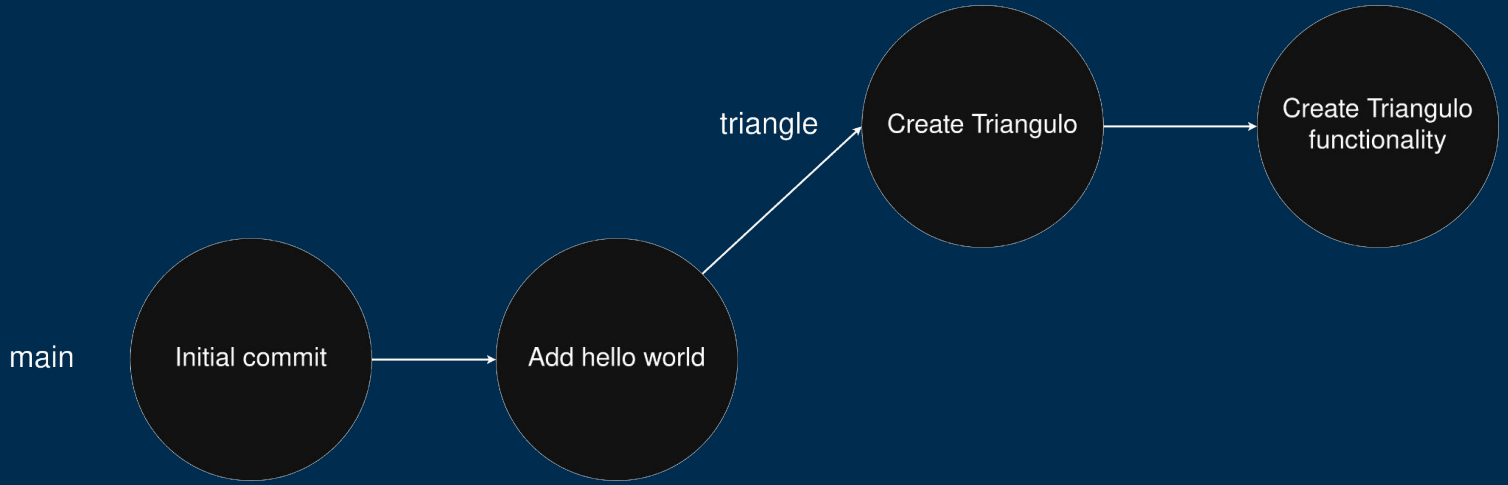
```java
public class Triangulo {

    private float base;
    private float altura;

    public Triangulo(float base, float altura) {
        this.base = base;
        this.altura = altura;
    }
}
```

TERMINAL

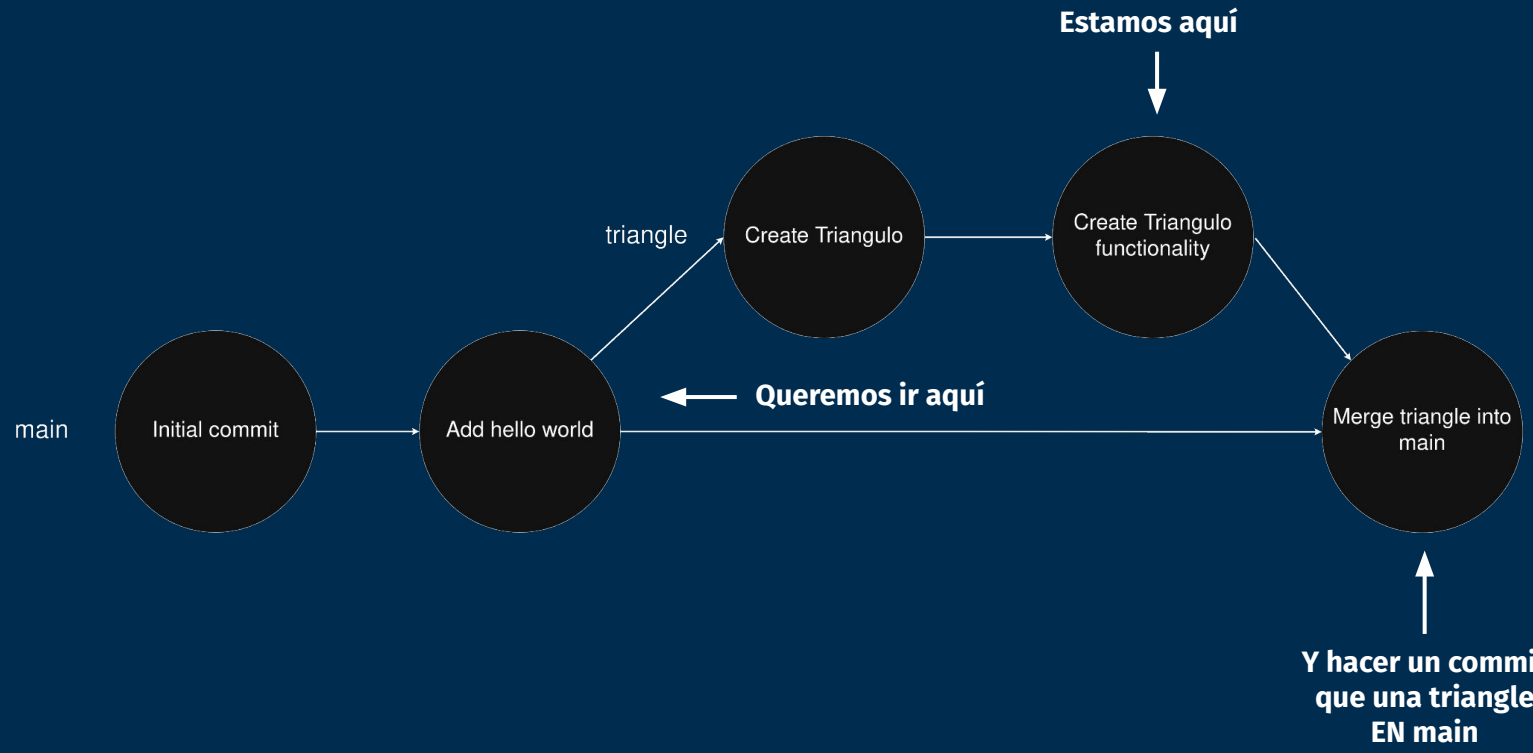
bash

```
[markel@smint Proyecto]$ git add Triangulo.java
[markel@smint Proyecto]$ git commit -m "Create Triangulo functionality"
[triangle f232ba9] Create Triangulo functionality
 1 file changed, 10 insertions(+)
[markel@smint Proyecto]$
```

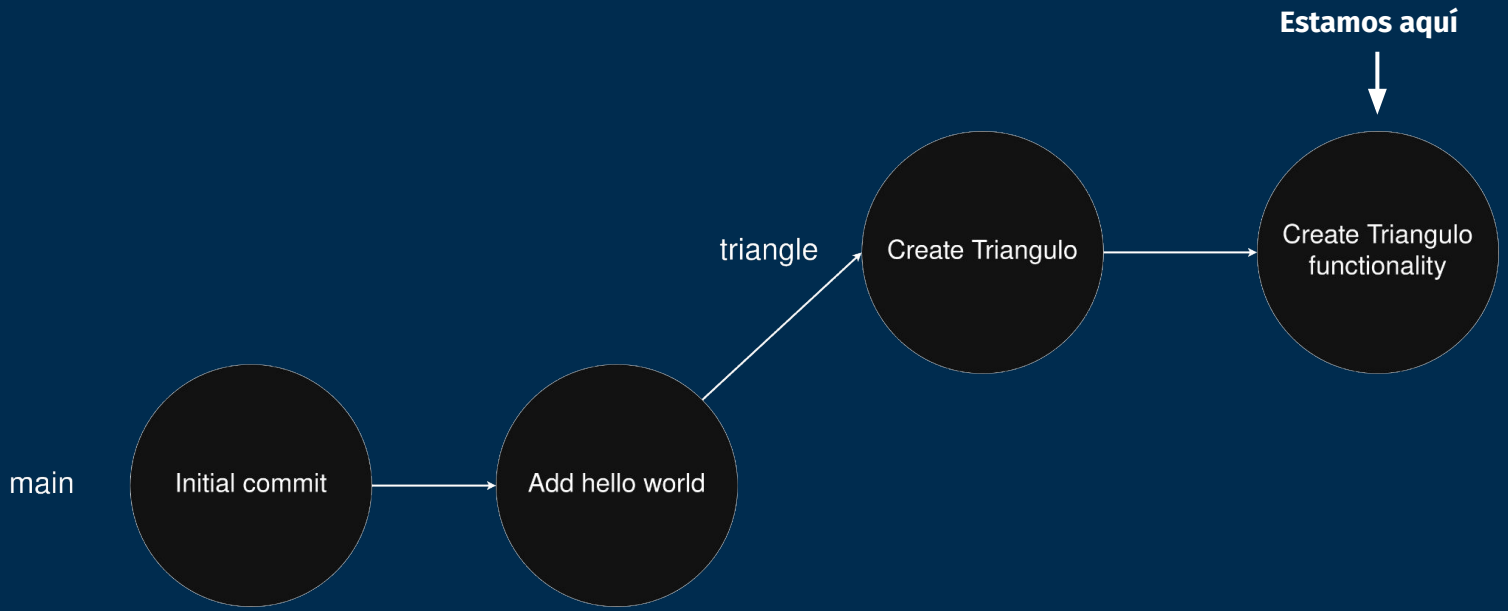
Branches


triangle
Create Triangulo
Create Triangulo functionality
main
Initial commit
Add hello world

Merge

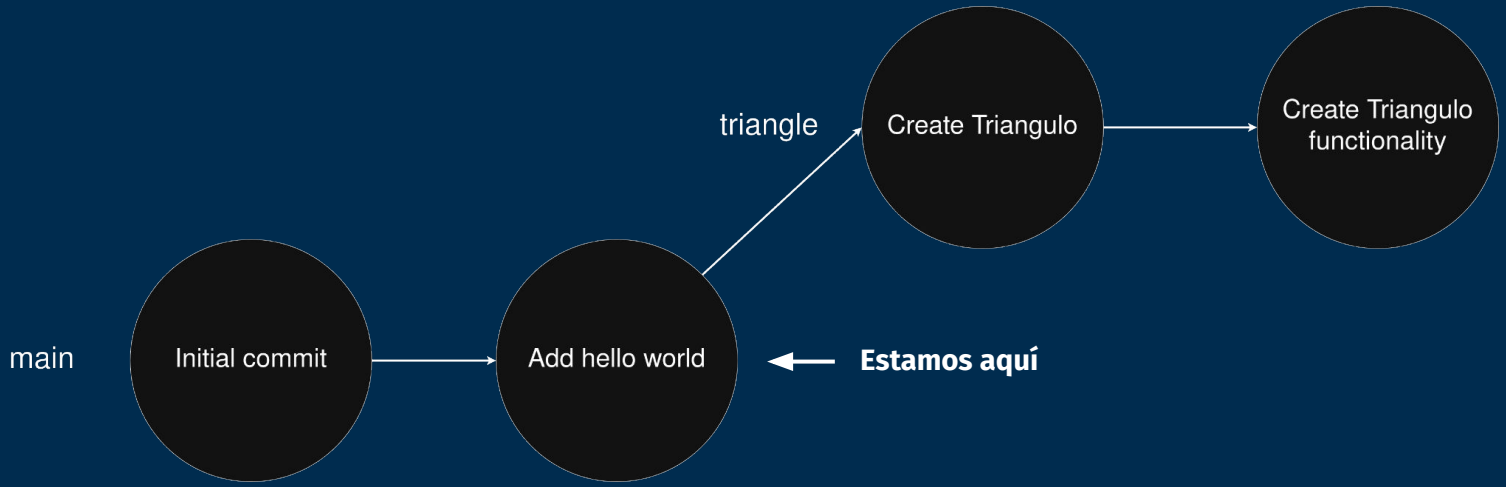

Estamos aquí
triangle
Create Triangulo
Create Triangulo functionality
main
Initial commit
Add hello world
Queremos ir aquí
Merge triangle into main
Y hacer un commit que una triangle EN main

Merge


Estamos aquí
triangle
Create Triangulo
Create Triangulo functionality
main
Initial commit
Add hello world

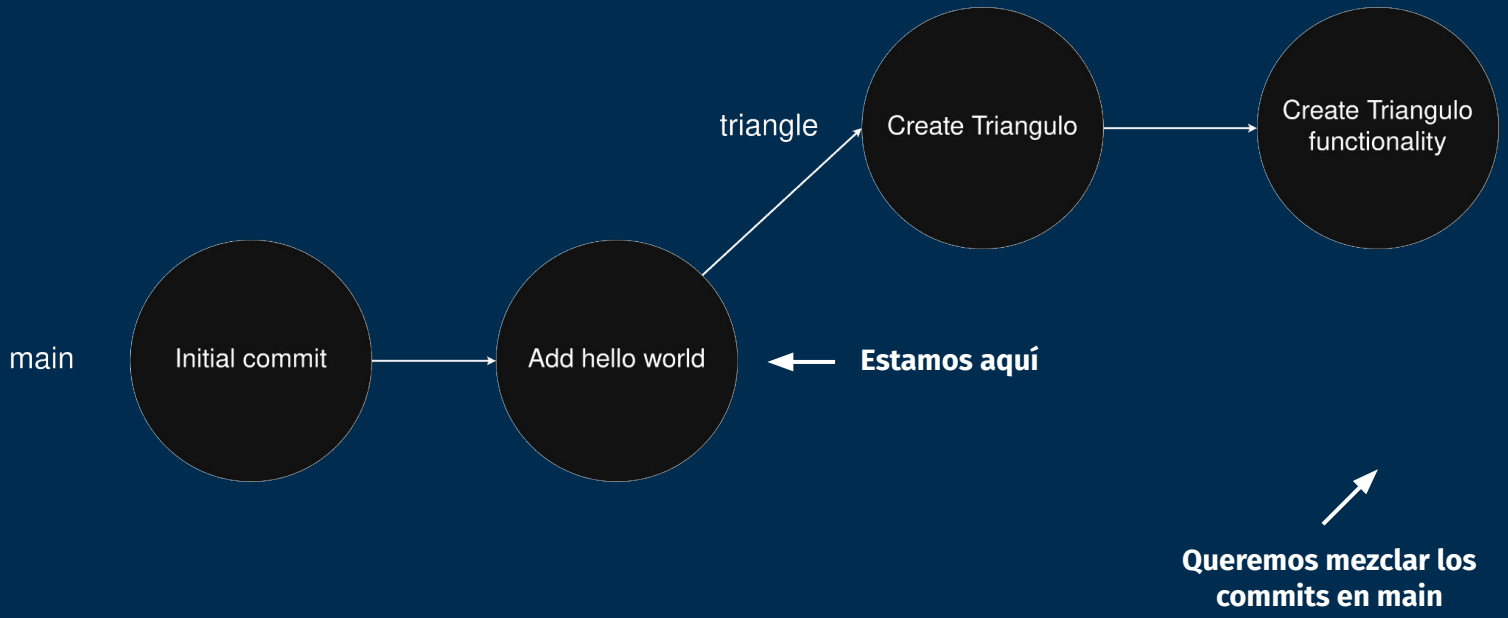
Merge

```
TERMINAL                                                    bash
[markel@smint Proyecto]$ git checkout main
Switched to branch 'main'
[markel@smint Proyecto]$
```

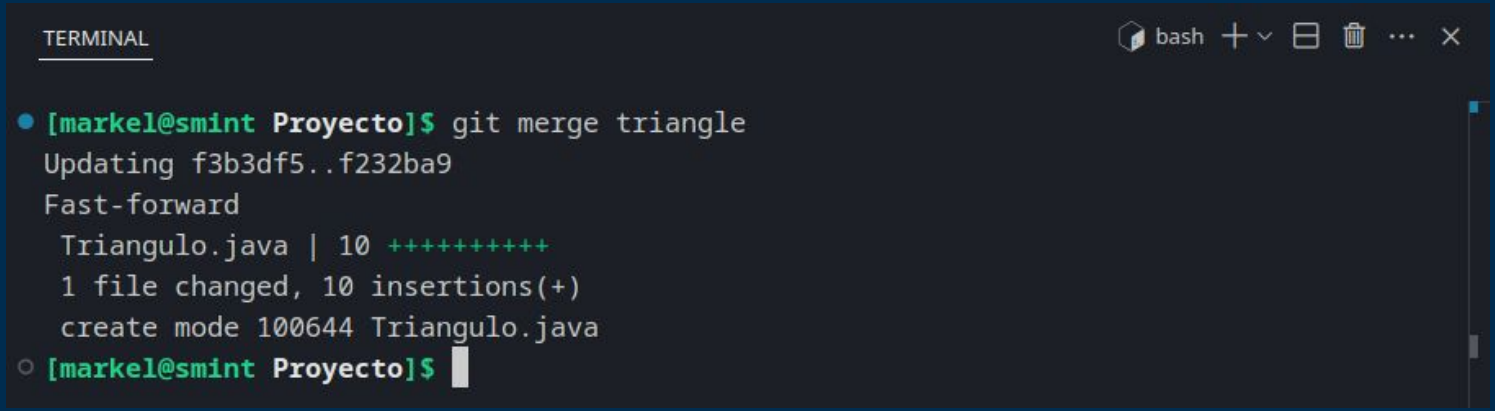


$ git checkout main

Merge


triangle
Create Triangulo
Create Triangulo functionality
main
Initial commit
Add hello world
Estamos aquí
Queremos mezclar los commits en main

Merge



(Estando en la rama destino)

$ git merge <rama origen>

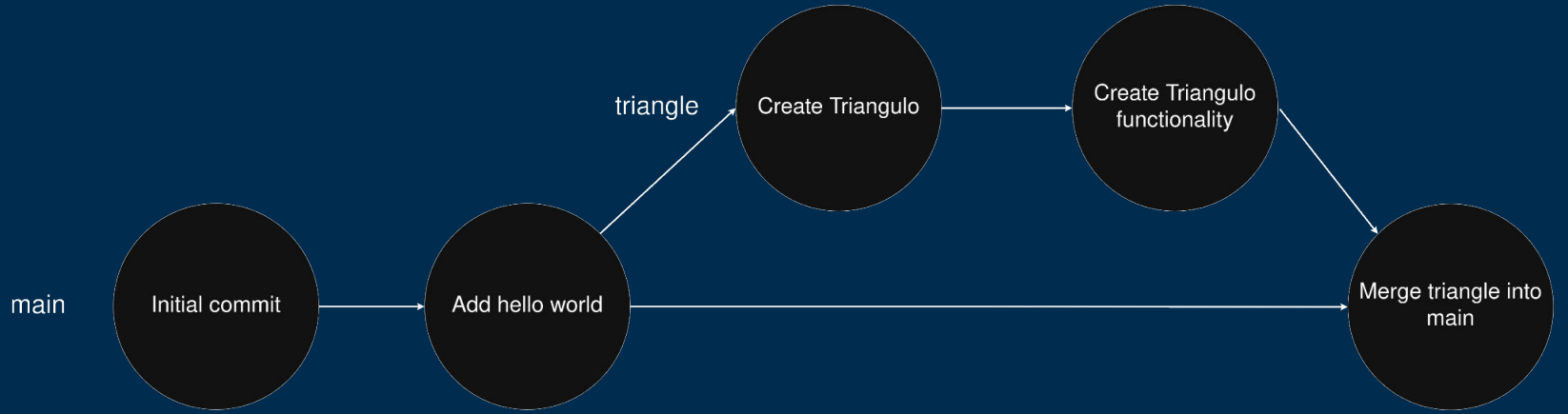
Merge

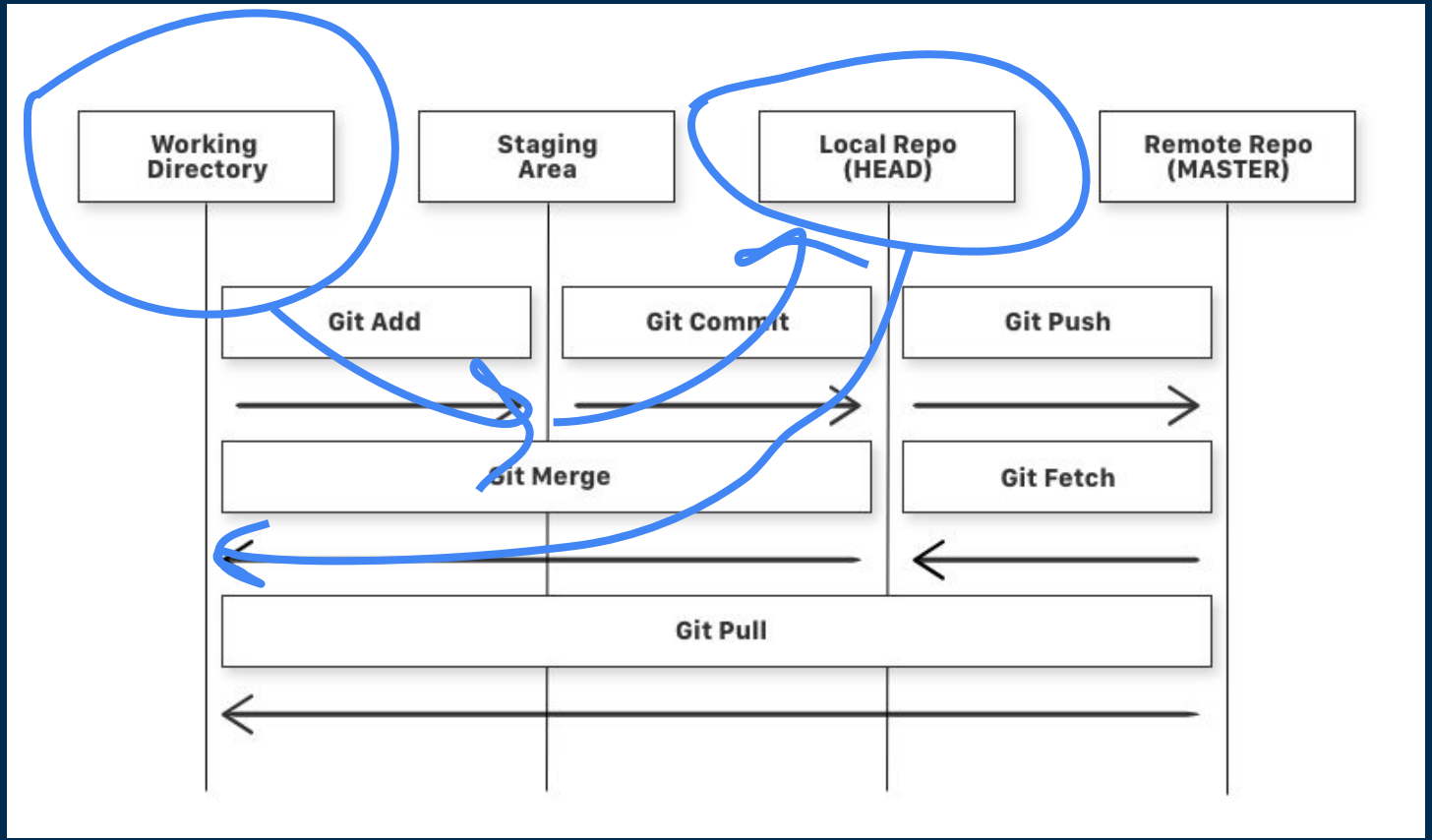
TERMINAL

bash

```
[markel@smint Proyecto]$ git log
commit f232ba9346d4e34eab3aaac60fbd260e51988c10 (HEAD -> triangle, main)
Author: Markel Mencía <markel.mnc@gmail.com>
Date:   Wed Jun 18 01:10:43 2025 +0200

    Create Triangulo functionality

commit b3a47a9429fea7b5e3fdb1823b49e8130b64084e
Author: Markel Mencía <markel.mnc@gmail.com>
Date:   Wed Jun 18 00:58:11 2025 +0200

    Create Triangulo

commit f3b3df552d69670b1d56055524b535d57daaedb6
Author: Markel Mencía <markel.mnc@gmail.com>
Date:   Tue Jun 17 19:13:28 2025 +0200

    Add hello world

commit 284eb79a54fca9094abaa072a3b1b63e00f9a680
Author: Markel Mencía <markel.mnc@gmail.com>
Date:   Tue Jun 17 19:02:25 2025 +0200

    Initial commit
[markel@smint Proyecto]$
```

Merge


triangle
Create Triangulo
Create Triangulo functionality
main
Initial commit
Add hello world
Merge triangle into main


Working Directory
Staging Area
Local Repo (HEAD)
Remote Repo (MASTER)
Git Add
Git Commit
Git Push
Git Merge
Git Fetch
Git Pull


Tenemos que establecer esta conexión
Repositorio Local
Repositorio Remoto
GitHub
Y crear el repositorio remoto

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner *

markelmencia /

Repository name *

Proyecto

Proyecto is available.

Great repository names are short and memorable. Need inspiration? How about **redesigned-parakeet** ?

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.

Initialize this repository with:

- [ ] Add a README file
  This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

You are creating a public repository in your personal account.

Create repository

Proyecto Public

Pin | Watch 0 | Fork 0 | Star 0

Set up GitHub Copilot

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Get started with GitHub Copilot

Add collaborators to this repository

Search for people using their GitHub username or email address.

Invite collaborators

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

HTTPS SSH https://github.com/markelmencia/Proyecto.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

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

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

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

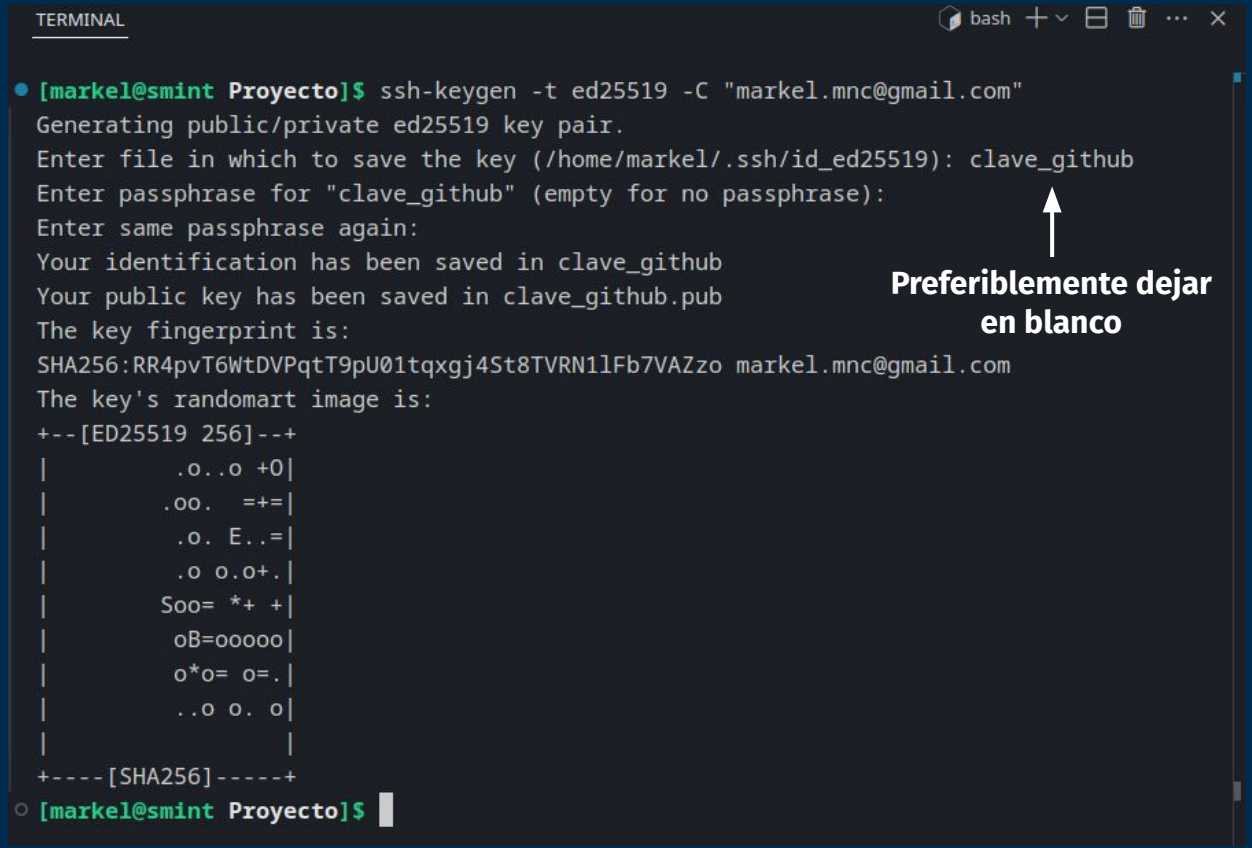
```
git remote add origin https://github.com/markelmencia/Proyecto.git
git branch -M main
git push -u origin main
```

ProTip! Use the URL for this page when adding GitHub as a remote.

ssh

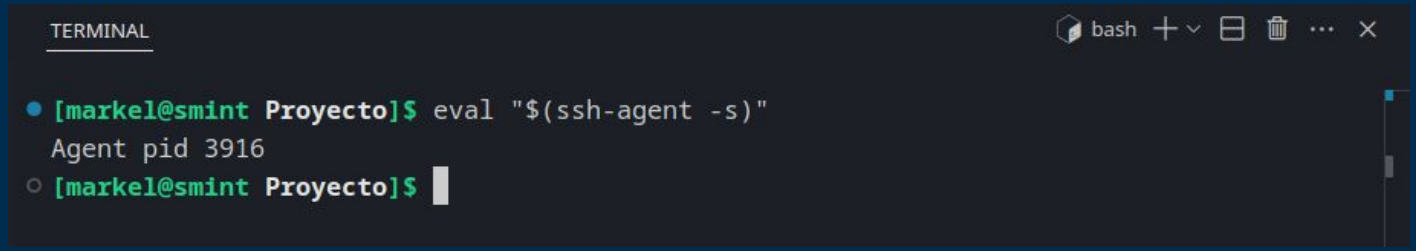
GitHub

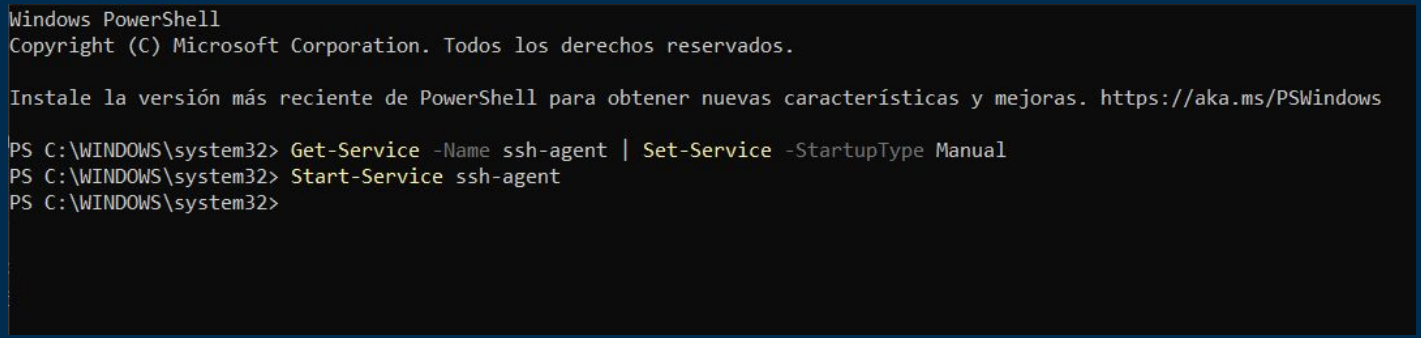
Generar par de claves



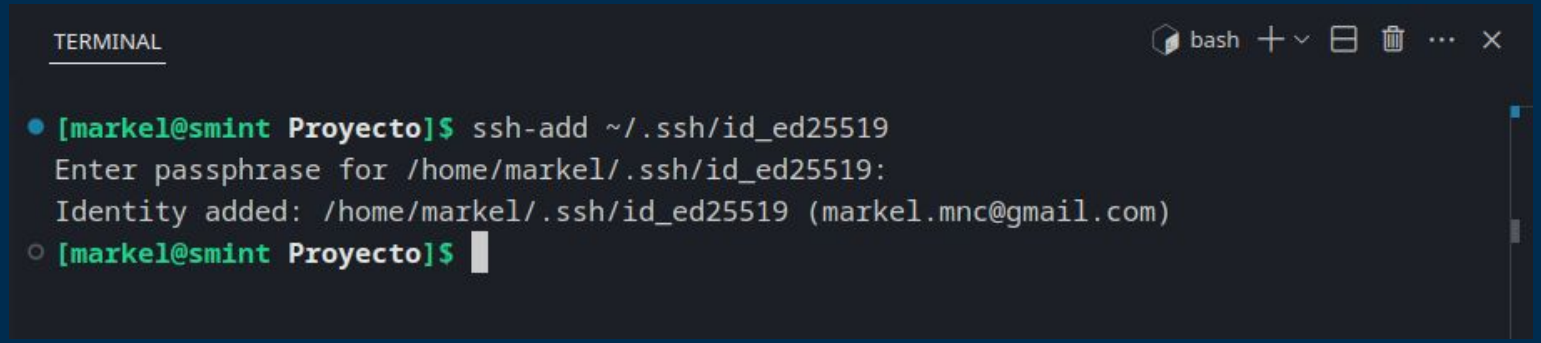
$ ssh-keygen -t ed25519 -C <email>

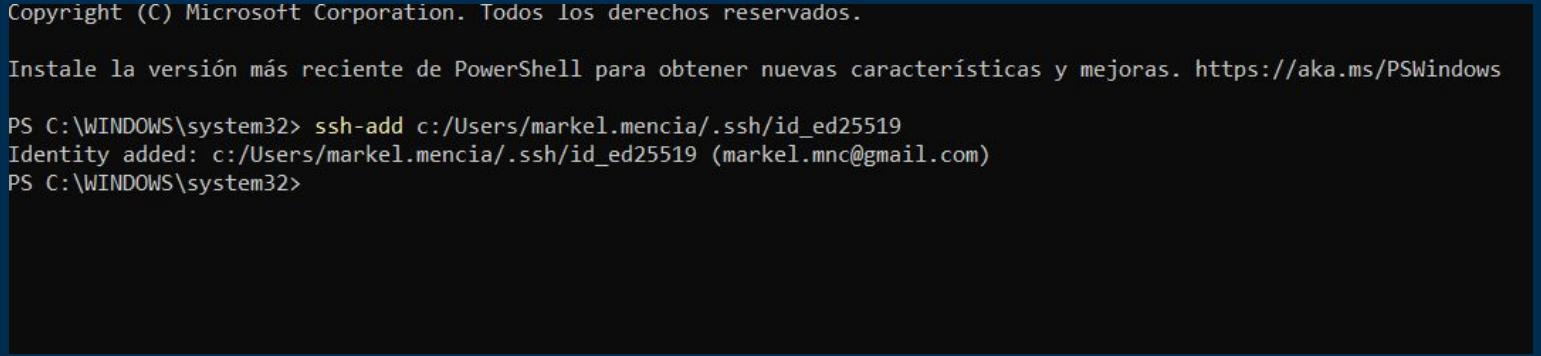
Iniciar el agente ssh





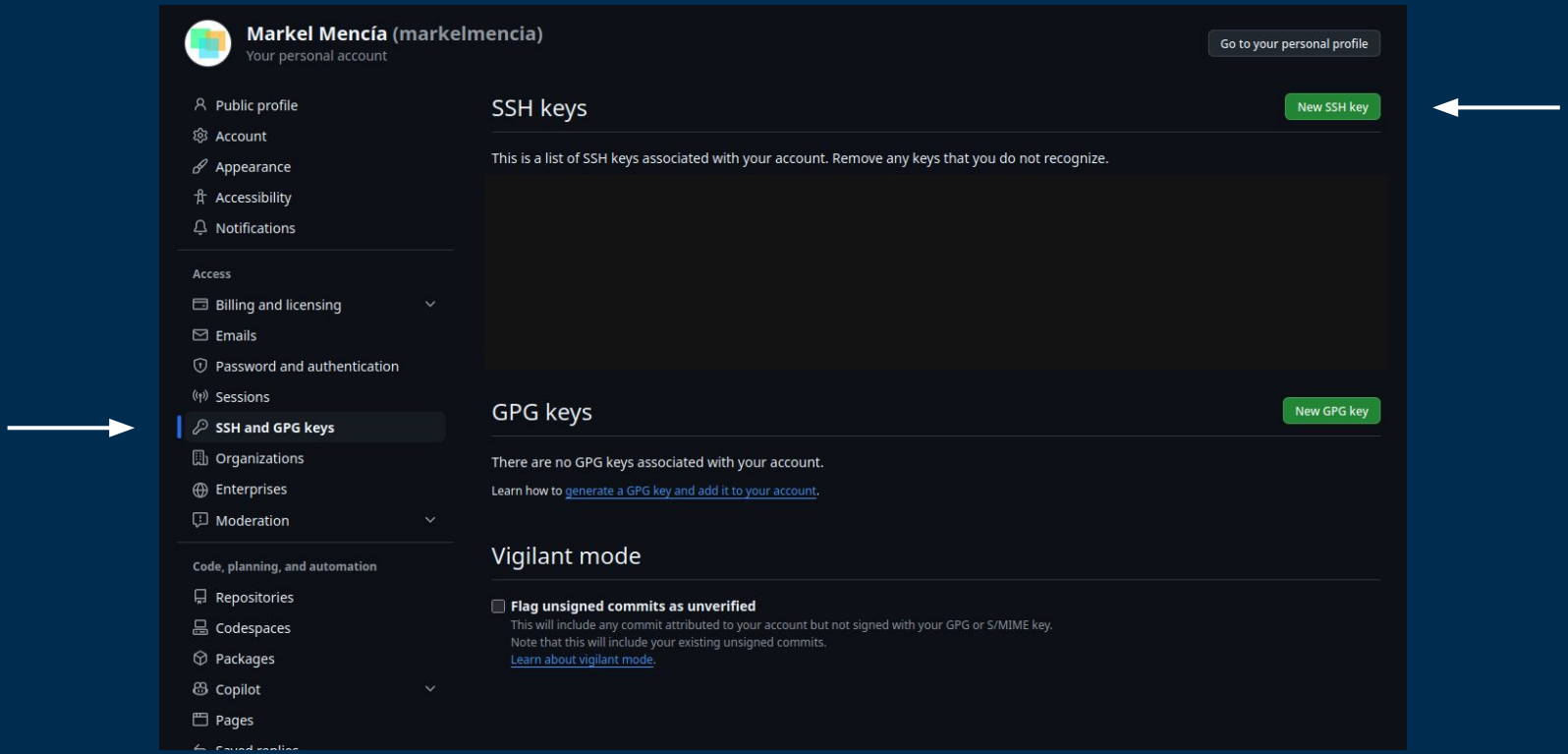
Añadir el par de claves a ssh





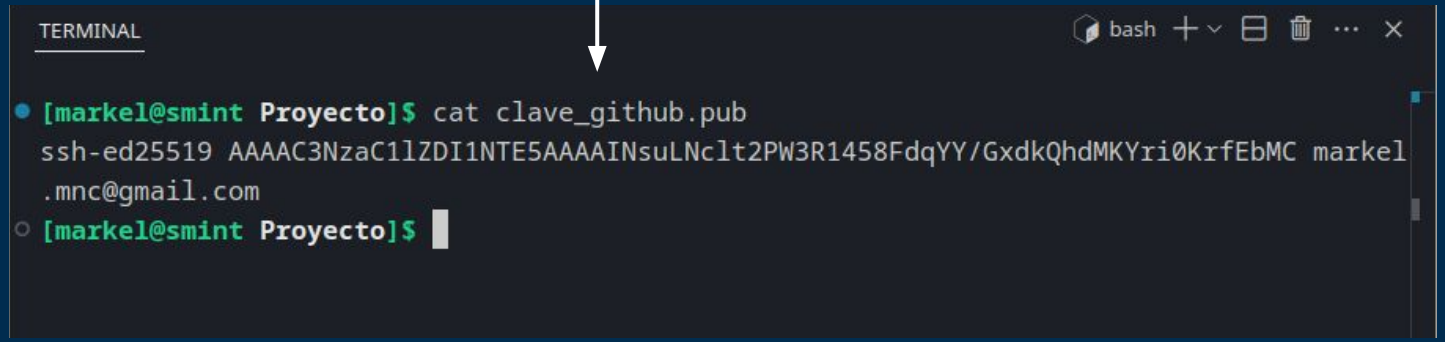
$ ssh-add ~/.ssh/id_ed25519”

Añadir la clave pública a GitHub



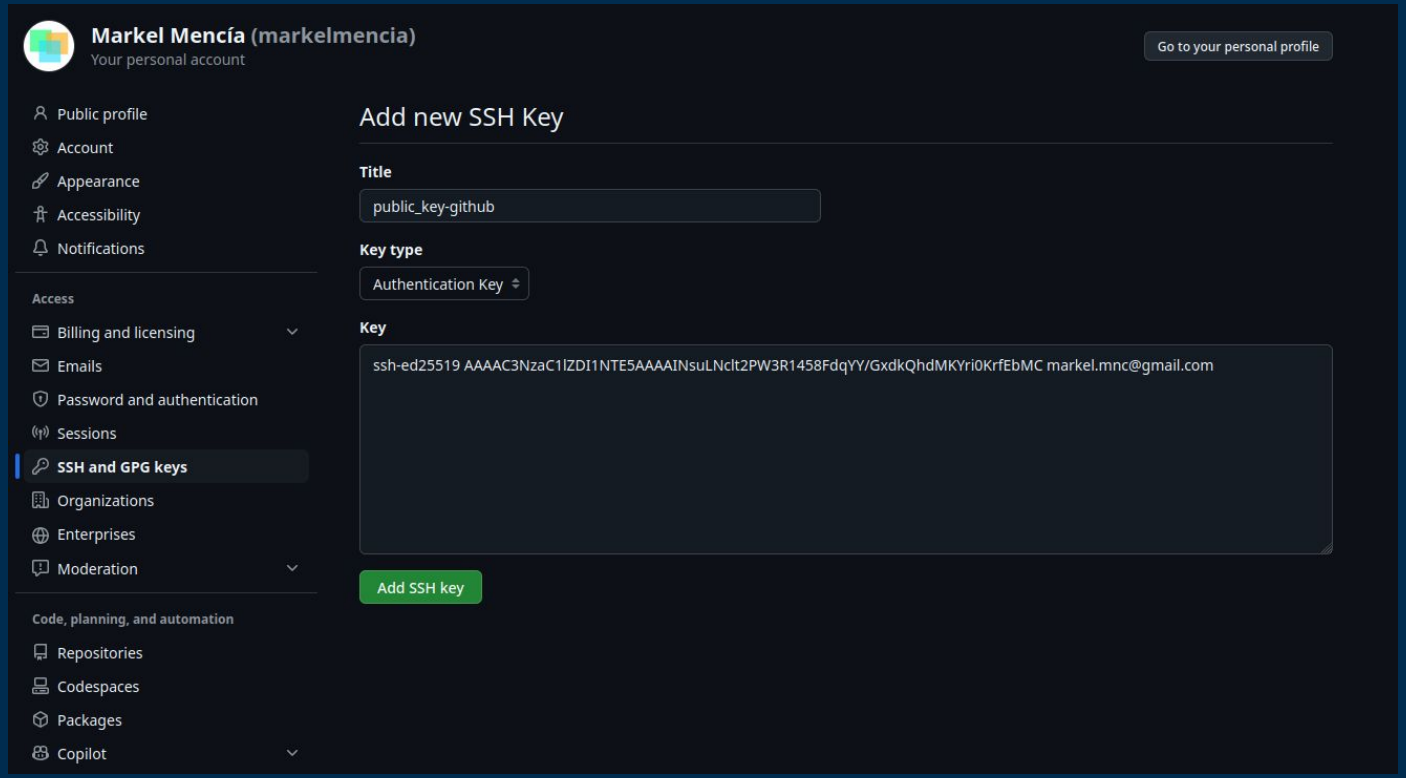
Añadir la clave pública a GitHub

~/.ssh/id_ed25519.pub



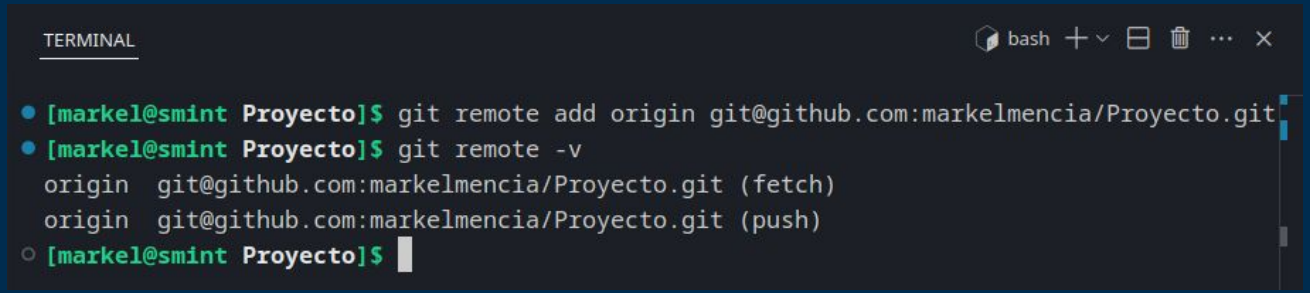
$ cat ~/.ssh/id_ed25519.pub

Añadir la clave pública a GitHub



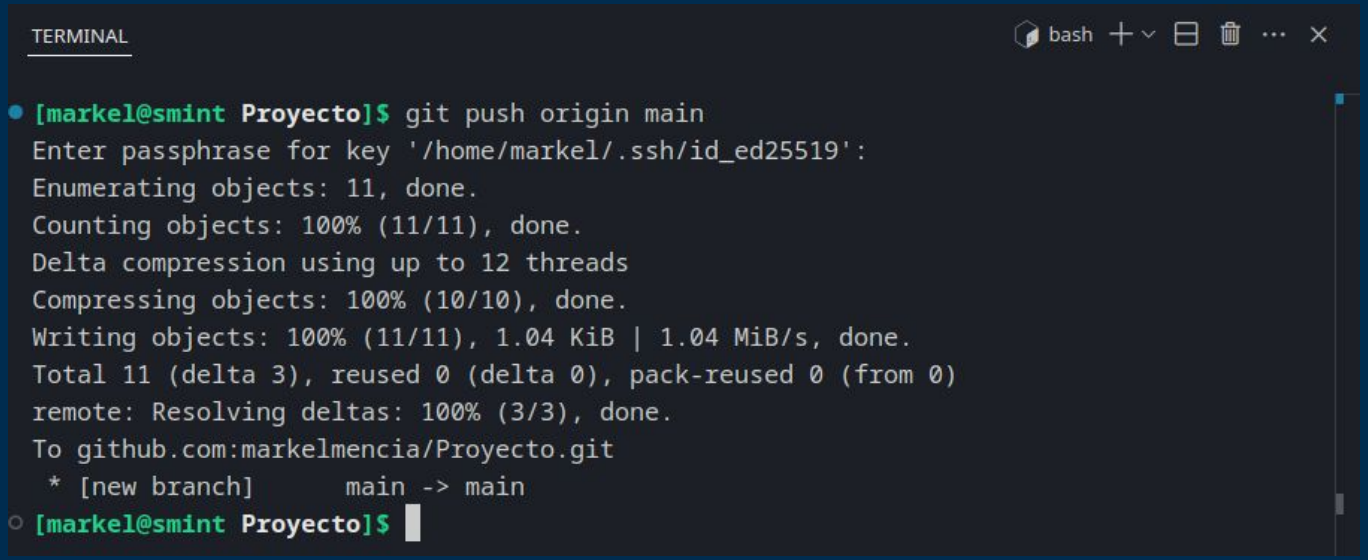

Ya tenemos la conexión ssh establecida
Repositorio Local
Repositorio Remoto
GitHub
Y hemos creado el repositorio remoto

Vincular repositorio local con remoto

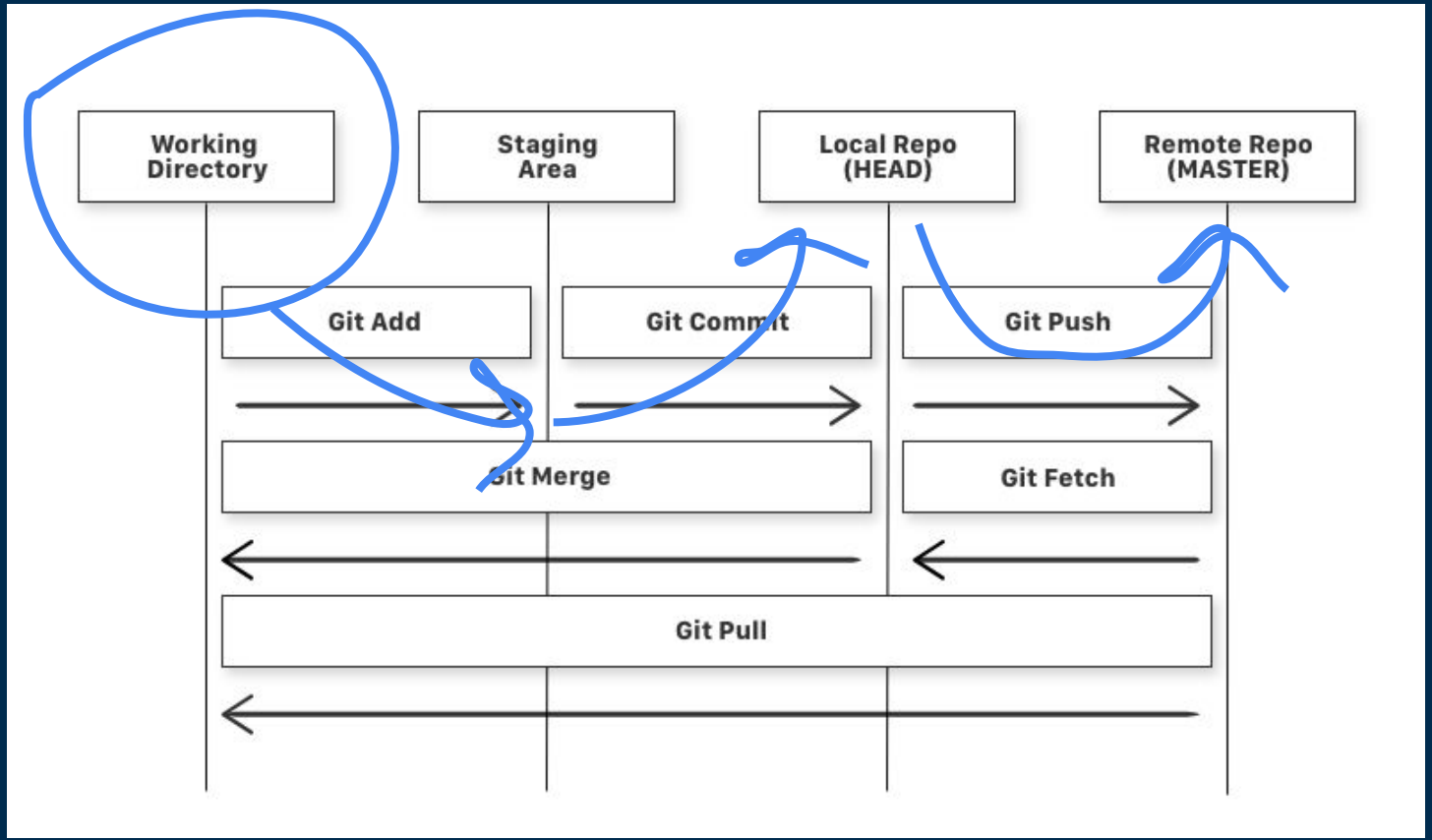


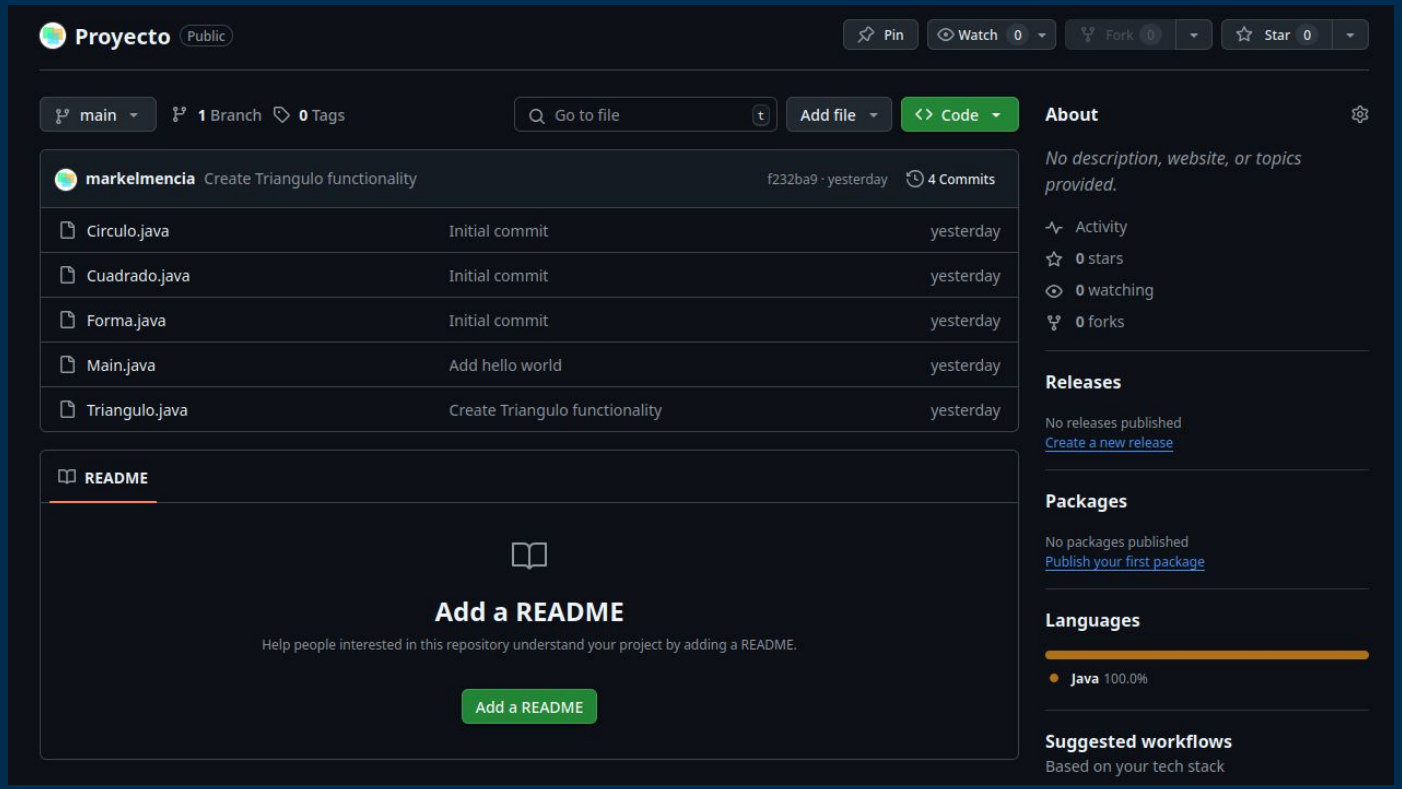
$ git remote add origin <enlace ssh>

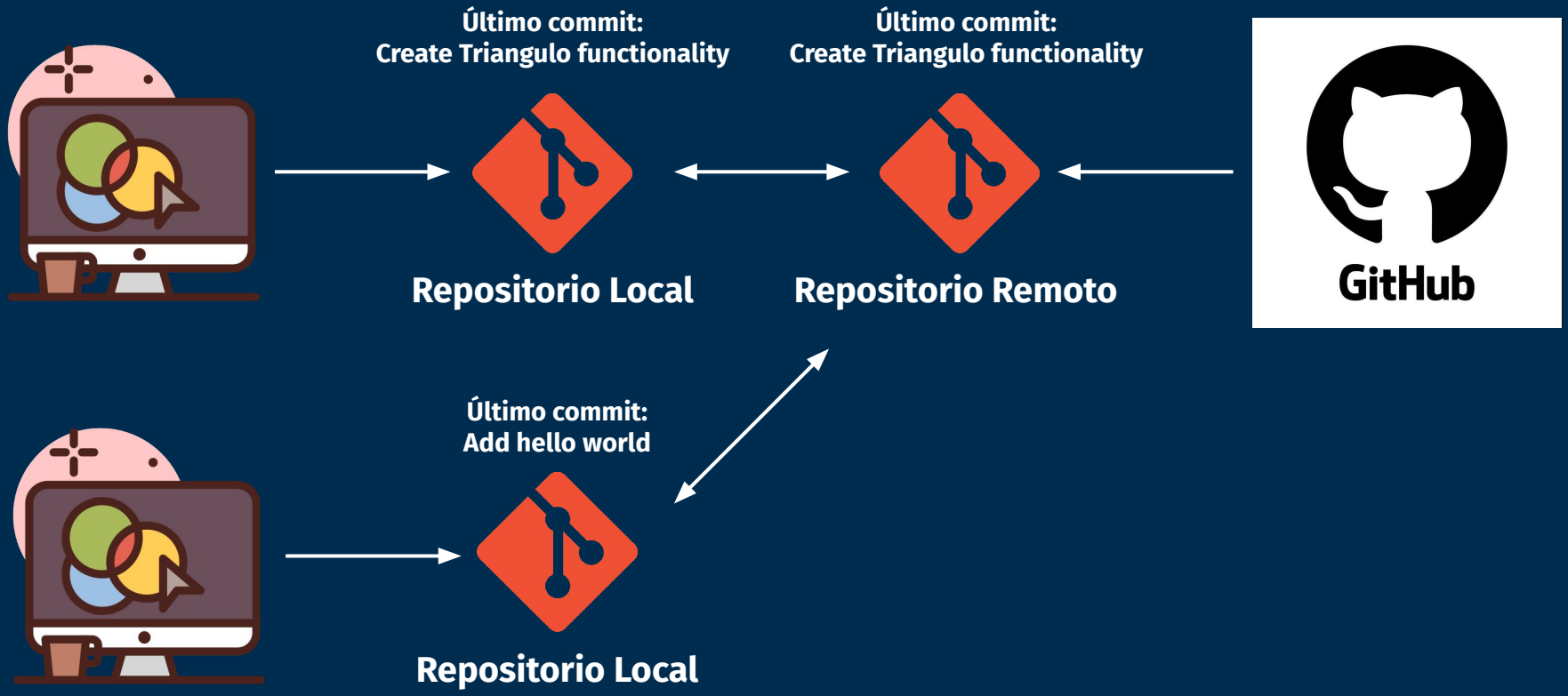
Push

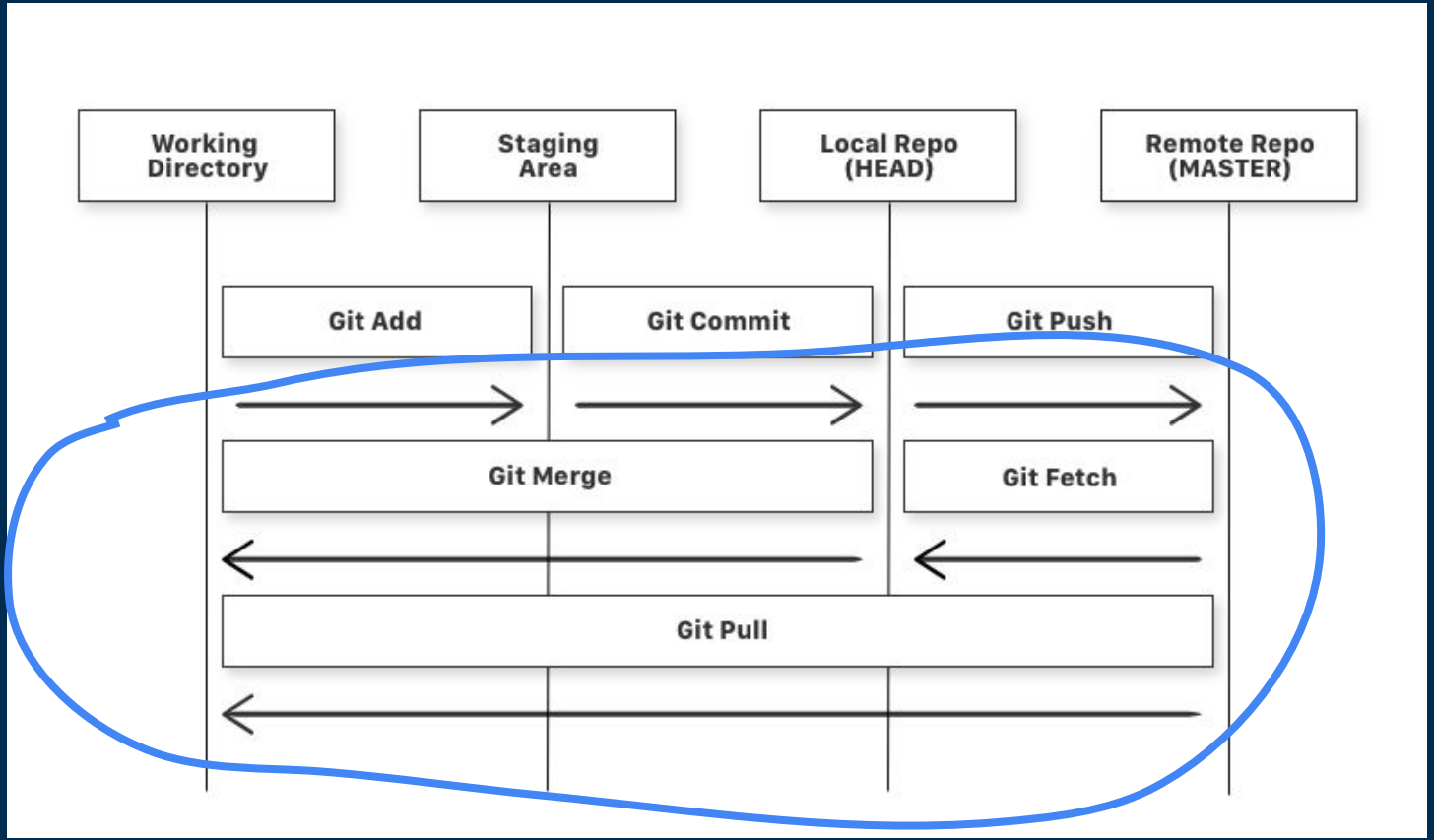


$ git push origin <branch>


Working Directory
Staging Area
Local Repo (HEAD)
Remote Repo (MASTER)
Git Add
Git Commit
Git Push
Git Merge
Git Fetch
Git Pull


Proyecto
Public
Pin
Watch 0
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
t
Add file
Code
markelmencia Create Triangulo functionality
f232ba9 · yesterday
4 Commits
Circulo.java Initial commit yesterday
Cuadrado.java Initial commit yesterday
Forma.java Initial commit yesterday
Main.java Add hello world yesterday
Triangulo.java Create Triangulo functionality yesterday
README
Add a README
Help people interested in this repository understand your project by adding a README.
Add a README
About
No description, website, or topics provided.
Activity
0 stars
0 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Languages
Java 100.0%
Suggested workflows
Based on your tech stack


Último commit:
Create Triangulo functionality
Último commit:
Create Triangulo functionality
Repositorio Local
Repositorio Remoto
GitHub
Último commit:
Add hello world
Repositorio Local


Working Directory
Staging Area
Local Repo (HEAD)
Remote Repo (MASTER)
Git Add
Git Commit
Git Push
Git Merge
Git Fetch
Git Pull

Fetch

```
TERMINAL                                                    bash
[markel@smint Proyecto]$ git fetch
Enter passphrase for key '/home/markel/.ssh/id_ed25519':
[markel@smint Proyecto]$
```

Local pregunta a Remoto si ha habido actualizaciones

Repositorio Local

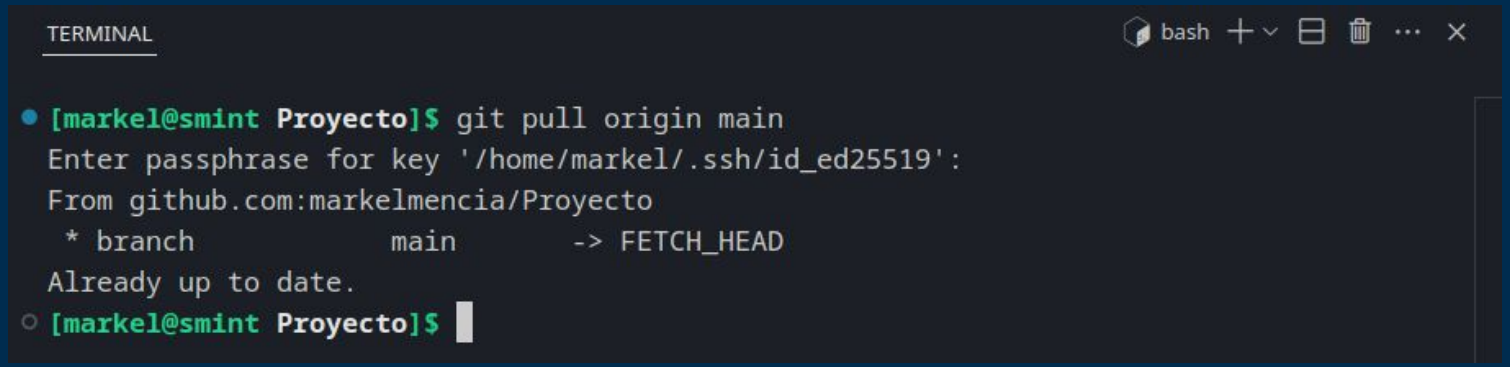
Repositorio Remoto

GitHub

```
TERMINAL                                                    bash
[markel@smint Proyecto]$ git merge origin/main
Already up to date.
[markel@smint Proyecto]$
```

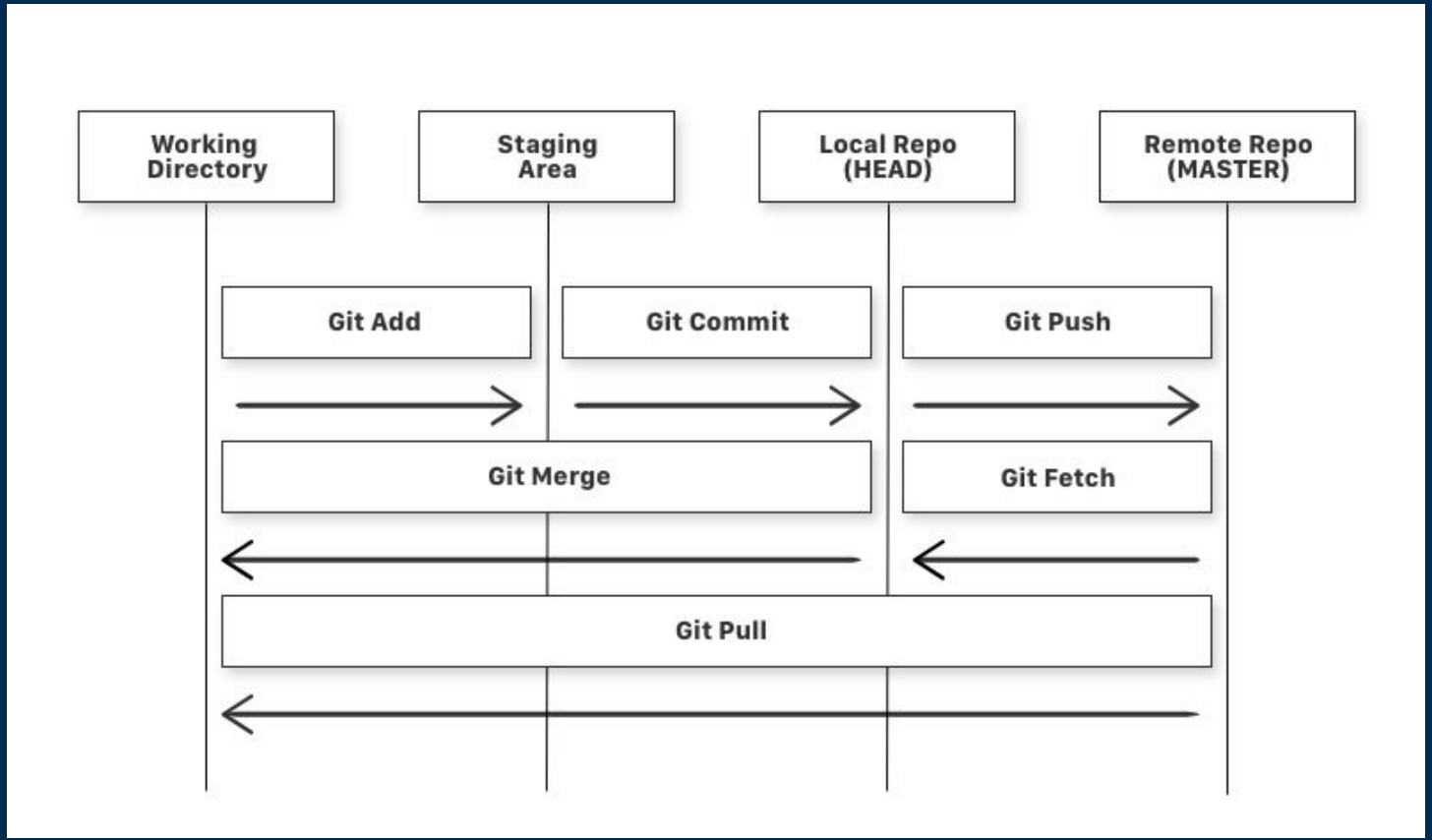
$ git fetch
$ git merge origin/<branch>

Pull



Pull = Fetch + Merge

$ git pull origin <branch>

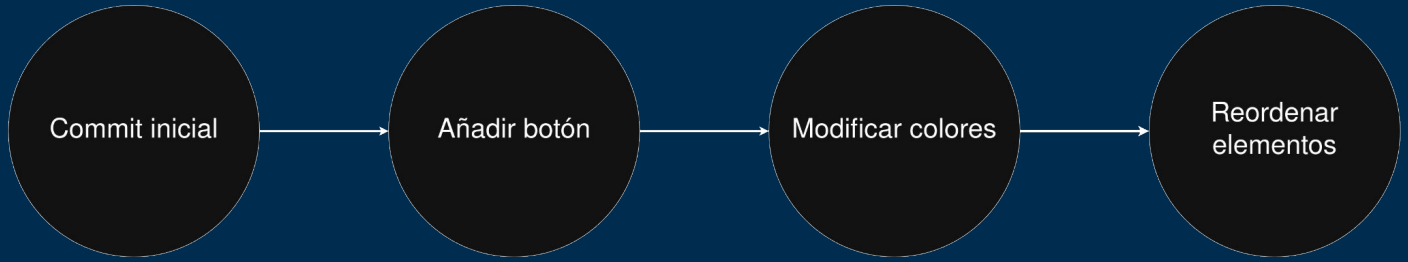

Working Directory
Staging Area
Local Repo (HEAD)
Remote Repo (MASTER)
Git Add
Git Commit
Git Push
Git Merge
Git Fetch
Git Pull

Buenas prácticas

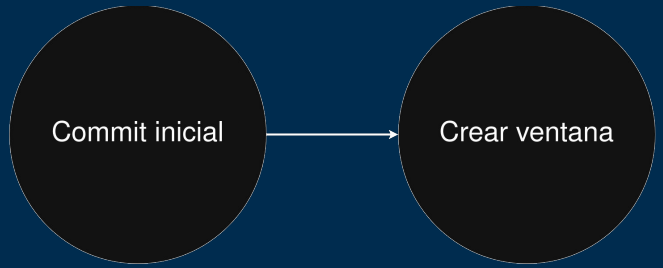
(Y recomendaciones)

Commits cortos y específicos

Do:

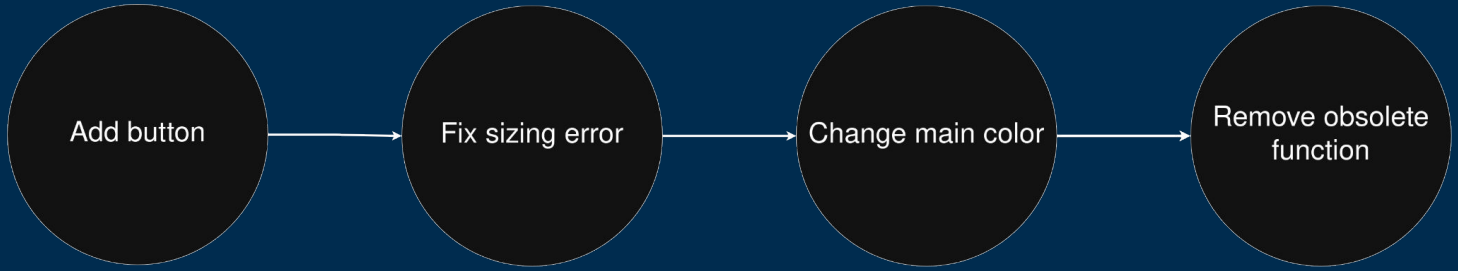


Don't:

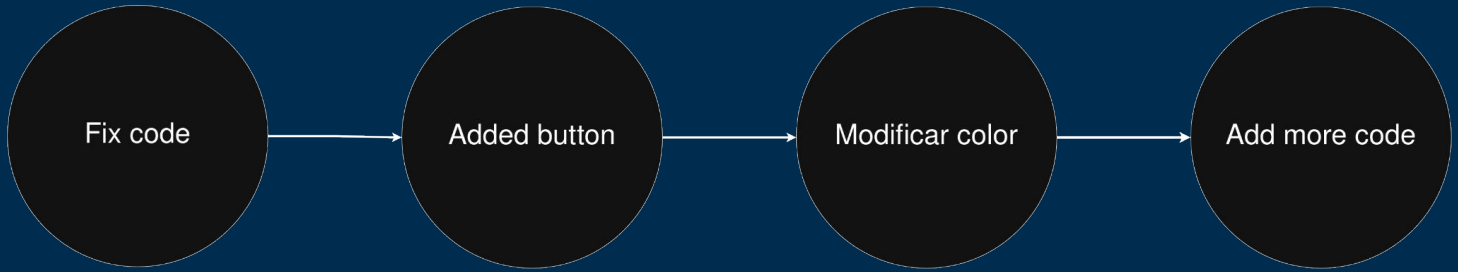


Mensajes de commit estructurados

Do:



Don’t:



Solo pushear lo necesario

Pushea:

- Código fuente
- Ficheros complementarios al código fuente
- Documentación
- Tests
- Ejemplos

No pushees:

- Ejecutables
- Ficheros de bases de datos
- Carpetas de IDEs
- Logs / ficheros temporales
- Pares de claves / claves de APIs (porfi)

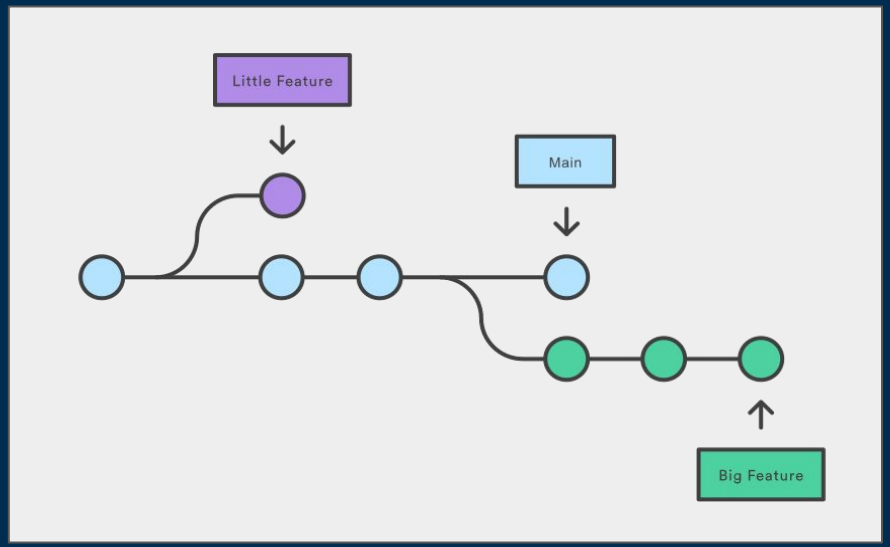
Utilizar un .gitignore

.gitignore

```
# Ejecutables
*.exe
*.bin
*.out

# Logs
logs/
*.log

# Directorio de binarios
bin/

# Directorios que crean los IDEs
.idea/
.vscode/

# Ficheros de configuración
*.conf
*.env

```

Estructurar branches

Do:



Don’t:

Escribir código legible

```go
func main() {
    // Setup sockets
    var sock, tlssock net.Listener
    sockets := 1

    sock = setupConn()
    if useTLS {
        tlssock = setupTLSConn()
        sockets += 1
    }

    // Set up database logging file only
    // if the logging level is INFO or more
    var dblog *stdlog.Logger
    if log.Level >= log.INFO {
        f := logFile()
        defer f.Close()
        dblog = stdlog.New(f, "", stdlog.LstdFlags)
    }

    // Setup database
    database := db.Connect(dblog)
    sqldb, _ := database.DB()
    defer sqldb.Close()
```

Documentar repositorio

Do:

README | Code of conduct | MIT license | Security

React • license MIT | npm v19.1.0 | (Runtime) Build and Test failing | (Compiler) TypeScript passing | PRs welcome

React is a JavaScript library for building user interfaces.

- **Declarative:** React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes. Declarative views make your code more predictable, simpler to understand, and easier to debug.
- **Component-Based:** Build encapsulated components that manage their own state, then compose them to make complex UIs. Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep the state out of the DOM.
- **Learn Once, Write Anywhere:** We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code. React can also render on the server using Node and power mobile apps using React Native.

Learn how to use React in your project.

Installation

React has been designed for gradual adoption from the start, and **you can use as little or as much React as you need**:

- Use Quick Start to get a taste of React.
- Add React to an Existing Project to use as little or as much React as you need.
- Create a New React App if you're looking for a powerful JavaScript toolchain.

Documentation

You can find the React documentation on the website.

Check out the Getting Started page for a quick overview.

The documentation is divided into several sections:

- Quick Start
- Tutorial
- Thinking in React
- Installation
- Describing the UI

Don’t:

qr Public | Watch 1

main | 1 Branch | 0 Tags | Go to file t | Add file | Code

Markiiuz Updated README.md — 75b2e37 · 2 years ago — 4 Commits

README.md	Updated README.md	2 years ago
qr.py	Test functions added	2 years ago

README

QR Code generator

This program will allow you to create qr codes easily

¡Gracias!