

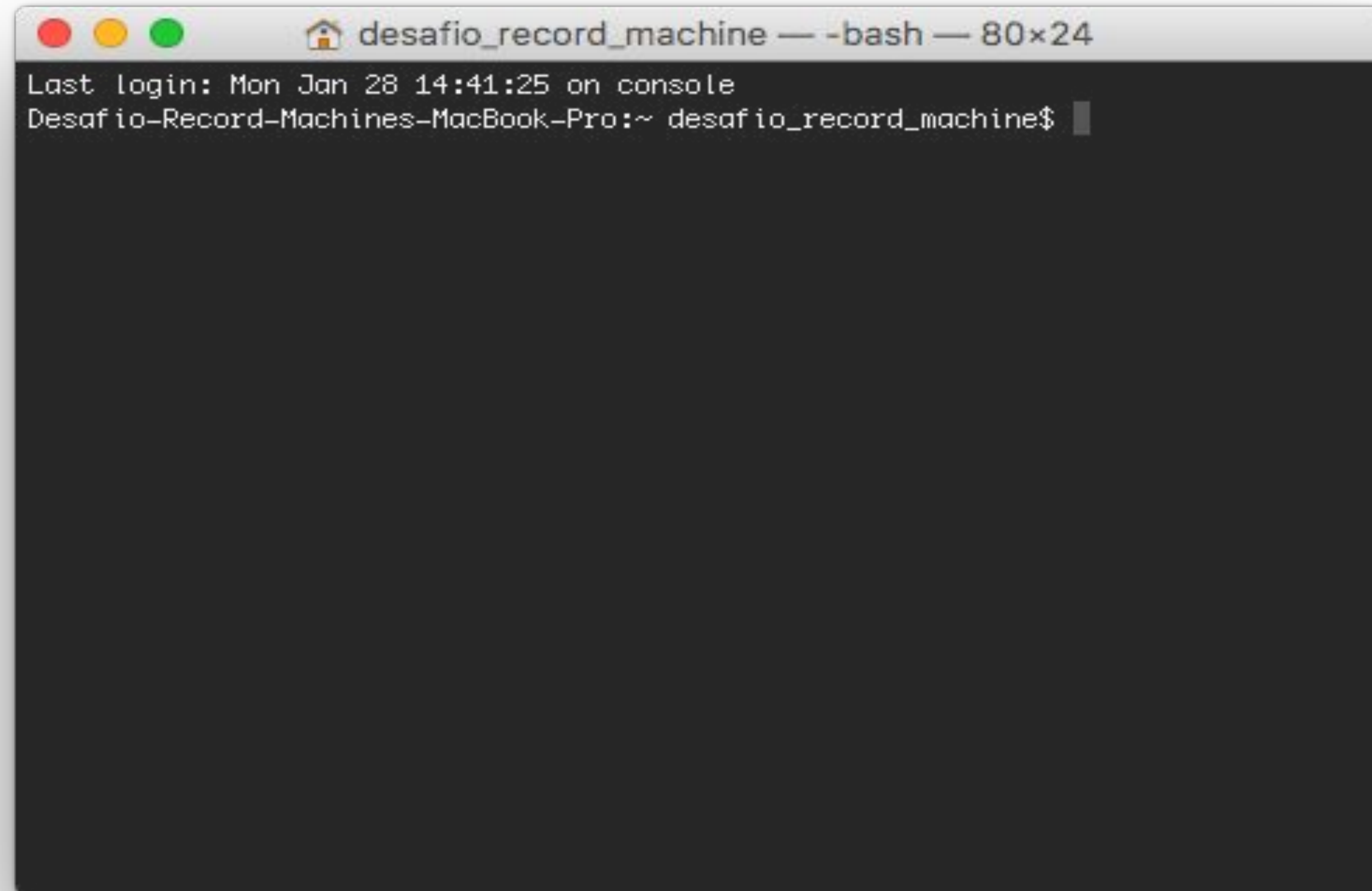
***{desafío}***  
***latam\_***



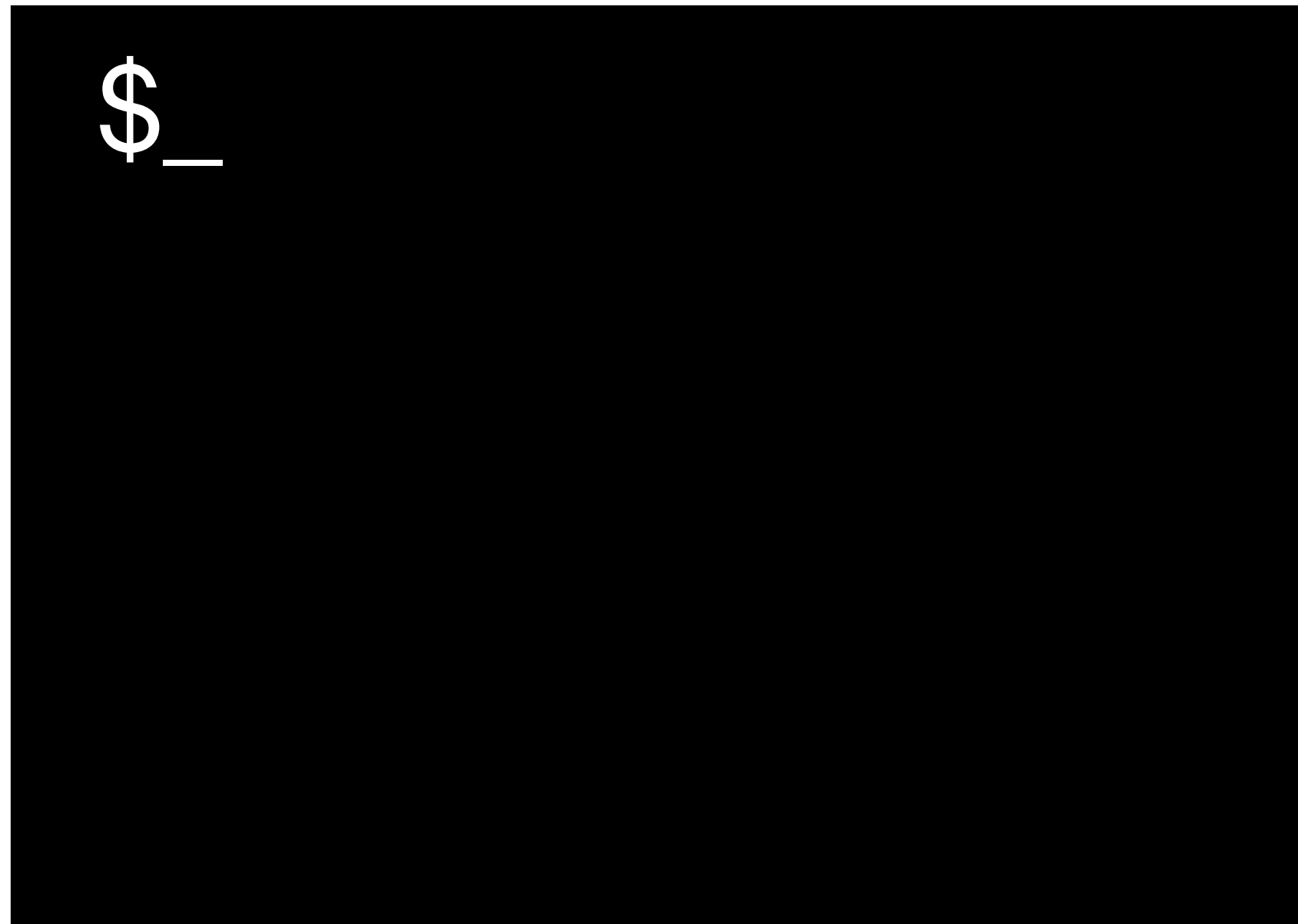
# Terminal, Git, GitHub y GitHub Pages \_

# Terminal

# ¿Que es Terminal?



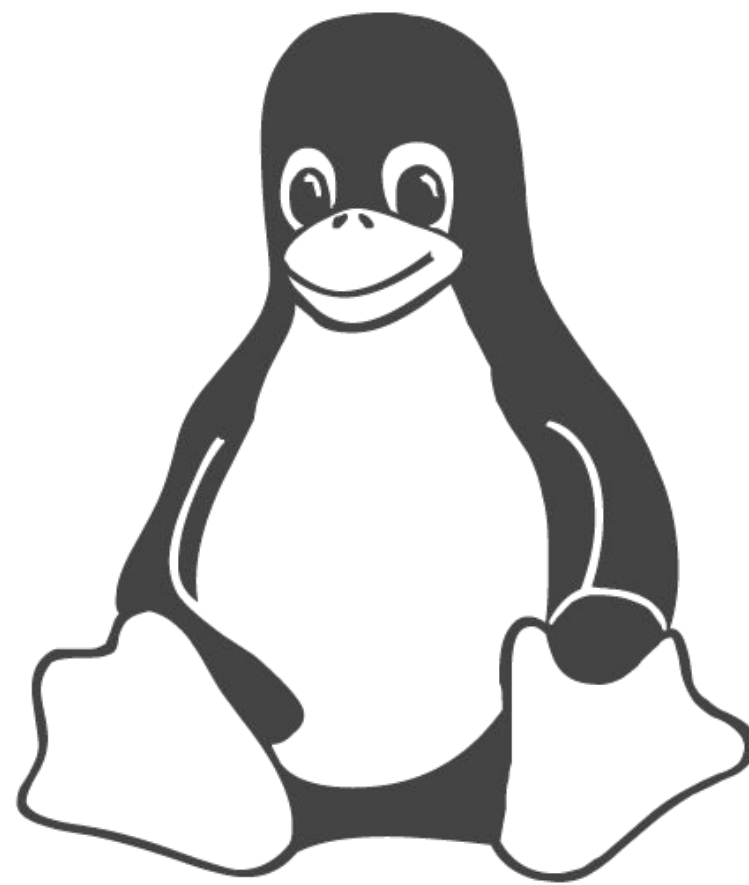
# ¿Qué pasa con Windows?



1. Pausa el video.
2. Descarga la terminal desde el siguiente link → [gitforwindows](https://github.com/gitforwindows/gitforwindows)
3. Luego, instala la terminal en tu computador.
4. Y, ¡listo!

# Iniciar Terminal

## LINUX



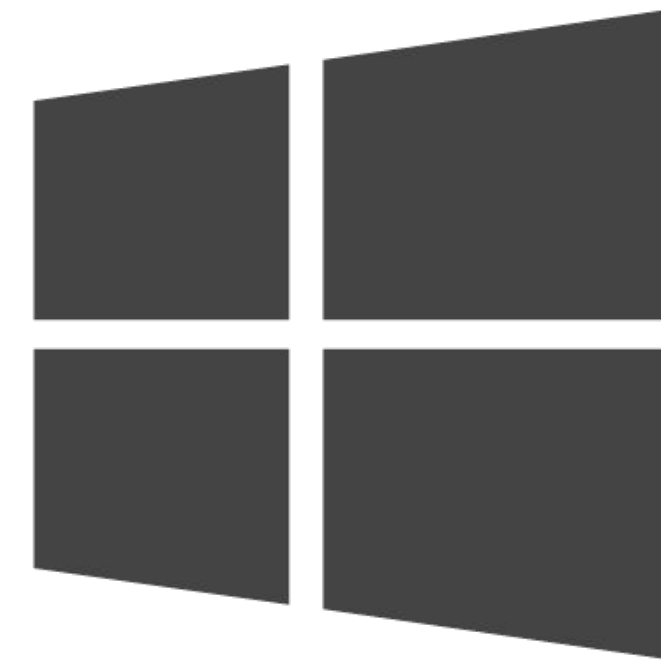
Presiona ctrl + alt + t

## MAC



⌘ + espacio, busca por  
spotlight la "terminal"

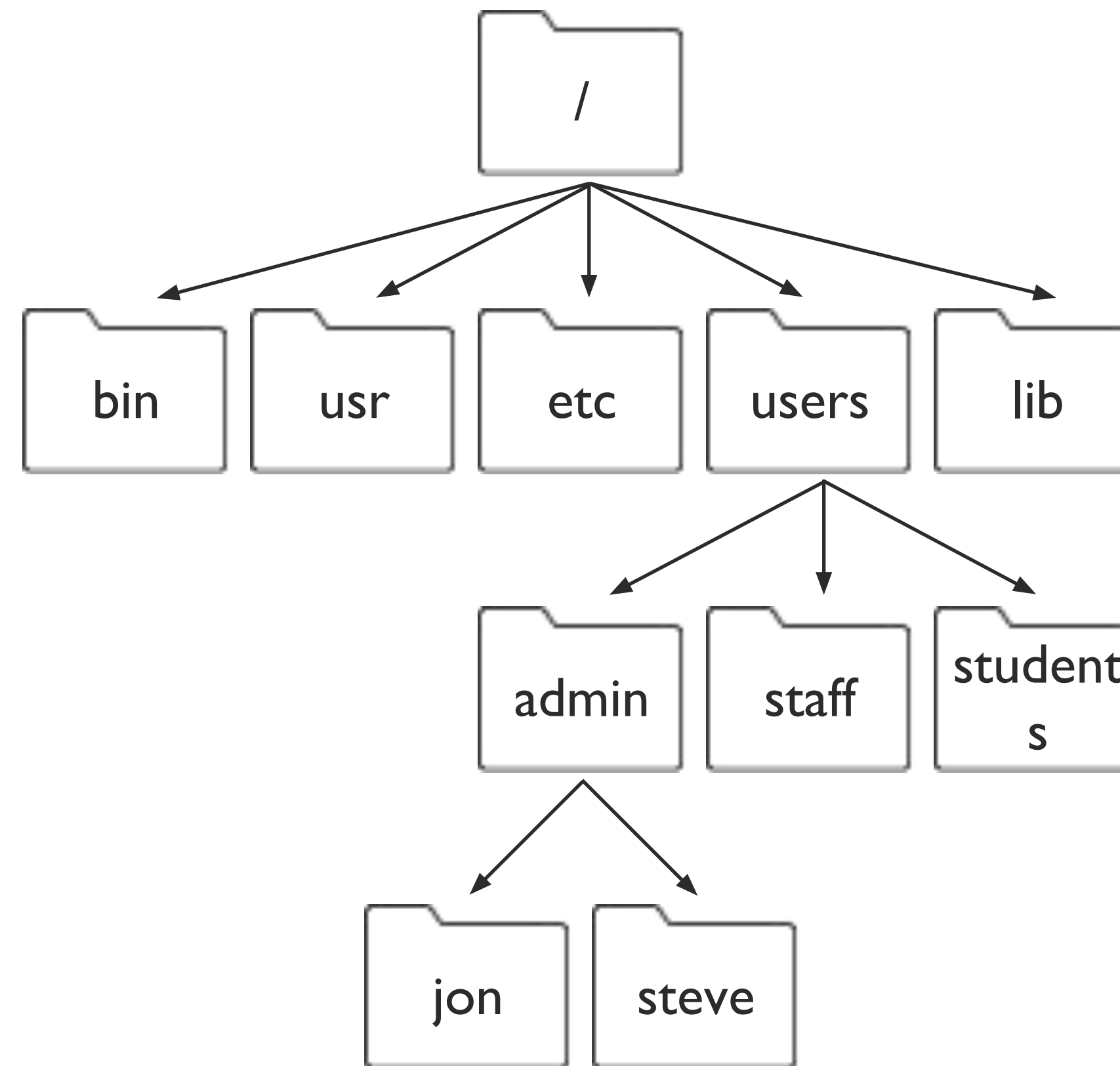
## WINDOWS



Busca el programa  
"git bash" y ábrelo

# Comandos

# Estructura de Directorios



# pwd

Comando que sirve para saber en qué directorio nos encontramos

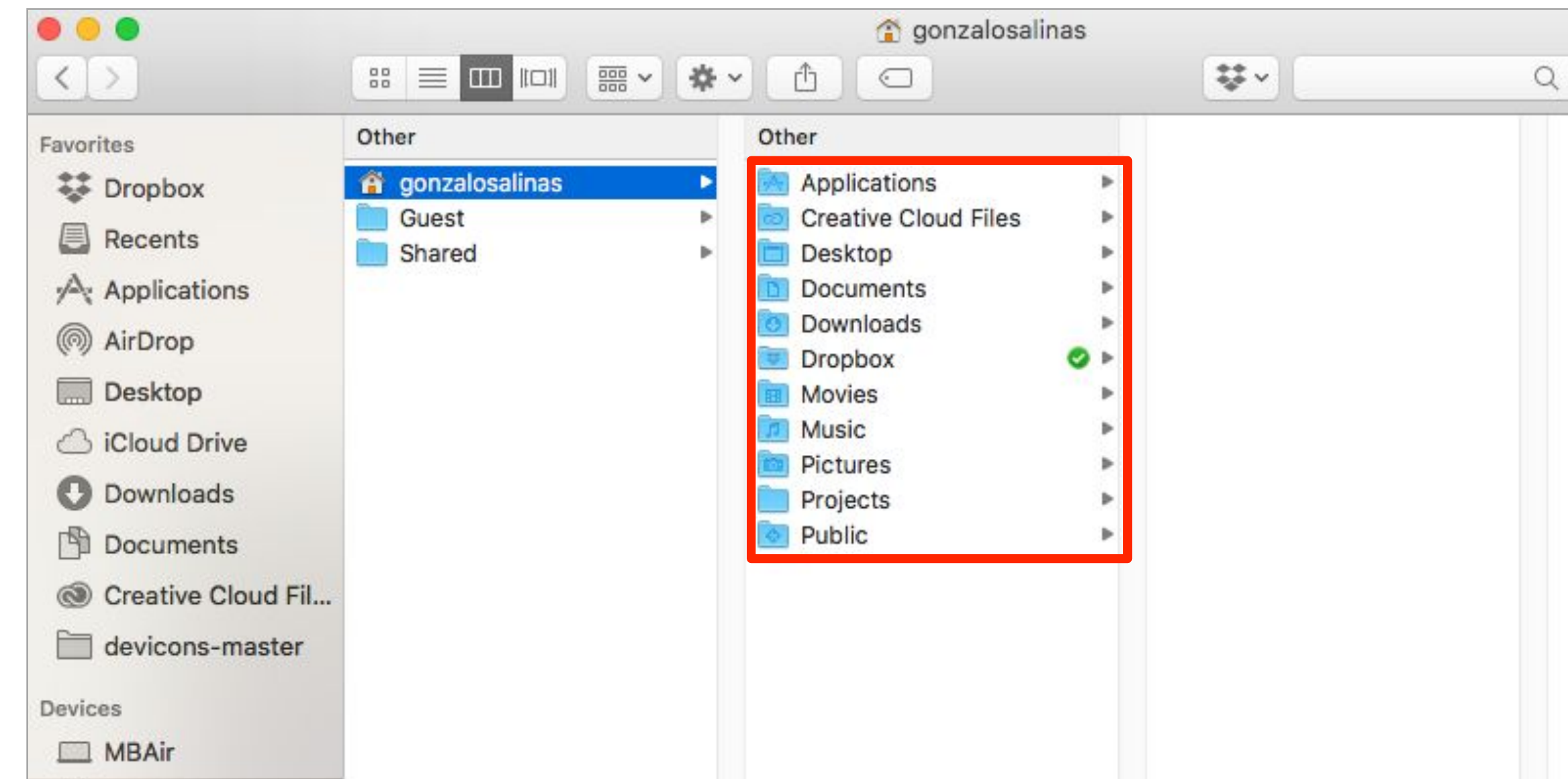
```
$ pwd  
users/mi_usuario
```



# ls

```
$ ls  
Applications  
Desktop  
Download  
Movie  
Music  
Pictures  
Public
```

- Muestra una lista de los archivos que hay dentro de un directorio específico.



# ls -a

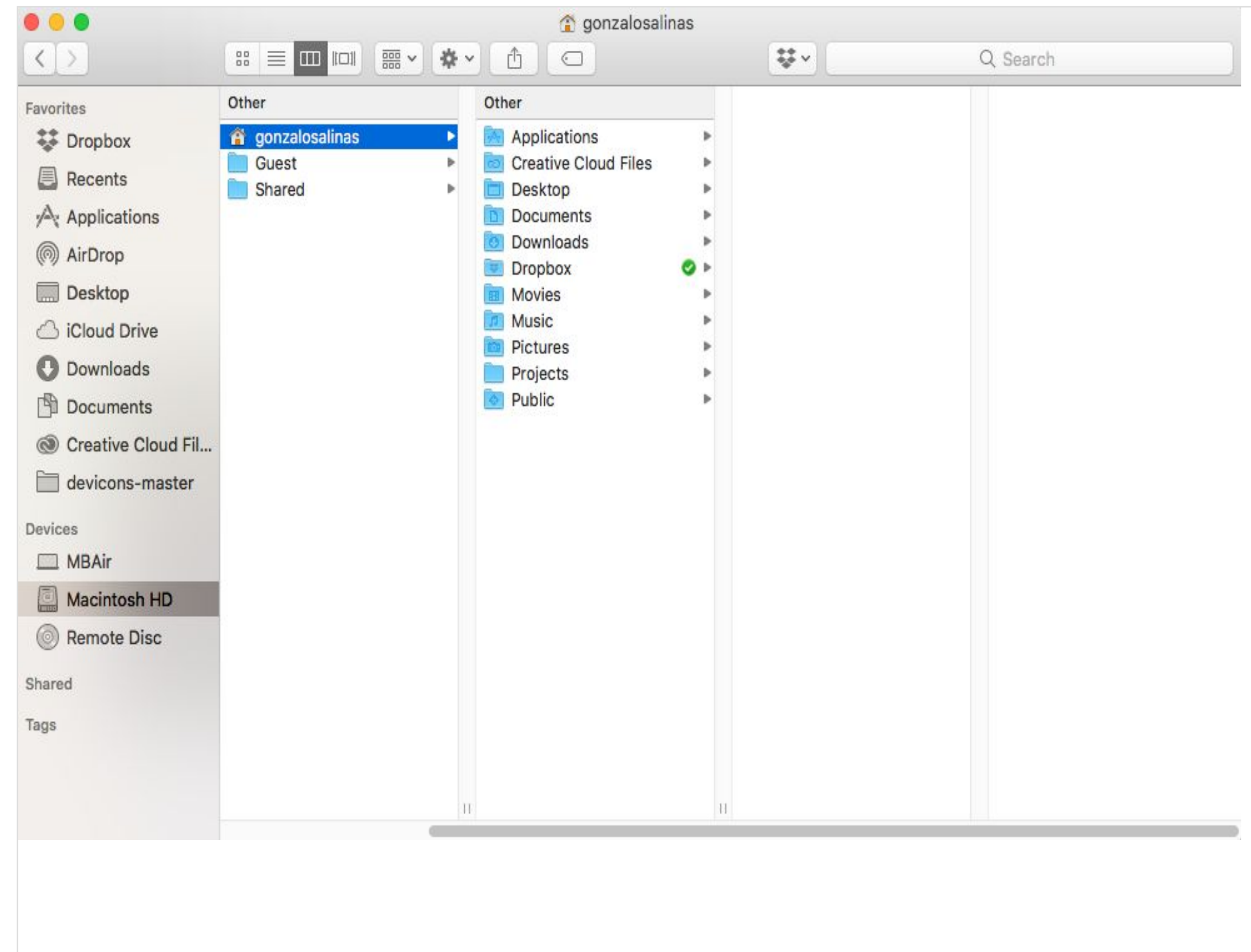
```
$ ls -a  
Applications  
Desktop  
Download  
.DS_Store  
Movie  
Music  
Pictures  
Public
```

- Este comando nos mostrará el listado de todos los archivos, incluyendo los archivos ocultos.
- Los archivos ocultos aparecen siempre con un “.” delante de ellos.

# cd

```
$ cd
```

- Se utiliza para navegar entre directorios.

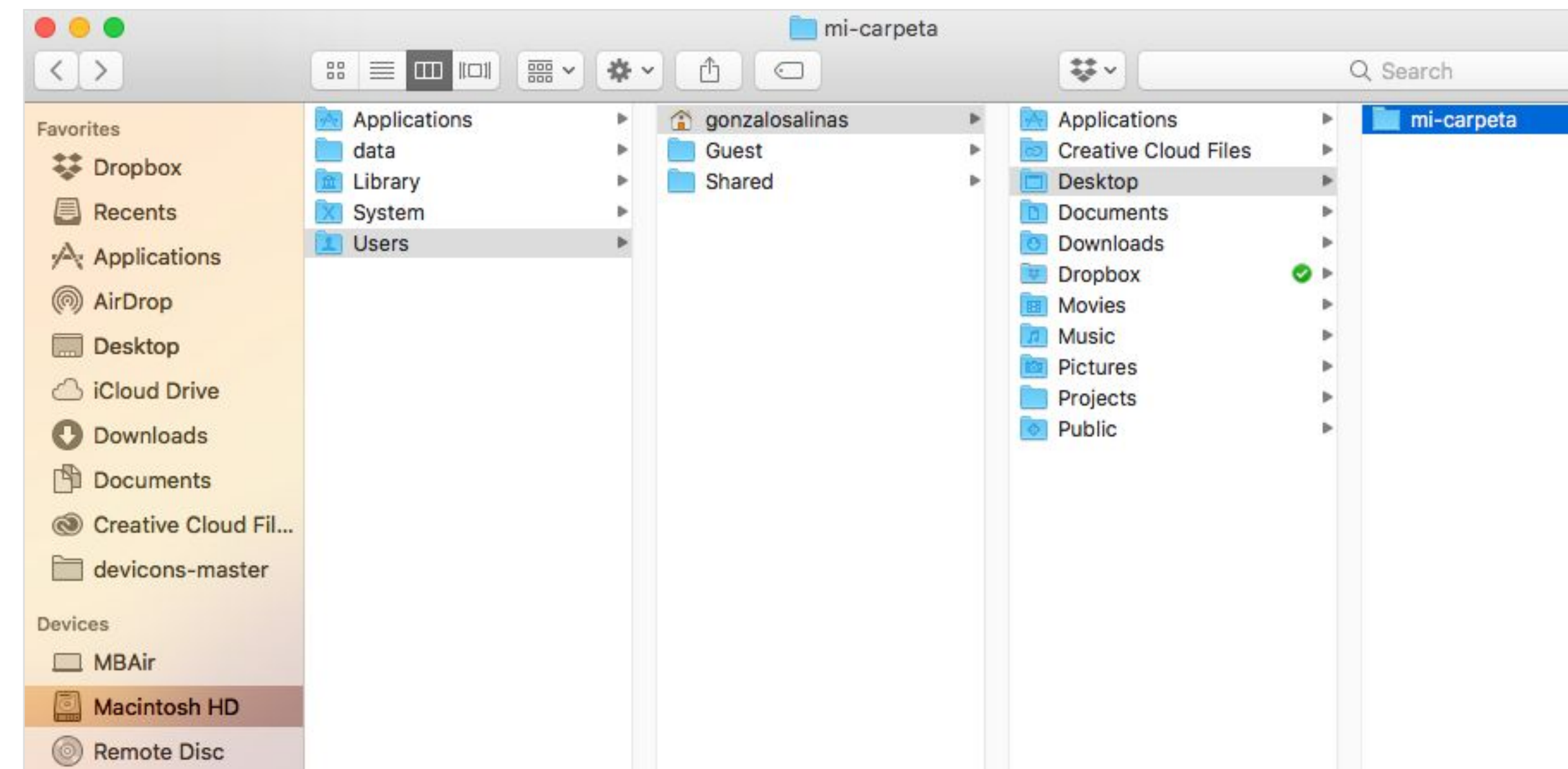


# cd [ruta a directorio]

```
$ cd Desktop/mi-carpeta
```

ruta: /Desktop/mi carpeta

- Este comando nos llevará a una carpeta que especifiquemos.
- Este comando es igual a elegir una carpeta en el navegador de archivos.

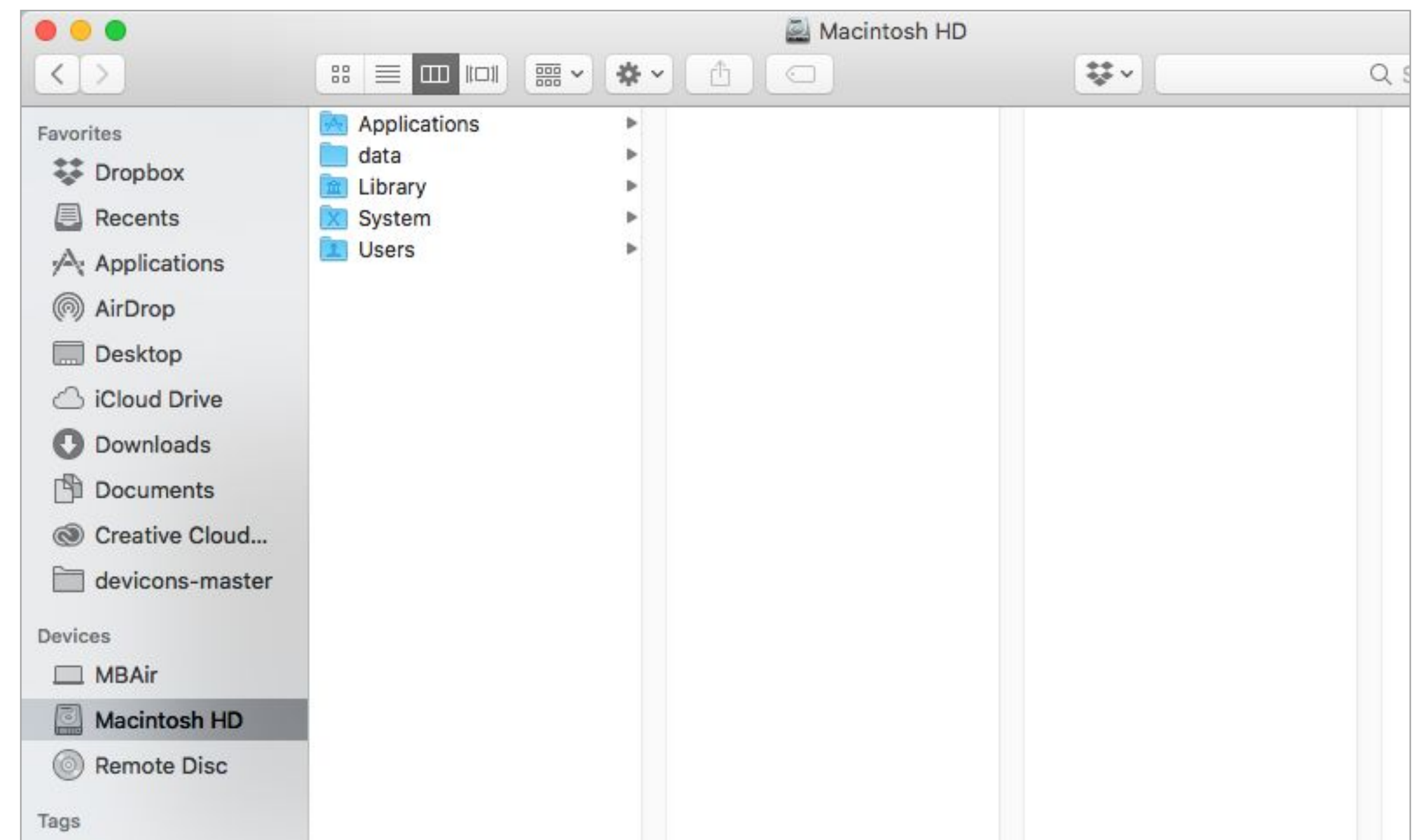




# cd /

```
$ cd /  
ruta: /
```

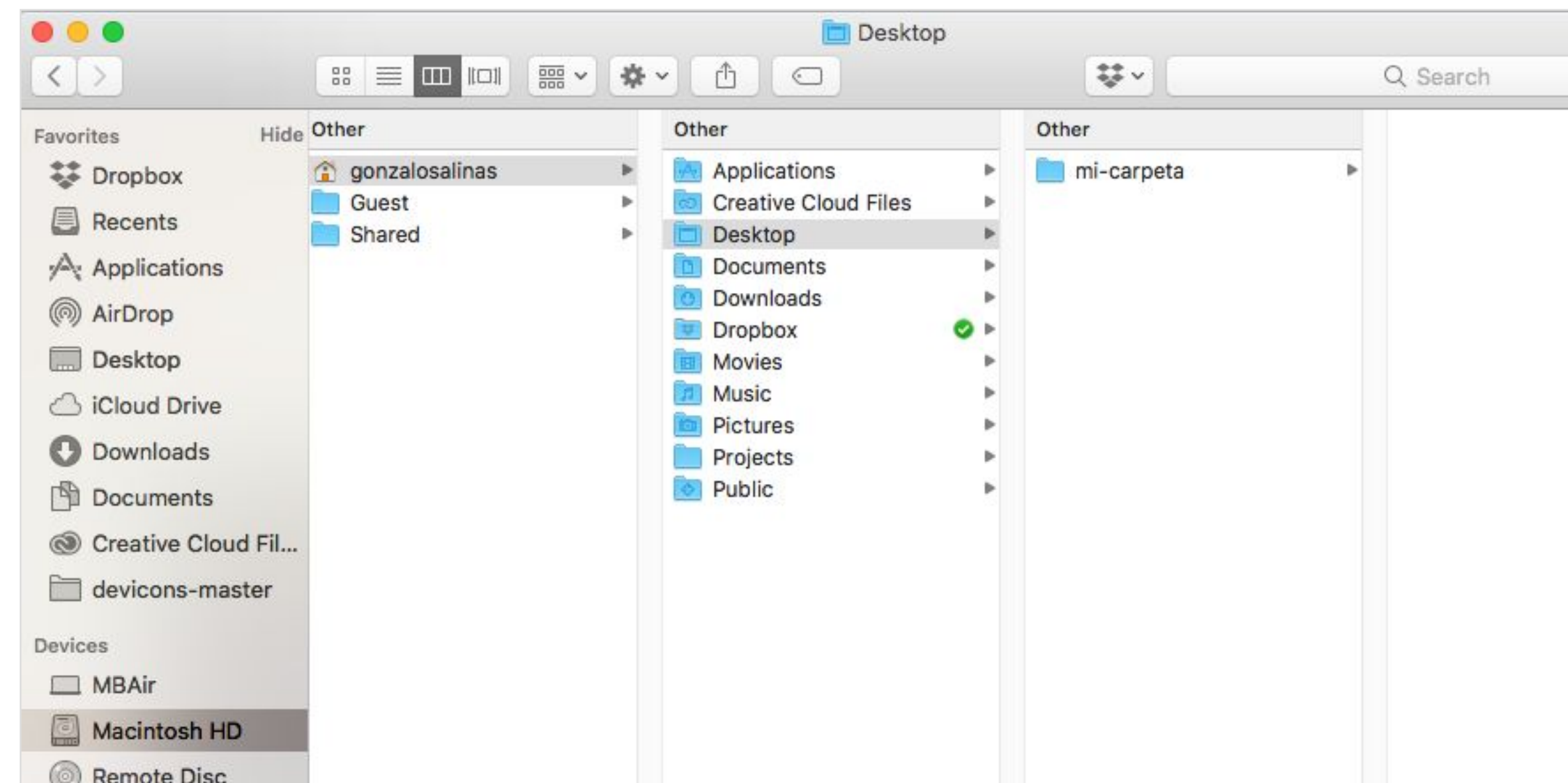
- Este comando nos llevará a la carpeta raíz.



# cd ..

```
$ cd ..
```

- Sirve para ir hacia una carpeta atrás (carpeta contenedora).



# Anatomía de comandos

# Anatomía de un comando

```
$ pwd
```

Nombre de comando



# Anatomía de un comando

```
$ cd Desktop
```

Argumento

# Anatomía de un comando

```
$ ls -a
```

Opción

**Nuestro sistema operativo puede  
ser sensible a las mayúsculas**

**cd** es distinto a **CD**, **cD** o **Cd**

**cd hola** es distinto a **cd Hola**

# Administración de archivos y directorios

Dentro del terminal existen varios comandos que sirven para manejar archivos y directorios

# mkdir [directorio]

```
$ mkdir carpeta_1  
(ruta: users/mi_usuario/Desktop)
```

- Comando que crea un directorio vacío.

# touch [archivo]

```
$ touch index.html  
(ruta: users/mi_usuario/Desktop)
```

- Comando que crea un archivo vacío.
- El archivo creado debe contener un nombre y la extensión a la cual pertenece.



# cp

```
$cp index.html  
(ruta: users/mi_usuario/Desktop)
```

- Comando que sirve para copiar archivos y directorios.

# mv

Comando que sirve para cambiar el nombre de un archivo  
y/o mover un archivo de un directorio a otro

```
$ mv
```

# **mv [archivo] [nuevo nombre]**

Con esta combinación podemos cambiar el nombre de un archivo

```
$ mv index.html archivo.html  
(ruta: users/mi_usuario/Desktop)
```

Para hacerlo debemos indicar el nombre actual del archivo,  
con el nuevo nombre que quieras agregar

# **mv [ruta archivo] [nueva ruta archivo]**

```
$ mv archivo.html ../Desktop/mi-  
carpeta  
(ruta:  
users/mi_usuario/Desktop/mi-carpeta)
```

- Con esta combinación podemos mover un archivo de un directorio a otro.
- Para cambiar el archivo debes escribir la ruta del archivo, espacio, y luego la nueva ruta del archivo.

# rm [nombre archivo]

Comando que sirve para eliminar un archivo especificado por su nombre

```
$ rm index.html  
(ruta: users/mi_usuario/Desktop)
```

# **rm -r [nombre directorio]**

Comando que sirve para eliminar por completo un directorio

```
$ rm index.html  
(ruta: users/mi_usuario/Desktop)
```

# Recapitulación de comandos

Comando	Explicación
<b>pwd</b>	Muestra en que directorio te encuentras
<b>cd</b>	Cambia de directorio a uno especificado
<b>cd ..</b>	Permite ir a un directorio anterior al actual
<b>cd /</b>	Lleva a la raíz
<b>cd ~</b>	Lleva al directorio \$home
<b>ls</b>	Muestra todos los archivos y directorios de la carpeta actual
<b>ls -a</b>	Muestra los archivos, los archivos ocultos y los directorios de la carpeta actual
<b>touch</b>	Crea un archivo
<b>mv</b>	Renombra y mueve un archivo
<b>rm</b>	Borra un archivo
<b>mkdir</b>	Crea un directorio
<b>rmdir</b>	Elimina un directorio vacío
<b>rm -r</b>	Elimina todos los archivos de un directorio, incluido el directorio



# ¿Qué es Git?



Git es un software de control de versiones open source, diseñado para mantener archivos de manera eficaz y confiable.

Google

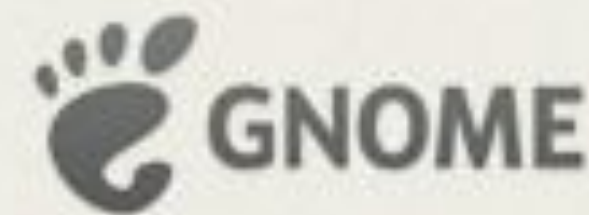
facebook

Microsoft

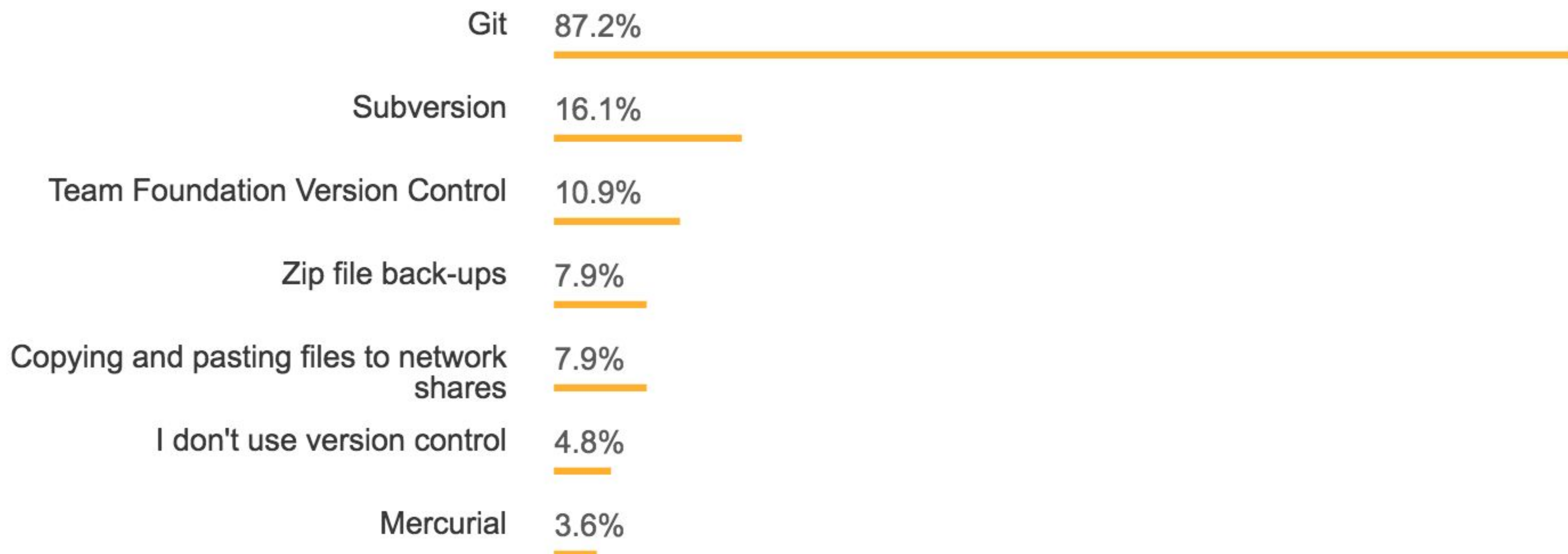
twitter

LinkedIn

NETFLIX



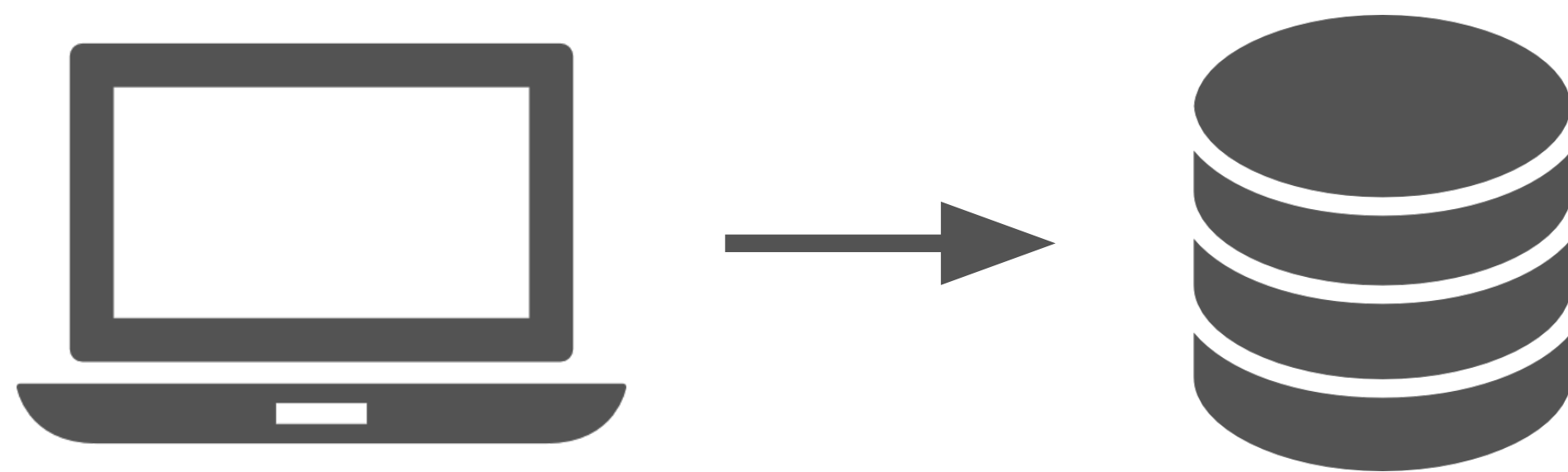
{desafío}  
latam\_



# Git nos permite:

- Recuperar versiones anteriores de nuestro código
- Recuperar archivos borrados
- Ayudar a gestionar cambios realizados por otras personas
- Administrar un proyecto donde trabajan múltiples desarrolladores

# ¿Qué es control de versiones?



Es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante.

# ¿Cuándo utilizar Git?



Se recomienda utilizar **siempre**.

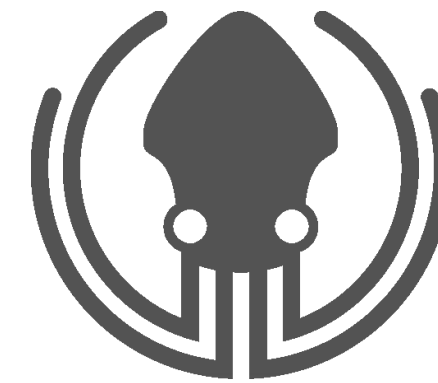
# Formas de utilizar Git



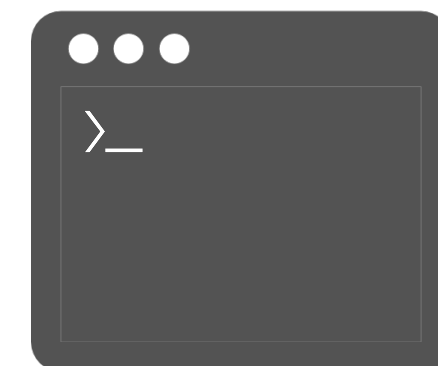
**GIT**



Editores de  
texto



Programas de  
versionamiento



Terminal

# Instalar Git



# Git en Mac

```
$ git --version  
git version 2.14.3
```

1. Ingresar al terminal y escribir **git --version** para verificar si está instalado.
2. Si no está instalado, descarga la última versión desde el siguiente [link](#).
3. Seguir las instrucciones de instalación.
4. Volver al terminal y escribir nuevamente **git --version** para verificar que todo resultó bien.

# Git en Windows

```
$ sudo apt-get update  
$ sudo apt-get install git
```

```
$ git --version  
git version 2.14.3
```

1. Ingresar al terminal y escribe **sudo apt-get update**.
2. Luego escribe **sudo apt-get install git**.
3. Verificar que la instalación resultó bien escribiendo **git - -version**.

Anteriormente descargamos *Git for Windows*  
e instalamos el terminal en nuestro computador.

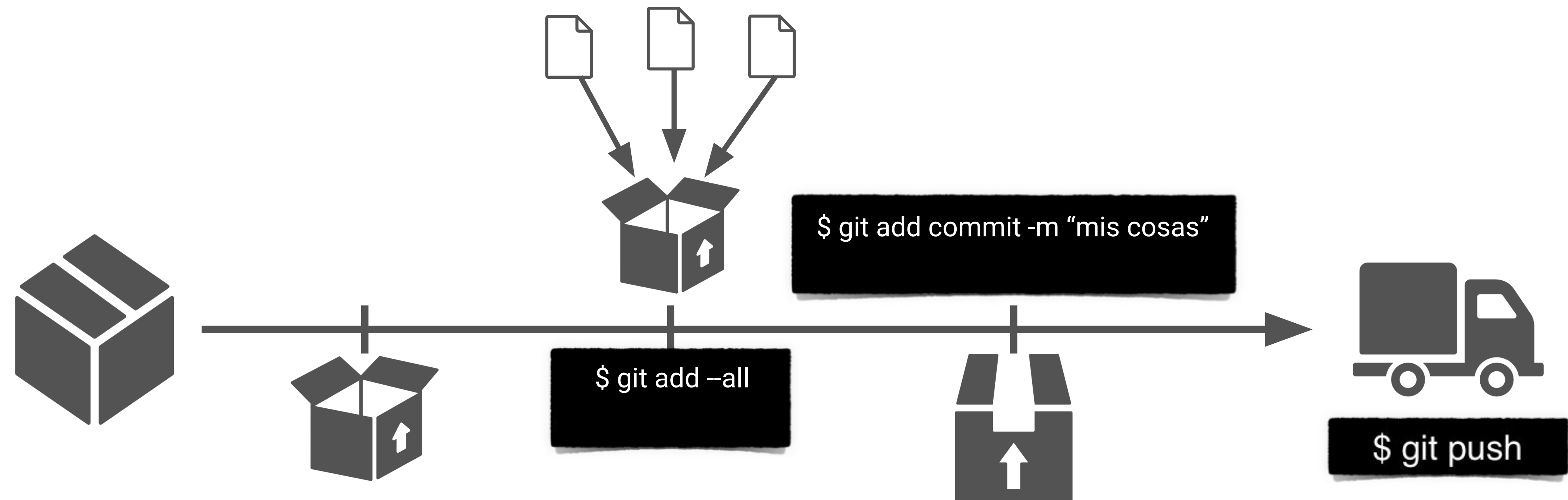
# Configurar Git

```
$ git config --global user.name "Tu nombre"
```

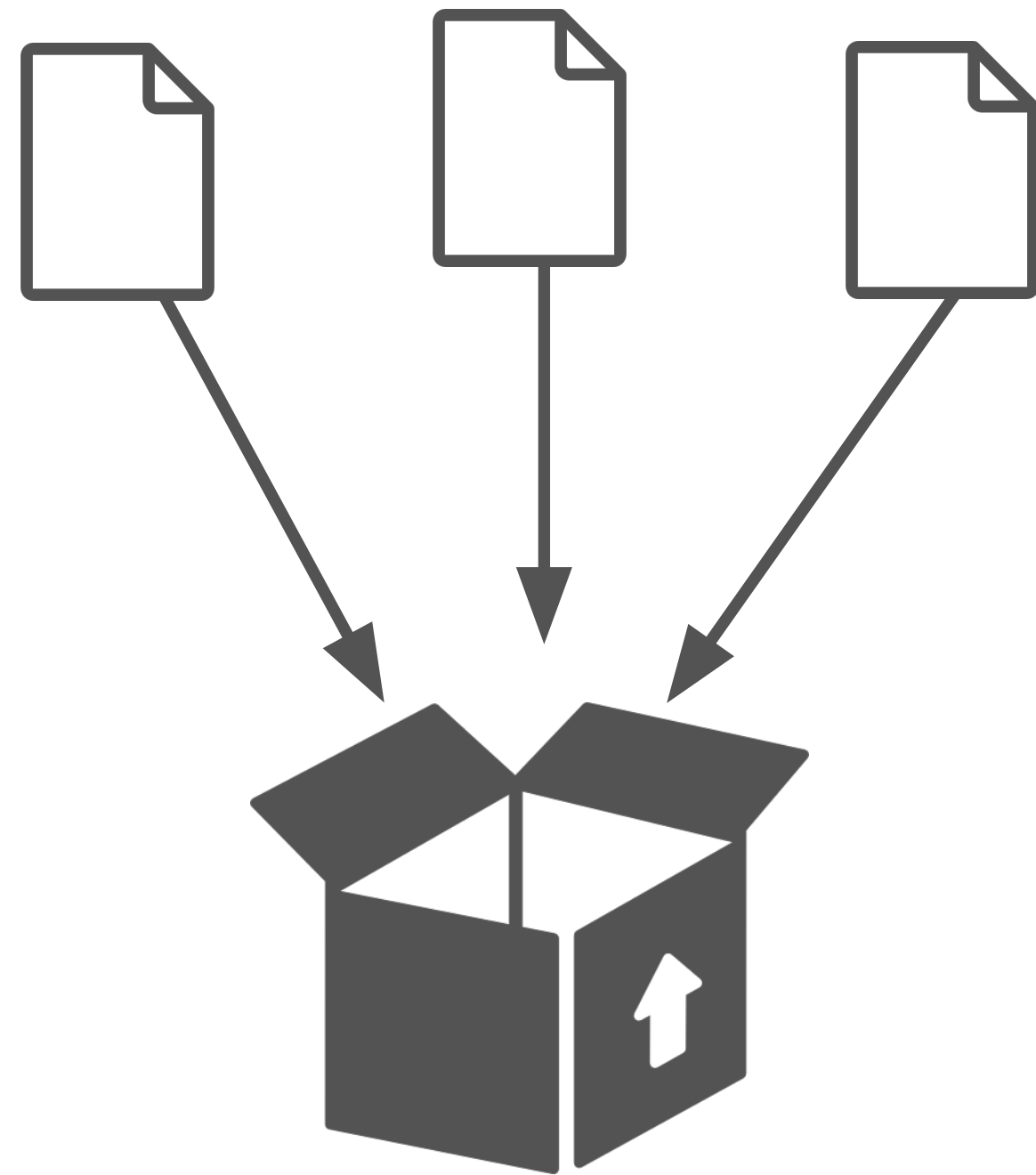
```
$ git config - -list  
user.name= tu nombre  
user.mail= tucorreo@mail.com
```

- Escribir en el terminal el comando **git config --global user.name**, seguido de *tu nombre* dentro de comillas dobles.
- Revisar si la configuración está correcta usando el comando **git config - -list**.

# Flujo de trabajo en Git



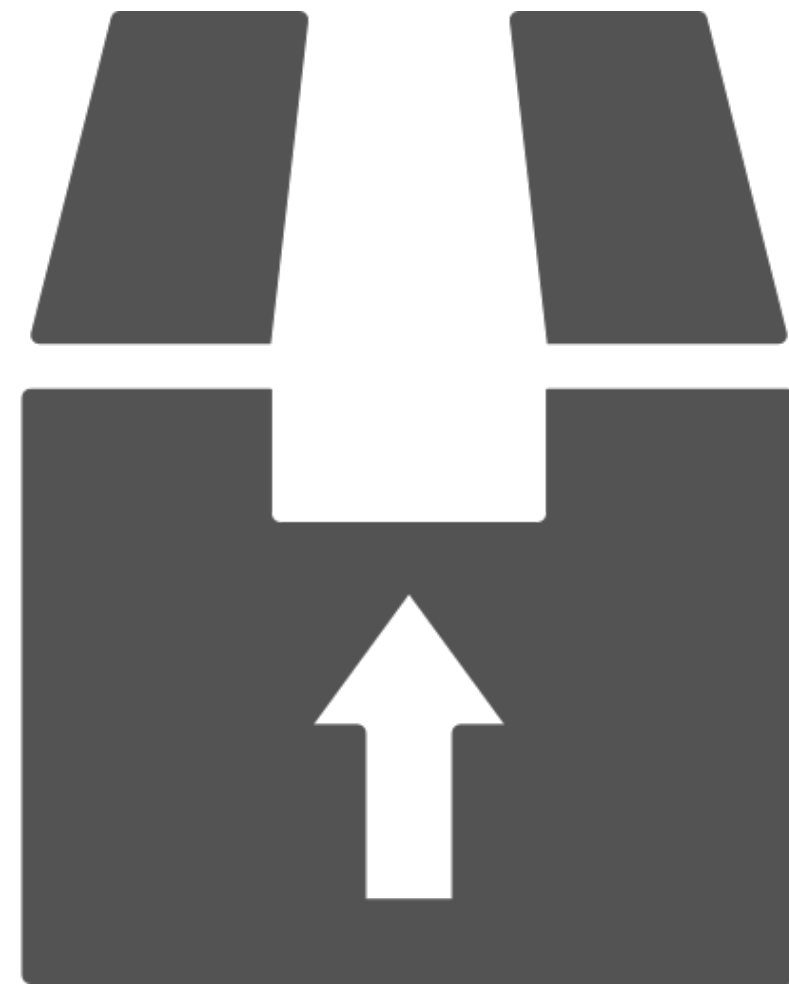
# Agregar cosas a la caja



- Ahora que está abierta, es momento de agregar todas nuestras cosas a la caja.
- Esta acción es igual a escribir **git add --all** o uno a uno con **git add [nombre\_de\_archivo]** .

```
$ git add --all
```

# Cerrar y etiquetar la caja



```
$ git commit -m "mis cosas"
```

- Luego de agregar todas nuestras cosas, habrá que etiquetar y enlistar las cosas de nuestra caja.
- Esta acción es igual a escribir **git commit -m "mis cosas"**.



# Enviar la caja



- A continuación debemos enviar la caja a destino.
- Esta acción es igual a escribir **git push origin master**.

```
$ git push origin master
```

# Revisar la caja



```
$ git status
```

```
On branch master  
nothing to commit, working tree clean/  
mi-usuario/Desktop/mi-carpeta)
```

- **git status** nos ayudará a saber en qué parte del flujo de trabajo nos encontramos.
- Es similar a revisar el proceso dentro de una lista.

# ¿Local o remoto?

Dentro de estos servicios, existen varios tipos de repositorios remotos, las más usadas son:



**Github**



**Bitbucket**



**Gitlab**

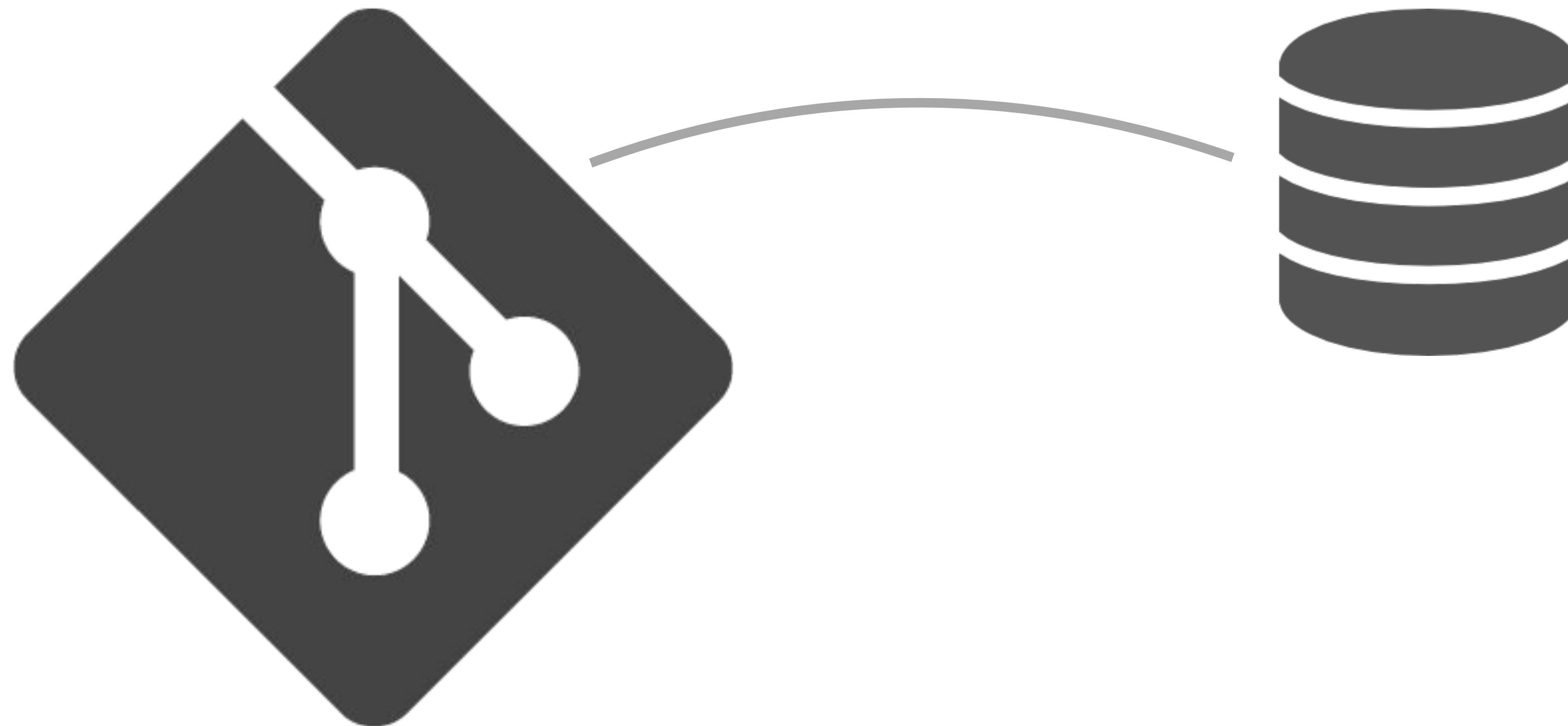
En nuestro caso usaremos uno de los servicios  
de repositorios remotos más conocida,  
llamado **GitHub**

# ¿Qué es GitHub?



GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git.

# ¿Qué es un repositorio remoto?



# Protocolo SSH



# ¿Qué es SSH?

SSH es un protocolo que nos permite conectarnos y autenticarnos a servidores y servicios remotos como GitHub.

Ya teniendo nuestra cuenta creada, es momento de configurar el traspaso de datos entre nuestro *computador y GitHub*, a través una llave cifrada **SSH**.

# ¿Cómo saber si tengo una llave SSH en mi computador?

Ingresa al terminal y escribe:

```
$ ls -al ~/.ssh
```

Si no encontramos ninguna llave, debemos generar una llave SSH.

`d_dsa.pub`

`id_ecdsa.pub`

`id_ed25519.pub`

`id_rsa.pub`

# Generar una nueva llave SSH



1

```
$ ssh-keygen -t rsa -b 4096 -C  
"tu-correo@mail.com"
```

2

```
. Enter a file in which to save the key  
(/Users/you/.ssh/id_rsa): [Press  
enter]
```

3

```
Enter passphrase (empty for no  
passphrase): [Type a passphrase]
```

4

```
Enter passphrase (empty for no  
passphrase): [Type a passphrase]
```

1. Abrir el terminal y escribir el comando **ssh-keygen**.
2. Cuando se solicite guardar el archivo, presiona enter.
3. Cuando se solicite la clave presiona enter.
4. Cuando se solicite la clave nuevamente presiona enter.

# Agregar una llave SSH al SSH-Agent

1

```
$ eval "$(ssh-agent -s)"  
Agent pid 59566
```

2

MAC 

```
$ ssh-add -K ~/.ssh/id_rsa
```

Windows 

```
$ ssh-add ~/.ssh/id_rsa
```

Linux 

```
$ ssh-add ~/.ssh/id_rsa
```

1. Iniciar el administrador de ssh, escribiendo en el terminal **eval "\$(ssh-agent -s)"**

1. Agregar el SSH que creamos anteriormente usando el comando:

**ssh-add -K ~/.ssh/id\_rsa**



# Agregar una llave SSH a GitHub

**MAC** `$ pbcopy < ~/.ssh/id_rsa.pub`

**Windows** `$ clip < ~/.ssh/id_rsa.pub`

**Linux** `$ sudo apt-get install xclip`  
# Descarga e instala xclip para  
copiar el ssh.  
`$ xclip -sel clip < ~/.ssh/  
id_rsa.pub`

1. Primero debemos copiar la clave ssh desde el portapapeles usando alguna de estas opciones de copiado.

2. Inicia sesión en GitHub. Cuando estés dentro presiona tu avatar y luego elige la opción settings.

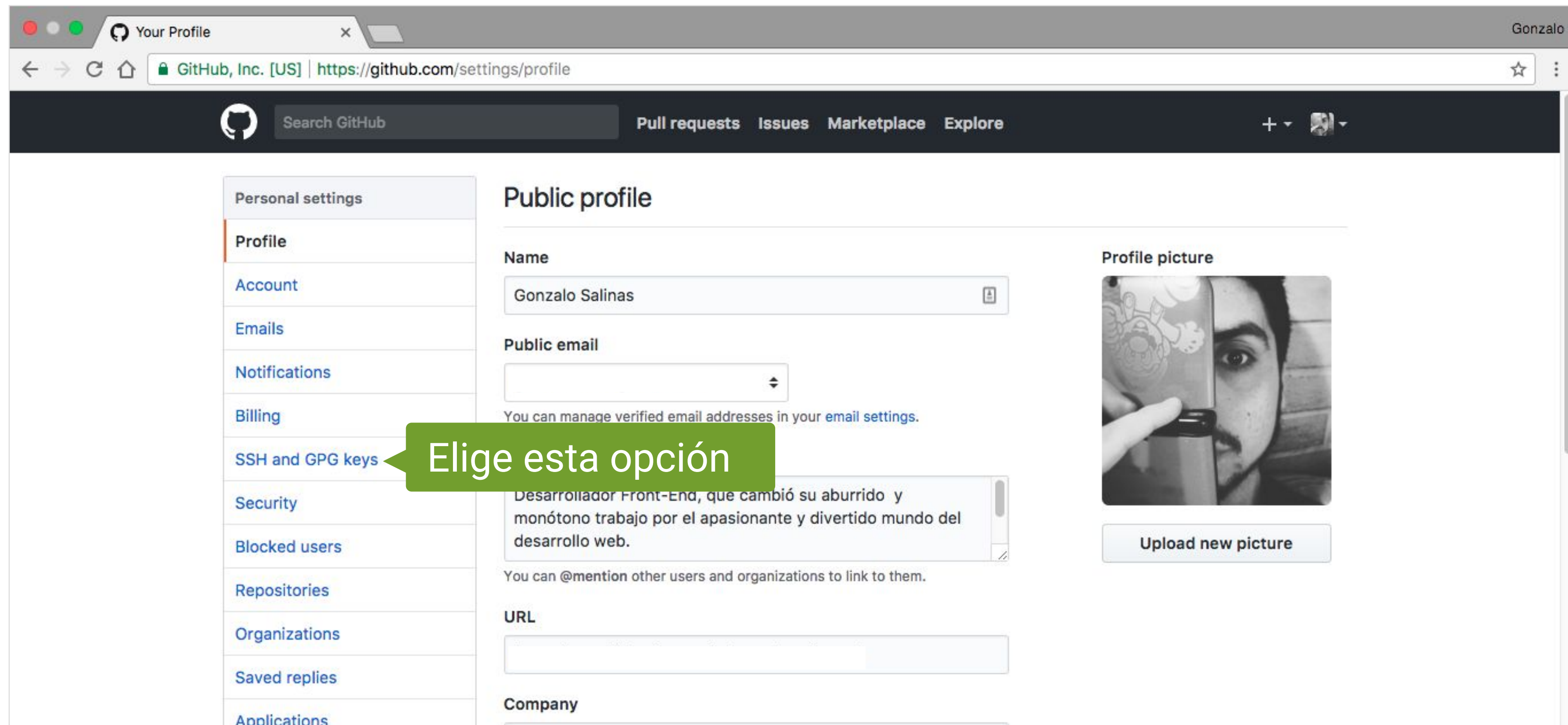
The screenshot shows the GitHub profile page for user 'gsalinase' (Gonzalo Salinas). The browser address bar shows 'https://github.com/gsalinase'. The page header includes navigation links: Pull requests, Issues, Marketplace, and Explore. The user's profile information is on the left, including a profile picture, name 'Gonzalo Salinas', username 'gsalinase', and bio: 'Desarrollador Front-End, UI & UX que cambió su aburrido y monótono trabajo por el apasionante y divertido mundo del desarrollo web. @gsalinase'. The main content area shows 'Pinned repositories' with five items: 'landing-desafio', 'asperger-chile', 'Politips', 'mini-twitter', and 'chile\_con\_sentido'. A dropdown menu is open from the user's avatar in the top right corner, showing options: 'Signed in as gsalinase', 'Your profile', 'Your stars', 'Your Gists', 'Help', 'Settings' (highlighted in blue), and 'Sign out'. Two green callout boxes with arrows point to the avatar and the 'Settings' option, with the text 'Presiona tu avatar' and 'Presiona Settings' respectively.

Presiona tu avatar

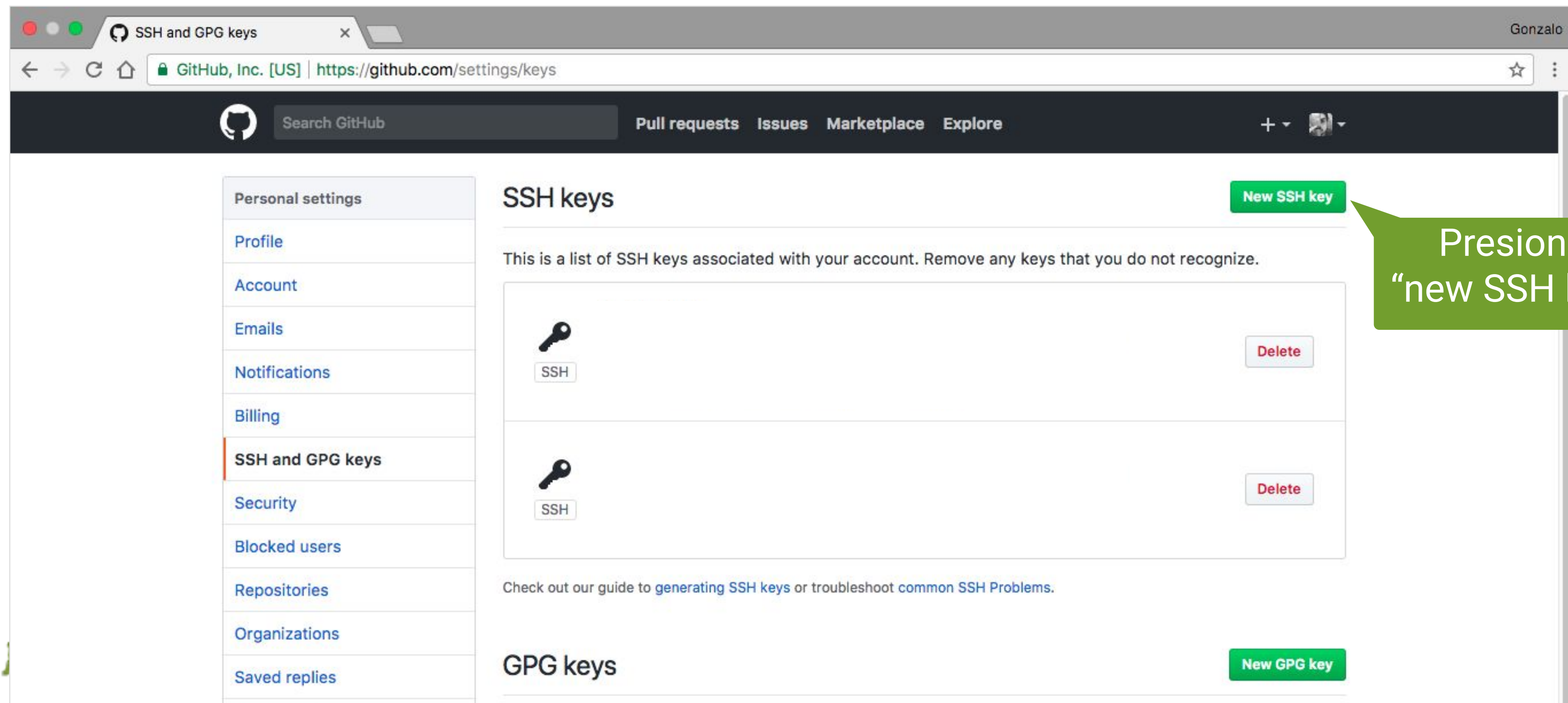
Presiona Settings



3. Dentro de la configuración personal de tu cuenta, presiona la opción “llave SSH y GPG”.

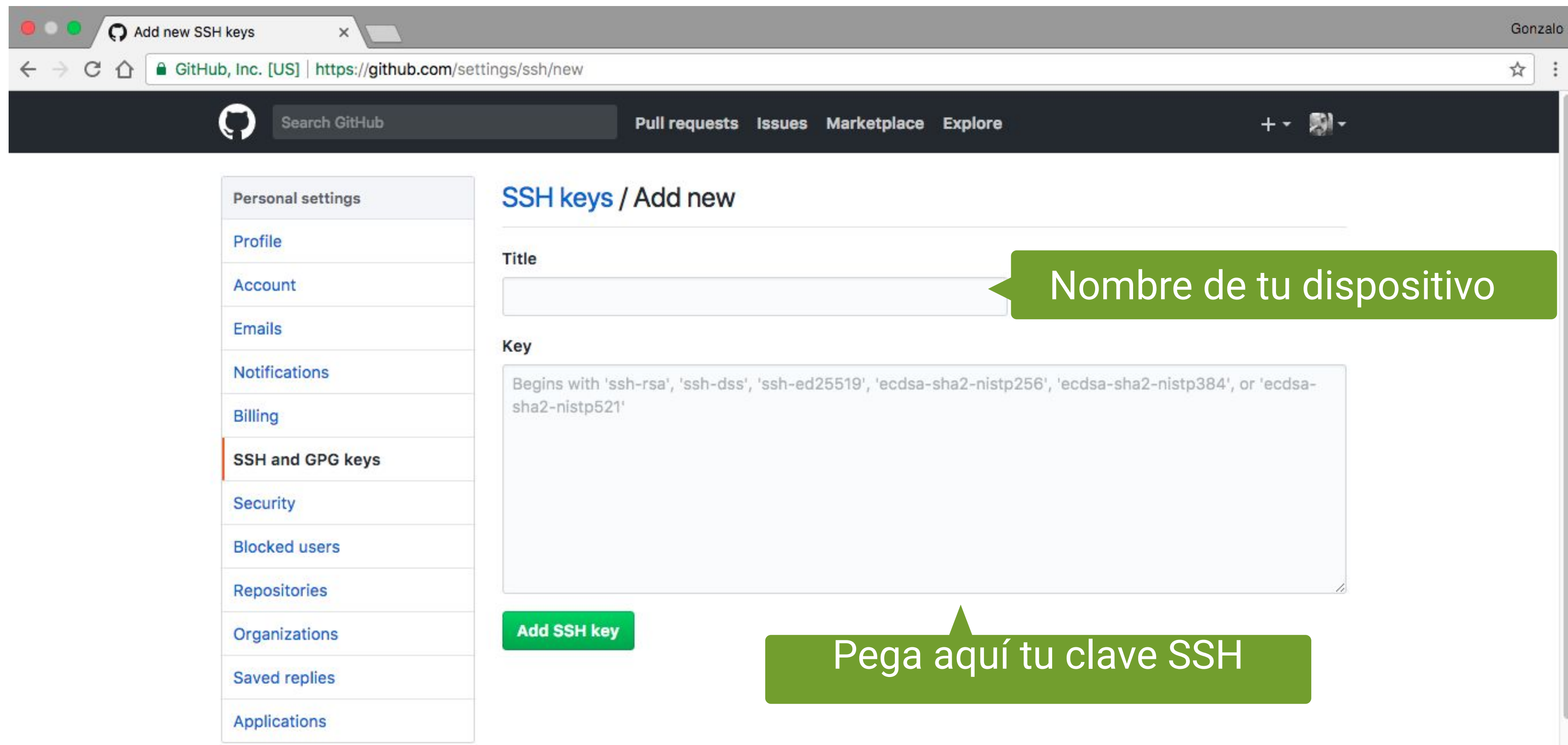


4. Luego, presiona en “new SSH key”.



The screenshot shows the GitHub 'SSH and GPG keys' settings page. The left sidebar contains a list of settings: Personal settings, Profile, Account, Emails, Notifications, Billing, SSH and GPG keys (highlighted), Security, Blocked users, Repositories, Organizations, and Saved replies. The main content area is titled 'SSH keys' and includes a green 'New SSH key' button in the top right. Below the title, there is a list of SSH keys, each with a key icon, the label 'SSH', and a 'Delete' button. A green callout bubble with the text 'Presiona "new SSH key"' points to the 'New SSH key' button. At the bottom of the page, there is a section for 'GPG keys' with a 'New GPG key' button.

5. Finalmente, dale un nombre a tu dispositivo y pega la clave SSH.



The screenshot shows the GitHub 'Add new SSH keys' page. On the left is a sidebar with 'Personal settings' and various options, with 'SSH and GPG keys' highlighted. The main area is titled 'SSH keys / Add new' and contains a 'Title' text input field and a 'Key' text area. A green callout bubble points to the 'Title' field with the text 'Nombre de tu dispositivo'. Another green callout bubble points to the 'Key' text area with the text 'Pega aquí tu clave SSH'. Below the key area is a green 'Add SSH key' button. The browser's address bar shows the URL 'https://github.com/settings/ssh/new'.

Personal settings

- Profile
- Account
- Emails
- Notifications
- Billing
- SSH and GPG keys**
- Security
- Blocked users
- Repositories
- Organizations
- Saved replies
- Applications

### SSH keys / Add new

Title

Nombre de tu dispositivo

Key

Begins with 'ssh-rsa', 'ssh-dss', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521'

Add SSH key

Pega aquí tu clave SSH



# Testear la conexión SSH

```
$ ssh -T git@github.com  
Hi username! You've successfully  
authenticated, but GitHub does not  
provide shell access.
```

1. Abre el terminal y escribe **ssh -T git@github.com**
1. Si funciona la conexión entre nuestro computador y GitHub debiese aparecer el siguiente mensaje, *"Hi user! You've successfully authenticated, but GitHub does not support access"*.

# Nuestro primer repositorio



gsalinase (Gonzalo Salinas) x Gonzalo

← → ↻ 🏠 🔒 GitHub, Inc. [US] | https://github.com/gsalinase ☆ ⋮

🐱 Search GitHub Pull requests Issues Marketplace Explore +

**New repository**  
Import repository  
New gist  
New organization

Elige "New repository"

Overview Repositories 30 Stars 1 Followers 5 Following

**Pinned repositories**

≡ **landing-desafio**

Landing Page desarrollado para una actividad en Desafío Latam.

● HTML

≡ **asperger-chile**

Proyecto hecho para renovar la web de la Fundación Asperger Chile.

● HTML

≡ **Politips**

Forked from Alekinpad/Politips

Politips es una aplicación web que informa a los ciudadanos sobre el nuevo sistema electoral -- AngelHacks Hackaton

● CSS

≡ **mini-twitter**

Forked from DesafioLatam/Blearring-Frontend-E08CP1A1

Mini Twitter

● HTML

≡ **chile\_con\_sentido**

Forked from sofiaprogramadora/chile\_con\_sentido

Web App destinada a encontrar eventos culturales y gratuitos con sentido -- Hackaton Desafío Latam

● Ruby

You can now pin up to 6 repositories.

**Gonzalo Salinas**  
gsalinase

Desarrollador Front-End, que cambió su aburrido y monótono trabajo por el apasionante y divertido mundo del desarrollo web. @gsalinase

📍 Santiago, CL  
✉ gsalinase91@gmail.com  
🌐 https://www.linkedin.com/in/...

**Organizations**

https://github.com/new

Create a New Repository x Gonzalo

← → ↻ 🏠 **GitHub, Inc. [US]** <https://github.com/new> 🔍 ☆ ⋮


View site information GitHub Pull requests Issues Marketplace Explore + ▾ 👤 ▾

## Create a new repository

A repository contains all the files for your project, including the revision history.

---

**Owner** **Repository name**

 gsalinase ▾ / nombre-del-repositorio ✓

Great repository names are short and memorable. Need inspiration?

**Description (optional)**

Escribe una breve descripción del contenido del repositorio

---

☒ **Public**  
Anyone can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

---

☐ **Initialize this repository with a README**  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾ | Add a license: **None** ▾ ⓘ

**Create repository**

Dale nombre al repositorio


Escribe una breve descripción





Crea el repositorio presionando este botón



gsalinase/nombre-del-repositorio x Gonzalo



← → ↻ 🏠 [GitHub, Inc. \[US\]](#) | <https://github.com/gsalinase/nombre-del-repositorio> 🔍 ☆ ⋮

 This repository Search Pull requests Issues Marketplace Explore + ▾ 👤 ▾

 [gsalinase](#) / [nombre-del-repositorio](#)  Watch ▾ 0  Star 0  Fork 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Insights](#) [Settings](#)


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

 Set up in Desktop or [HTTPS](#) [SSH](#)  

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).


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

```
echo "# nombre-del-repositorio" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:gsalinase/nombre-del-repositorio.git
git push -u origin master
```



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

```
git remote add origin git@github.com:gsalinase/nombre-del-repositorio.git
git push -u origin master
```



**...or import code from another repository**

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

Copia el link al portapapeles

```
$ git init  
Initialized empty Git repository  
(ruta: Users/mi-usuario/Desktop/mi-repo/)
```

## Iniciar Git

- Crea una carpeta.
- Entra a la carpeta e inicia Git con git init.

```
$ git - -all
```

## Agregar archivos a Git

- Crea un archivo html.
- Agrega los archivos usando git - -all.

```
$ git commit -m "first commit"  
[master (root-commit) f78f6f4] first commit  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 index.html
```

## Guardar los cambios

- Guarda tus cambios usando el comando git commit -m.



```
$ git remote add origin  
git@github.com:mi-usuario/  
nombre-del-repositorio.git
```

## Agrega el repositorio de GitHub en tu computador

- Escribe en el terminal git remote add origin.
- Luego, pega la dirección del repositorio al lado del comando escrito.

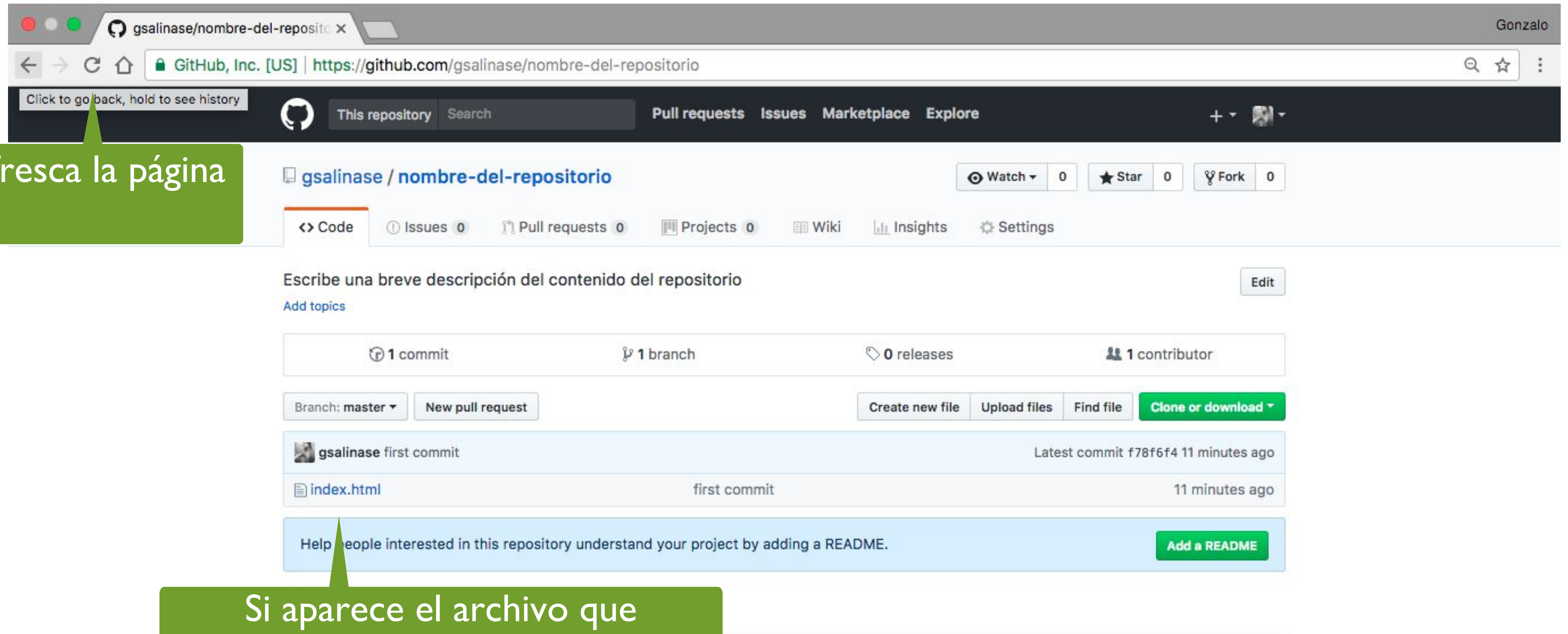


```
$ git push origin master  
Counting objects: 3, done.  
Writing objects: 100% (3/3), 219 bytes |  
219.00 KiB/s, done.  
Total 3 (delta 0), reused 0 (delta 0)
```

## Envía los cambios a GitHub

- Finalmente, envía los cambios a GitHub usando escribiendo **git push origin master**.

# Revisa si llegaron los cambios a GitHub



Refresca la página

Si aparece el archivo que creamos el proceso funcionó correctamente

# Trabajando con GitHub



# Subir y bajar archivos

```
$ git push origin master
```

Subir cambios

```
$ git pull origin master
```

Bajar cambios

# Manejo de repositorios

```
$ git remote
```

Revisar repositorios  
remotos

```
$ git remote -v
```

Conocer URL de  
repositorio

# Manejo de repositorios

```
$ git remote add origin user/  
mirepo.git
```

Añadir un repositorio

```
$ git remote show origin
```

Obtener información del  
repositorio remoto

```
$ git remote show nombre
```

Obtener información de repositorio remoto

```
$ git remote rename origin  
nuevoRepo
```

Renombrar el repositorio

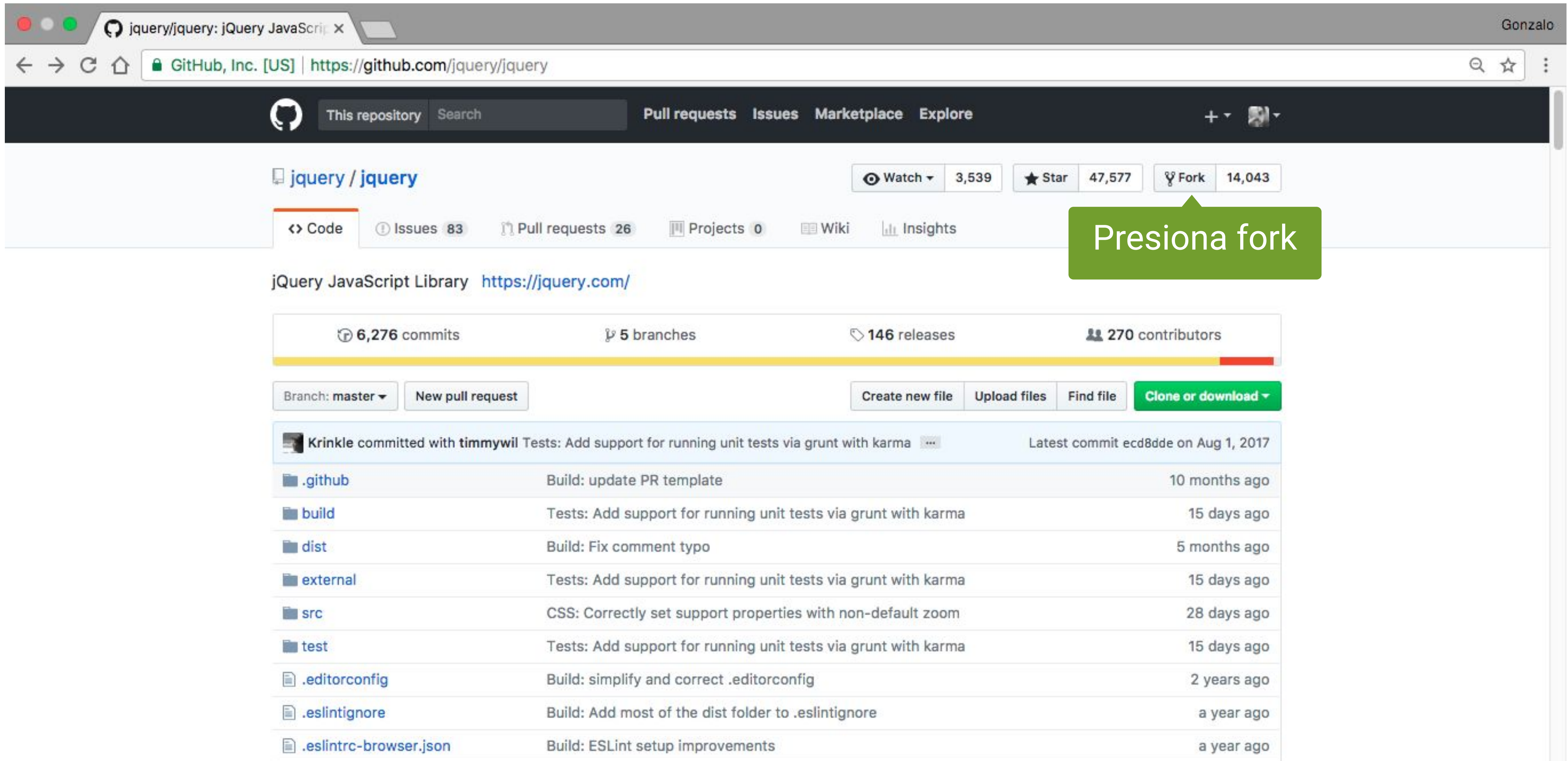
```
$ git rm nuevoRepo
```

Eliminar el repositorio

# Forkeando un repositorio



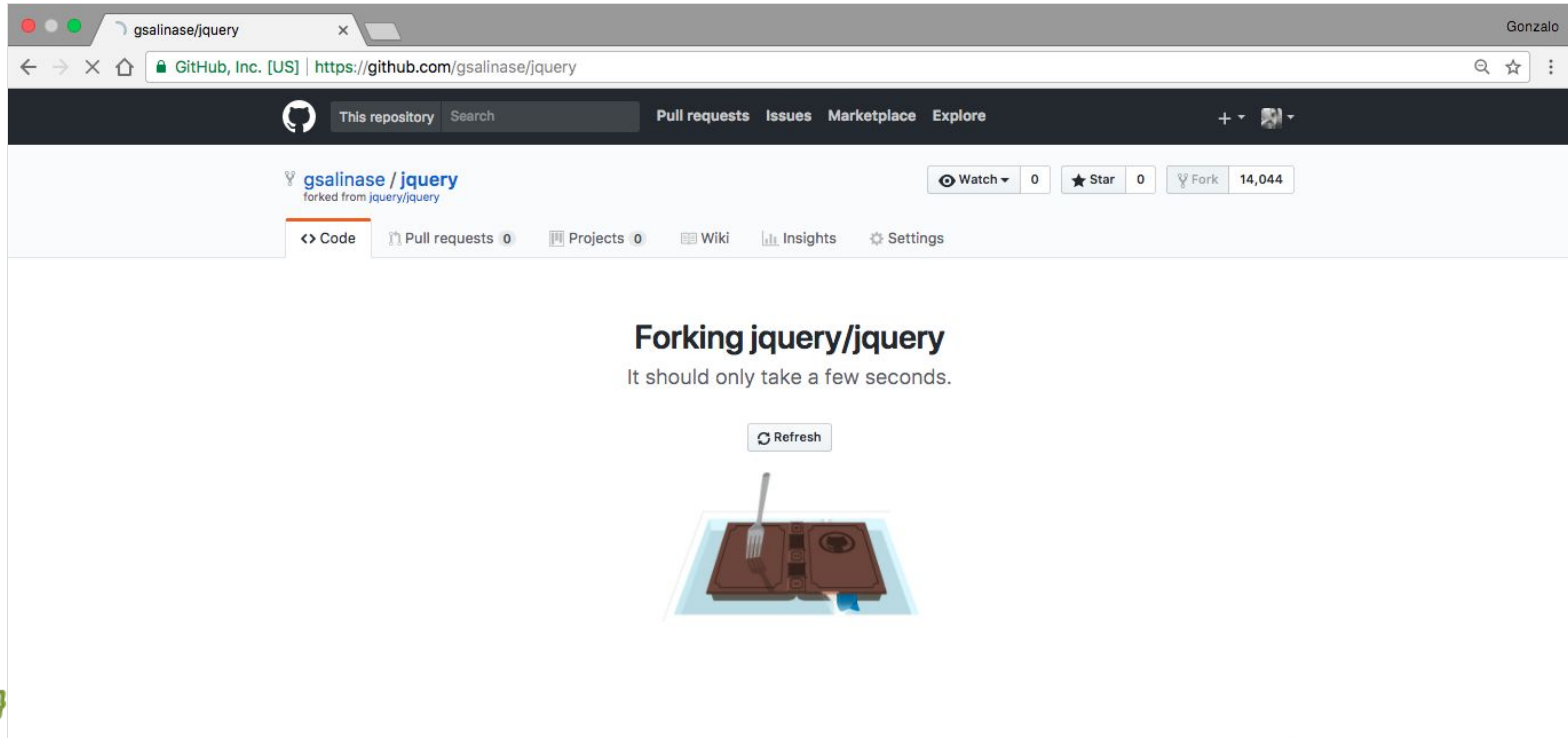
# 1. Presiona fork



The screenshot shows the GitHub repository page for jQuery. The browser's address bar displays the URL `https://github.com/jquery/jquery`. The repository name is `jquery / jquery`. The page shows 3,539 watches, 47,577 stars, and 14,043 forks. A green callout box with the text "Presiona fork" points to the "Fork" button. Below the repository name, there are tabs for Code, Issues (83), Pull requests (26), Projects (0), Wiki, and Insights. The repository description is "jQuery JavaScript Library" with a link to `https://jquery.com/`. The repository statistics show 6,276 commits, 5 branches, 146 releases, and 270 contributors. The "Clone or download" button is highlighted in green. The commit history table is as follows:

Commit	Message	Time
Krinkle committed with timmywil	Tests: Add support for running unit tests via grunt with karma	Latest commit ecd8dde on Aug 1, 2017
.github	Build: update PR template	10 months ago
build	Tests: Add support for running unit tests via grunt with karma	15 days ago
dist	Build: Fix comment typo	5 months ago
external	Tests: Add support for running unit tests via grunt with karma	15 days ago
src	CSS: Correctly set support properties with non-default zoom	28 days ago
test	Tests: Add support for running unit tests via grunt with karma	15 days ago
.editorconfig	Build: simplify and correct .editorconfig	2 years ago
.eslintignore	Build: Add most of the dist folder to .eslintignore	a year ago
.eslintrc-browser.json	Build: ESLint setup improvements	a year ago

## 2. Espera...





### 3. Revisa el repositorio forkeado

Si aparece  
“tu-usuario/  
el-repositorio  
clonado”,  
el fork fue todo  
un éxito

The screenshot shows a web browser window with the URL `https://github.com/gsalinase/jquery`. The page displays the repository details for 'gsalinase / jquery', which is a fork of 'jquery/jquery'. The repository has 14,044 forks, 0 stars, and 0 watches. The main content area shows the repository's history and a list of recent commits. The commits are as follows:

Commit	Author	Message	Time
Krinkle committed with timmywil	timmywil	Tests: Add support for running unit tests via grunt with karma	Latest commit ecd8dde on Aug 1, 2017
.github	Build: update PR template		10 months ago
build	Tests: Add support for running unit tests via grunt with karma		15 days ago
dist	Build: Fix comment typo		5 months ago
external	Tests: Add support for running unit tests via grunt with karma		15 days ago
src	CSS: Correctly set support properties with non-default zoom		28 days ago
test	Tests: Add support for running unit tests via grunt with karma		15 days ago
.editorconfig	Build: simplify and correct .editorconfig		2 years ago

# Clonando repositorios

# 1. Copia el link al portapapeles

The screenshot shows a web browser window displaying a GitHub repository page for 'gsalinase/nombre-del-repositorio'. The browser's address bar shows the URL 'https://github.com/gsalinase/nombre-del-repositorio'. The repository page includes a header with navigation links like 'This repository', 'Search', 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the header, the repository name is followed by 'Watch', 'Star', and 'Fork' buttons, each with a count of 0. A secondary navigation bar contains links for 'Code', 'Issues', 'Pull requests', 'Projects', 'Wiki', 'Insights', and 'Settings'. The main content area prompts the user to 'Escribe una breve descripción del contenido del repositorio' and shows repository statistics: '1 commit', '1 branch', '0 releases', and '1 contributor'. A 'Clone or download' button is visible. A dropdown menu is open from this button, showing 'Clone with SSH' as the selected option. The SSH link 'git@github.com:gsalinase/nombre-del-re' is highlighted with a blue selection box. A green callout box with a white arrow points to this link, containing the text 'Copia el enlace del repositorio'. Other options in the dropdown include 'Use HTTPS', 'Open in Desktop', and 'Download ZIP'.

gsalinase/nombre-del-reposito x Gonzalo

GitHub, Inc. [US] | https://github.com/gsalinase/nombre-del-repositorio

This repository Search Pull requests Issues Marketplace Explore

gsalinase / nombre-del-repositorio Watch 0 Star 0 Fork 0

<> Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Escribe una breve descripción del contenido del repositorio Edit

Add topics

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

gsalinase first commit

index.html first commit

Help people interested in this repository understand your project by adding a README.

Clone with SSH Use HTTPS

Use an SSH key and passphrase from account.

git@github.com:gsalinase/nombre-del-re

Open in Desktop Download ZIP

Copia el enlace del repositorio

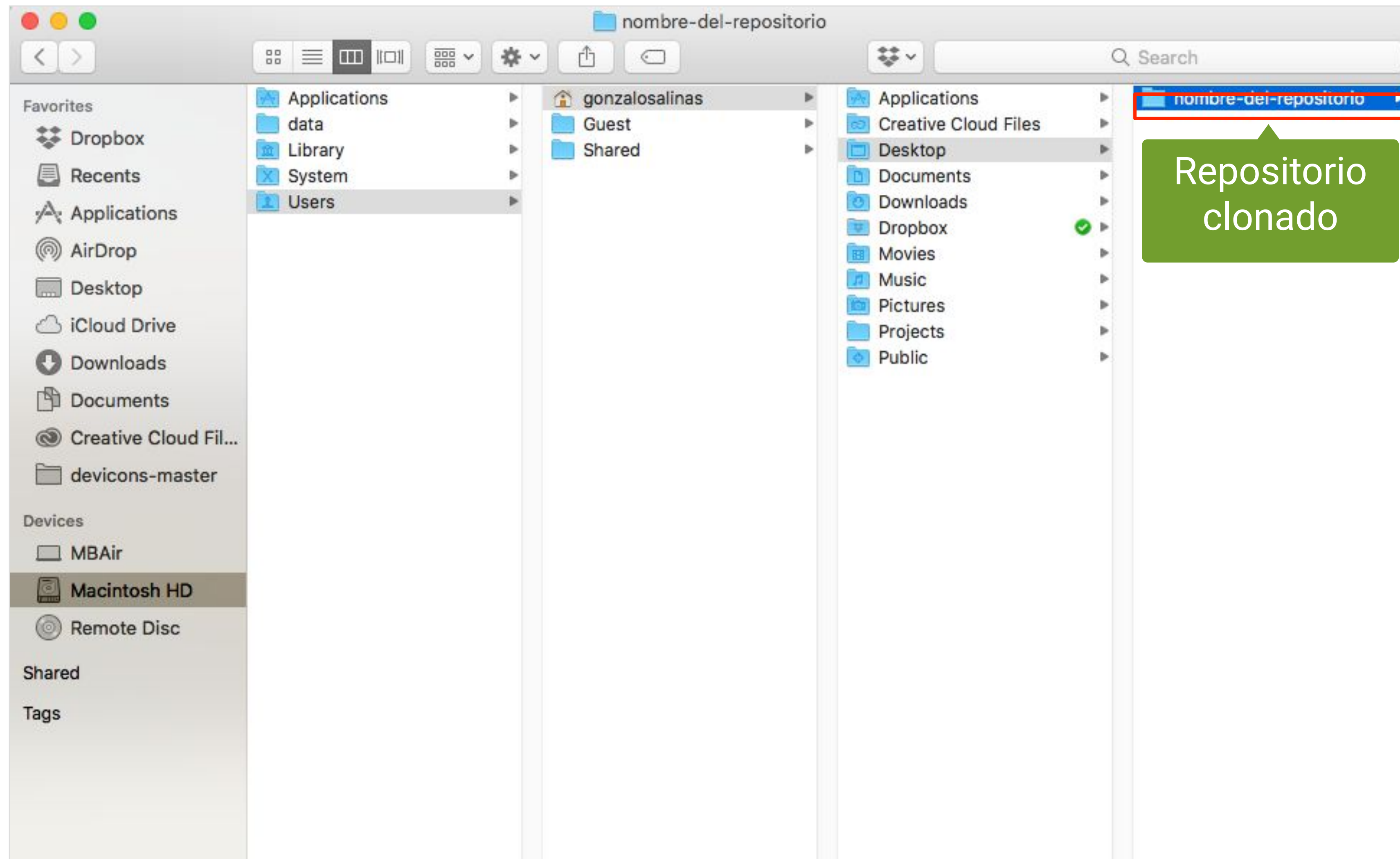


## 2. Clona el repositorio en tu computador

```
$ git clone git@github.com:mi-  
usuario/nombre-del-repositorio.git  
Cloning into 'nombre-del-repositorio'...  
remote: Counting objects: 3, done.  
remote: Total 3 (delta 0), reused 3 (delta 0),  
pack-reused 0  
Receiving objects: 100% (3/3), done.  
(ruta: Users/mi-usuario/Desktop/)
```

- Abre el terminal y escribe git clone, junto con el enlace copiado.

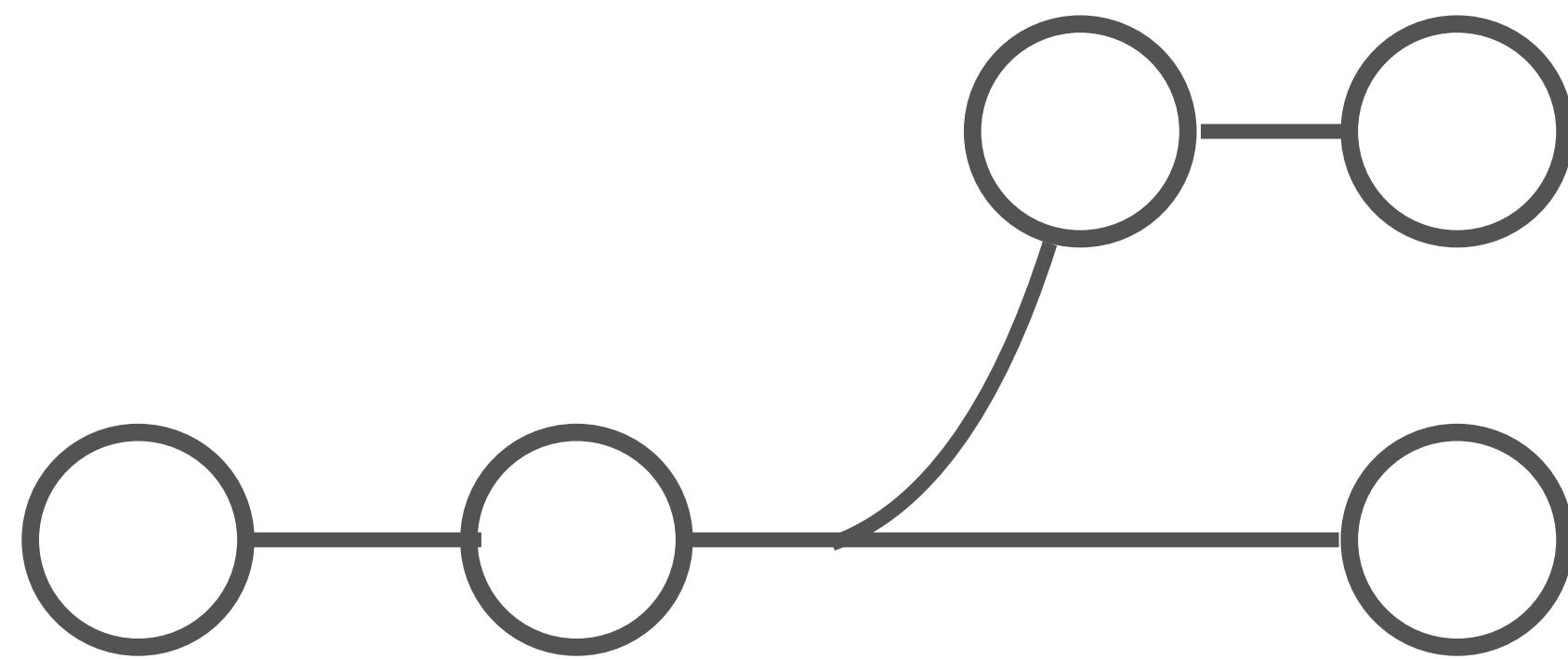
### 3. Revisa el repositorio en tu computador



# Conceptos importantes

- **Git remote:** Links directos hacia otros repositorios.  
Puedes crear, ver eliminar las conexiones hacia otros repositorios.
- **Git fetch:** Importa los cambios de un repositorio remoto hacia un repo local. Git fetch agrupa todas las ramas, y cambios de la repo remota sin combinarlas.
- **Git merge:** Combina los cambios de un repositorio remoto hacia un repo local.
- **Git pull:** Es una combinación de git fetch y git pull.  
Es importante revisar los cambios locales antes de hacer un pull debido a que esto puede causar conflictos entre el trabajo local y el remoto.

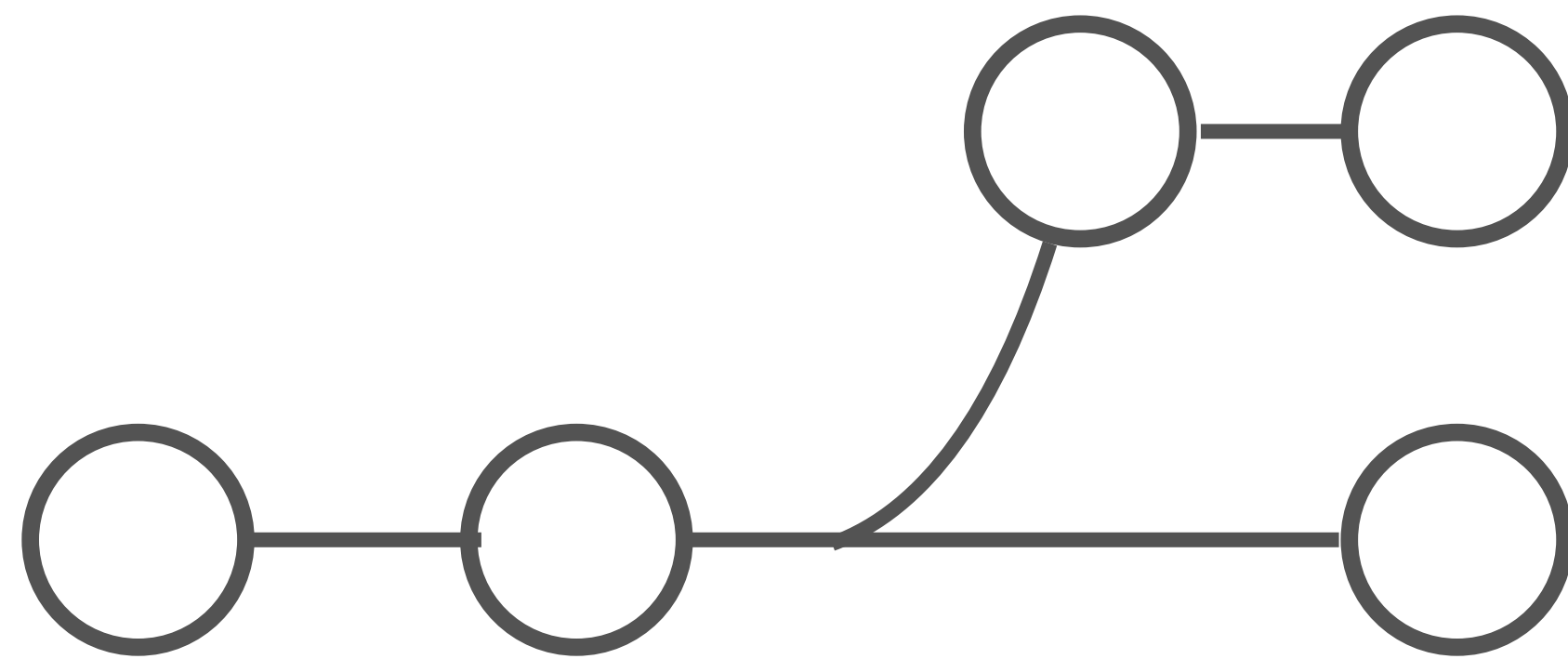
# Git Branch



- Git branch representa a una línea de desarrollo en las cuales se pueden editar y hacer cambios.
- Estás son importantes a la hora de trabajar con otras personas ya que dan un espacio de trabajo único para cada desarrollador.



# ¿Para qué sirven los branches?



- Para coordinar el trabajo con varias personas
- Para implementar cambios complejos y si no resultan destruirlos fácilmente
- Para poder implementar hotfixes mientras estamos trabajando en partes complejas que no hemos terminado de implementar



# Comandos para branches

- **git branch**: Muestra todas las ramas del proyecto.
- **git branch [nombre rama]**: Crea la rama especificada.
- **git checkout [nombre rama]**: Cambia a la rama especificada.
- **git checkout -b [nombre rama]**: Crea y cambia a la rama especificada.
- **git merge [nombre rama]**: Une una rama especificada a la rama donde estás situado.
- **git branch -d [nombre rama]**: Borra una rama ya mergeada.
- **git branch -D [nombre rama]**: Borra una rama no mergeada.

# GitHub Pages

# ¿Qué es GitHub Pages?

GitHub Pages es una funcionalidad de GitHub con la que podrás subir sitios web estáticos de manera gratuita.

# Crear el branch “gh-pages”

```
$ git branch gh-pages
```

(ruta: Users/mi-usuario/Desktop/mi-repo/)

- En el terminal debemos escribir  
git branch gh-pages

# Cambiar de rama a gh-pages

```
$ git checkout gh-pages
Switched to branch 'gh-pages'
$ git branch

*gh-pages
master
```

- Cambiemos de rama usando el comando `git checkout gh-pages`.
- Luego, veamos en que rama estamos usando `git branch`.

# Hacer un push para gh-pages

```
$ git push gh-pages
```

```
Total 0 (delta 0), reused 0 (delta 0)
```

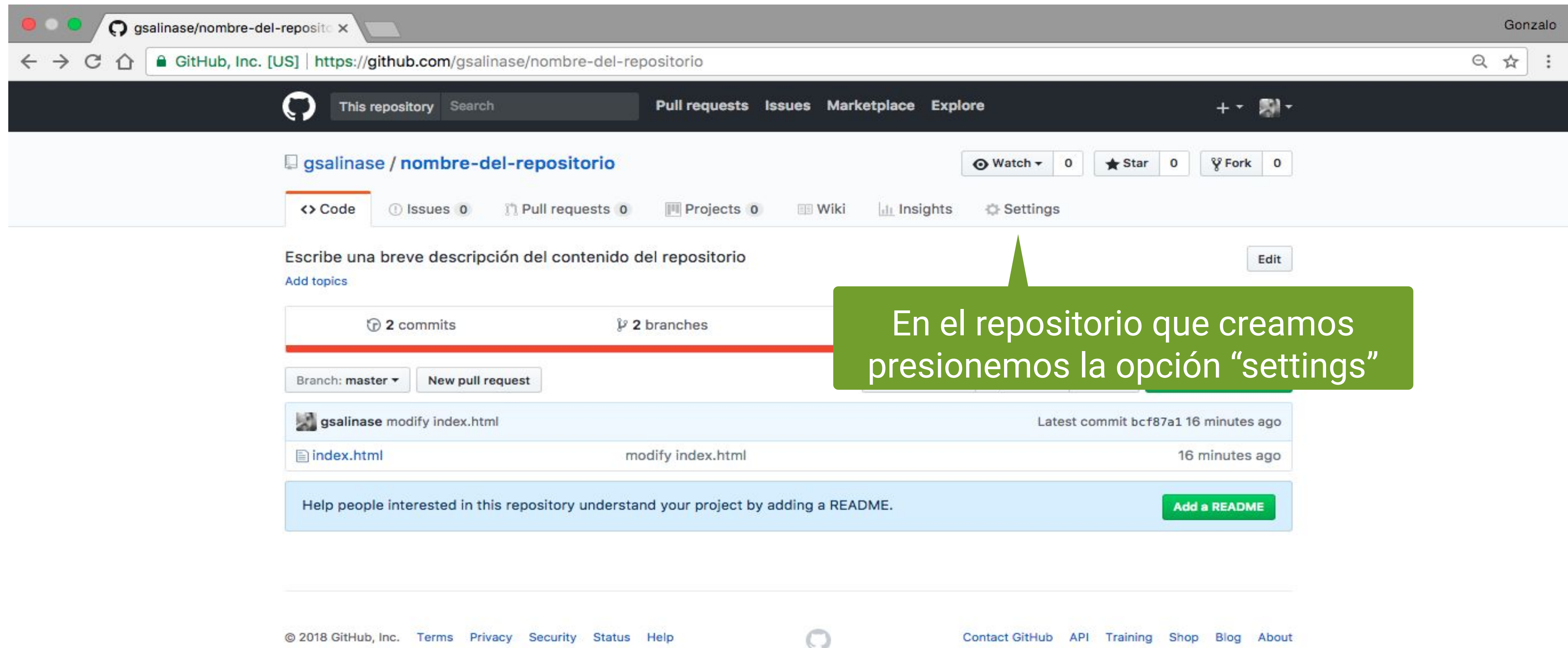
```
To
```

```
github.com:gsalinase/nombre-del-reposito  
rio.git
```

```
* [new branch]    gh-pages -> gh-pages
```

- Realizar un push de los cambios en la nueva rama usando git push gh-pages

# Revisar si se creó la página en GitHub Pages



The screenshot shows a web browser window displaying a GitHub repository page. The browser's address bar shows the URL `https://github.com/gsalinase/nombre-del-repositorio`. The repository name is `gsalinase / nombre-del-repositorio`. The page includes navigation tabs for `Code`, `Issues`, `Pull requests`, `Projects`, `Wiki`, `Insights`, and `Settings`. A green callout box with a pointer highlights the `Settings` tab, containing the text: "En el repositorio que creamos presionemos la opción 'settings'". The repository details show 2 commits and 2 branches. A commit by `gsalinase` is listed, modifying `index.html` 16 minutes ago. At the bottom, there is a prompt to "Add a README" with a green button labeled `Add a README`.



Options x Gonzalo

← → ↻ 🏠 🔒 GitHub, Inc. [US] | <https://github.com/gsalinase/nombre-del-repositorio/settings> 🔍 ☆ ⋮

## GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

✓ Your site is published at <https://gsalinase.github.io/nombre-del-repositorio/>

**Source**  
Your GitHub Pages site is currently being built from the gh-pages branch. [Learn more.](#)

gh-pages branch ▾ Save

**Theme Chooser**  
Select a theme to publish your site with a Jekyll theme. [Learn more.](#)

Choose a theme

**Custom domain**  
Custom domains allow you to serve your site from a domain other than gsalinase.github.io. [Learn more.](#)

Save

☒ **Enforce HTTPS**  
— Required for your site because you are using the default domain (gsalinase.github.io)

HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site. When HTTPS is enforced, your site will only be served over HTTPS. [Learn more.](#)

Copia el enlace  
y pégalo en el  
navegador

Haz scroll  
hasta que  
encuentres  
la sección  
“GitHub Pages”

- Si todo salió bien verás la página con el enlace que pegamos.
- Es importante saber que los links de github pages son de la siguiente manera: **`https://usuario.github.io/nombre-del-repositorio/`**

# Dominio personalizado

# ¿Qué es un dominio?

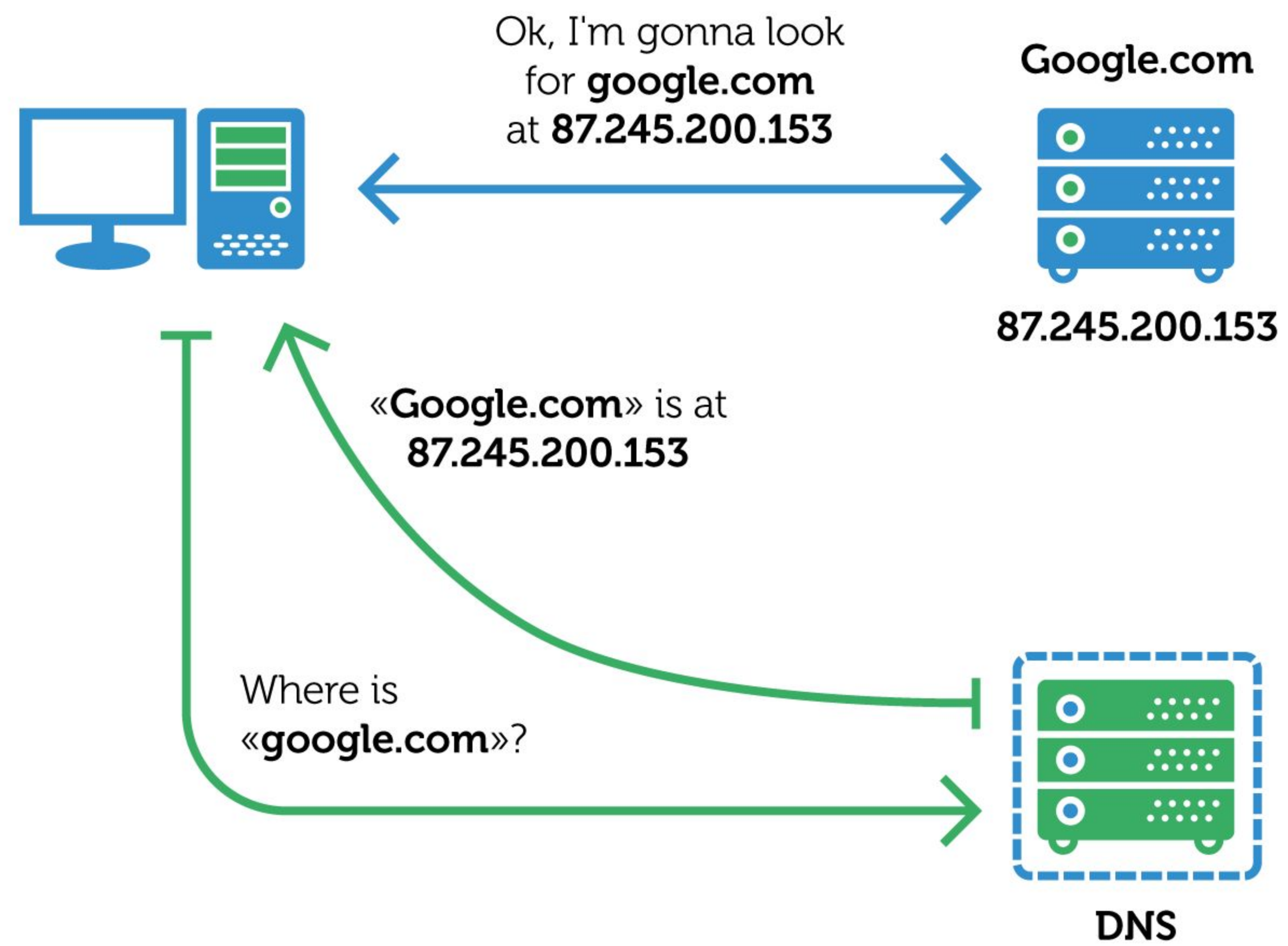
En internet, un dominio es el nombre único y exclusivo que se le da a un sitio web, con él podemos visitar el sitio desde nuestro navegador. Es la identidad del sitio.

**DNS = Sistema de nombres de dominio**

**DNS asocia el dominio con una IP**



**IP = dirección numérica que identifica un dispositivo conectado a una red**



© 2016 AO Kaspersky Lab. All Rights Reserved.

**Un dominio es un recurso limitado**

# Servicio de asignación de dominio



**{desafío}**  
**latam\_**

*Academia de  
talentos digitales*

[www.desafiolatam.com](http://www.desafiolatam.com)