

Summer School 2025

Repositorios de git

Qué son, cómo usarlos y buenas prácticas

Markel Mencía











¿Cómo funciona?

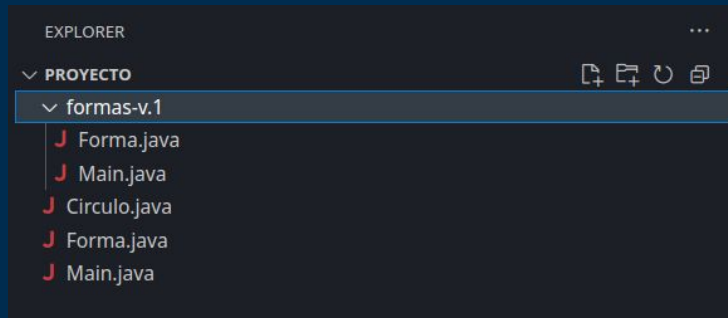


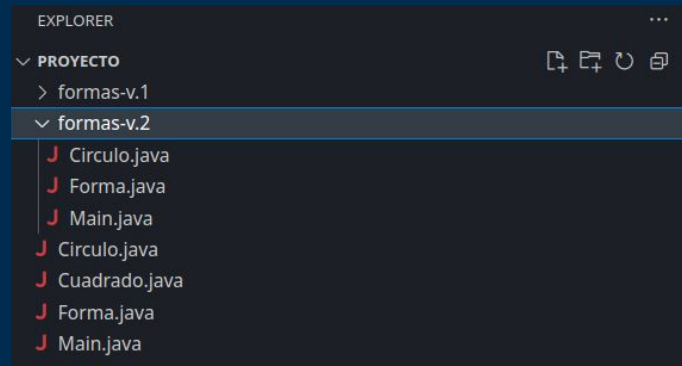
EXPLORER

▼ **PROYECTO**

 Forma.java

 Main.java





Commit inicial



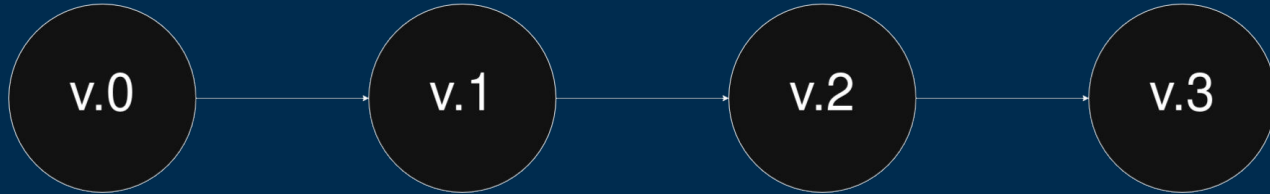
Añadir Circulo



Añadir Cuadrado



Añadir Triángulo



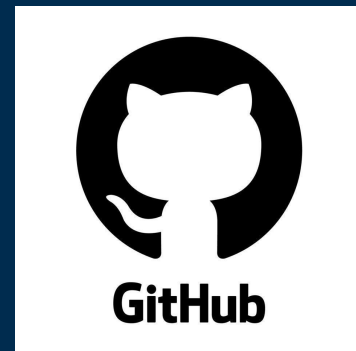


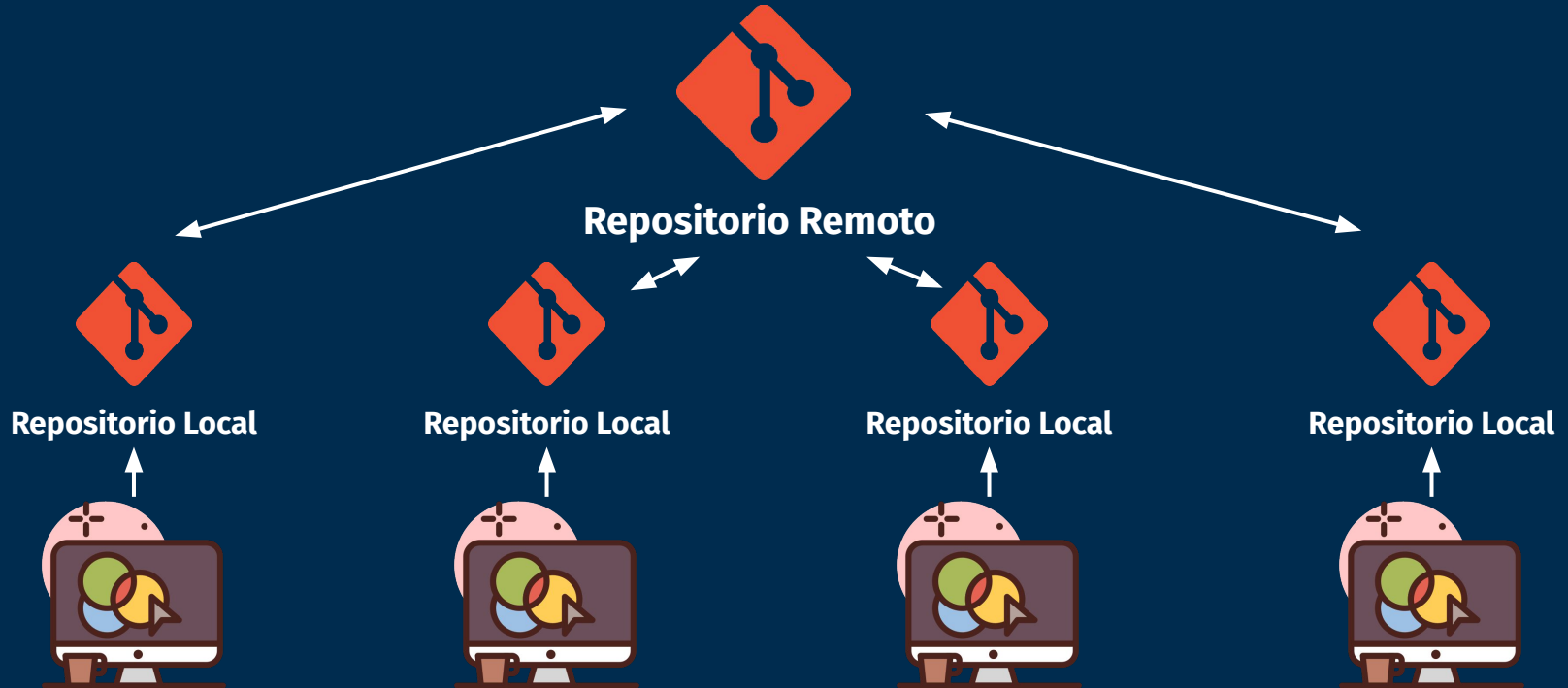


Repositorio Local

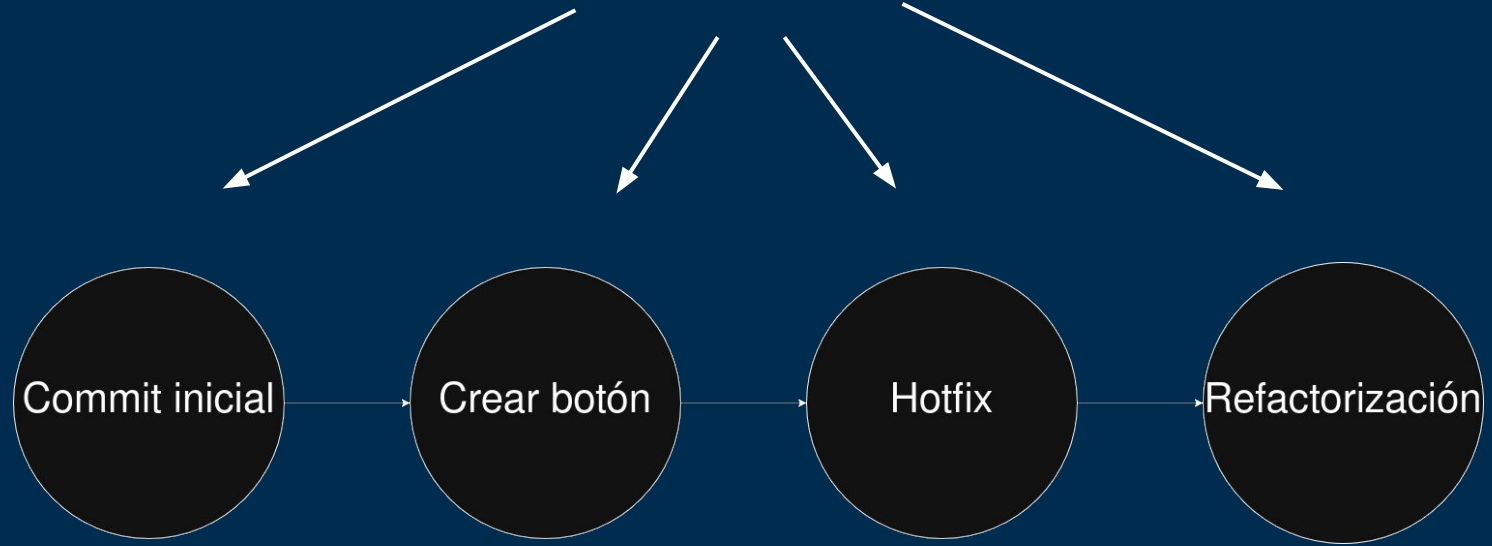


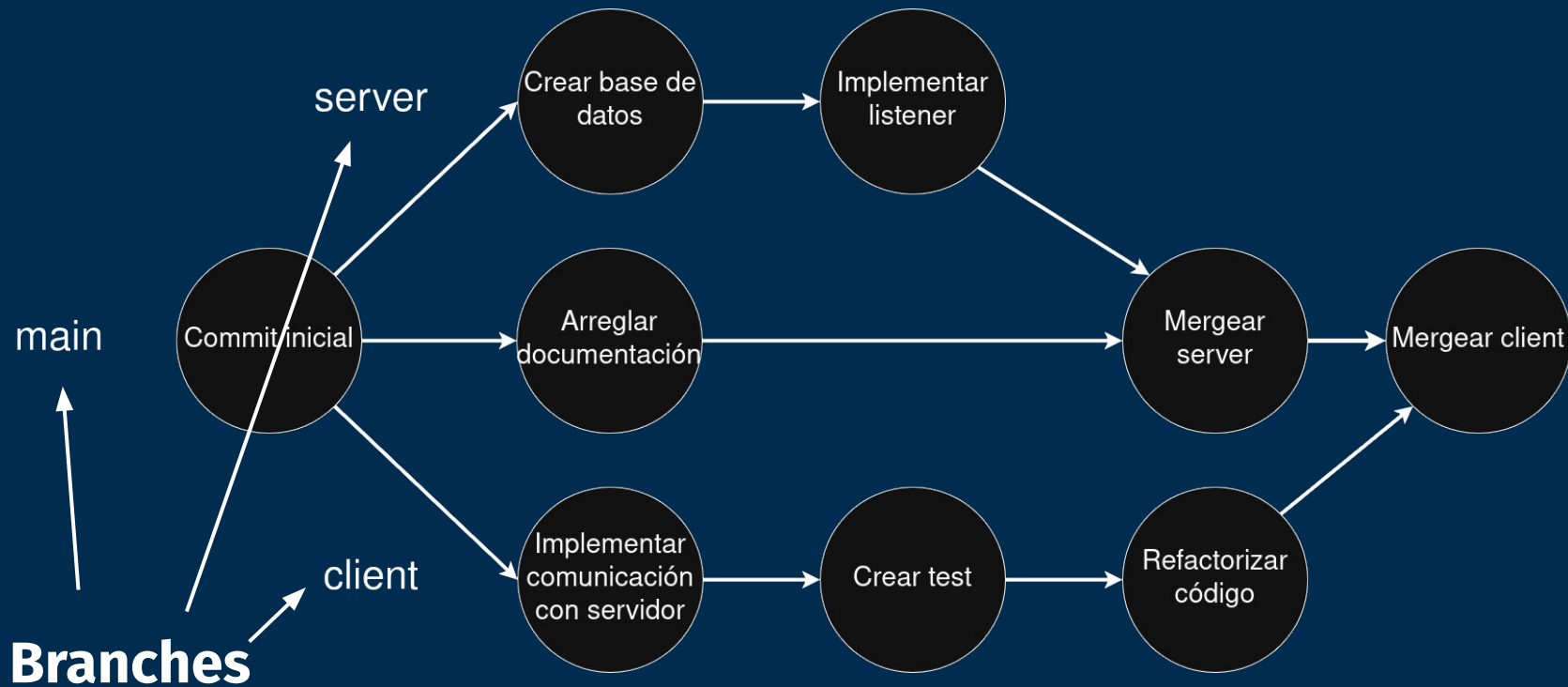
Repositorio Remoto





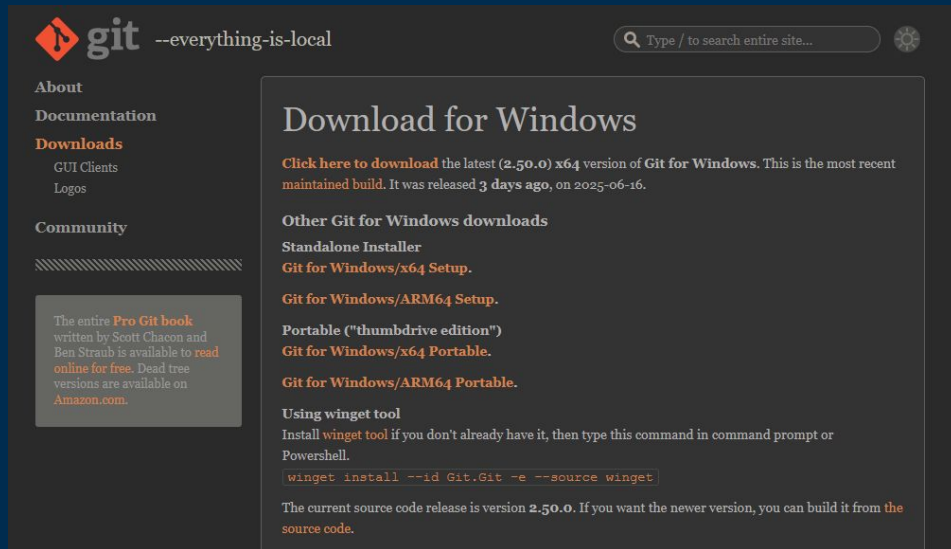
Commits







Instalación

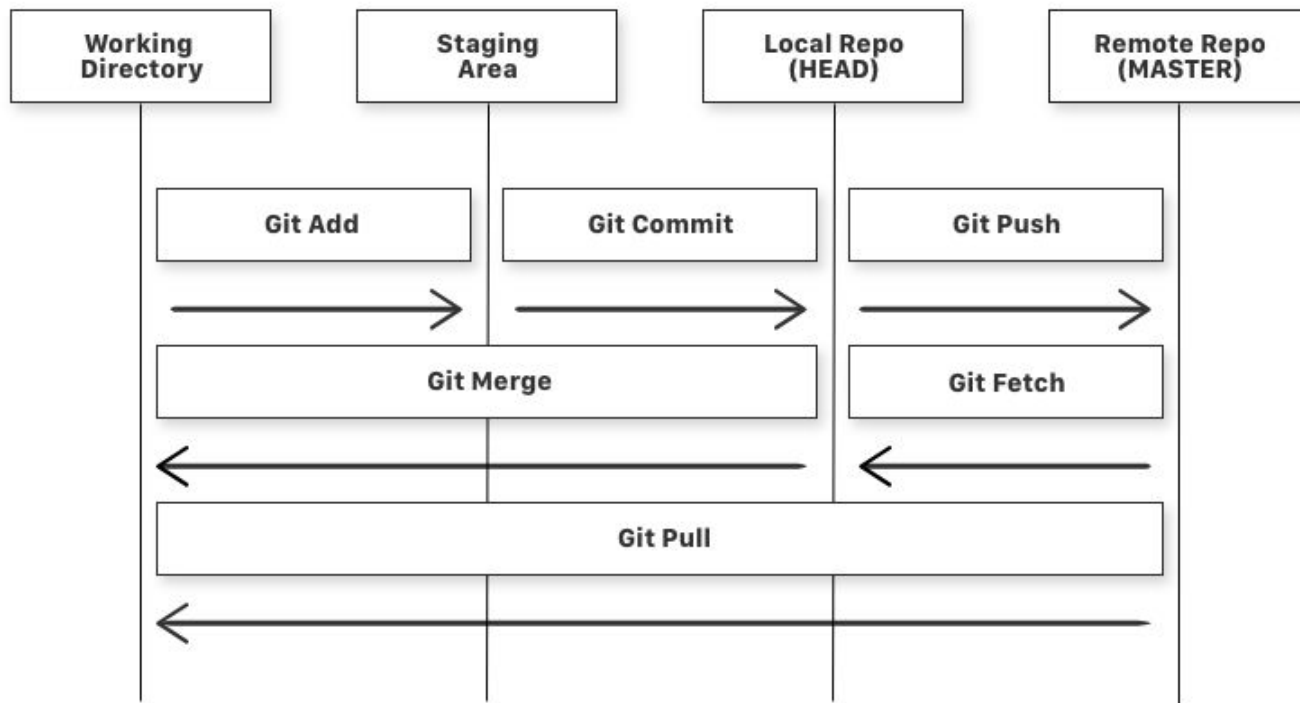


The screenshot shows the Git website's 'Download for Windows' page. The header includes the Git logo and the tagline '--everything-is-local'. A search bar is present with the placeholder text 'Type / to search entire site...'. The left sidebar contains navigation links: 'About', 'Documentation', 'Downloads' (highlighted), 'GUI Clients', 'Logos', and 'Community'. Below these links is a decorative separator line. A small box in the sidebar contains text about the 'Pro Git book'. The main content area is titled 'Download for Windows' and contains the following text: 'Click here to download the latest (2.50.0) x64 version of Git for Windows. This is the most recent maintained build. It was released 3 days ago, on 2025-06-16.' Below this, there is a section titled 'Other Git for Windows downloads' which lists: 'Standalone Installer', 'Git for Windows/x64 Setup.', 'Git for Windows/ARM64 Setup.', 'Portable ("thumbdrive edition")', 'Git for Windows/x64 Portable.', and 'Git for Windows/ARM64 Portable.' A section titled 'Using winget tool' follows, with instructions to install the winget tool and a code block showing the command: `winget install --id Git.Git -e --source winget`. The page concludes with a note about the current source code release being version 2.50.0 and a link to the source code.

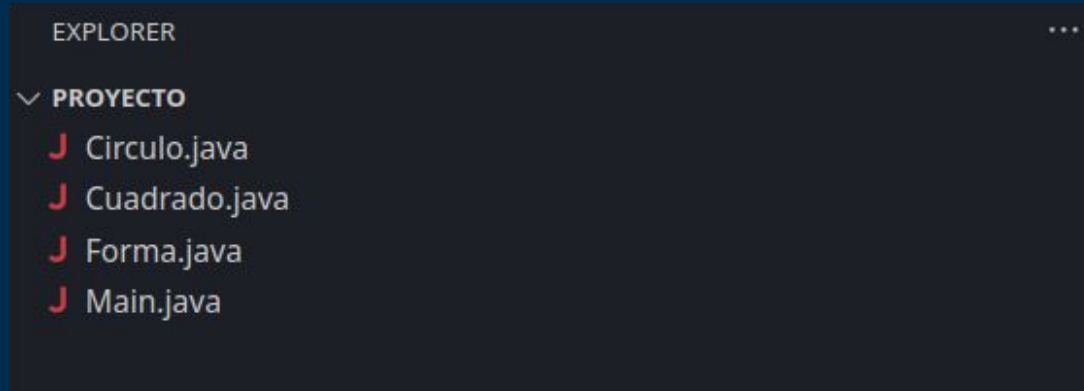
\$ sudo apt install git

\$ sudo pacman -S git

\$ sudo dnf install git



Espacio de trabajo



Espacio de trabajo

✓ PROYECTO

> .git

J Circulo.java

U

J Cuadrado.java

U

J Forma.java

U

J Main.java

U

\$ git init

Espacio de trabajo

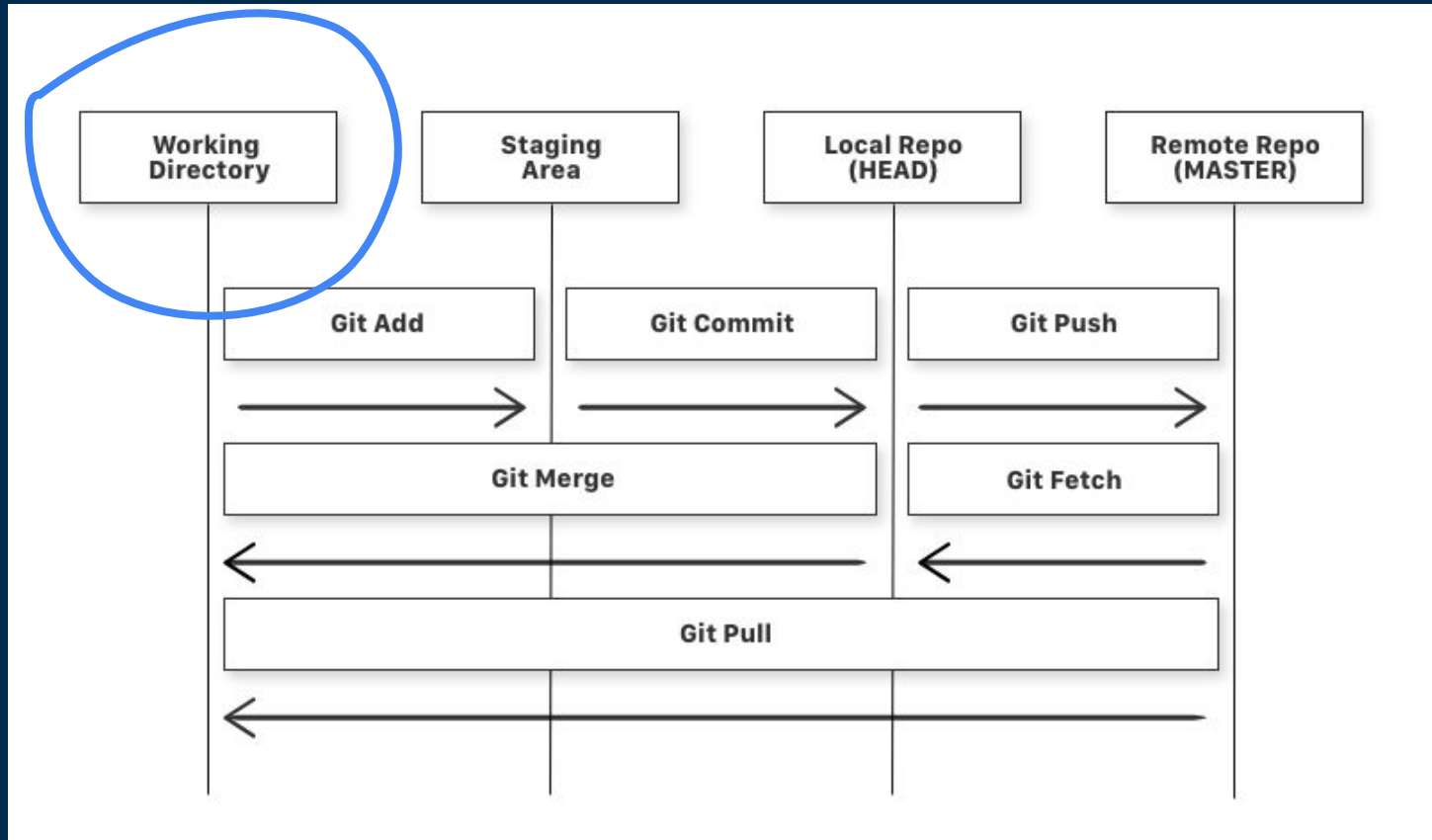
```
TERMINAL
[markel@smint Proyecto]$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Circulo.java
    Cuadrado.java
    Forma.java
    Main.java

nothing added to commit but untracked files present (use "git add" to track)
[markel@smint Proyecto]$
```

\$ git status



Área de staging

```
TERMINAL
bash + - [ ] [ ] ... X

• [markel@smint Proyecto]$ git add Main.java
• [markel@smint Proyecto]$ git status
On branch main

No commits yet

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

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Circulo.java
    Cuadrado.java
    Forma.java

○ [markel@smint Proyecto]$
```

\$ git add <fichero>

Área de staging

```
TERMINAL bash + v [icon] [icon] ... X

• [markel@smint Proyecto]$ git add --all
• [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

○ [markel@smint Proyecto]$
```

\$ git add --all

Área de staging

J Main.java M X

J Main.java

```
1 public class Main {  
2     public static void main() {  
3         System.out.println("Estoy modificando un fichero stageado")  
4     }  
5 }
```

TERMINAL

bash + v [icon] [icon] ... X

• [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
```

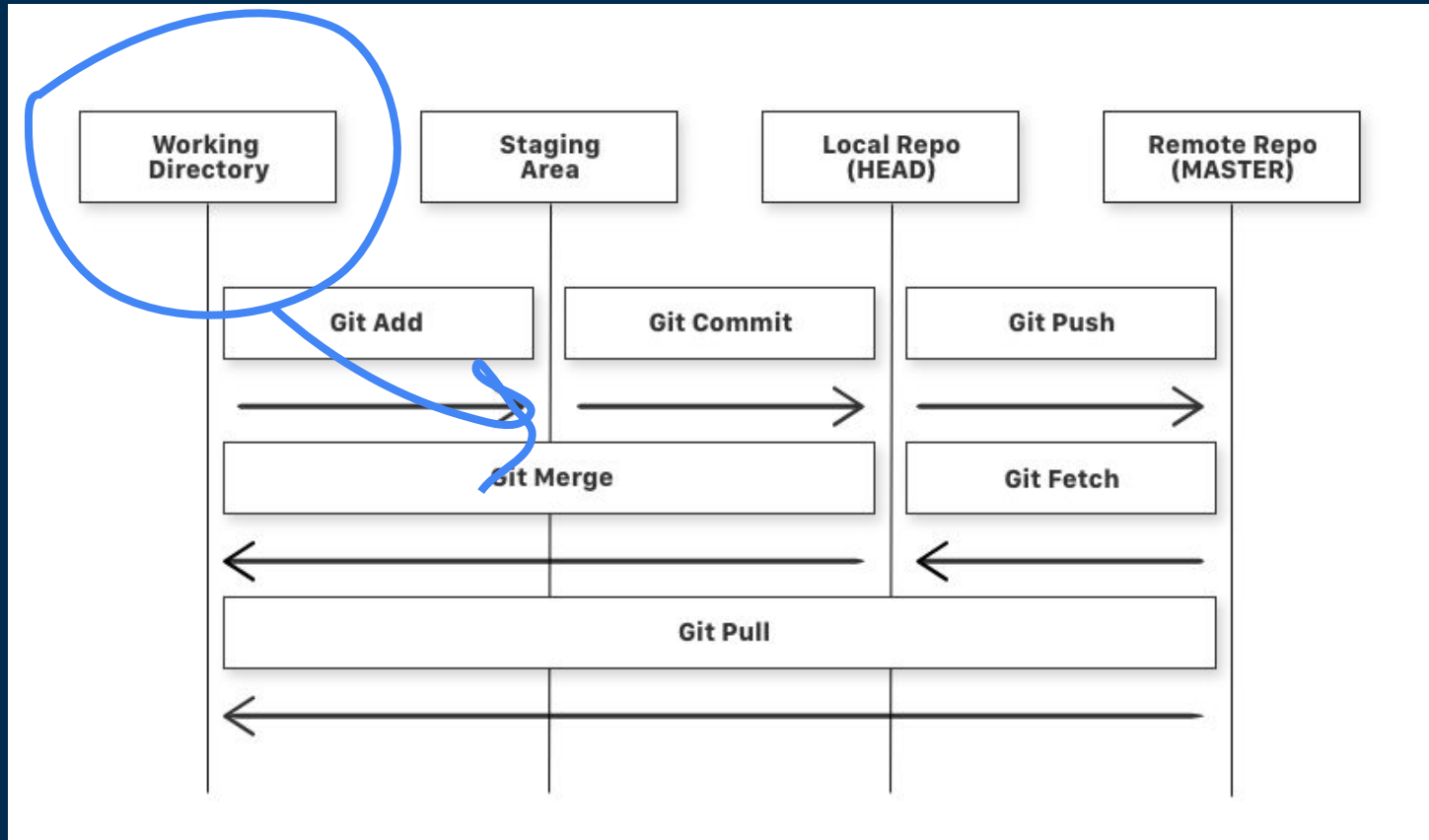
Changes not staged for commit:

(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]\$



Commits

```
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

• [markel@smint Proyecto]$ git commit -m "Initial commit"
[main (root-commit) 284eb79] Initial commit
4 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Circulo.java
create mode 100644 Cuadrado.java
create mode 100644 Forma.java
create mode 100644 Main.java

• [markel@smint Proyecto]$ git status
On branch main
nothing to commit, working tree clean

○ [markel@smint Proyecto]$
```

\$ git commit -m <nombre del commit>

Commits

J Main.java M X

J Main.java

```
1 public Class Main {  
2     public static int main() {  
3         System.out.println("hello world");  
4     }  
5 }
```

TERMINAL

bash + ▾ [] [] ... X

- [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

```
TERMINAL bash + v [icon] [icon] ... x

• [markel@smint Proyecto]$ git log
commit f3b3df552d69670b1d56055524b535d57daaedb6 (HEAD -> main)
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]$
```

Hash del commit

\$ git log

Commits

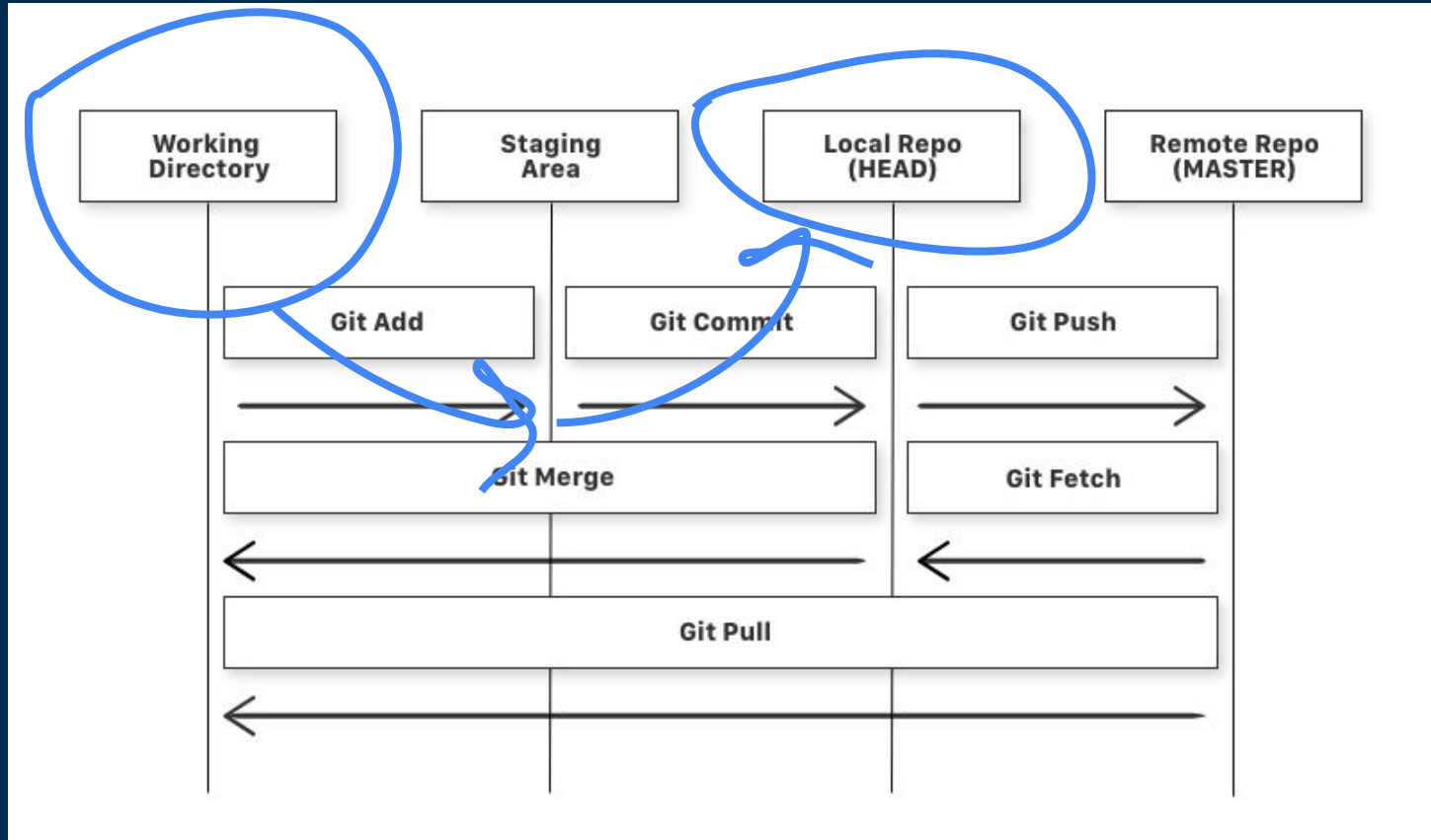
```
TERMINAL
bash + - [icon] [icon] [icon] [icon] [icon] [icon]

• [markel@smint Proyecto]$ git show f3b3df552d69670b1d56055524b535d57daaedb6
commit f3b3df552d69670b1d56055524b535d57daaedb6 (HEAD -> main)
Author: Markel Mencía <markel.mnc@gmail.com>
Date:   Tue Jun 17 19:13:28 2025 +0200

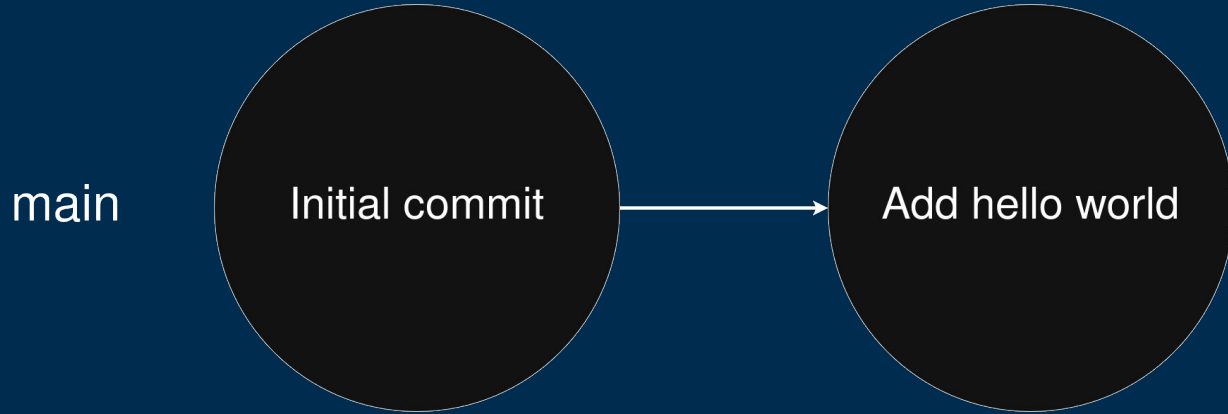
    Add hello world

diff --git a/Main.java b/Main.java
index e69de29..c9b3a5c 100644
--- a/Main.java
+++ b/Main.java
@@ -0,0 +1,5 @@
+public Class Main {
+    public static int main() {
+        System.out.println("hello world");
+    }
+}
\ No newline at end of file
○ [markel@smint Proyecto]$
```

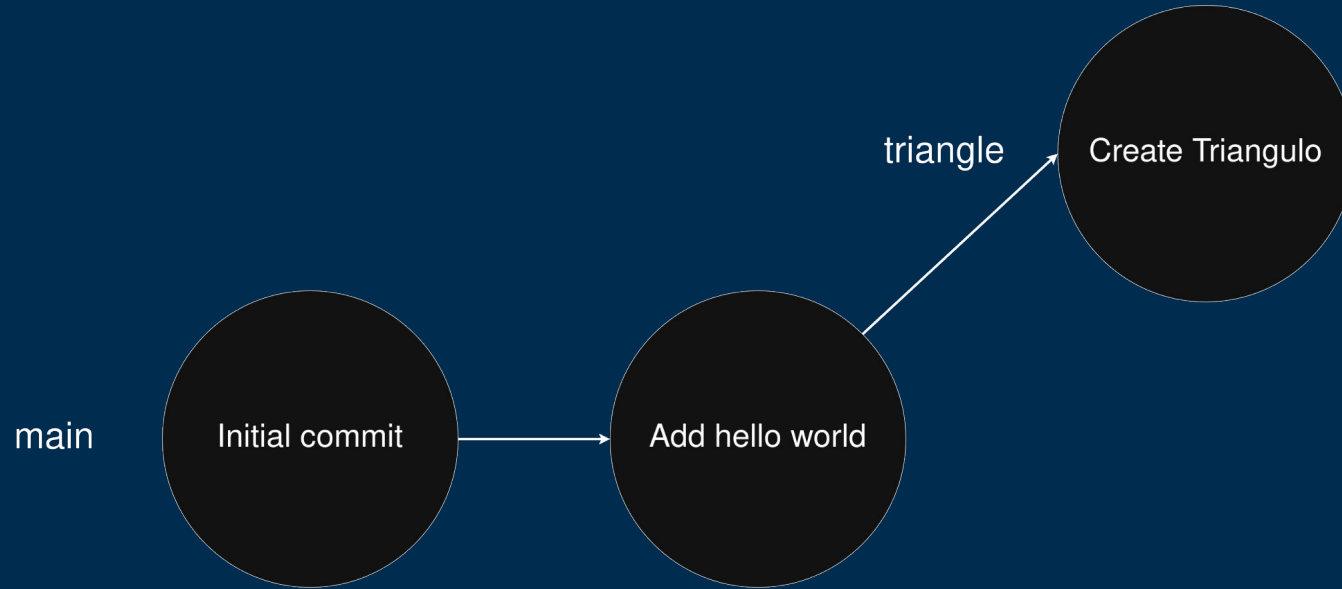
\$ git show <hash del commit>



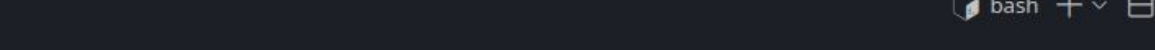
Branches



Branches



Branches



A terminal window titled "TERMINAL" with a dark background. The prompt is `[markel@smint Proyecto]$`. The first command is `git branch triangle`. The second command is `git branch`, which lists the branches: `* main` and `triangle`. A white arrow points to the `* main` line. The third command is `[markel@smint Proyecto]$` with a cursor.

```
TERMINAL
[markel@smint Proyecto]$ git branch triangle
[markel@smint Proyecto]$ git branch
* main
  triangle
[markel@smint Proyecto]$
```

Branch actual

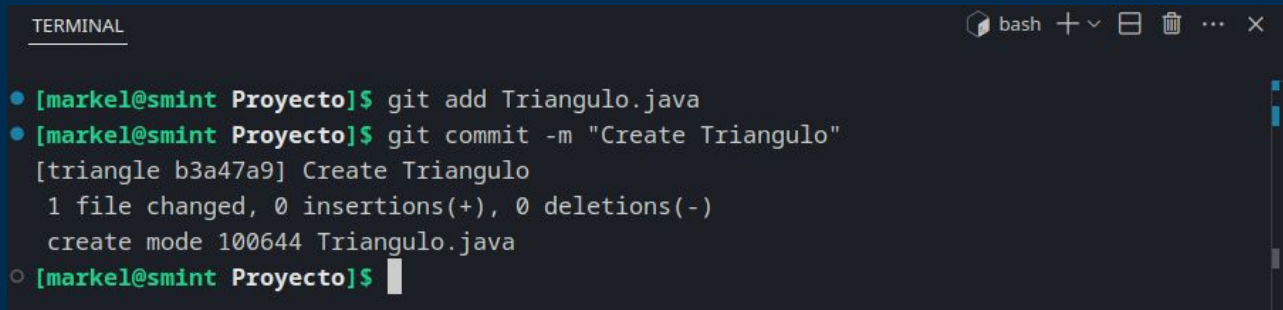
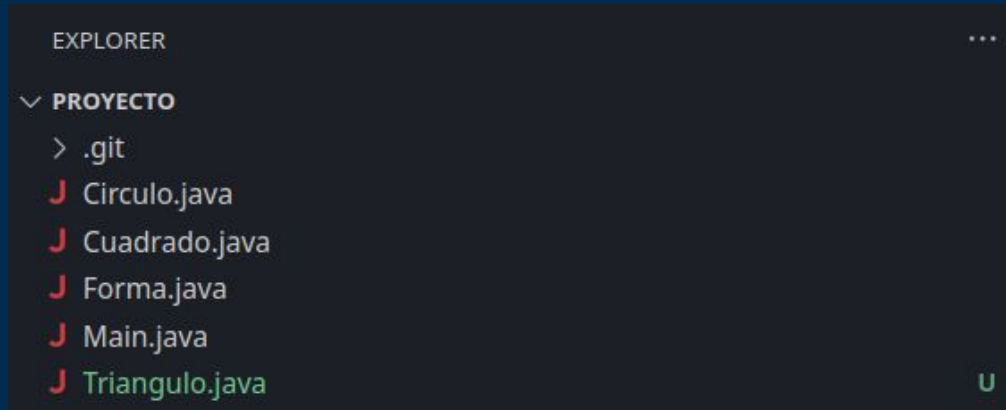
\$ git branch <nueva branch>

Branches

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

\$ git checkout <branch>

Branches



Branches

J Triangulo.java M X

J Triangulo.java

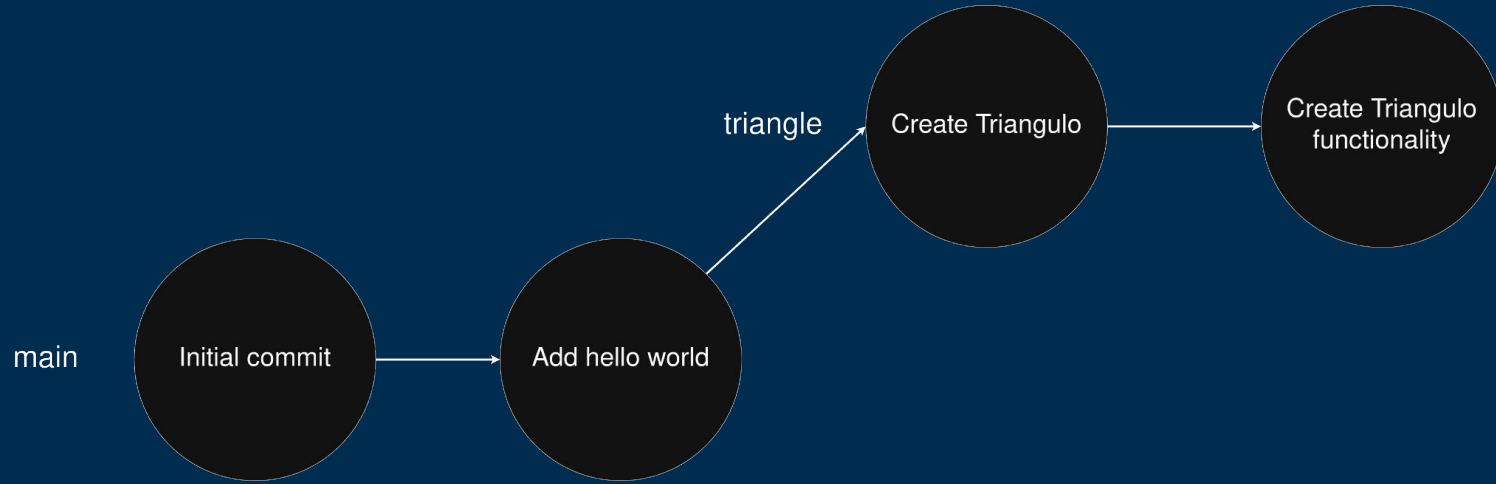
```
1 public class Triangulo {  
2  
3     private float base;  
4     private float altura;  
5  
6     public Triangulo(float base, float altura) {  
7         this.base = base;  
8         this.altura = altura;  
9     }  
10 }
```

TERMINAL

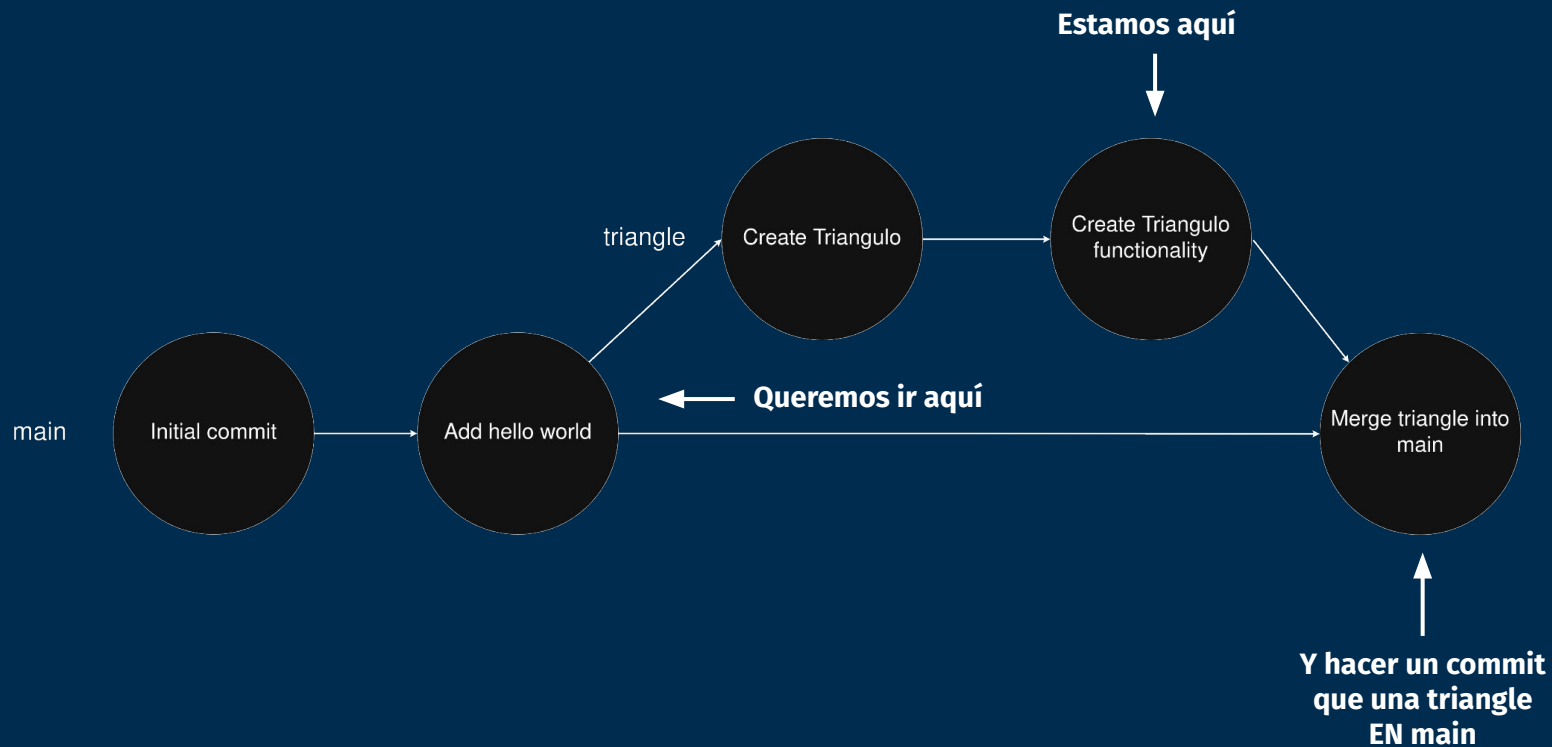
bash + v [icon] [icon] ... X

- [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]\$

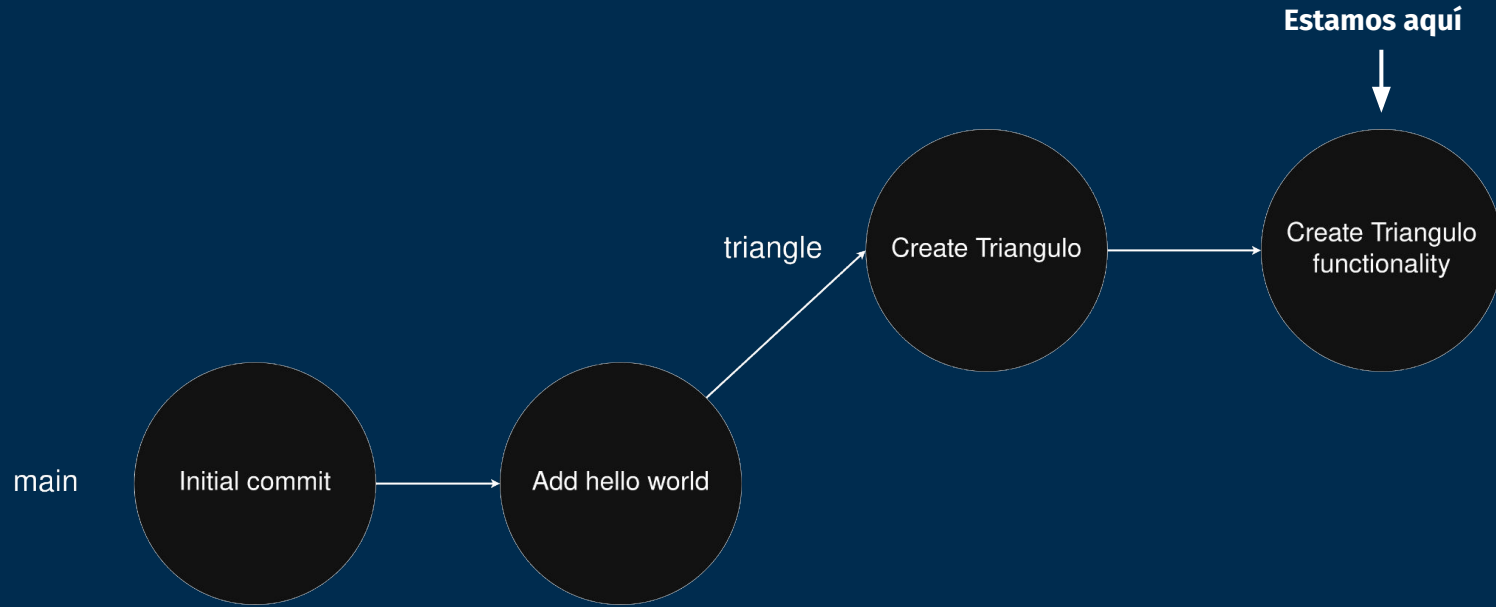
Branches



Merge

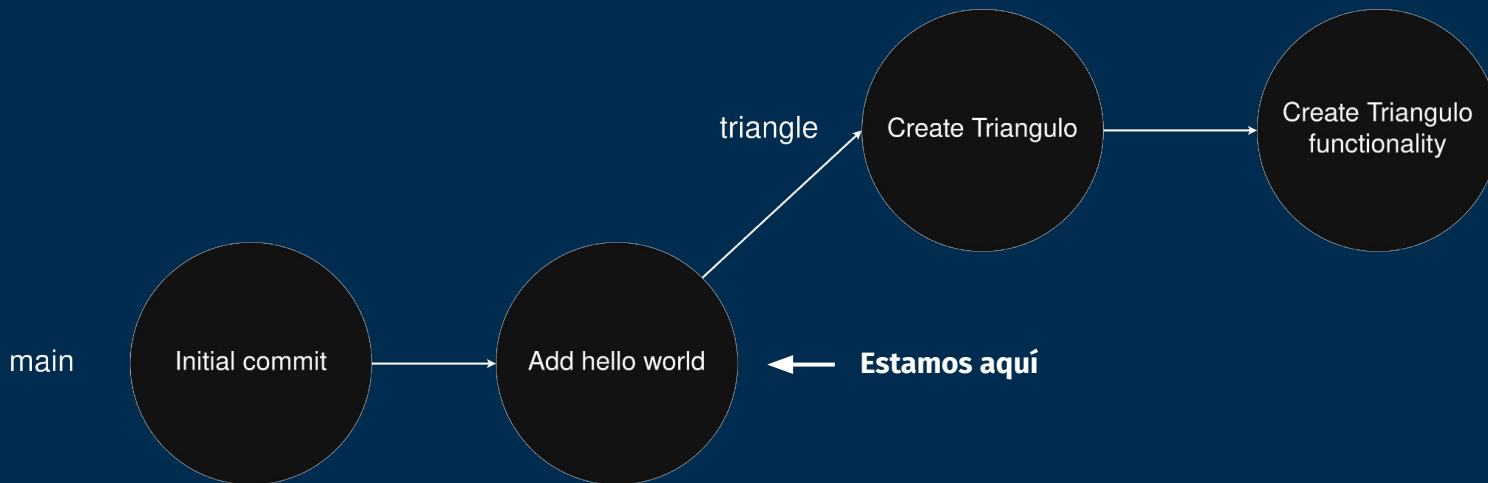


Merge



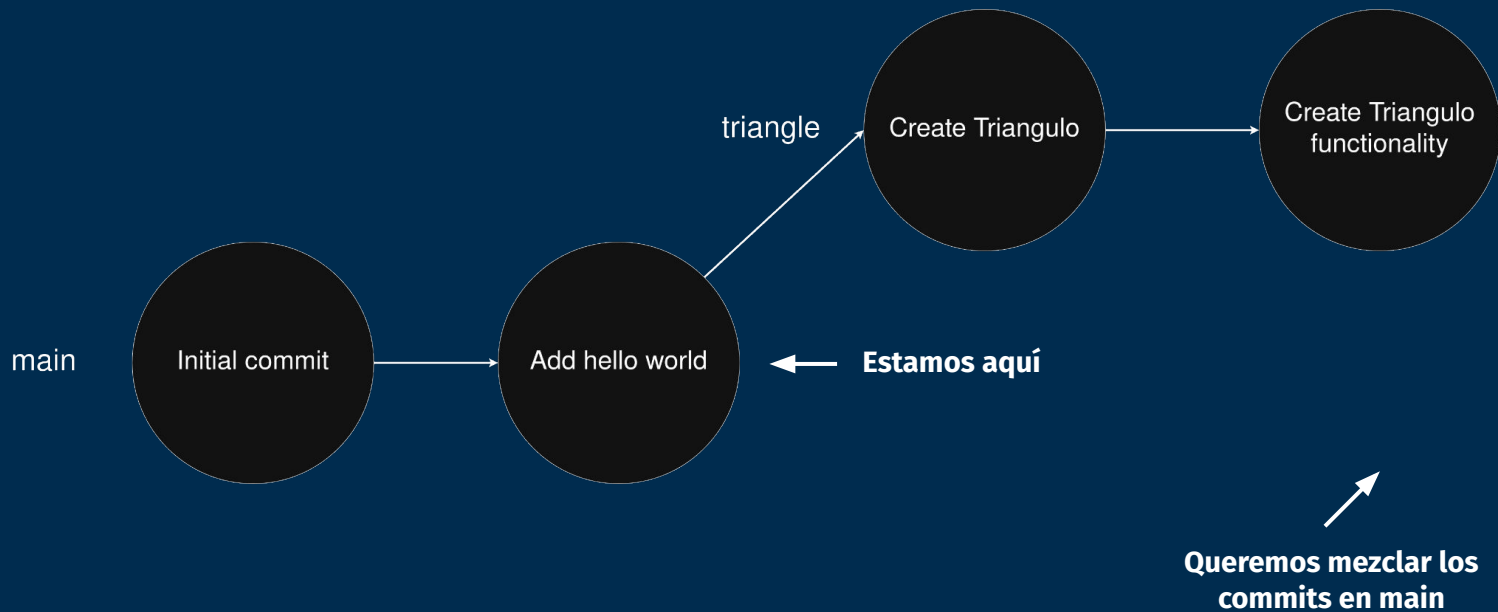
Merge

```
TERMINAL bash + - [x]
• [markel@smint Proyecto]$ git checkout main
| Switched to branch 'main'
○ [markel@smint Proyecto]$
```



\$ git checkout main

Merge



Merge

```
TERMINAL bash + - [ ] [ ] ... X

• [markel@smint Proyecto]$ git merge triangle
Updating f3b3df5..f232ba9
Fast-forward
 Triangulo.java | 10 ++++++++
 1 file changed, 10 insertions(+)
 create mode 100644 Triangulo.java
○ [markel@smint Proyecto]$
```

(Estando en la rama destino)

\$ git merge <rama origen>

Merge

```
TERMINAL bash + v [icons] ... X

• [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

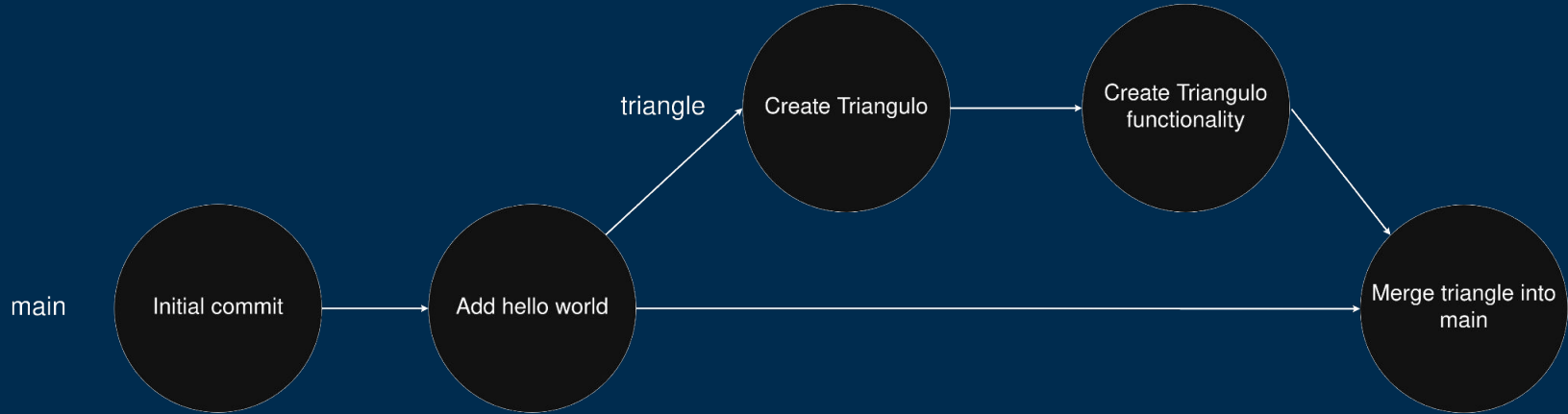
commit f3b3df552d69670b1d56055524b535d57daaedb6
Author: Markel Mencía <markel.mnc@gmail.com>
Date:   Tue Jun 17 19:13:28 2025 +0200

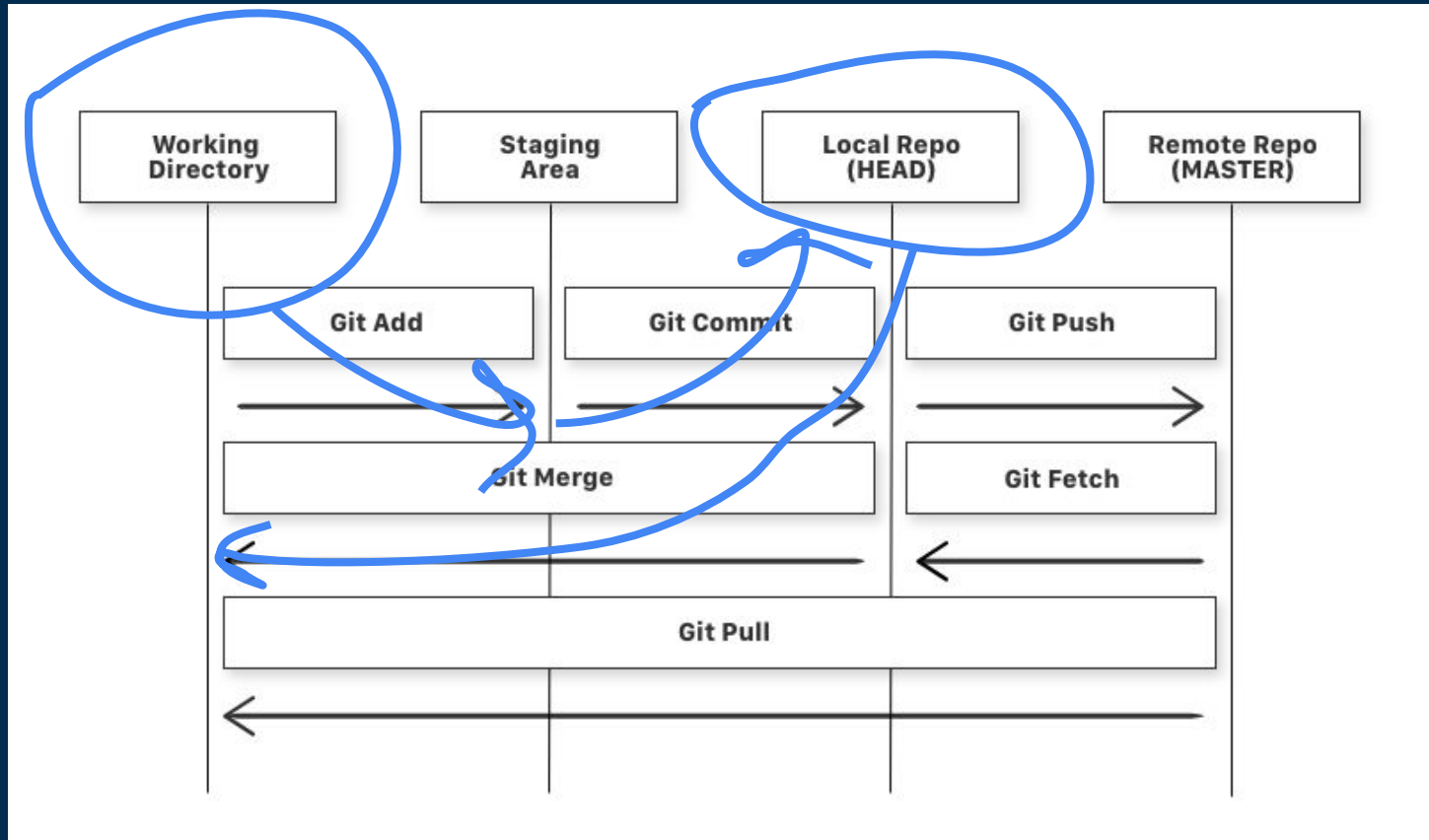
    Add hello world

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

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

Merge





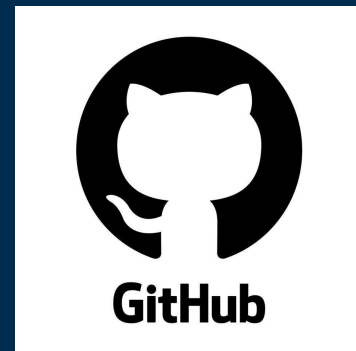




Repositorio Local



Repositorio Remoto



Tenemos que establecer esta conexión



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



Projecto 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
```

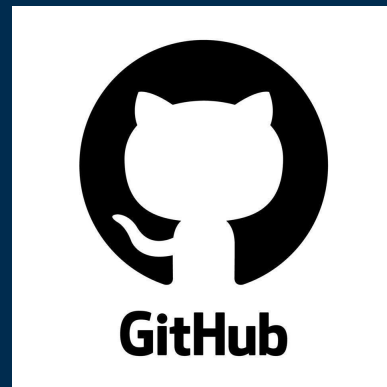


...or push an existing repository from the command line

```
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.



Generar par de claves

```
TERMINAL bash + v [icon] [icon] ... x

• [markel@smint Proyecto]$ ssh-keygen -t ed25519 -C "markel.mnc@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/markel/.ssh/id_ed25519): clave_github
Enter passphrase for "clave_github" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in clave_github
Your public key has been saved in clave_github.pub
The key fingerprint is:
SHA256:RR4pvT6WtDVPqtT9pU01tqxgj4St8TVRN1lFb7VAZzo markel.mnc@gmail.com
The key's randomart image is:
+---[ED25519 256]---+
|      .o..o +O|
|      .oo.  =+=|
|      .o. E..=|
|      .o o.o+.|
|      Soo= *+ +|
|      oB=ooooo|
|      o*o= o=.|
|      ..o o. o|
|              |
+-----[SHA256]-----+

○ [markel@smint Proyecto]$
```

↑
**Preferiblemente dejar
en blanco**

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

Iniciar el agente ssh

```
TERMINAL bash + - [ ] [ ] ... X

• [markel@smint Proyecto]$ eval "$(ssh-agent -s)"
  Agent pid 3916
○ [markel@smint Proyecto]$
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\WINDOWS\system32> Get-Service -Name ssh-agent | Set-Service -StartupType Manual
PS C:\WINDOWS\system32> Start-Service ssh-agent
PS C:\WINDOWS\system32>
```

Añadir el par de claves a ssh

TERMINAL

bash + ▾ ☐ 🗑️ ⋮ ✕

```
● [markel@smint Proyecto]$ ssh-add ~/.ssh/id_ed25519
Enter passphrase for /home/markel/.ssh/id_ed25519:
Identity added: /home/markel/.ssh/id_ed25519 (markel.mnc@gmail.com)
○ [markel@smint Proyecto]$
```

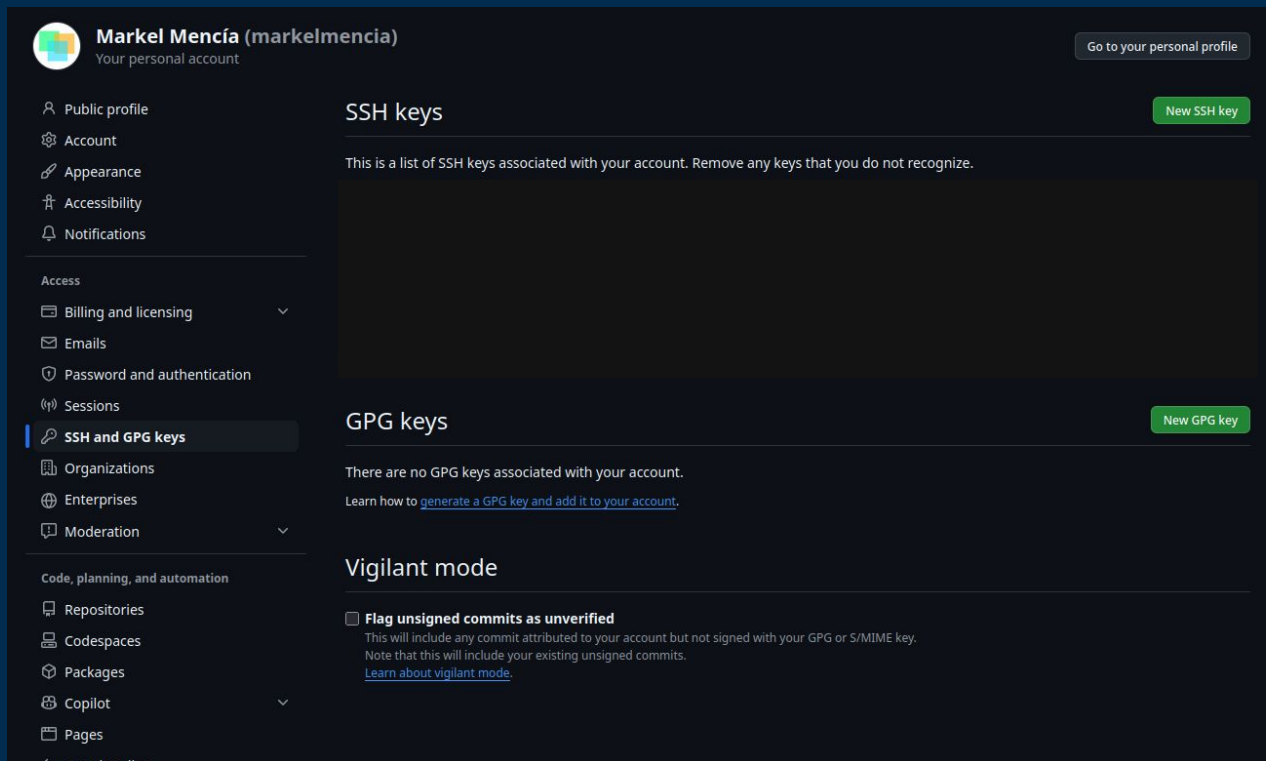
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. <https://aka.ms/PSWindows>

```
PS C:\WINDOWS\system32> ssh-add c:/Users/markel.mencia/.ssh/id_ed25519
Identity added: c:/Users/markel.mencia/.ssh/id_ed25519 (markel.mnc@gmail.com)
PS C:\WINDOWS\system32>
```

\$ ssh-add ~/.ssh/id_ed25519"

Añadir la clave pública a GitHub



Markel Mencía (markelmencia)
Your personal account

Go to your personal profile

Public profile
Account
Appearance
Accessibility
Notifications

Access

Billing and licensing
Emails
Password and authentication
Sessions

SSH and GPG keys

Organizations
Enterprises
Moderation

Code, planning, and automation

Repositories
Codespaces
Packages
Copilot
Pages
Saved replies

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

GPG keys

New GPG key

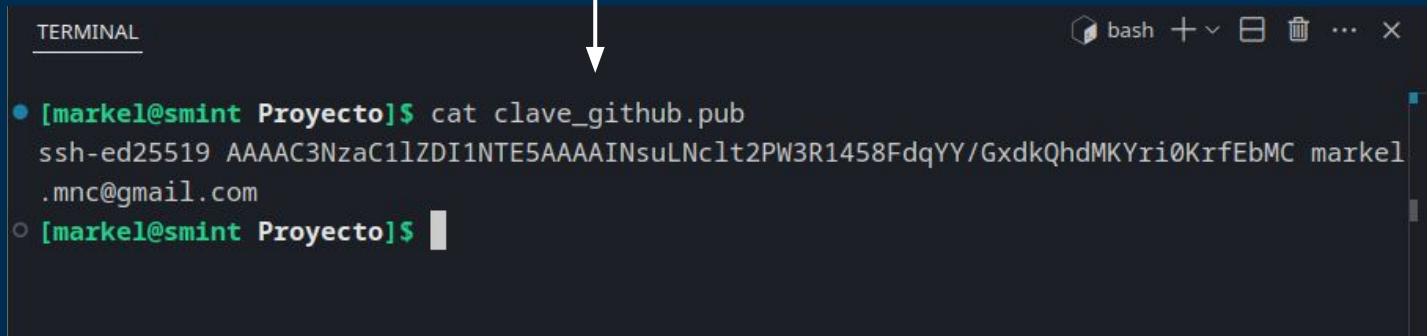
There are no GPG keys associated with your account.
Learn how to [generate a GPG key and add it to your account](#).

Vigilant mode

☐ **Flag unsigned commits as unverified**
This will include any commit attributed to your account but not signed with your GPG or S/MIME key.
Note that this will include your existing unsigned commits.
[Learn about vigilant mode.](#)

Añadir la clave pública a GitHub

~/.ssh/id_ed25519.pub




A terminal window titled "TERMINAL" with a dark background. The window shows a command prompt where the user has entered `cat clave_github.pub`. The output of the command is displayed on the next line: `ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAINsuLNc1t2PW3R1458FdqYY/GxdkQhdMKYri0KrfEbMC markel.mnc@gmail.com`. The prompt is now `[markel@smint Proyecto]$` with a cursor. Above the terminal window, the text `~/.ssh/id_ed25519.pub` is shown with a white arrow pointing down to the terminal.

```
TERMINAL
[markel@smint Proyecto]$ cat clave_github.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAINsuLNc1t2PW3R1458FdqYY/GxdkQhdMKYri0KrfEbMC markel.mnc@gmail.com
[markel@smint Proyecto]$
```

\$ cat ~/.ssh/id_ed25519.pub

Añadir la clave pública a GitHub

**Markel Mencía** (markelmencia)
Your personal account

[Go to your personal profile](#)

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and licensing

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Enterprises

Moderation

Code, planning, and automation

Repositories

Codespaces

Packages

Copilot

Add new SSH Key

Title
public_key-github

Key type
Authentication Key

Key
ssh-ed25519 AAAAC3NzaC1lZD11NTE5AAAAInsuLNclt2PW3R1458FdqYY/GxdkQhdMKYri0KrfEbMC markel.mnc@gmail.com

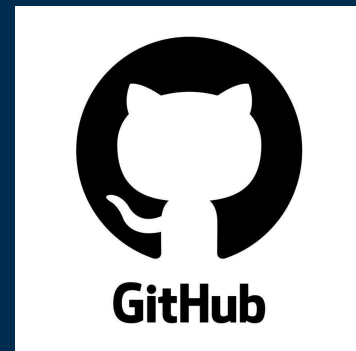
Add SSH key



Repositorio Local



Repositorio Remoto



Ya tenemos la conexión ssh establecida



Y hemos creado el repositorio remoto



Vincular repositorio local con remoto

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

HTTPS

SSH

git@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](#).

TERMINAL

bash + v [icon] [icon] ... x

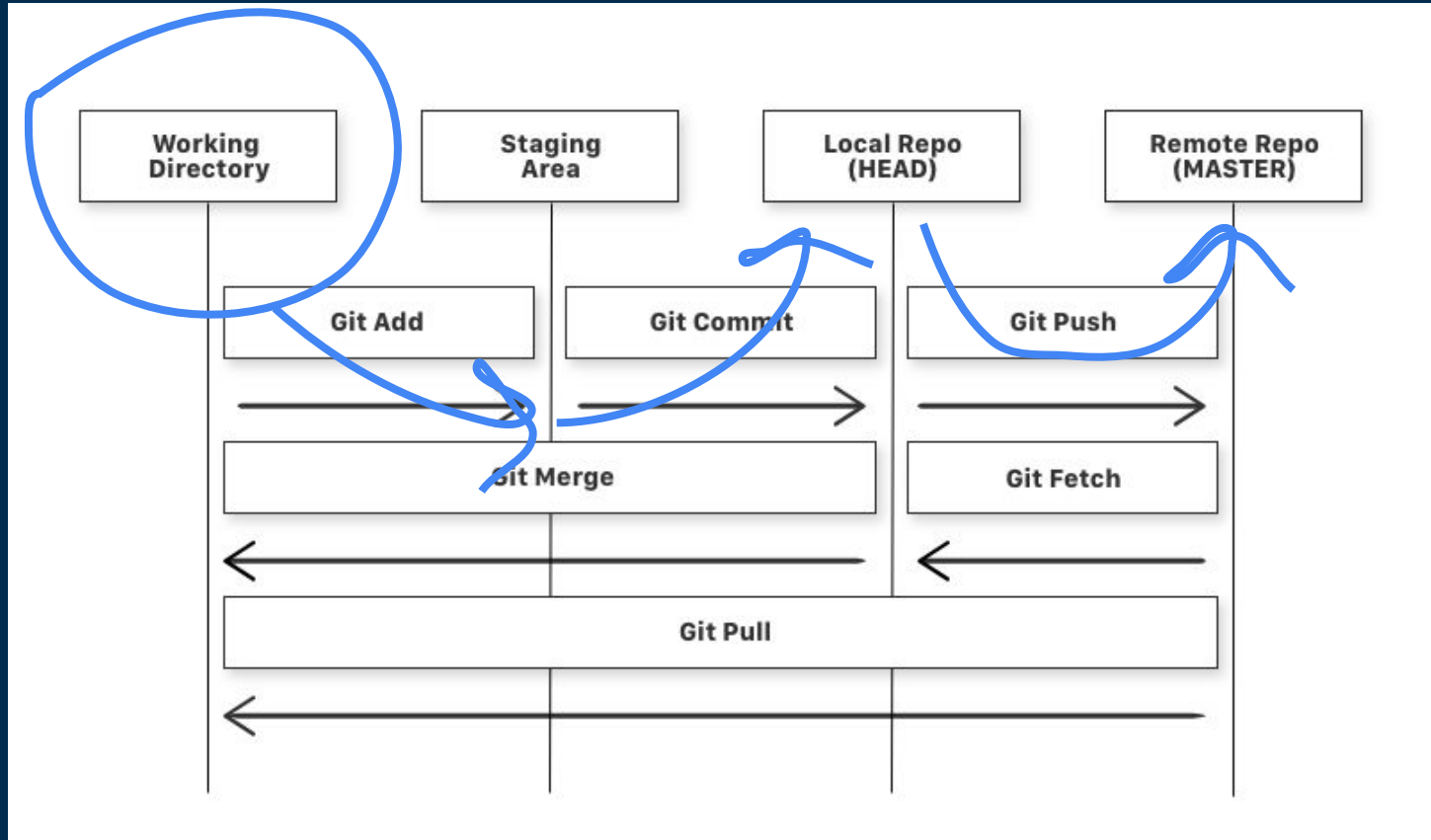
```
• [markel@smint Proyecto]$ git remote add origin git@github.com:markelmencia/Proyecto.git
• [markel@smint Proyecto]$ git remote -v
origin git@github.com:markelmencia/Proyecto.git (fetch)
origin git@github.com:markelmencia/Proyecto.git (push)
○ [markel@smint Proyecto]$
```

\$ git remote add origin <enlace ssh>

Push

```
TERMINAL
[markel@smint Proyecto]$ git push origin main
Enter passphrase for key '/home/markel/.ssh/id_ed25519':
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (11/11), 1.04 KiB | 1.04 MiB/s, done.
Total 11 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), done.
To github.com:markelmencia/Proyecto.git
 * [new branch]      main -> main
[markel@smint Proyecto]$
```

\$ git push origin <branch>





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

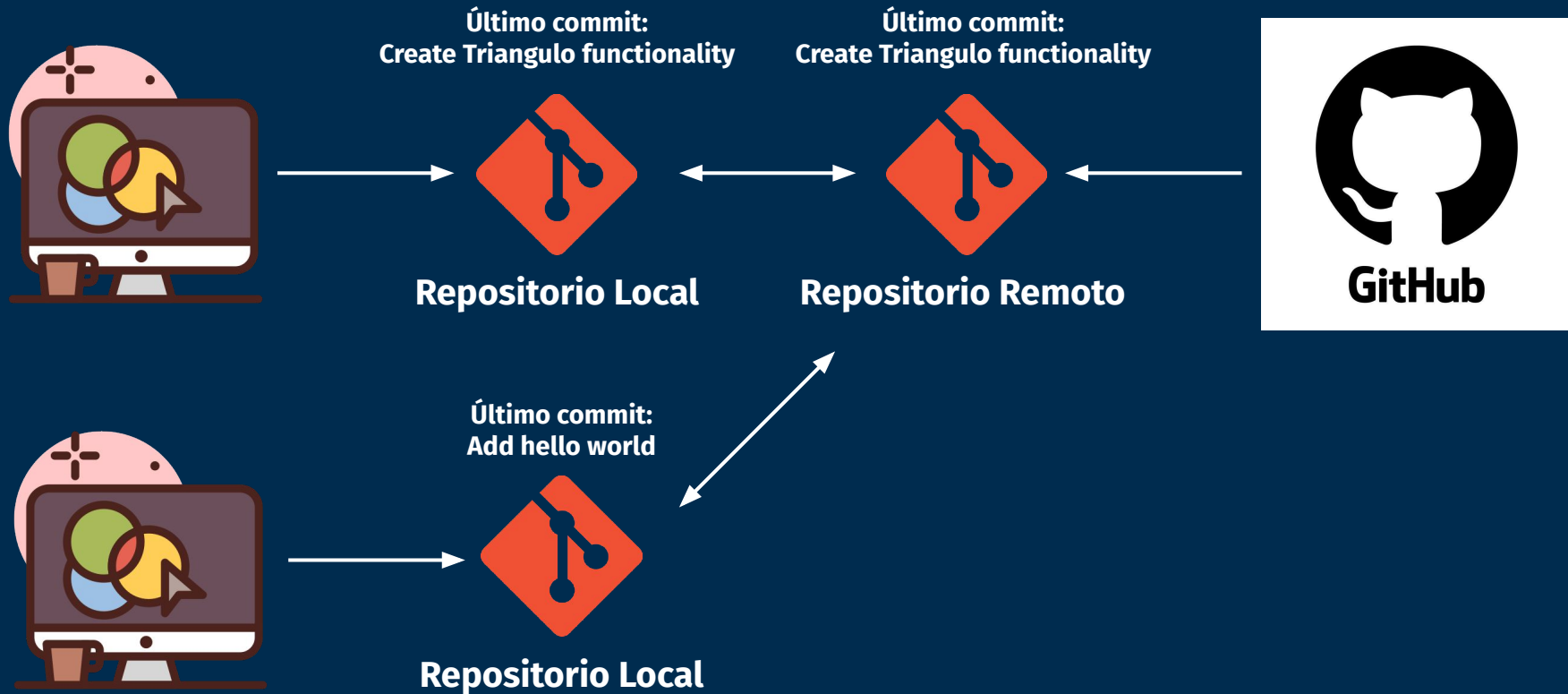
[Publish your first package](#)

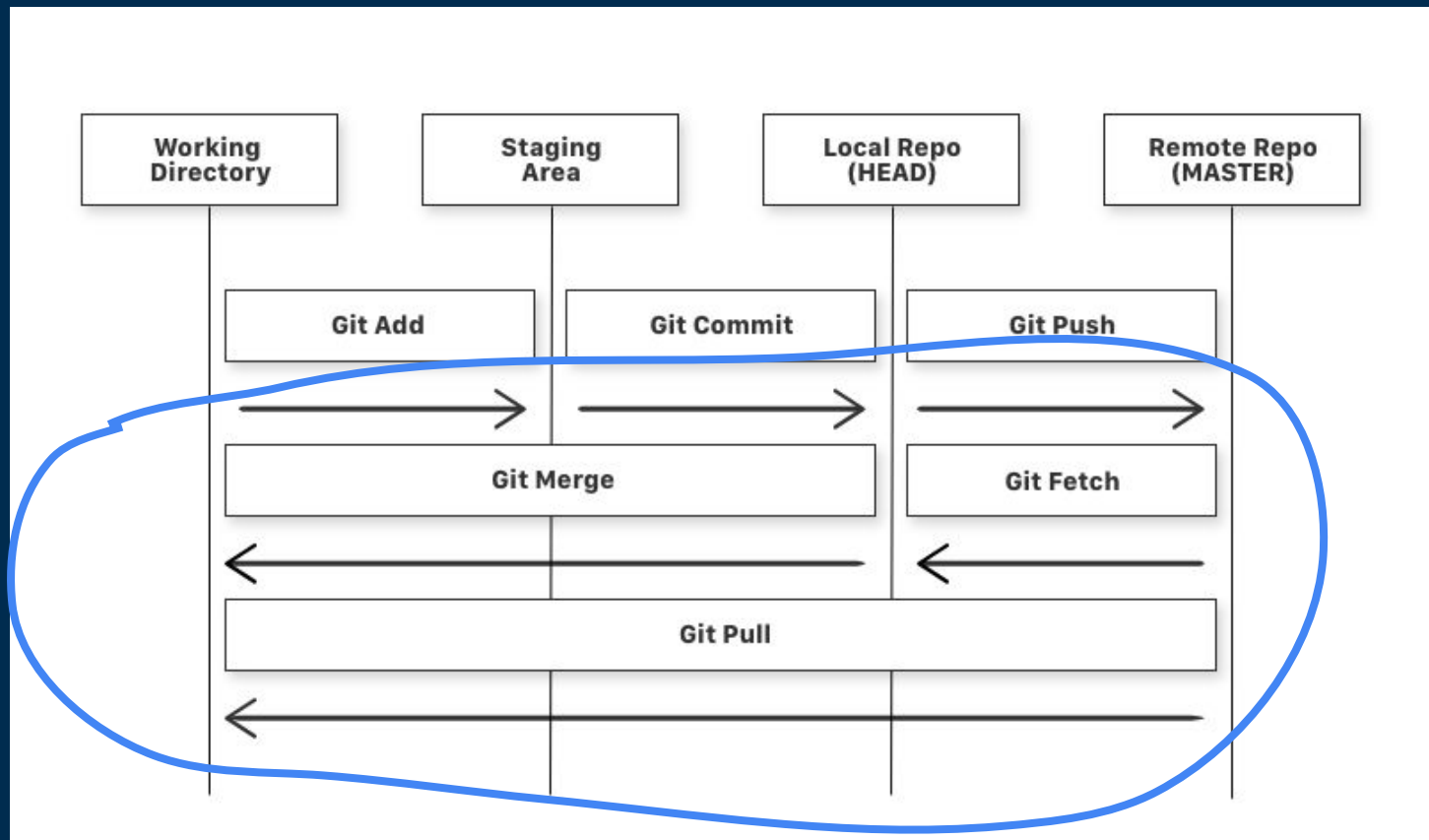
Languages

java 100.0%

Suggested workflows

Based on your tech stack





Fetch

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

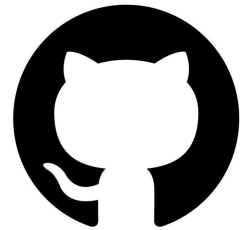


Repositorio Local

Local pregunta a Remoto
si ha habido
actualizaciones



Repositorio Remoto



GitHub

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

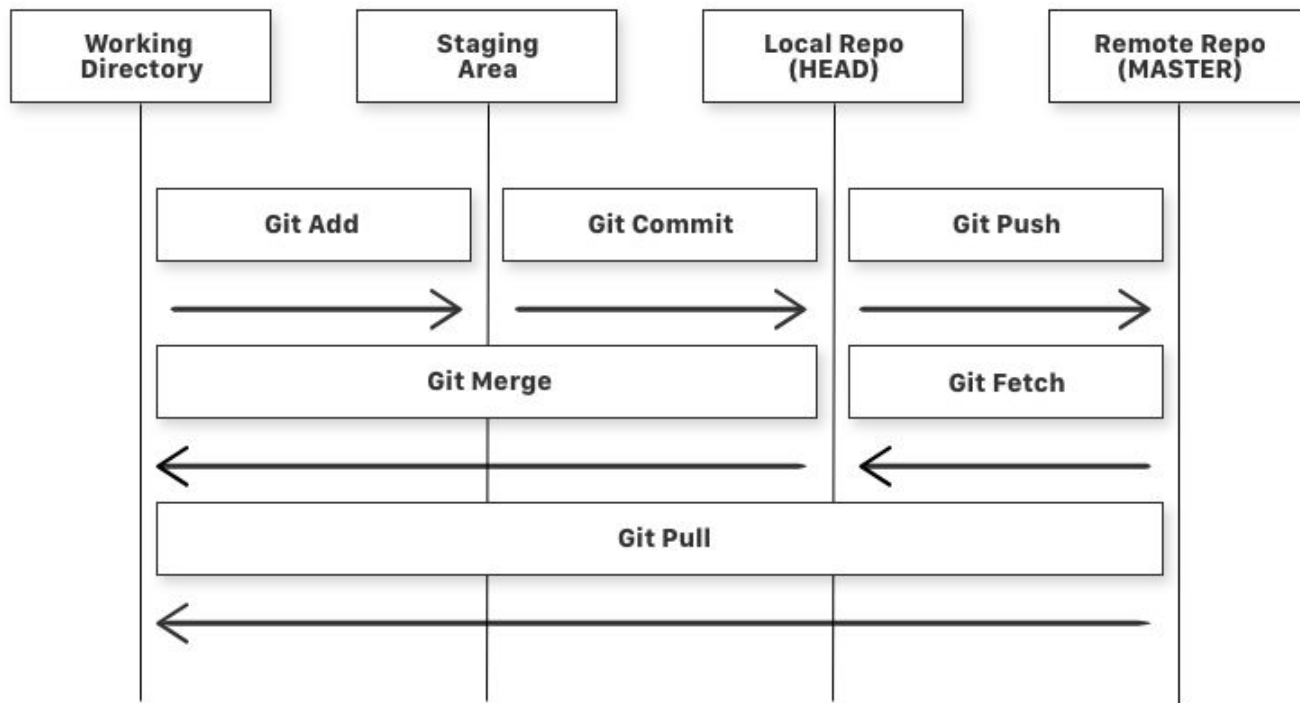
\$ git fetch
\$ git merge origin/<branch>

Pull

```
TERMINAL bash + ▾ ☰ 🗑️ ⋮ ✕  
● [markel@smint Proyecto]$ git pull origin main  
Enter passphrase for key '/home/markel/.ssh/id_ed25519':  
From github.com:markelmencia/Proyecto  
* branch          main          -> FETCH_HEAD  
Already up to date.  
○ [markel@smint Proyecto]$
```

Pull = Fetch + Merge

\$ git pull origin <branch>



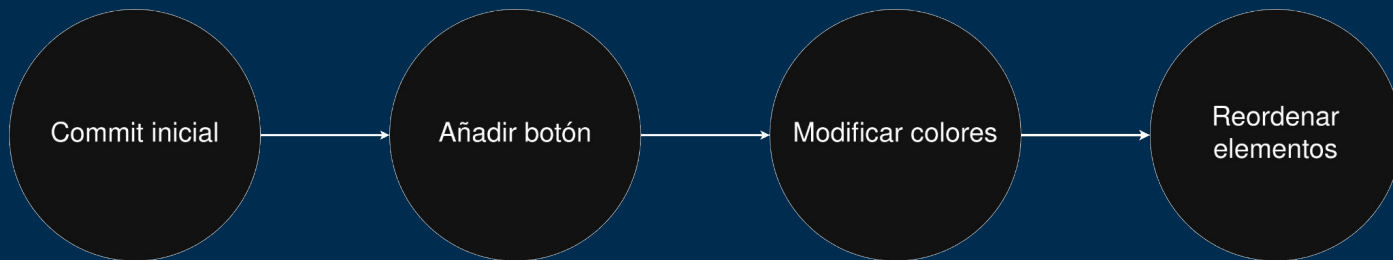


Buenas prácticas

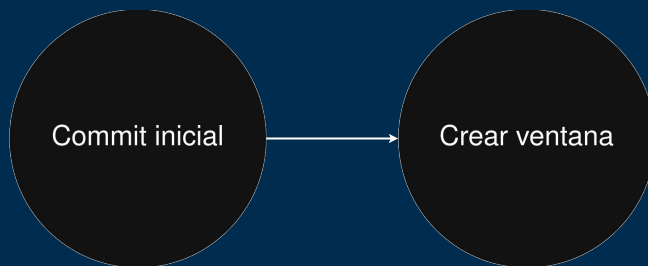
(Y recomendaciones)

Commits cortos y específicos

Do:

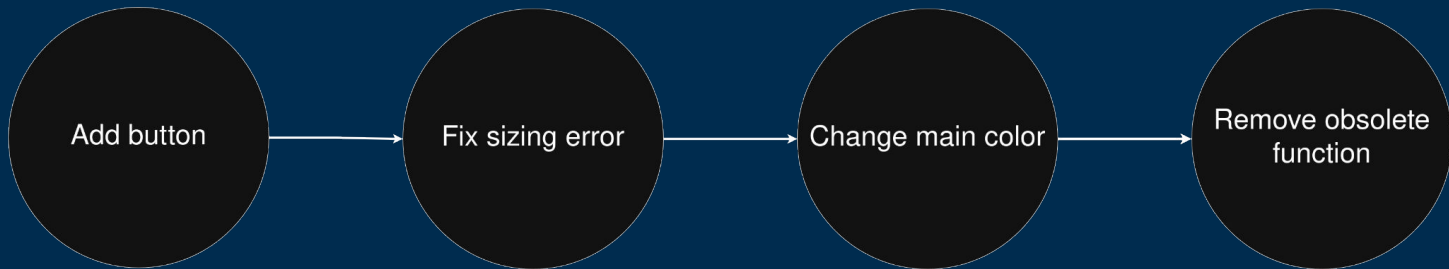


Don't:

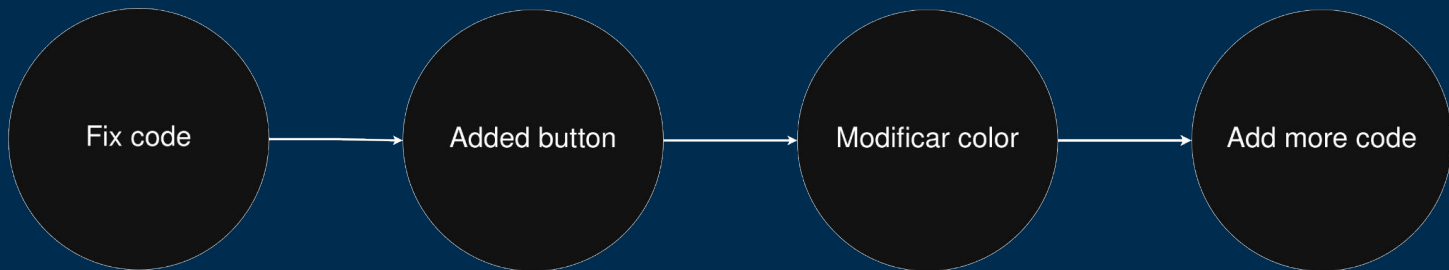


Mensajes de commit estructurados

Do:



Don't:



Solo pushear lo necesario

Pushea:

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

No pushees:

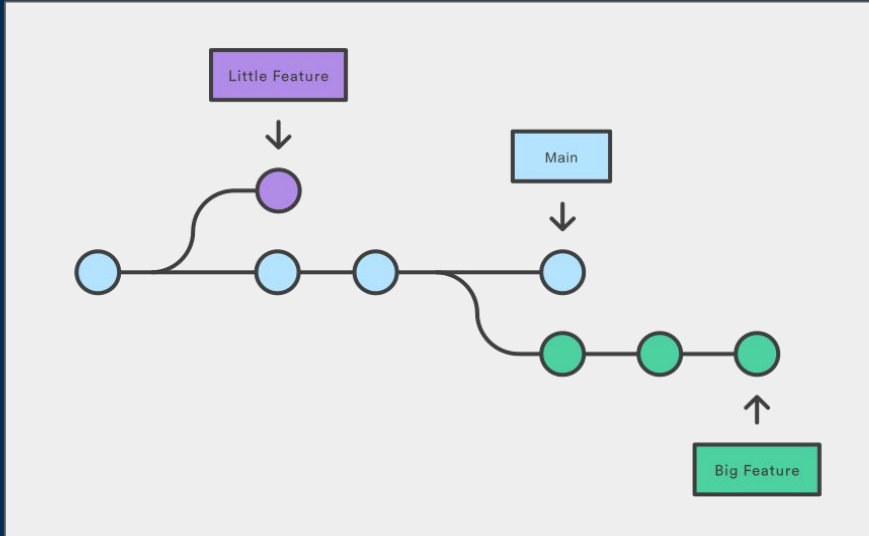
- Ejecutables
- Ficheros de bases de datos
- Carpetas de IDEs
- Logs / ficheros temporales
- Pares de claves / claves de APIs (porfi)

Utilizar un .gitignore

```
◆ .gitignore
1  # Ejecutables
2  *.exe
3  *.bin
4  *.out
5
6  # Logs
7  logs/
8  *.log
9
10 # Directorio de binarios
11 bin/
12
13 # Directorios que crean los IDEs
14 .idea/
15 .vscode/
16
17 # Ficheros de configuración
18 *.conf
19 *.env
20
```

Estructural branches

Do:



Don't:



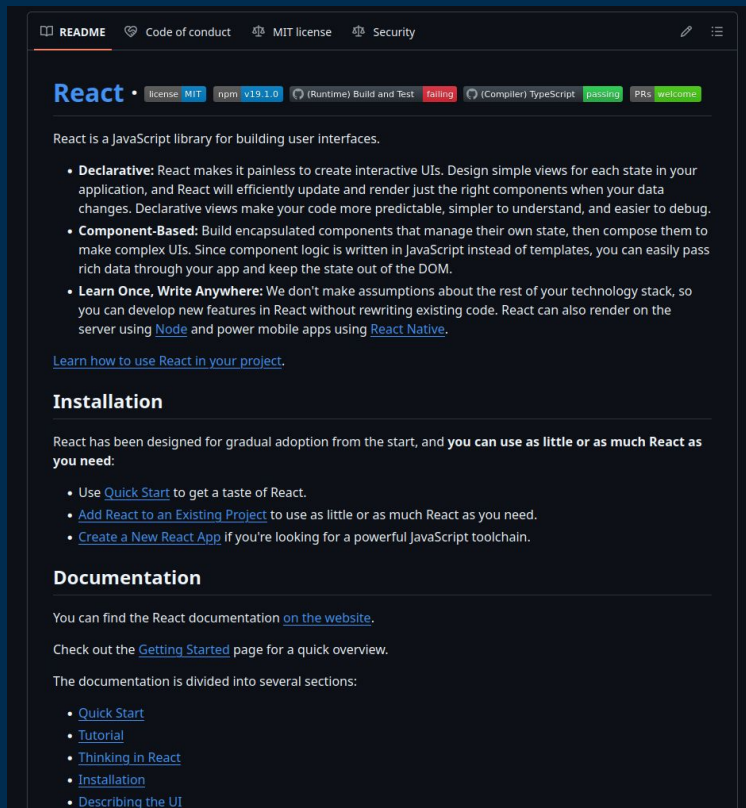
Escribir código legible

```
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()  
}
```

Do:

Documentar repositorio

Don't:



The screenshot shows the README for the React.js repository. It features a dark theme with a top navigation bar containing links for README, Code of conduct, MIT license, and Security. The main heading is 'React' followed by a series of status badges: license MIT, npm v19.1.0, (Runtime) Build and Test failing, (Compiler) TypeScript passing, and PRs welcome. The text describes React as a JavaScript library for building user interfaces. It lists three key features: Declarative (painless to create interactive UIs), Component-Based (encapsulated components), and Learn Once, Write Anywhere (no assumptions about the rest of the technology stack). A link points to 'Learn how to use React in your project.' Below this is an 'Installation' section stating that React is designed for gradual adoption and can be used as little or as much as needed. It provides three links: Quick Start, Add React to an Existing Project, and Create a New React App. The 'Documentation' section follows, stating that documentation can be found on the website and that it is divided into several sections: Quick Start, Tutorial, Thinking In React, Installation, and Describing the UI.

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.

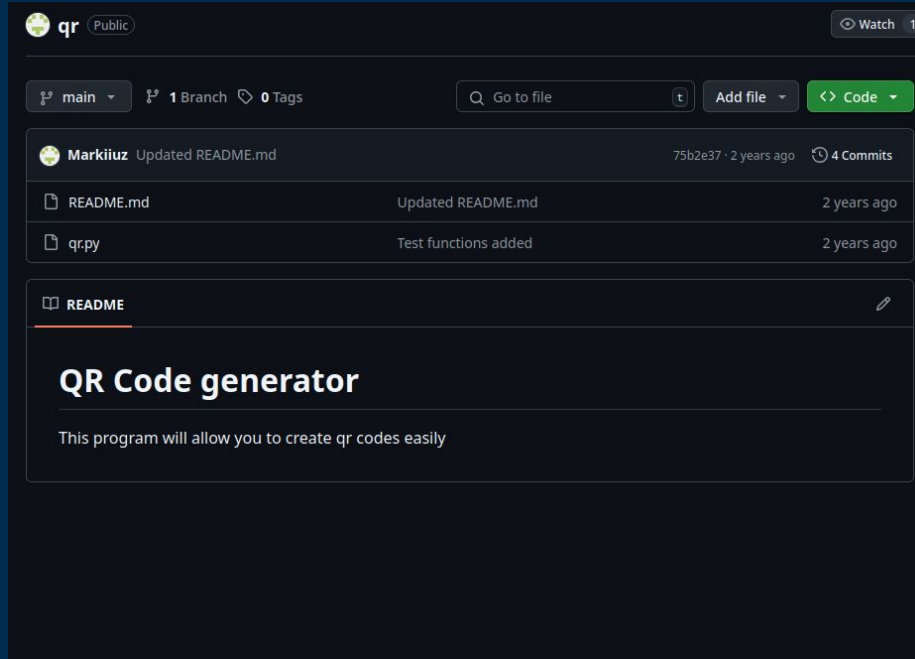
Documentation

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](#)



The screenshot shows the GitHub page for a repository named 'qr'. The repository is public and has 1 branch and 0 tags. It was updated by Markiiuz 75b2e37 · 2 years ago with 4 commits. The file list shows README.md (Updated README.md, 2 years ago) and qr.py (Test functions added, 2 years ago). Below the file list is a section titled 'QR Code generator' with a description: 'This program will allow you to create qr codes easily'.

qr Public

Watch 1

main 1 Branch 0 Tags

Go to file

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

QR Code generator

This program will allow you to create qr codes easily

¡Gracias!