

FABLAB.UV



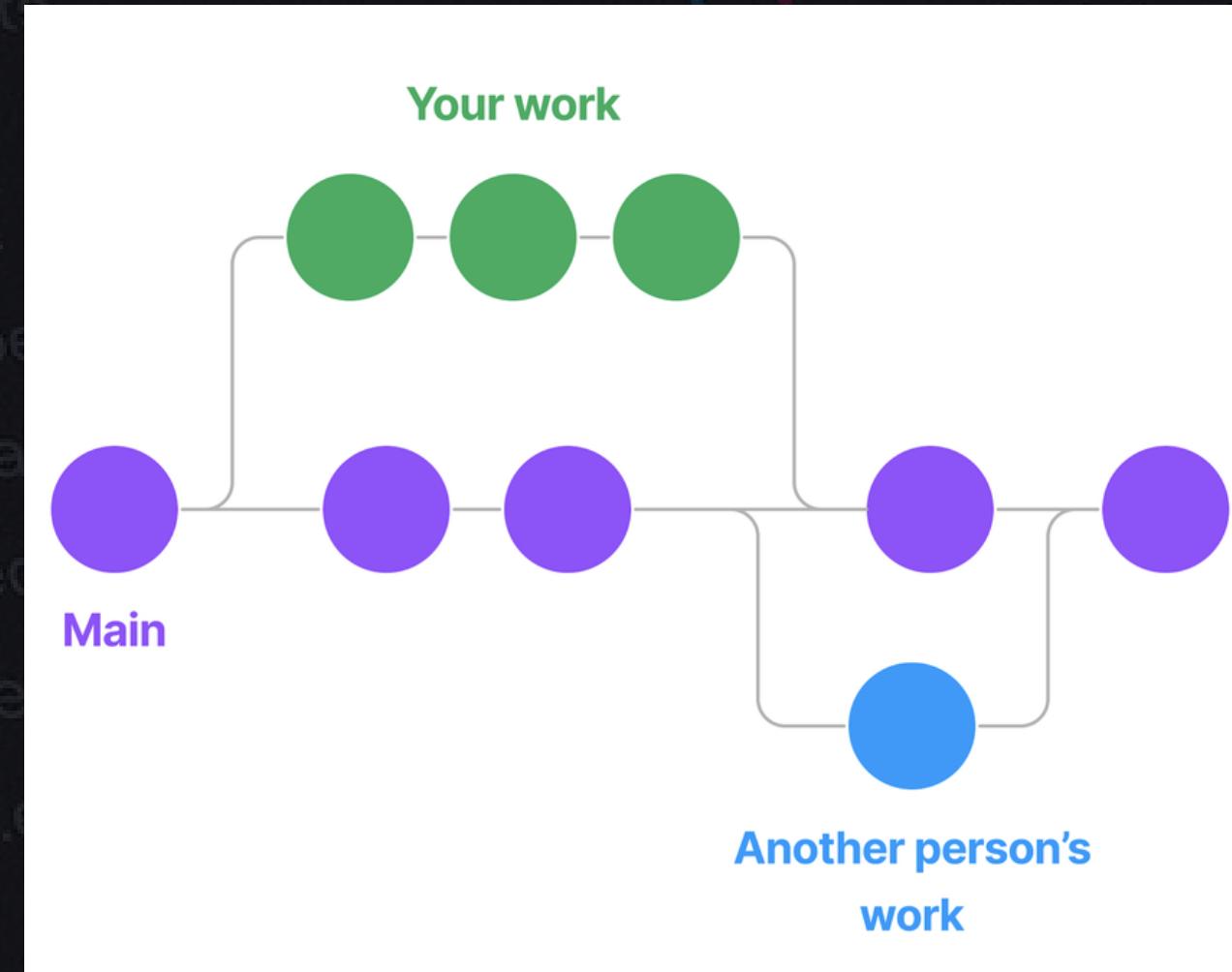
Taller de Git: Dominando el Control de Versiones

ALEJANDRO DIAZ CANTILLANO
EST. INGENIERÍA CIVIL INFORMATICA

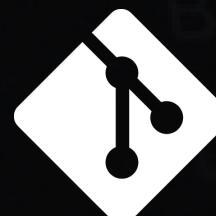


Fundamentos de Git

¿Qué es Git?



Git permite el desarrollo simultaneo de un software, control de versiones y trabajo en equipo



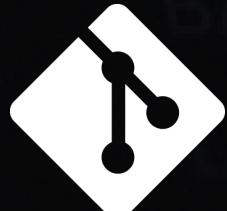
Dominando el Control de Versiones



Fundamentos de Git

Configuración inicial: instalación y configuración básica

The screenshot shows the official Git website (git-scm.com/). The main navigation bar includes links for 'About', 'Documentation', 'Downloads' (which is highlighted in red), and 'Community'. The 'Downloads' section provides links for 'GUI Clients' and 'Logos'. The central content area is titled 'Download for Windows' and features a call-to-action button: 'Click here to download the latest (2.41.0) 64-bit version of Git for Windows. This is the most recent maintained build. It was released about 1 month ago, on 2023-07-13.' Below this, there are links for 'Other Git for Windows downloads' including 'Standalone Installer', '32-bit Git for Windows Setup.', '64-bit Git for Windows Setup.', 'Portable ("thumbdrive edition")', '32-bit Git for Windows Portable.', and '64-bit Git for Windows Portable.'. A section on using the 'winget tool' is also present. At the bottom, there's a 'Now What?' section with a link to the source code. Three circular icons at the bottom represent 'Books', 'Code', and 'Terminal'.



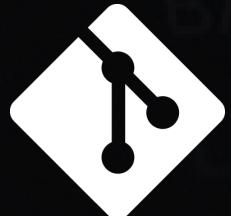
Dominando el Control de Versiones



Repositorio del taller



<https://github.com/ixyz022/FabLab-Taller-Git.git>



Dominando el Control de Versiones





Comandos Básicos de Git

Configuraciones iniciales

Configurar tu usuario

```
git config --global user.name "Tu Nombre"
```

```
git config --global user.email "tu@email.com"
```

Configurar nombre de la rama principal

```
git config --global init.defaultBranch main
```



Dominando el Control de Versiones



Comandos Básicos de Git

Nuestro primer repositorio

```
MINGW64:/c/Users/ixyz0/Personal/Nueva carpeta
ixyz0@DESKTOP-B1HB7H1 MINGW64 ~/Personal/Nueva carpeta
$ git init
Initialized empty Git repository in C:/Users/ixyz0/Personal/Nueva carpeta/.git/
ixyz0@DESKTOP-B1HB7H1 MINGW64 ~/Personal/Nueva carpeta (main)
$ git add .
ixyz0@DESKTOP-B1HB7H1 MINGW64 ~/Personal/Nueva carpeta (main)
$ git commit -am "HOLA"
[main (root-commit) 2565a8f] HOLA
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Nuevo documento de texto.txt
ixyz0@DESKTOP-B1HB7H1 MINGW64 ~/Personal/Nueva carpeta (main)
$
```

1. Inicializamos un repositorio
2. Añadir archivos a trabajar
3. Hacer un commit

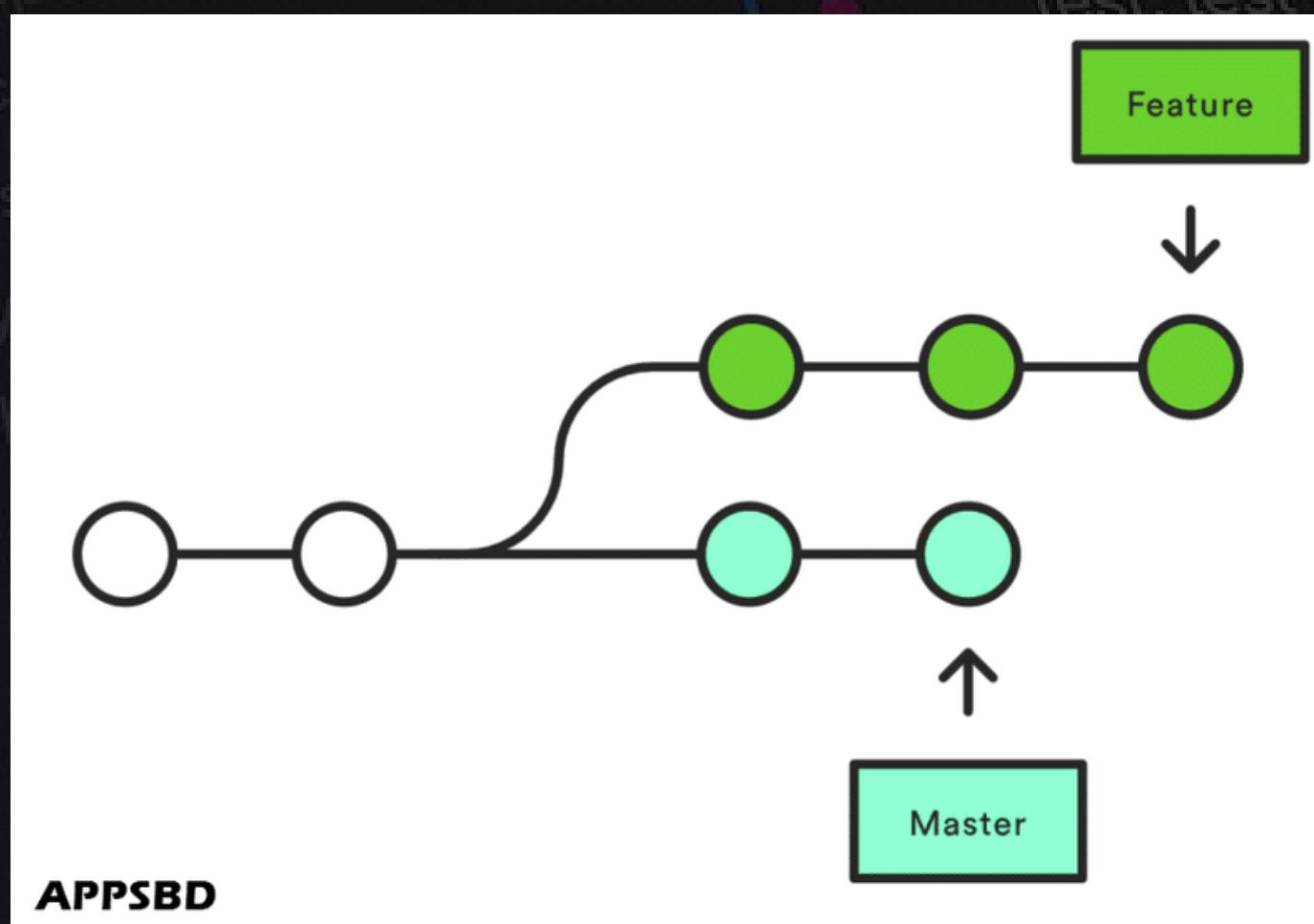


Dominando el Control de Versiones



Trabajando con Ramas

¿Que son las ramas?



En Git, las ramas son diferentes líneas de trabajo que te permiten desarrollar características o correcciones sin afectar la versión principal de tu proyecto.



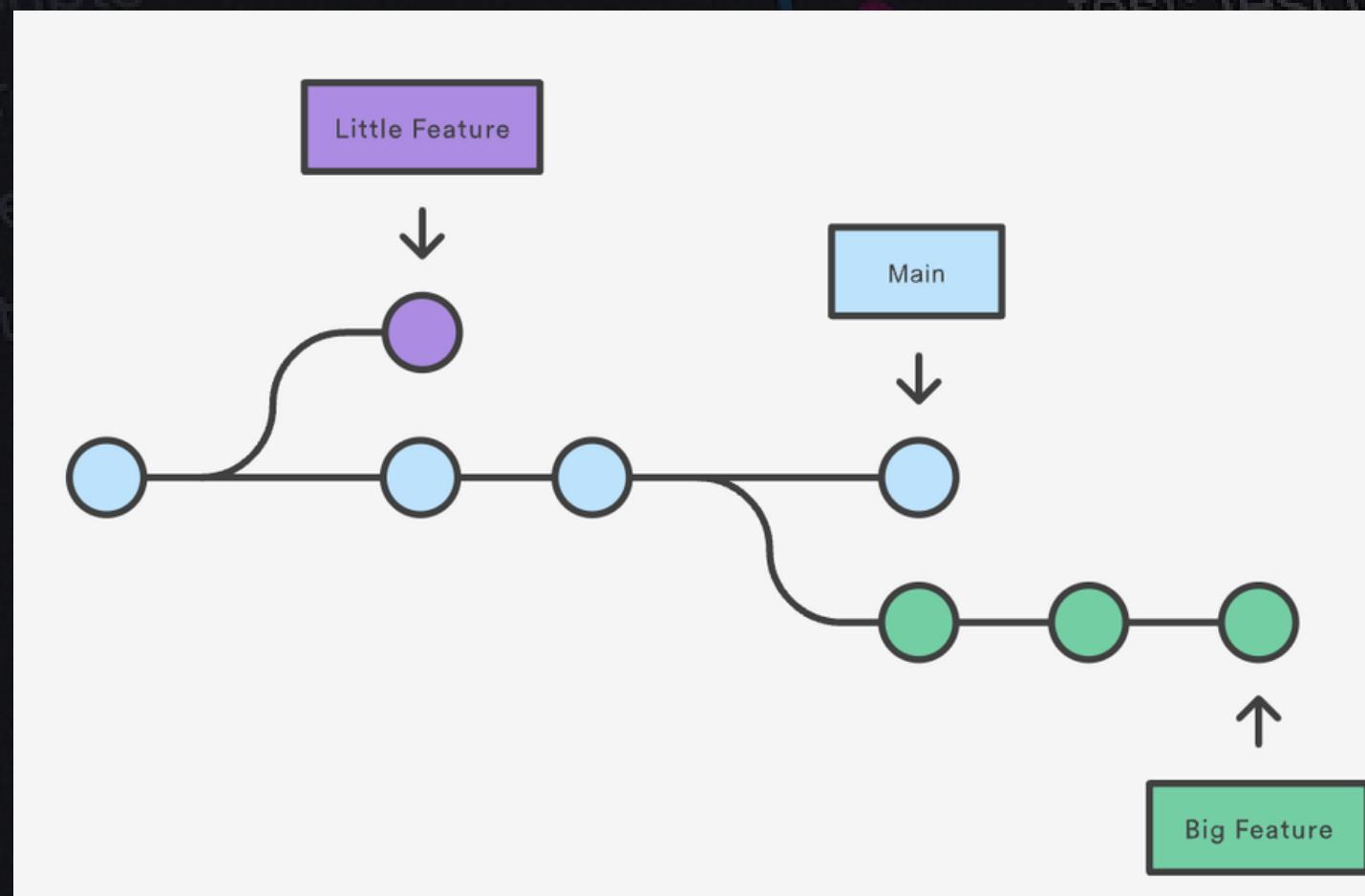
Dominando el Control de Versiones





Trabajando con Ramas

Visualizar y crear ramas



1. Visualizar las ramas disponibles

git branch

2. Crear una rama

git branch **nombre-rama**

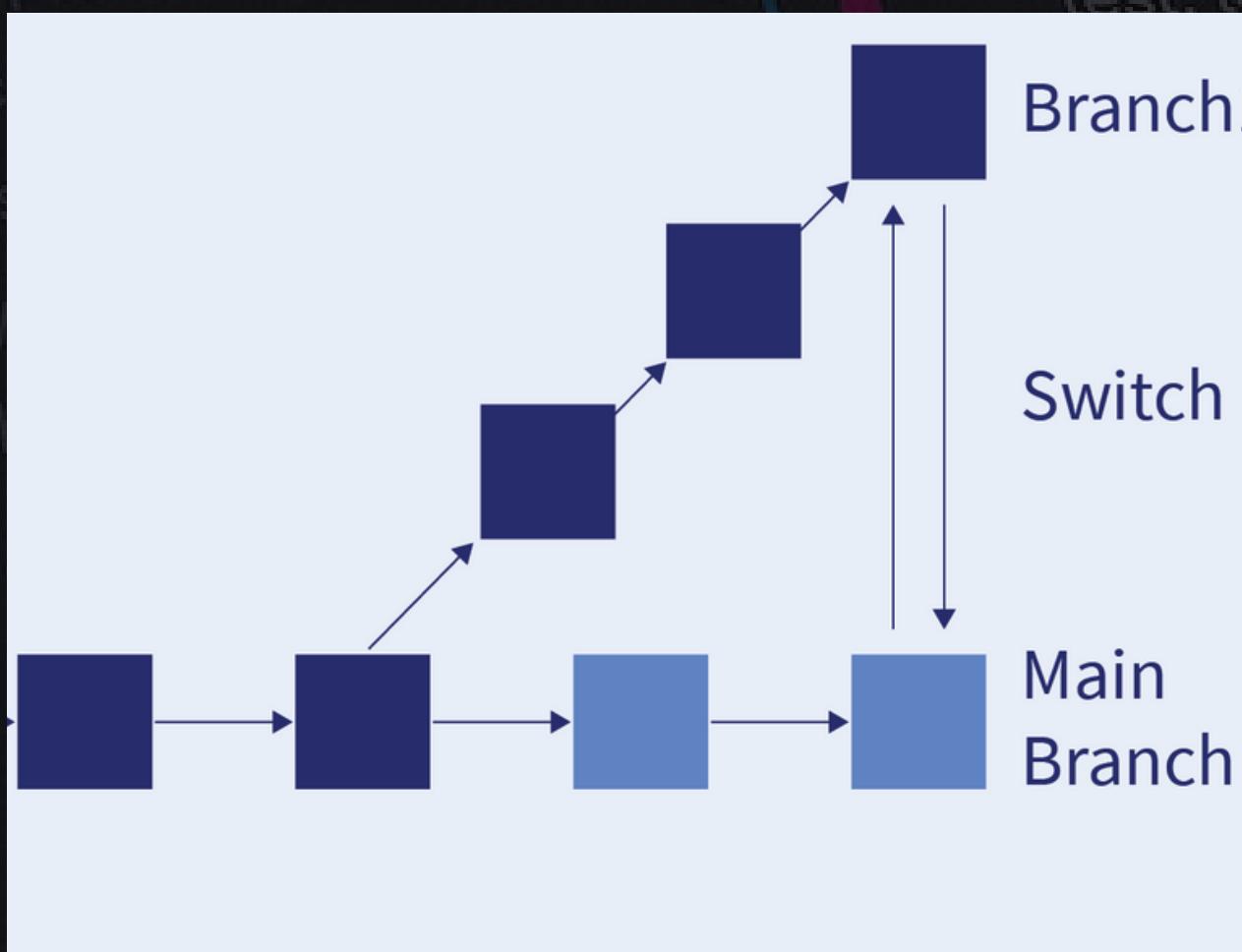


Dominando el Control de Versiones



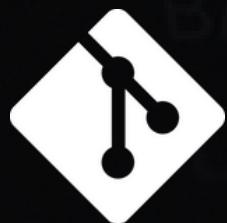
Trabajando con Ramas

Moverse entre ramas



git checkout <rama>.

Movernos en diferentes ramas de tu proyecto

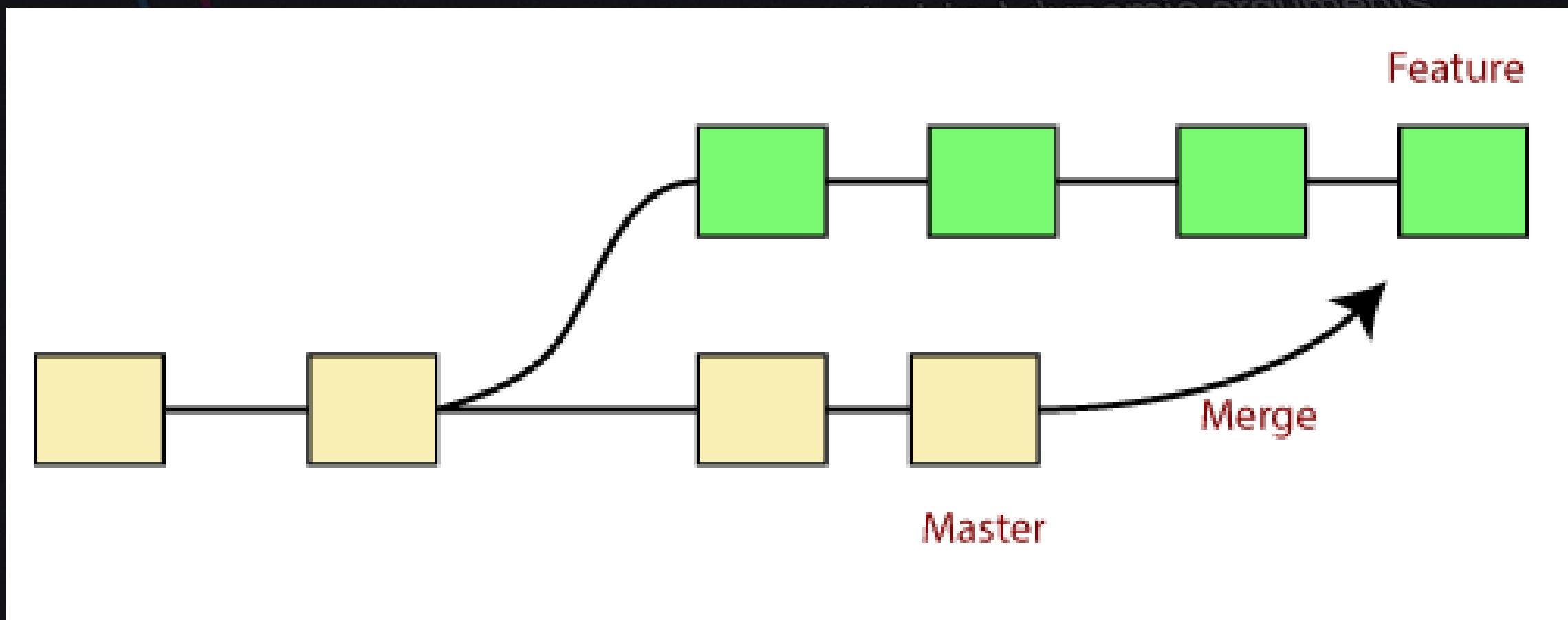


Dominando el Control de Versiones



Trabajando con Ramas

Unión de ramas

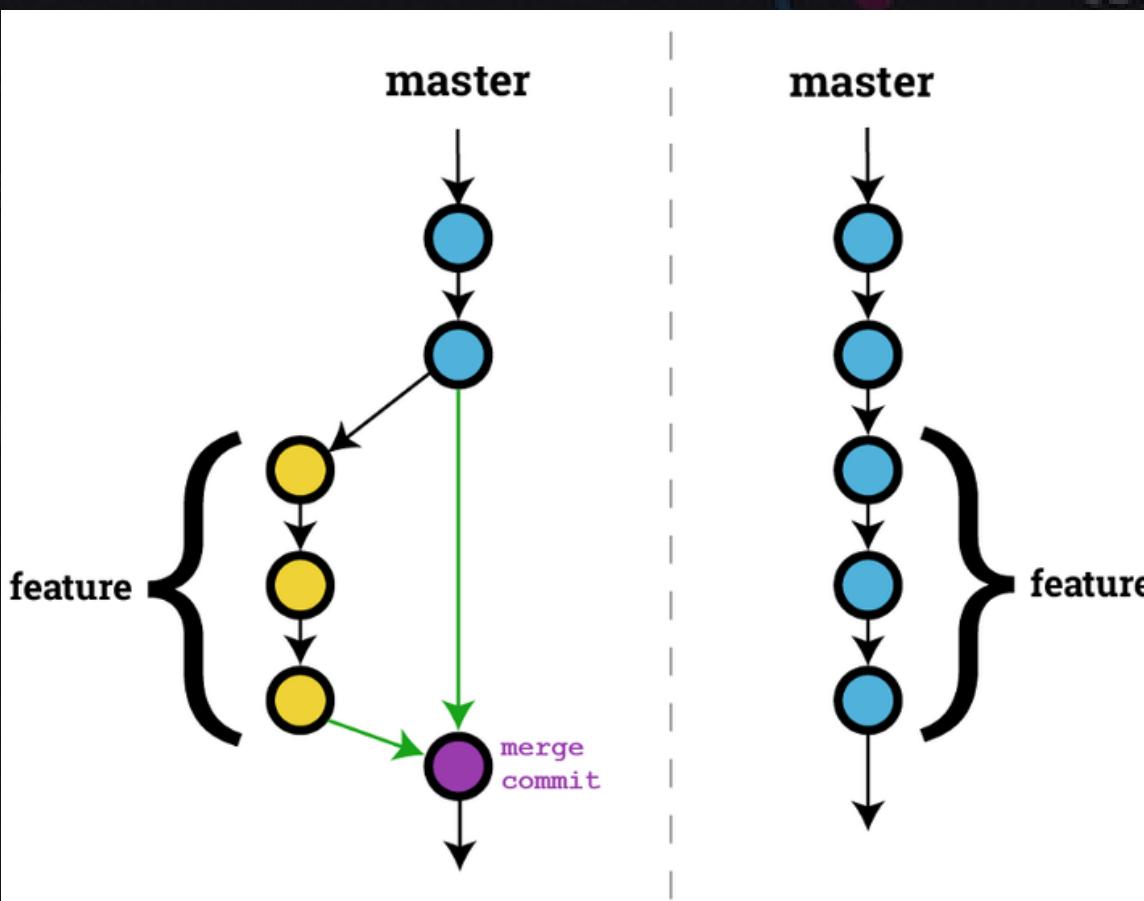


Dominando el Control de Versiones



Trabajando con Ramas

¿Como lo hacemos?



1. Asegurarse que estamos en la rama de origen(master)
2. Ejecutar el comando
git merge rama-destino



Dominando el Control de Versiones



Trabajando con Ramas

Resolver conflictos de fusión.



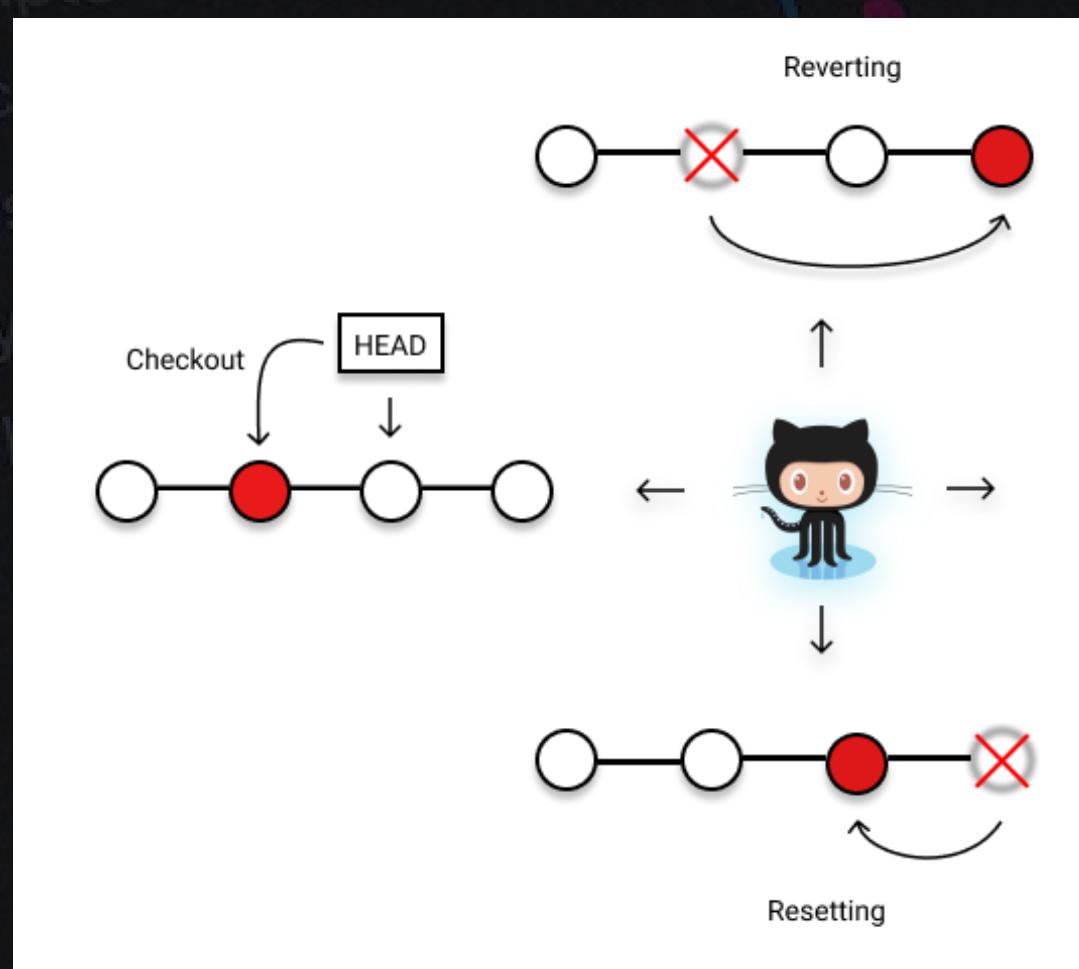
Dominando el Control de Versiones



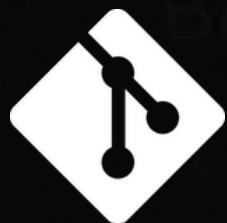


Revertir cambios

Restauremos nuestro código



Existen diversas maneras de deshacer cambios en nuestro código, pero en esta ocasión exploraremos dos de ellas.



Dominando el Control de Versiones

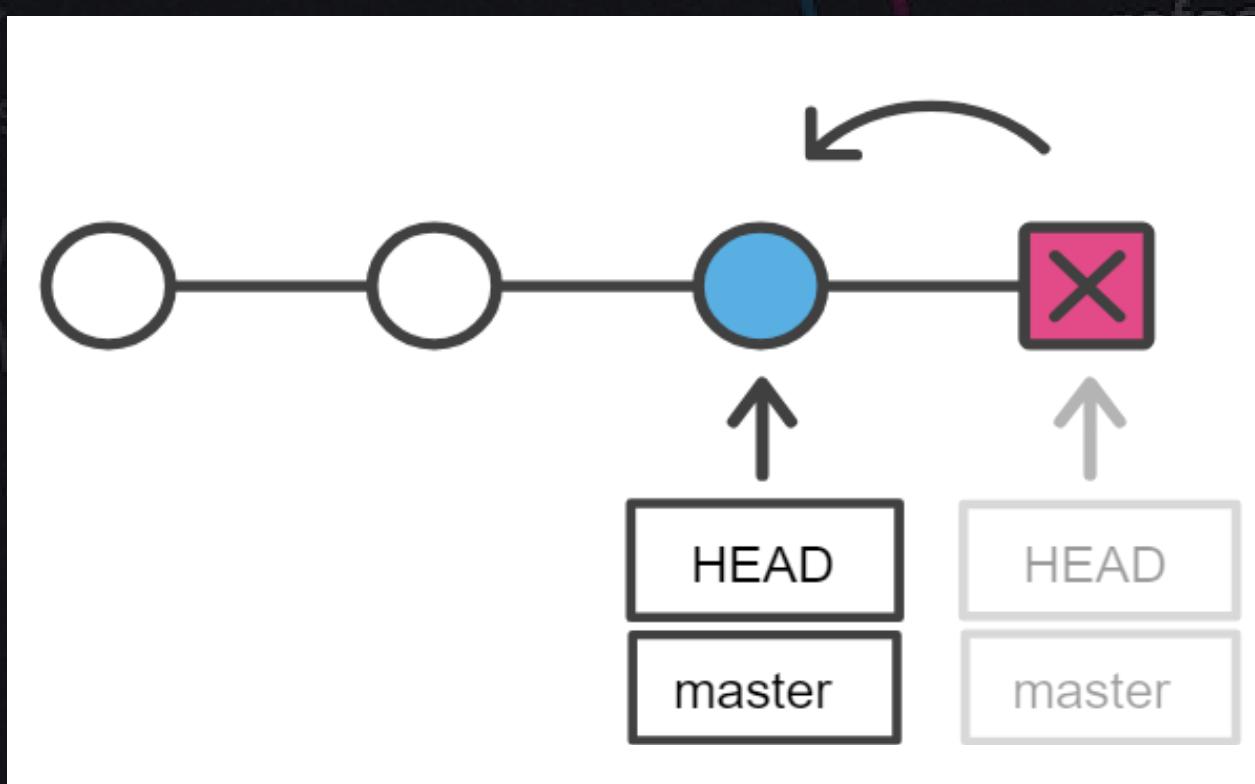
 **FABLUV**

 Universidad
de Valparaíso
CHILE



Revertir cambios

Resetear nuestro código al último commit



Git reset elimina cambios no confirmados y permite retroceder a commits anteriores. No es recomendable retroceder si ya se enviaron cambios al repositorio remoto.

`git reset --hard <hash_del_commit>`



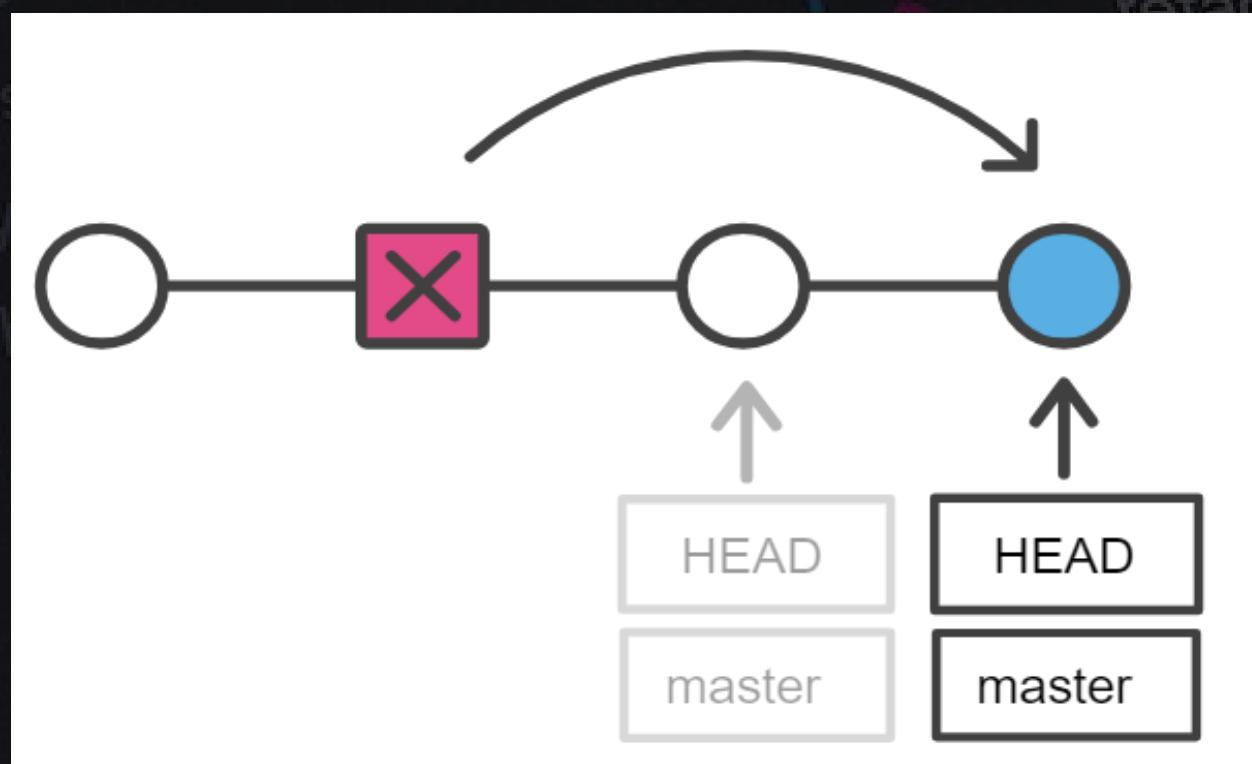
Dominando el Control de Versiones





Revertir cambios

Revertir cambios manteniendo un respaldo



`git revert <hash_del_commit>`

Crea un nuevo commit para deshacer los cambios de un commit anterior, preservando la integridad del historial de cambios

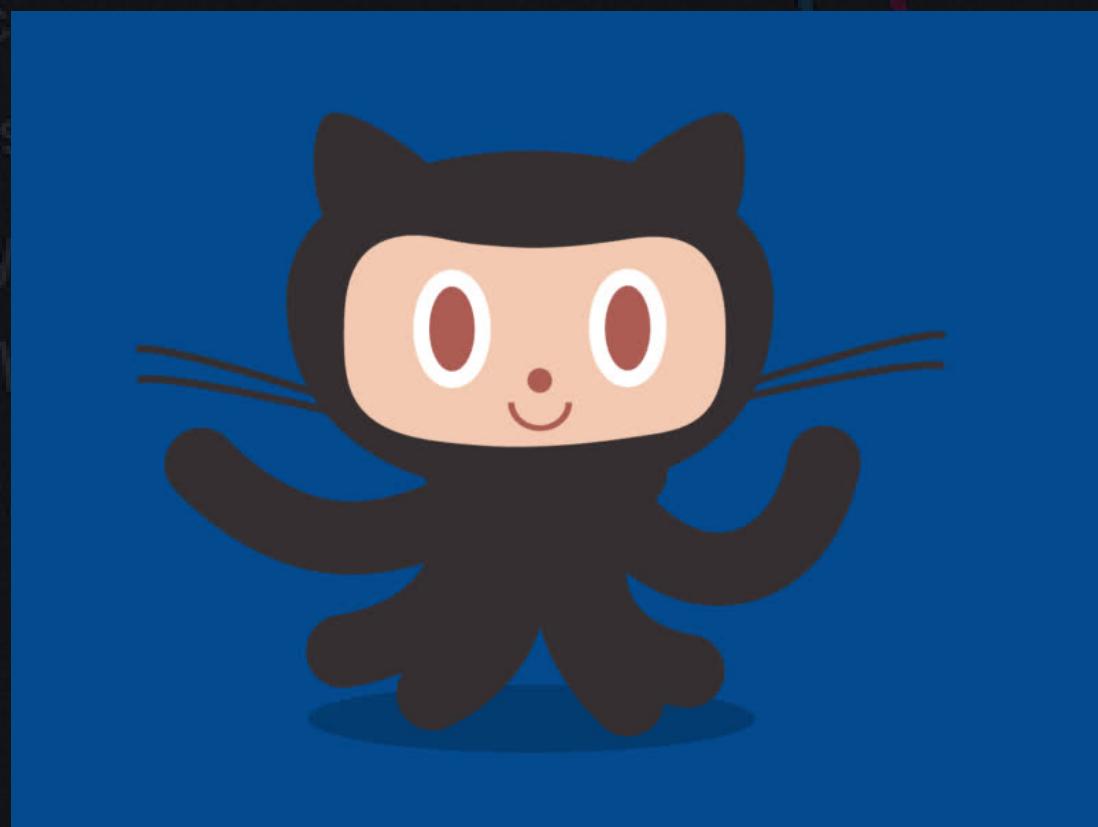


Dominando el Control de Versiones



Introducción a GitHub

¿Qué es GitHub?



GitHub es una plataforma de alojamiento web para el control de versiones de proyectos de desarrollo de software.



Dominando el Control de Versiones





Introducción a GitHub

Creación de un repositorio en GitHub

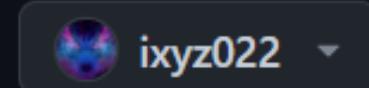
Create a new repository

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 *



Repository name *

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

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



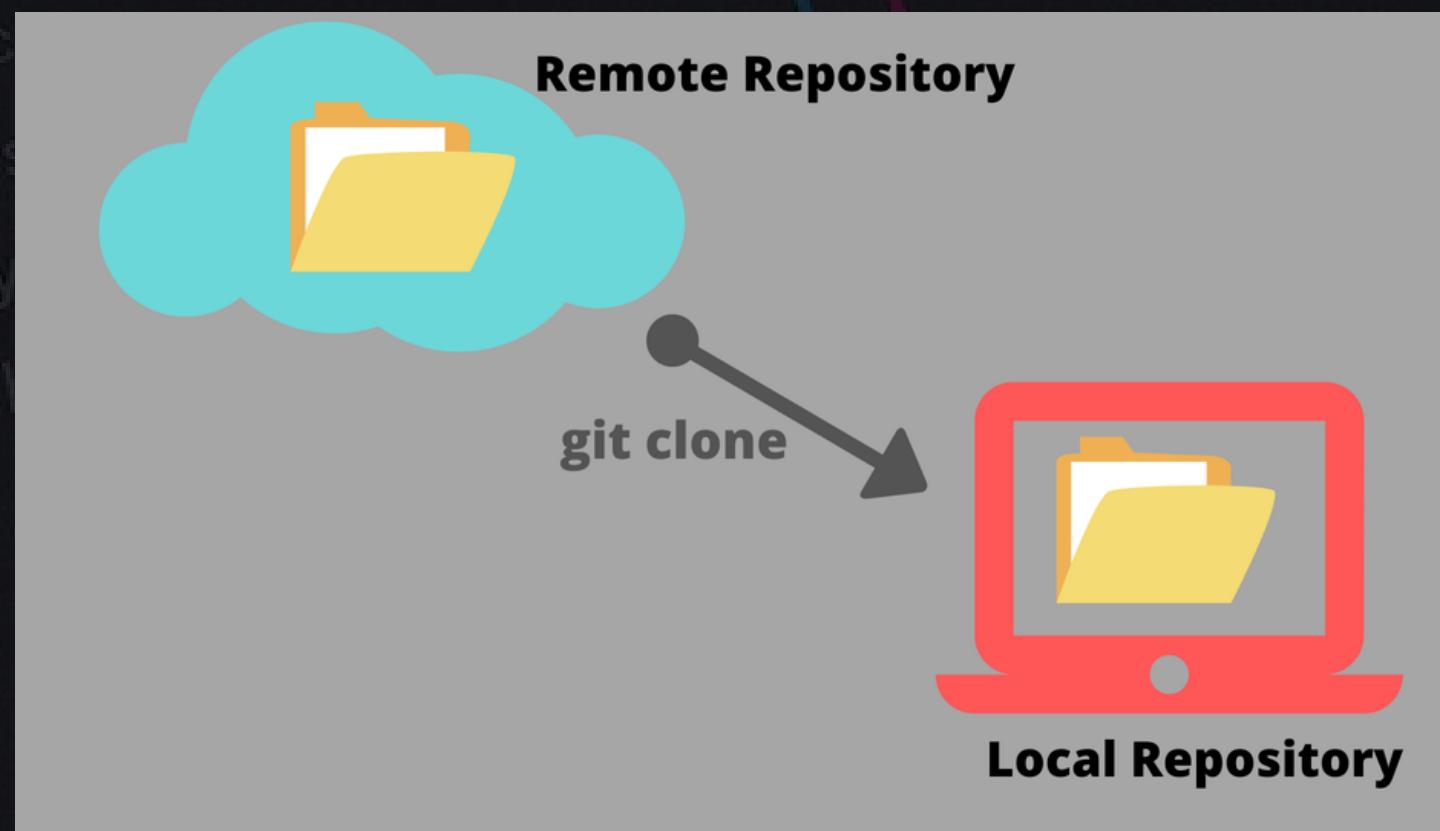
Dominando el Control de Versiones





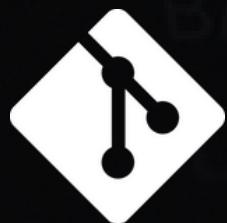
Colaboración con GitHub

Clonar un repositorio remoto



Para clonar un repositorio remoto en Git, utiliza el siguiente comando:

`git clone URL_del_repositorio`



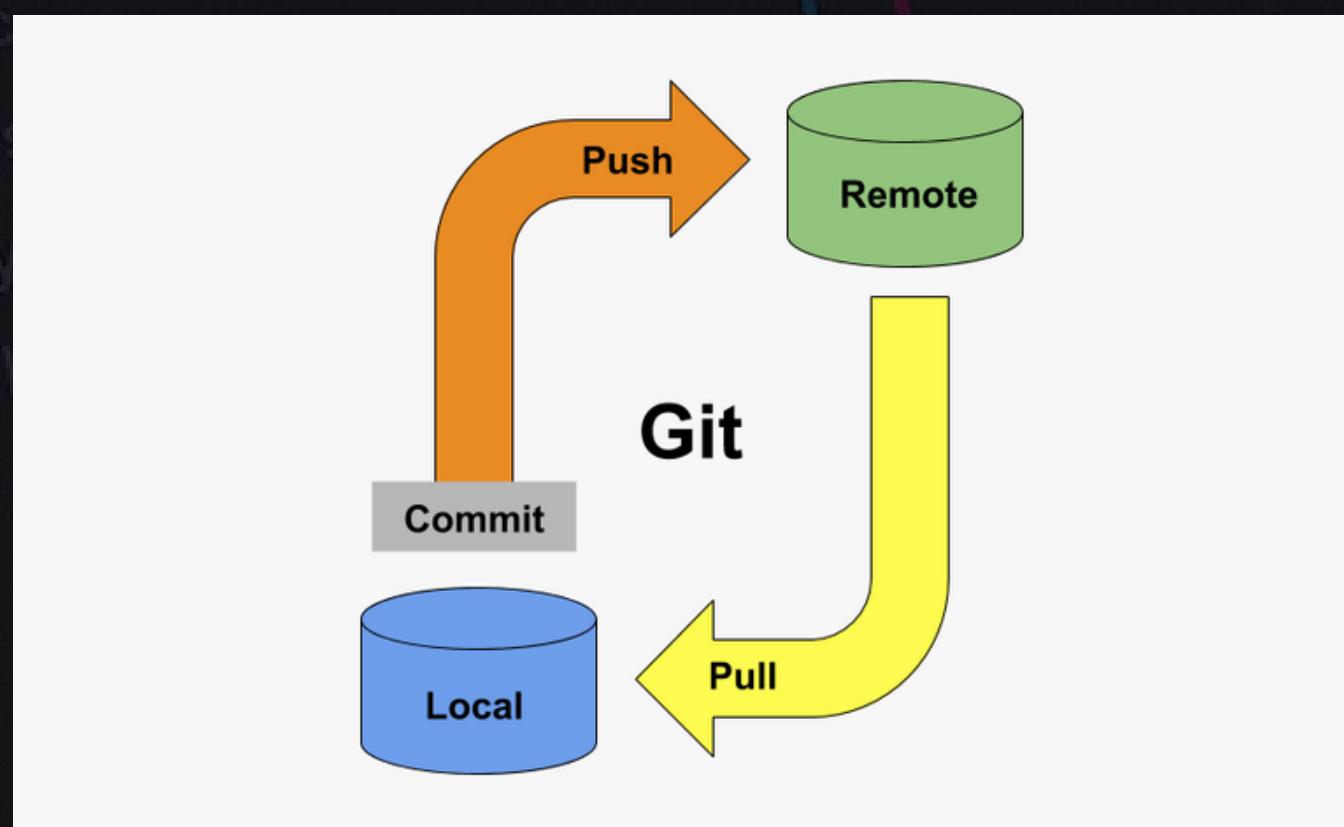
Dominando el Control de Versiones





Colaboración con GitHub

Sincronizar cambios



git push: Enviar los cambios locales de tu repositorio Git a un repositorio remoto

git pull: Traer y fusionar cambios desde un repositorio remoto en tu repositorio local.



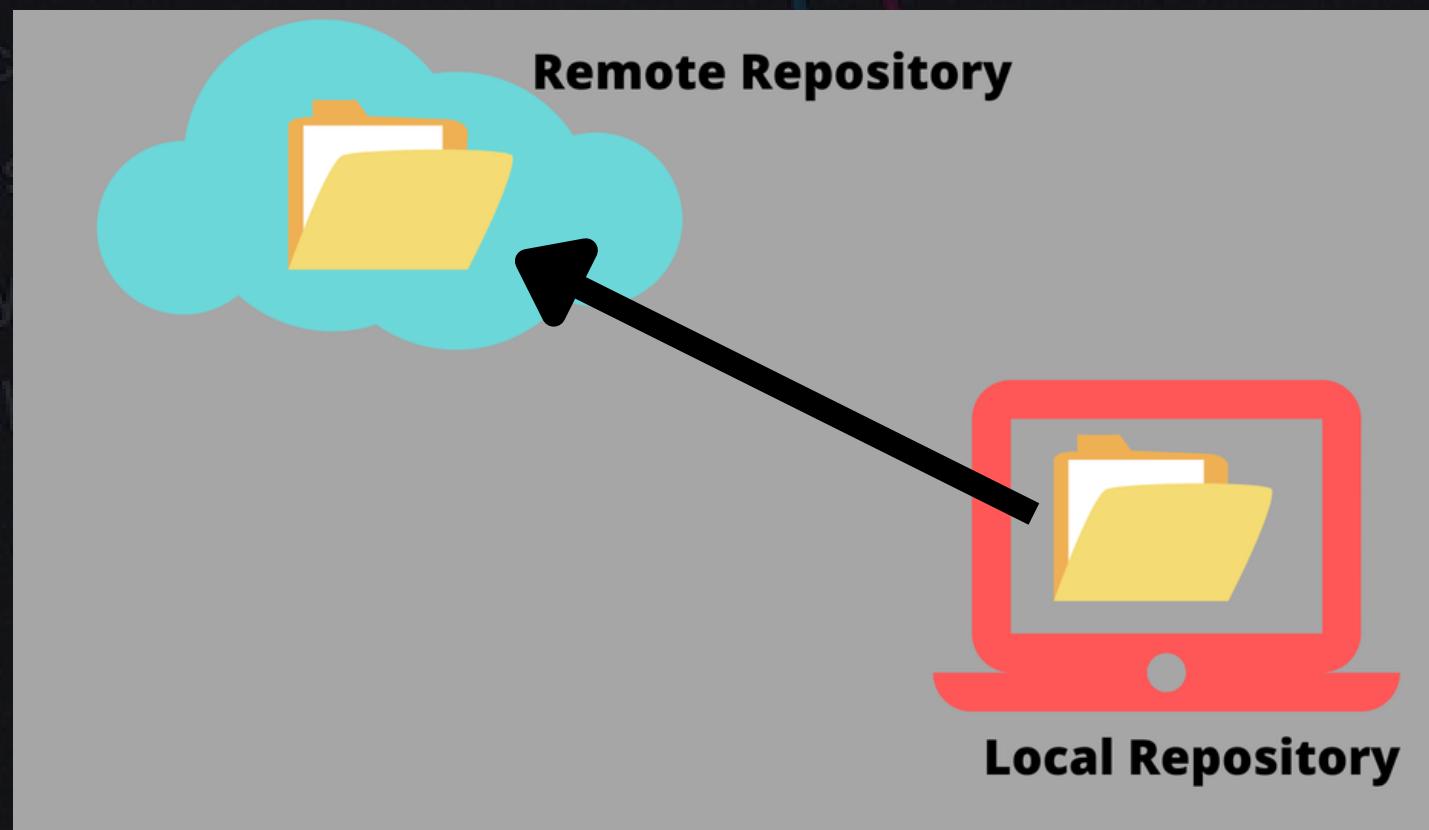
Dominando el Control de Versiones





Colaboración con GitHub

Subir un repositorio local

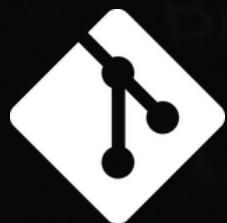


Vincular nuestro repositorio local a
uno remoto en Git

`git remote add origin`

`URL_del_repositorio_remoto`

`git push -u origin main`



Dominando el Control de Versiones





Buenas Prácticas y Recursos

Uso de .gitignore para ignorar archivos no deseados.

• gitignore.io

Create useful .gitignore files for your project

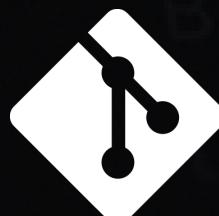
Android IntelliJ

Generate ▾

[Source Code](#)

[Command Line Docs](#)

[Watch Video Tutorial](#)



Dominando el Control de Versiones





Markdown

Documentación de tu repositorio

The image shows a dark-themed code editor window titled "Untitled 2 — Edited". The editor displays the following Markdown content:

```
#Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipisicing elit, quis
**nostrud exercitation** ullamco laboris nisi ut aliquip ex ea
commodo consequat. Duis aute irure dolor in reprehenderit in
*voluptate velit*.

###Code

```javascript
var foo = 'bar';
if(true) foo = 'foo';
```

###Tables

First Header	Second Header
Content from cell 1 | Content from cell 2
Content in the first column | Content in the second column

###Lists

- [x] @mentions, #refs, [links](), **formatting**, and <del>tags</del> supported
- [x] list syntax required (any unordered or ordered list supported)
- [x] this is a complete item
- [ ] this is an incomplete item
```

The rendered output on the right side of the editor shows:

Markdown

Code

```
var foo = 'bar';
if(true) foo = 'foo';
```

Tables

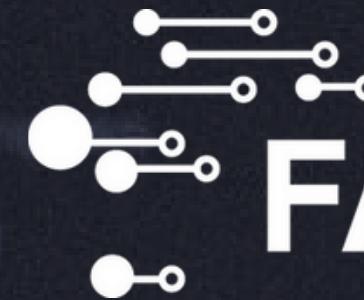
| First Header | Second Header |
|-----------------------------|------------------------------|
| Content from cell 1 | Content from cell 2 |
| Content in the first column | Content in the second column |

Lists

- @mentions, #refs, [links](#), **formatting**, and ~~tags~~ supported
- list syntax required (any unordered or ordered list supported)
- this is a complete item



Dominando el Control de Versiones



FABLAB.UV



Taller de Git: Dominando el Control de Versiones

ALEJANDRO DIAZ CANTILLANO
EST. INGENIERÍA CIVIL INFORMATICA



Extra: Crea tus comandos

```
v2.6.0-beta.2 build: release 2.6.0-beta.2
build: build 2.6.0-beta.2
origin/dynamic-directive-arguments feat: dynamic args for custom directives (#9371)
  detection accuracy (#9371)

MINGW64:/c/Users/ixyz0/OneDrive - alumnos.uv.cl/Trabajo/FabLab/Talleres/Fa...
/FabLab-Taller-Git (main)
$ git lg
* 08db2ba - (8 minutes ago) cambio - Ixyz.dev (HEAD -> main, origin/main, origin
/HEAD)
* 833795b - (9 minutes ago) revirtiendo - Ixyz.dev
* d7945e1 - (14 minutes ago) probando cambios - Ixyz.dev
* dc7db9f - (23 minutes ago) Initial commit - Ixyz

ixyz0@DESKTOP-RGSQGBR MINGW64 ~/OneDrive - alumnos.uv.cl/Trabajo/FabLab/Talleres
/FabLab-Taller-Git (main)
$ git status
On branch main
Your branch is up to date with 'origin/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: README.md

no changes added to commit (use "git add" and/or "git commit -a")

ixyz0@DESKTOP-RGSQGBR MINGW64 ~/OneDrive - alumnos.uv.cl/Trabajo/FabLab/Talleres
/FabLab-Taller-Git (main)
$ |
```

