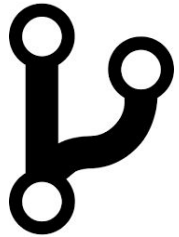


Git - GitHub



Sistema de gestión de
versiones



Inicializar un repo

¿Qué es un repositorio?



1. Iniciando desde una carpeta

```
MINGW64:/c/Expo_Git
Diego@NUC7i3BNHXF MINGW64 /c/Expo_Git
$ pwd
/c/Expo_Git
```

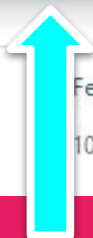
2. Ejecutamos `>git init`

```
MINGW64:/c/Expo_Git
Diego@NUC7i3BNHXF MINGW64 /c/Expo_Git
$ git init
Initialized empty Git repository in C:/Expo_Git/.git/
Diego@NUC7i3BNHXF MINGW64 /c/Expo_Git (master)
```

al (C:) > Expo_Git

Nombre	Fecha de modificaci3n
.git	10/4/2021 12:36

carpeta oculta .git



(master)

Inicializar un repositorio local

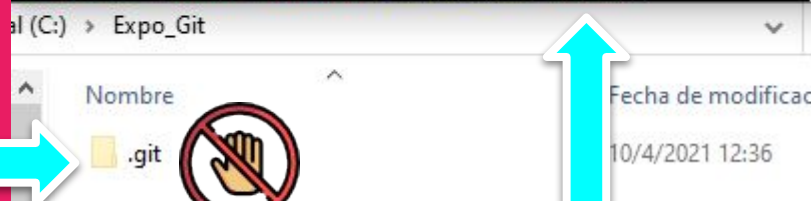


1. Iniciando desde una carpeta

```
MINGW64:/c/Expo_Git
Diego@NUC7i3BNHXF MINGW64 /c/Expo_Git
$ pwd
/c/Expo_Git
```

2. Ejecutamos `>git init`

```
MINGW64:/c/Expo_Git
Diego@NUC7i3BNHXF MINGW64 /c/Expo_Git
$ git init
Initialized empty Git repository in C:/Expo_Git/.git/
Diego@NUC7i3BNHXF MINGW64 /c/Expo_Git (master)
```



carpeta oculta .git

(master)

Repositorio local



NUNCA se debe borrar esa carpeta ni tampoco se debe acceder de forma directa a sus contenidos.

1. Ejecutamos >git config user.name “...”

```
Diego@NUC7i3BNHXF MINGW64 /c/Expo_Git (master)  
$ git config user.name "Diego Díaz"
```

2. Ejecutamos >git config user.email “...”

```
Diego@NUC7i3BNHXF MINGW64 /c/Expo_Git (master)  
$ git config user.name "Diego Díaz"  
  
Diego@NUC7i3BNHXF MINGW64 /c/Expo_Git (master)  
$ git config user.email "diego@dedosmedia.com"
```

3. Globalmente (solo se hace una vez)

```
>git config --global user.name “...”
```

```
>git config --global user.email “...”
```

```
Diego@NUC7i3BNHXF MINGW64 /c/Expo_Git (master)  
$ git log  
commit ab89a3bdfcb9e166e160a854671f7c5f39cd595b (HEAD -> master)  
Author: Diego Díaz <diego@dedosmedia.com>  
Date:   Sat Apr 10 14:50:37 2021 -0500  
  
    Nuestro primer commit al repo
```

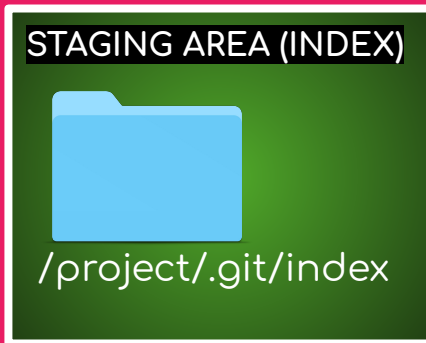
Configurando el usuario



Git



Los archivos que
ves en tu disco



Una zona intermedia



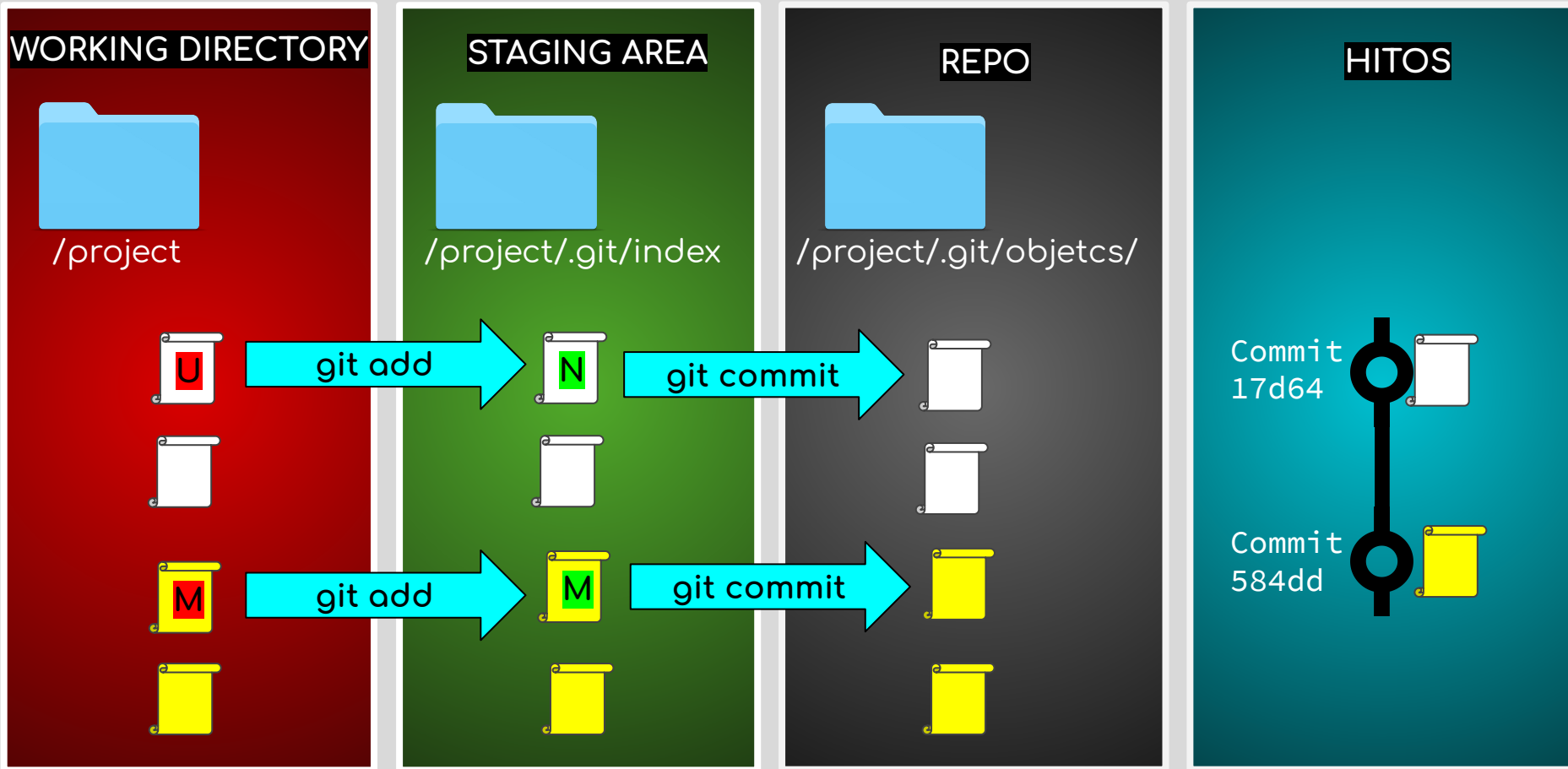
La historia de cambios

Podemos ver un repo como si fueran 3 carpetas

donde los archivos se van moviendo entre cada una de ellas



Flujo de trabajo entre las diferentes áreas



>git add .

>git add archivo1 archivo2 archivo3 ...

```
Diego@NUC7i3BNHXF MINGW64 /c/Expo_Git (master)  
$ git add .
```

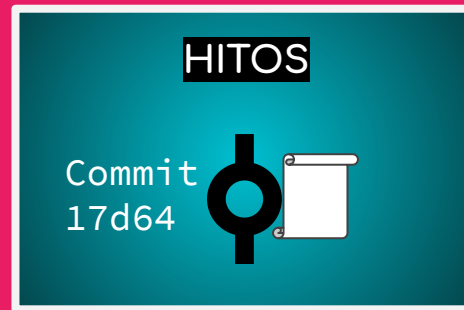
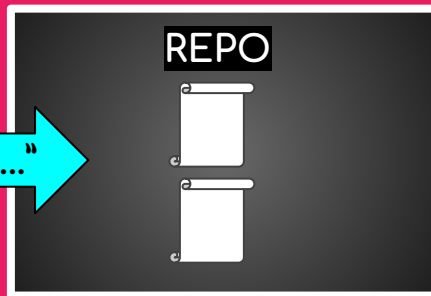
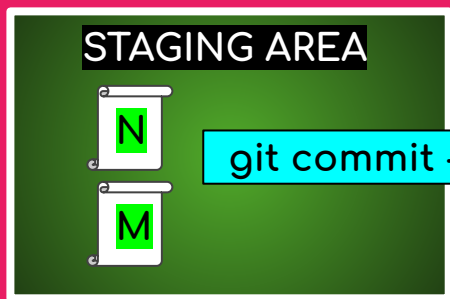


Agregamos archivos nuevos o modificados



>git commit -m "¿Uriel dónde estás?"

```
Diego@NUC7i3BNHXF MINGW64 /c/Expo_Git (master)
$ git commit -m "¿Uriel dónde estás?"
[master 3fef307] ¿Uriel dónde estás?
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 uriel.txt
```



Confirmando los cambios
se crea un hito en el repositorio



>git status

WORKING DIRECTORY



```
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to add to the staging area)
nuevo.txt
```

```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to add to the staging area)
  (use "git checkout -- <file>..." to discard changes in working directory)
modified:   nuevo.txt
```

STAGING AREA



```
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
new file:   nuevo.txt
```

```
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
modified:   nuevo.txt
```

Revisando el estado del repositorio local
podemos ver si han habido cambios en los archivos





GitHub

**GitHub, repositorios remotos y como subir
archivos.**

GitHub

¿Qué es? ¿Para qué sirve?

Es un sitio web, para almacenar los archivos y proyectos de manera gratuita.



GitHub

¿Cómo funciona?

A través de **repositorios**. Creo uno por proyecto. Aca vamos a crear los remotos



CREACIÓN DE UN REPOSITORIO REMOTO



valenanton-26

Edit profile

Joined 22 days ago

Overview

Repositories 6

Projects

Packages

Popular repositories

[Primer-Repo-IntroInf](#)

Primer repositorio creado de prueba para Introduccion a la informatica

[Creando-Repositorios](#)

Clase 4, creando repositorios en Git Hub

[Prueba](#)

Ejercicio del Playground para subir archivos

[Repo](#)

Repositorio en clase

[Front-End-1](#)

Carpeta de Front-End 1 del curso

HTML

[valenanton-26.github.io](#)

HTML

New repository

Import repository

New gist

New organization

New project

36 contributions in the last year

Contribution settings

Create a new repository

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

[Import a repository.](#)

Owner *



valenanton-26 ▾

Repository name *

/

**Elijo un nombre para el repositorio
Que sea claro**

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

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.

**Puedo poner una
descripción para
agregarle detalle al
repositorio**

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

☐ Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

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

Set up in Desktop

or

HTTPS

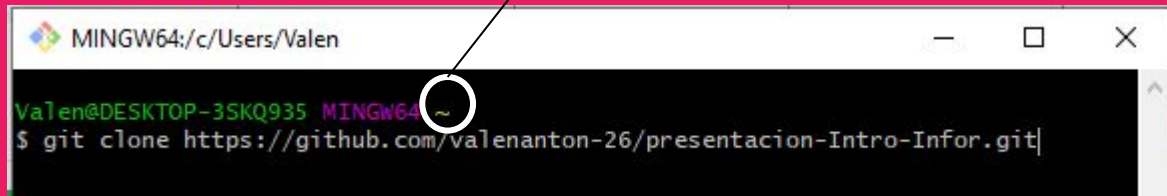
SSH

<https://github.com/valenanton-26/presentacion-Intro-Infor.git>



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

Donde aparece la carpeta de igual nombre



A terminal window titled 'MINGW64:/c/Users/Valen' with standard window controls. The prompt is 'Valen@DESKTOP-3SKQ935 MINGW64 ~'. The command '\$ git clone https://github.com/valenanton-26/presentacion-Intro-Infor.git' is entered. A white circle highlights the tilde '~' in the prompt, with an arrow pointing from the text 'Donde aparece la carpeta de igual nombre' to it.

```
MINGW64:/c/Users/Valen
Valen@DESKTOP-3SKQ935 MINGW64 ~
$ git clone https://github.com/valenanton-26/presentacion-Intro-Infor.git
```

Git clone

comandos

PRIMERO

Git pull origin main/master

SEGUNDO

Git push origin main/master



GitHub

Enlazar con un repositorio remoto



1. Ejecutamos

>git remote add origin "url al repo remoto"

```
Diego@NUC7i3BNHXF MINGW64 /c/Expo_Git (master)  
$ git remote add origin "http://github.com/repo"
```

2. Ejecutamos

>git remote -v

```
Diego@NUC7i3BNHXF MINGW64 /d/Dropbox (Personal)/DedosMedia/Digital House/Año 1/B  
imestre 1/Introduccion Informatica/Mochila/Cursada (main)  
$ git remote -v  
origin https://github.com/Digital-House-0321TDIIFM1C6LAED/Cursada.git (fetch)  
origin https://github.com/Digital-House-0321TDIIFM1C6LAED/Cursada.git (push)
```

Enlazando repo local y remoto

Para poder sincronizar el repositorio local con un repositorio en la nube. (Poder ejecutar pull/push)

Ejecutar `>git help remote`

GIT-REMOTE(1)

Git Manual

GIT-REMOTE(1)

NAME

`git-remote` - Manage set of tracked repositories

SYNOPSIS

```
git remote [-v | --verbose]  
git remote add [-t <branch>] [-m <master>] [-f] [--[no-]tags] [--mirror=<fetch|push>] <name> <url>  
git remote rename <old> <new>  
git remote remove <name>  
git remote set-head <name> (-a | --auto | -d | --delete | <branch>)  
git remote set-branches [--add] <name> <branch>...  
git remote get-url [--push] [--all] <name>  
git remote set-url [--push] <name> <newurl> [<oldurl>]  
git remote set-url --add [--push] <name> <newurl>  
git remote set-url --delete [--push] <name> <url>  
git remote [-v | --verbose] show [-n] <name>...  
git remote prune [-n | --dry-run] <name>...  
git remote [-v | --verbose] update [-p | --prune] [(<group> | <remote>)...]
```

DESCRIPTION

Manage the set of repositories ("remotes") whose branches you track.



Resolver conflictos

```
remote: Enumerating objects: 5, done.  
remote: Counting objects: 100% (5/5), done.  
remote: Compressing objects: 100% (2/2), done.  
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0  
Desempaquetando objetos: 100% (3/3), listo.  
Desde https://github.com/codelando/git-conflicts  
    e7e0905..3216f3f  master    -> origin/master  
Auto-fusionando styles.css  
CONFLICTO (contenido): Conflicto de fusión en styles.css  
Fusión automática falló; arregle los conflictos y luego realice un commit con el resultado.
```

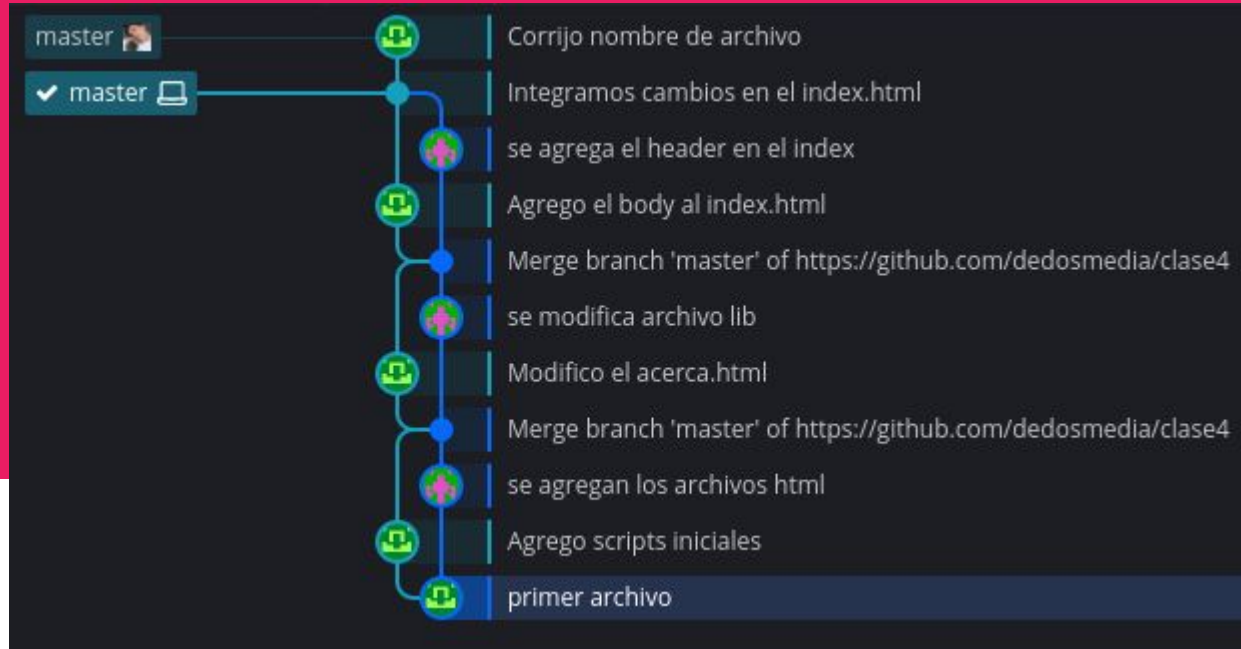


```
body {  
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes | Start Live Share Session  
<<<<<<<·HEAD (Current Change)  
  background-color: blue;  
=====  
  background-color: yellow;  
>>>>>>>·ca4ee67bbd25e4a5f77c9ed42ccec86d5e22a279 (Incoming Change)  
  font-family: monospace;  
}
```

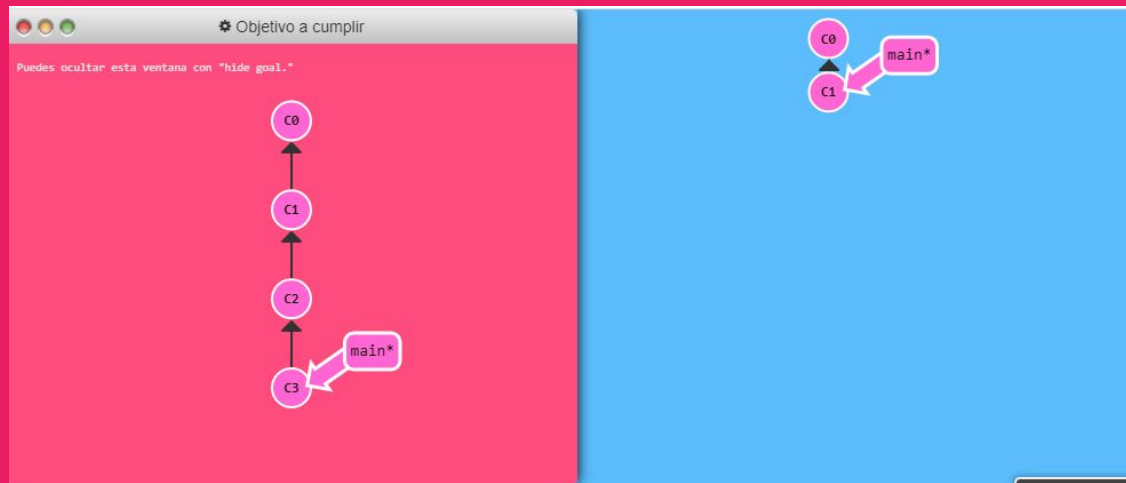
**Recuerda
siempre borrar
las líneas de
ayuda**

Visual Studio Code nos
ayuda a entender los errores
y tomar decisiones





Para practicar!



https://learngitbranching.js.org/?locale=es_ES

Practicar los conceptos vistos y además el concepto de branches (ramas)

