

# Introducción

## A git, Github y Github classroom

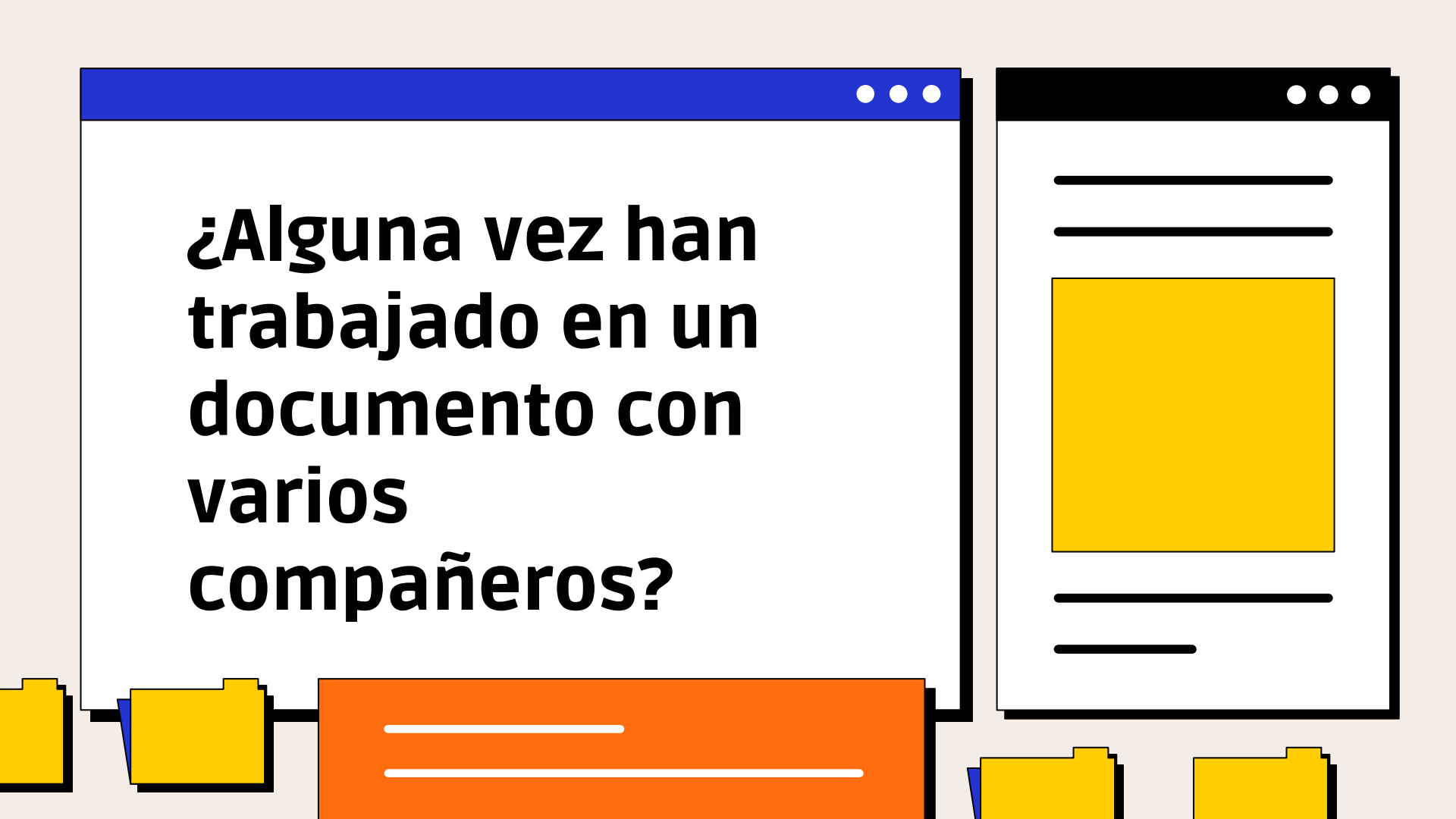
Alejandro Garcés  
Ramírez



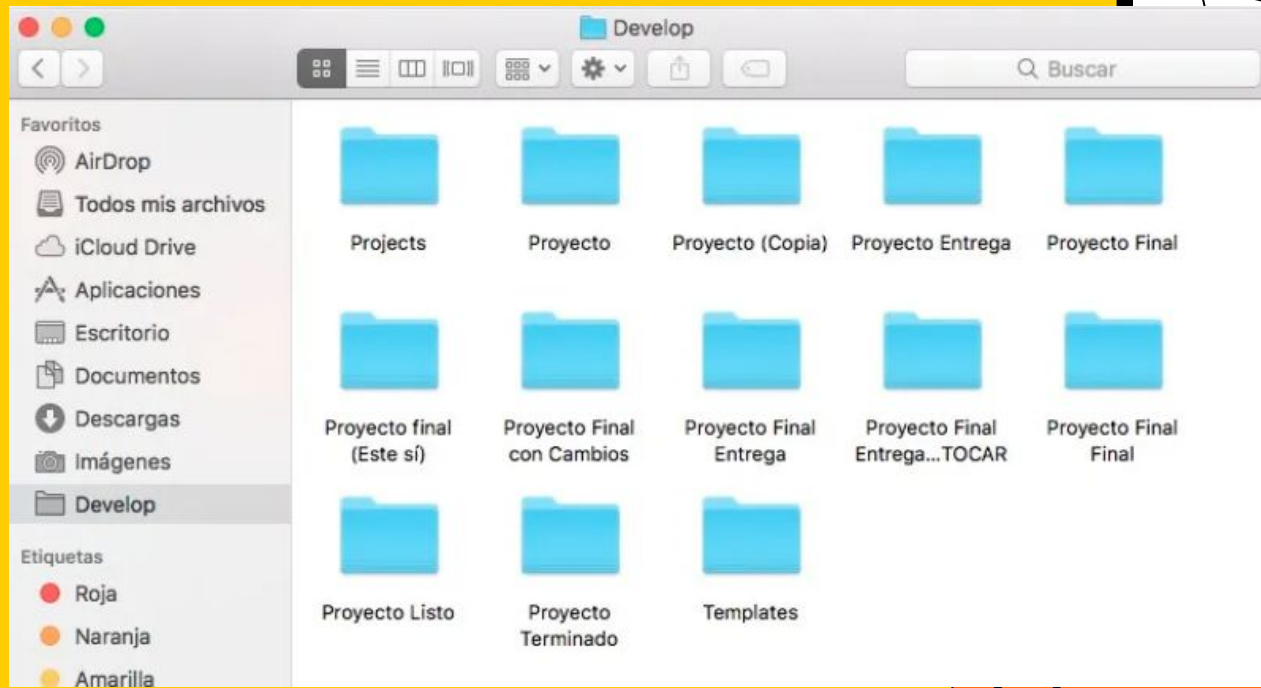
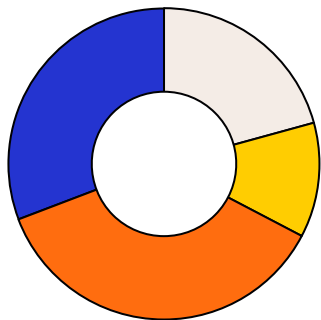
# Contenidos

## Guía de la presentación

<b>Introducción</b>	Presentación y preguntas
<b>Control de versiones</b>	Definición y tipos
<b>git</b>	Ventajas, conceptos básicos y actividad
<b>Github</b>	Ventajas, conceptos básicos y actividad
<b>Github classroom</b>	Ventajas, conceptos básicos y actividad
<b>Dudas-Despedida</b>	



**¿Alguna vez han  
trabajado en un  
documento con  
varios  
compañeros?**





# Control de versiones

Es un sistema que  
registra los cambios  
realizados sobre un  
archivo o conjunto de  
archivos a lo largo del  
tiempo

01

# Pequeña ilustración

**Cambio 1**

**Cambio 2**

**Cambio 3**



Index.html



Index.html



Index.html

<DOCTYPE html>

<DOCTYPE html>  
<html>  
<body>  
</body>  
</html>

<DOCTYPE html>  
<html>  
<head>  
...  
</head>  
<body>  
...  
</body>  
</html>

# Control de versiones



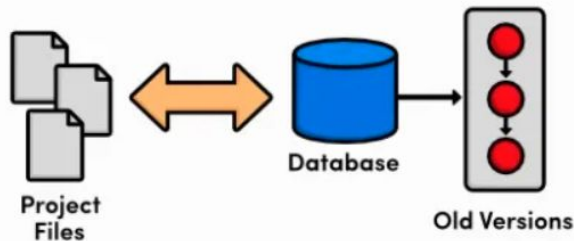
- Permite ver el historial completo de modificaciones en el proyecto.
- Facilita revertir a versiones anteriores si se cometen errores.
- Proporciona una copia de seguridad del proyecto en caso de pérdida de datos locales.
- Facilita el seguimiento de quién hizo qué cambios y cuándo.



Se divide en 3 tipos: Sistemas de Control de Versiones Locales, Centralizados y Distribuidos.

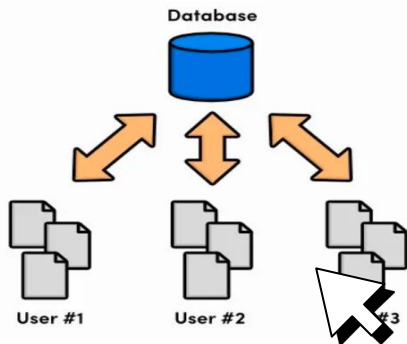


# Tipos



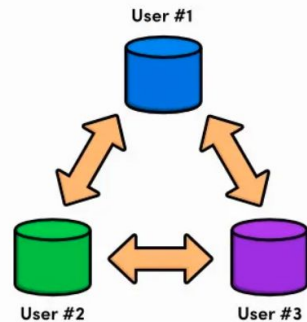
## Locales

Los cambios se guardan únicamente en la máquina local del usuario.



## Centrados

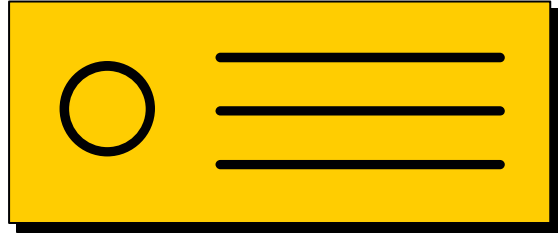
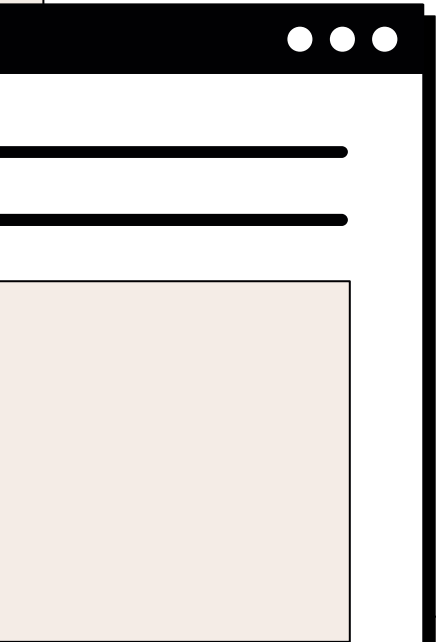
En vez de almacenar los cambios y versiones en el disco duro de los desarrolladores, estos se almacenaban en un servidor.



## Distribuidos

Optó por darle a cada desarrollador una copia local de todo el proyecto, en la que cada desarrollador podía trabajar de manera aislada





# ¿Qué es Git?

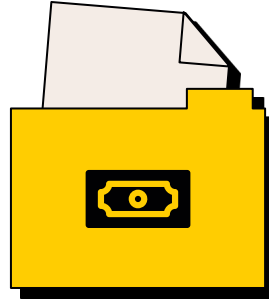
Es un sistema de control de versiones distribuido.

# ¿Por qué Git?



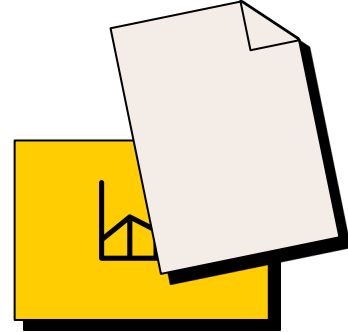
## Control granular

Ofrece herramientas para ver diferencias entre commits, ver historiales detallados y revertir cambios específicos.



## Distribuido

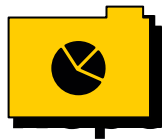
Cada desarrollador tiene una copia completa del repositorio, lo que permite trabajar sin necesidad de estar conectado a internet.



## Integración

Se integra bien con plataformas como GitHub, GitLab y Bitbucket, que ofrecen herramientas adicionales para colaboración.

# Conceptos básicos



## Repositori

0

Análogo a una caja fuerte.



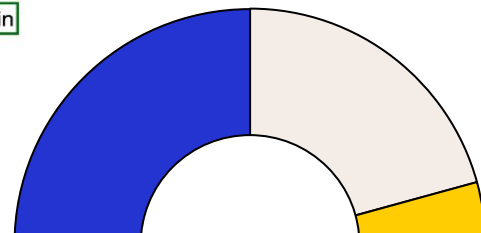
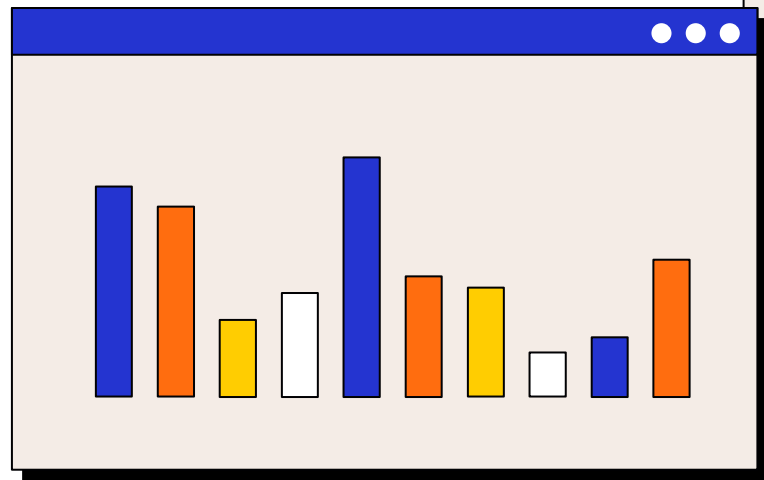
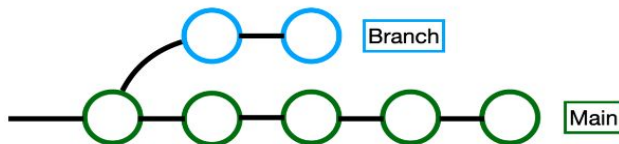
## Estados

Etapas del guardado.



## Ramas

Son como caminos paralelos en un proyecto. Puedes experimentar en ellos.

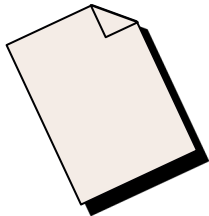


# Estados de los archivos



## Modificado

Se cambiado el archivo, pero aún no se ha guardado en el repositorio.



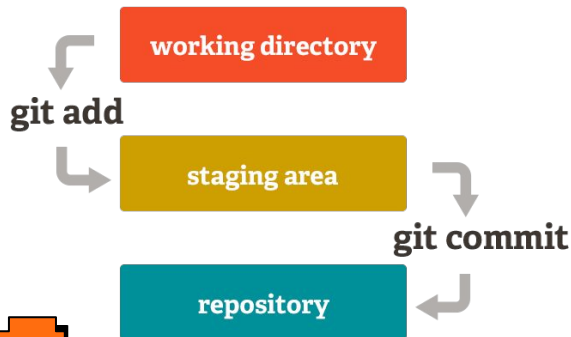
## Preparado

Se ha marcado el archivo para que se guarde en el próximo commit.



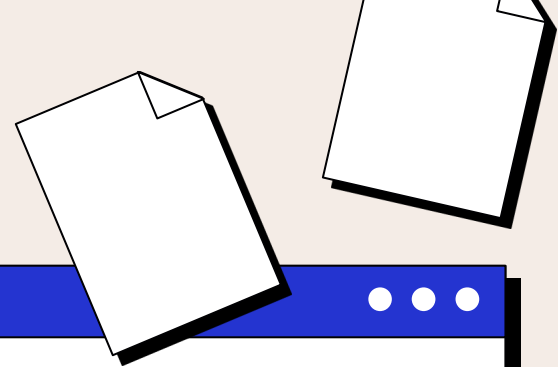
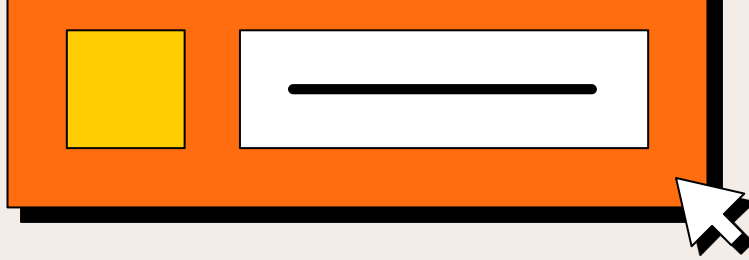
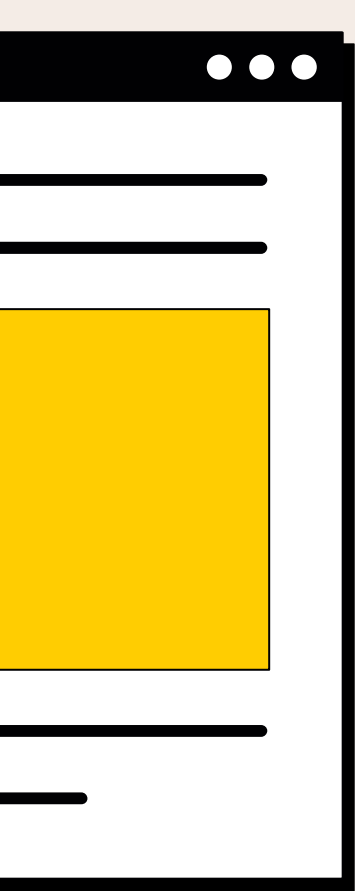
## Confirmado

El archivo se ha guardado de forma segura en el repositorio.





# Actividad



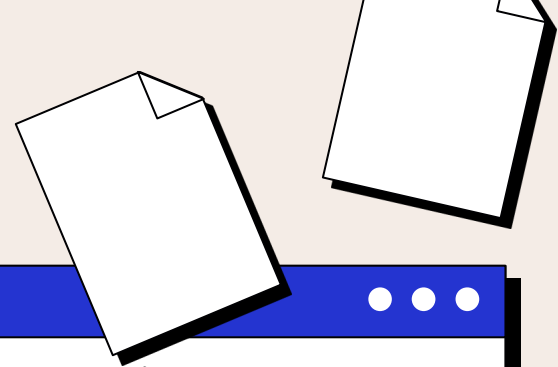
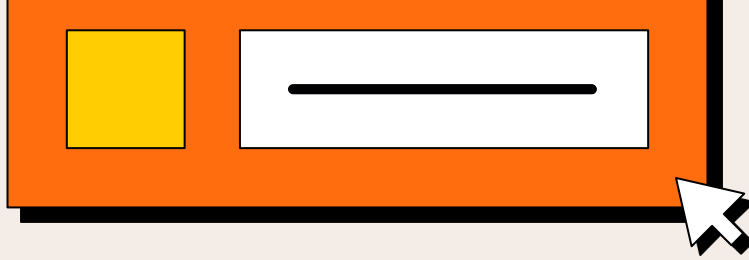
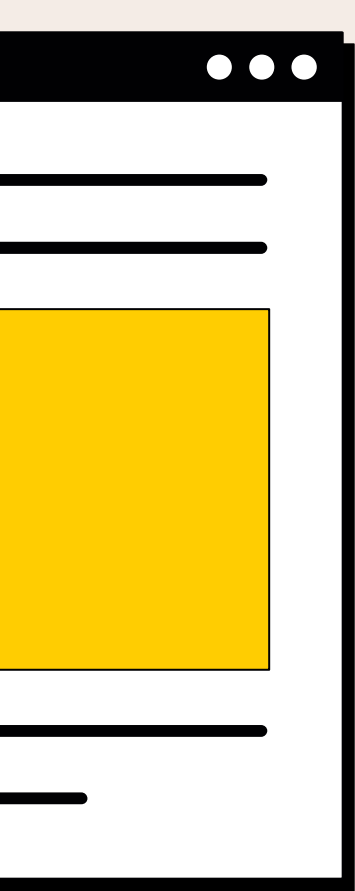
```
$ git config --global user.name "Su usuario"
$ git config --global user.email "Su correo"

$ cd Desktop
$ mkdir pruebaGit
$ cd pruebaGit

$ git init
$ touch miPrimerCommit.txt
$ git status

$ git add miPrimerCommit.txt
$ git commit -m "Este es mi primer commit"
$ git status
```





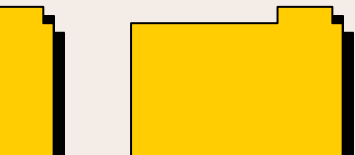
Ahora hacemos un cambio en el archivo miPrimerCommit.txt

```
$ git status  
$ git add miPrimerCommit.txt  
$ git commit -m "Mi primera modificación"
```

```
$ git log  
$ git log --graph --decorate --all --oneline  
$ git checkout ###  
$ git checkout master
```

Creamos una rama

```
$ git checkout -b nueva-rama
```





Hacemos un cambio en el archivo miPrimerCommit.txt

```
$ git status
```

```
$ git add miPrimerCommit.txt
```

```
$ git commit -m "Mi primera modificación en otra rama"
```

Unimos las 2 ramas

```
$ git checkout master
```

```
$ git merge nueva-rama
```

Podemos eliminar la otra rama con:

```
$ git branch -d nueva-rama
```



The background is a stylized illustration of a computer desktop. It features several overlapping windows and icons. In the top left, there's a yellow rectangular icon with a circle and three horizontal lines. Below it, on the left, is a window with a black title bar and three white dots. In the center, there's a large yellow square with the number '03' in black. To its right is a larger window with a blue title bar and three white dots, containing the main text. At the bottom left, there's an orange folder icon. A white mouse cursor is pointing at the bottom right corner of the yellow square with '03'.

**03**

# ¿Qué es Github?

Es como una red social para programadores. Puedes compartir tu código, colaborar con otros, ver las contribuciones de los demás.



# Conceptos clave



# Pull request

Es una petición para fusionar cambios de una rama a otra

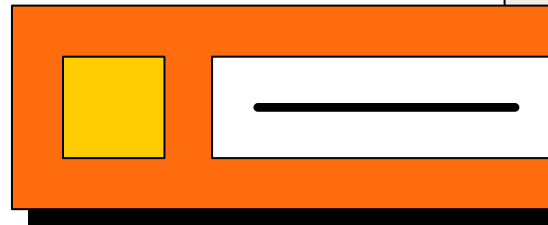
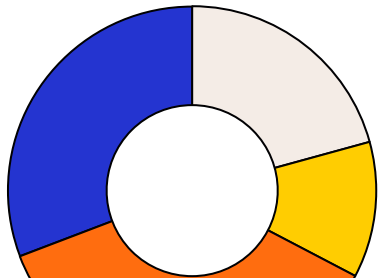


# Fork

Es una copia de un repositorio que se crea en tu cuenta de GitHub.


# Issue

Son tickets que se utilizan para gestionar tareas, reportar bugs, etc





Recorrido y actividad



Clonar el siguiente repositorio y visualizar lo contenido en el:

<https://github.com/AGR-23/Introduction-to-git-and-Github.git>

¿Cómo?

\$ git clone URL



A stylized graphic featuring several overlapping windows and folders. At the top left is a yellow rectangular window with a black circle icon and three horizontal lines. Below it is a black window with three white dots. To the right is a large light gray window with a blue header bar containing three white dots. In the center is a yellow square with the number '04' and a white mouse cursor arrow pointing at its bottom right corner. Below the yellow square is an orange folder icon. On the far left is a light gray rectangular area.

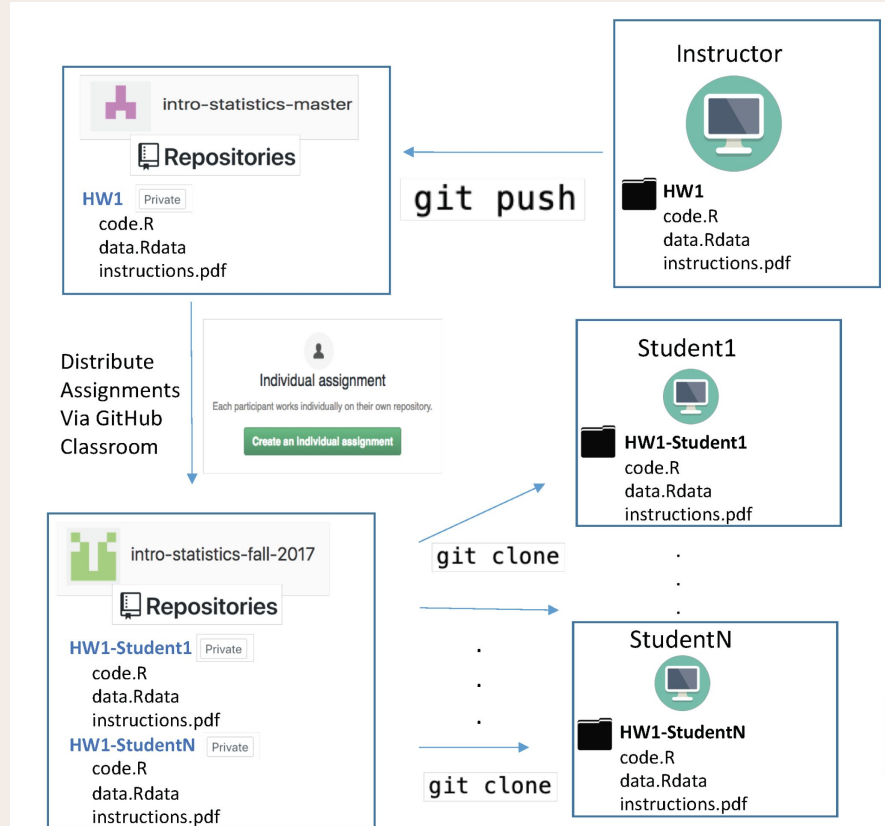
**04**

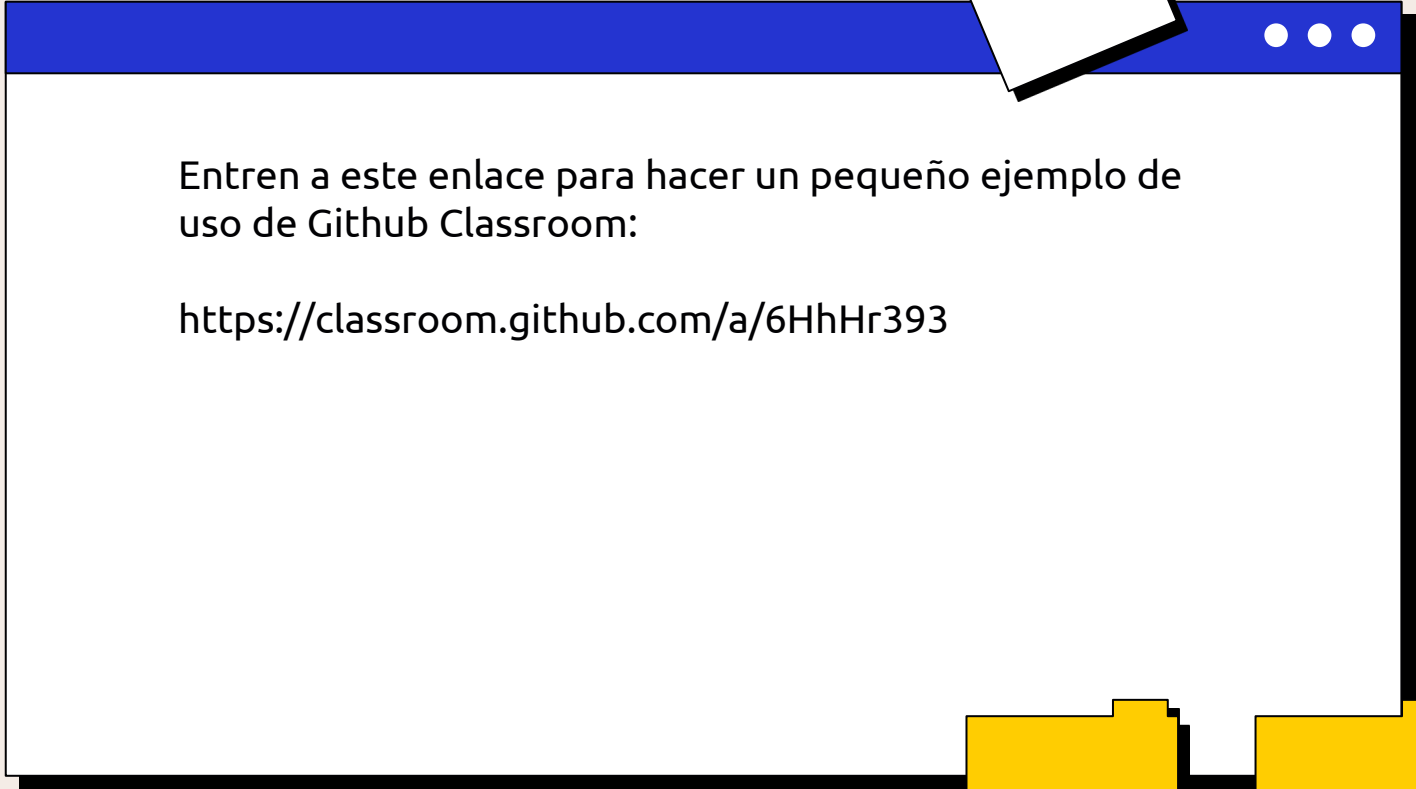
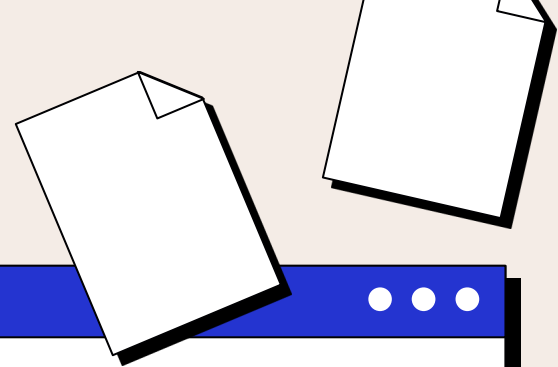
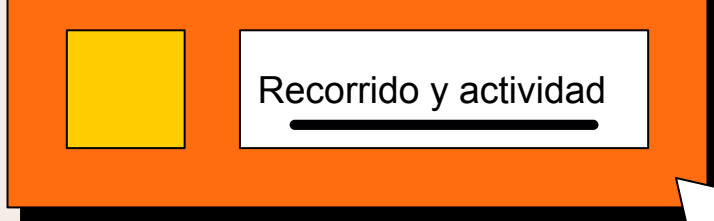
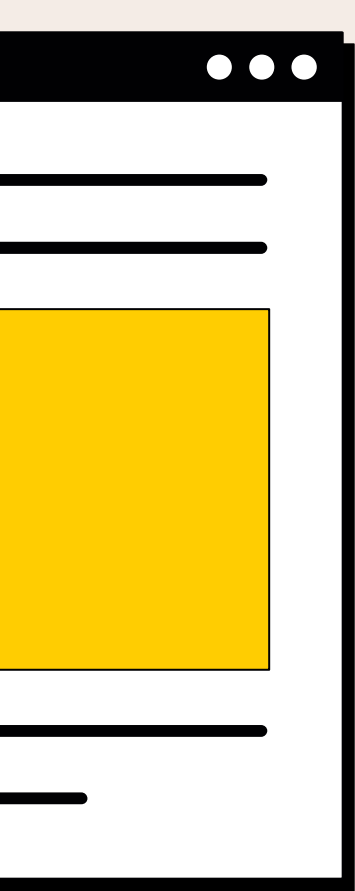
# ¿Qué es Github classroom?

Es una herramienta que permite a los docentes crear y administrar aulas y tareas digitales

# ¿Cómo funciona?

En esta plataforma, los estudiantes reciben copias individuales de un repositorio base creado por el instructor. Cada estudiante puede trabajar en su propia versión del proyecto y subir sus avances a un repositorio privado, facilitando la evaluación y el seguimiento del progreso.





Entren a este enlace para hacer un pequeño ejemplo de uso de Github Classroom:

<https://classroom.github.com/a/6HhHr393>





Recorrido y actividad

Al intentar subir cambios al repositorio se les pedirá una autenticación. Una manera de evitar que esta autenticación sea solicitada cada vez que se intente subir algo se usa el comando:

```
git config --global credential.helper store
```

Antes de hacer un push.

Esto hará que solo se autentifique por una vez





# Muchas gracias!

CREDITS: This presentation  
template was created by **Slidesgo**,  
including icons by **Flaticon**, and  
infographics & images by **Freepik**

# Resources

- [https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Fundamentos-de-Git#:~:text=Git%20tiene%20tres%20estados%20principales,%2C%20y%20preparado%20\(staged\).](https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Fundamentos-de-Git#:~:text=Git%20tiene%20tres%20estados%20principales,%2C%20y%20preparado%20(staged).)
- <https://aulasoftwarelibre.github.io/taller-de-git/>
- <https://www.youtube.com/watch?v=3GymExBkKjE>
- <https://bluuweb.github.io/tutorial-github/02-github/#crear-una-cuenta-en-github>