

fundación **esplai**  
ciudadanía comprometida

# T Systems

Let's power  
higher performance



**@mihifidem**

**creativity** is **intelligence** having **fun**

# Fundamentos de Git



# Fundamentos de Git

## ¿Qué es Git?

- Es un software de control de versiones, es decir, un software que nos permite llevar un registro de todos los cambios y avances del proyecto.



# Fundamentos de Git

## Historia de Git
























# Fundamentos de Git

¿Qué es un repositorio de Git?



¿Por qué nace Git?

Filename	Last modified
 index.html	2017-02-26 07:56:39
 index_.html	2009-07-20 09:23:19
 index_.html	2009-07-25 19:22:37
 index_.html	2009-07-28 10:05:07
 index_.html	2009-08-07 08:29:23
 index_.html	2009-08-08 03:24:56
 index_.html	2009-08-10 08:32:31
 index_.html	2009-08-17 22:07:20
 index_.html	2009-08-21 06:55:56
 index_.html	2009-09-05 01:30:24
 index_.html	2009-10-11 09:51:32
 index_.html	2009-10-14 07:11:03
 index_.html	2009-10-16 10:15:43
 index_.html	2009-10-23 22:59:09
 index_.html	2009-10-24 14:31:05
 index_.html	2009-10-26 00:06:11
 index_.html	2009-11-09 09:16:28
 index_.html	2009-11-23 10:02:03
 index_.html	2009-11-24 10:21:46
 index_.html	2009-11-30 22:58:23
 index_.html	2009-12-19 01:50:02
 index_.html	2010-04-23 23:14:22
 index_.html	2010-06-28 10:16:59
 index_.html	2010-08-19 08:19:31
 index_.html	2010-11-29 08:22:46
 index_.html	2010-12-28 16:28:04
 index_.html	2011-02-25 09:10:27
 index_.html	2011-03-19 10:48:02
 index_.html	2011-08-24 05:28:04
index_.html	2011-11-01 10:20:14
index_.html	2013-06-29 02:08:26

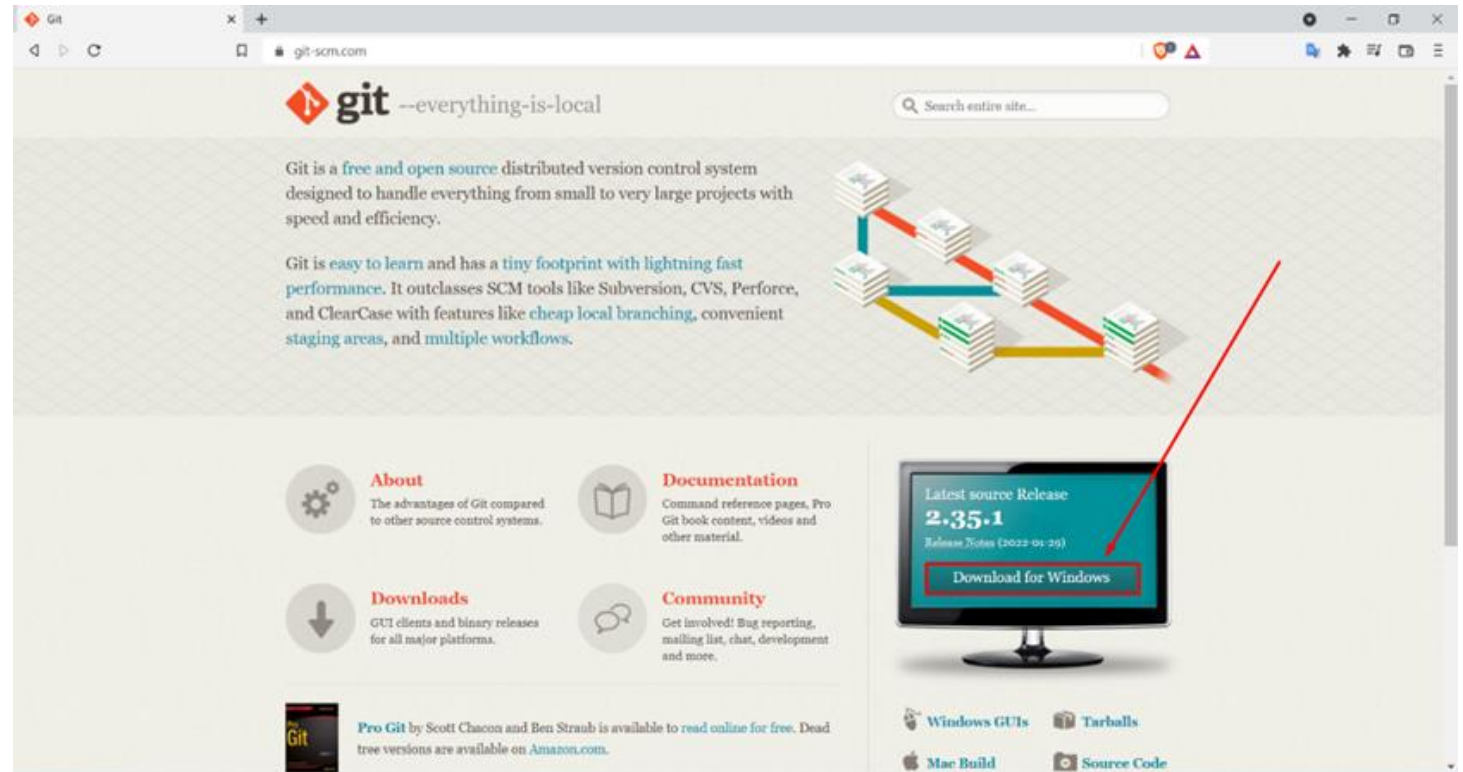
# Fundamentos de Git



# Fundamentos de Git

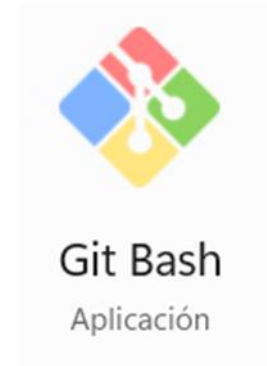
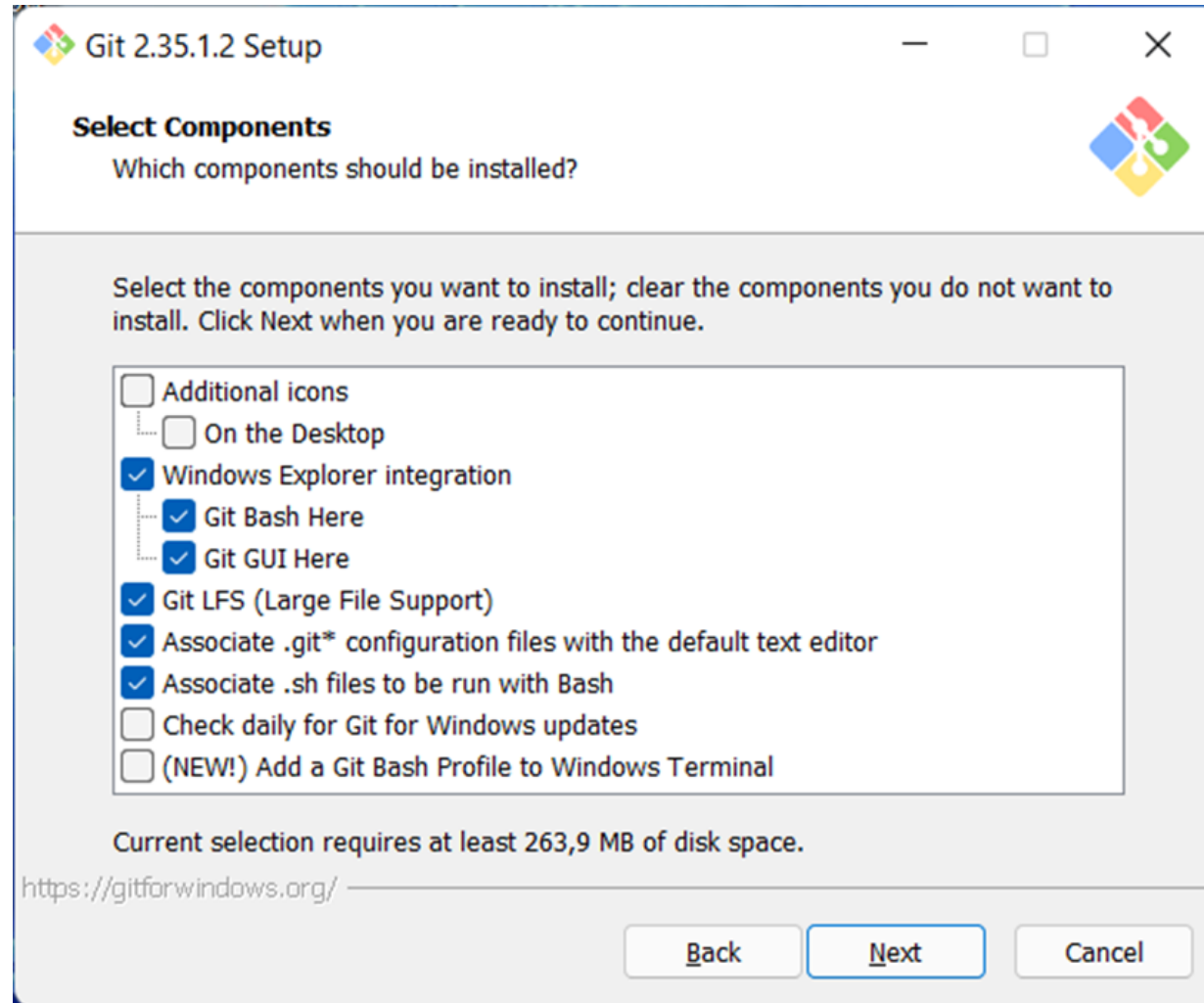
## Instalando Git

<https://git-scm.com/>



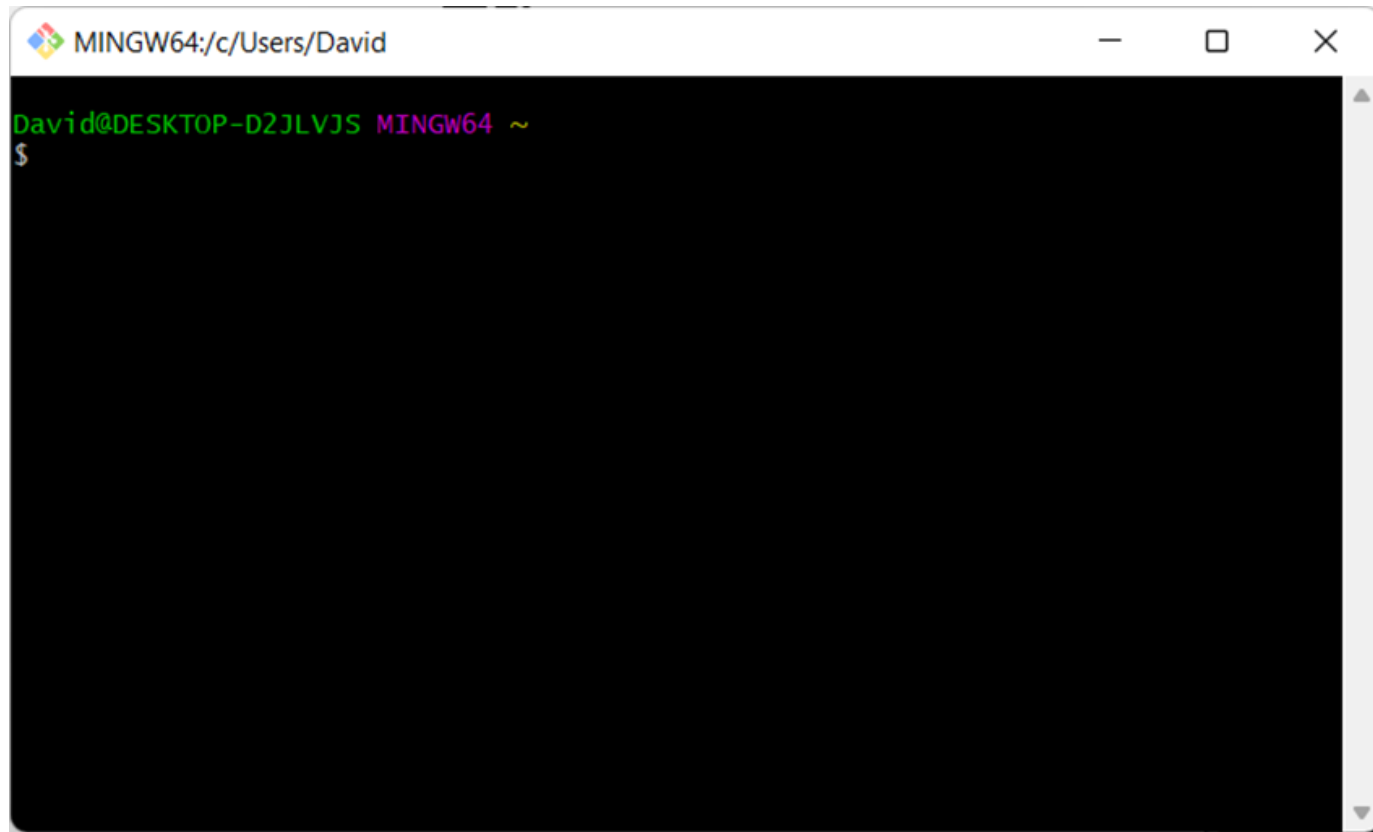
# Fundamentos de Git

## Instalando Git

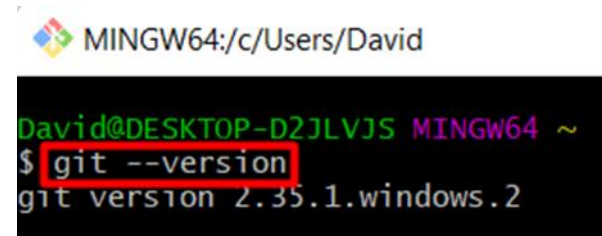


# Fundamentos de Git

## Instalando Git



```
MINGW64:/c/Users/David  
David@DESKTOP-D2JLVJS MINGW64 ~  
$
```

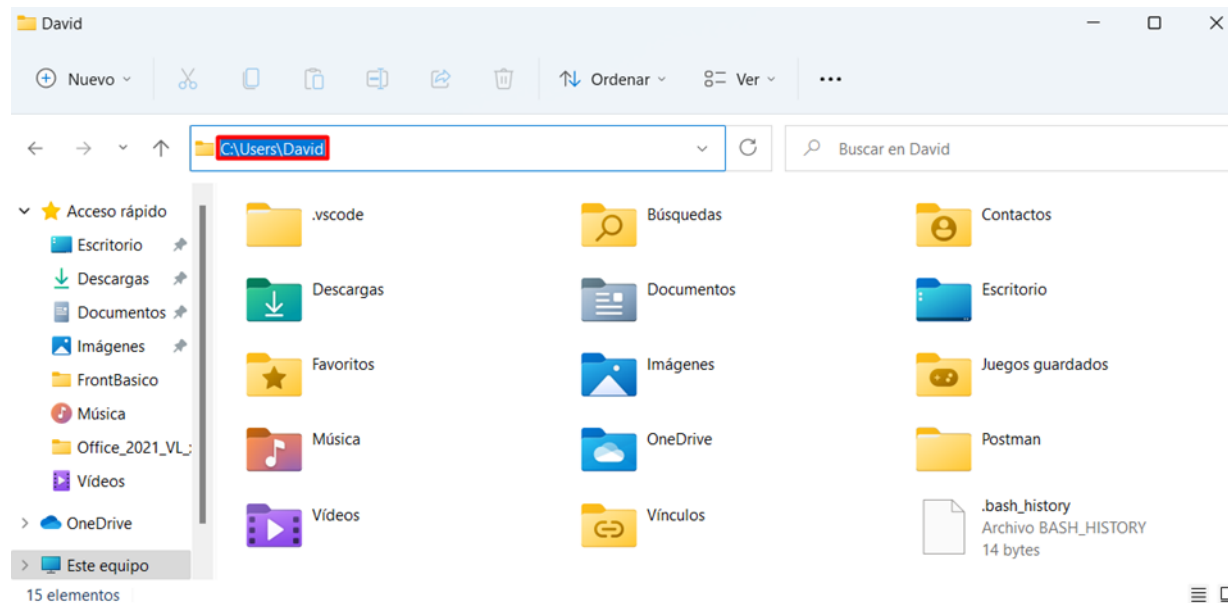


```
MINGW64:/c/Users/David  
David@DESKTOP-D2JLVJS MINGW64 ~  
$ git --version  
git version 2.35.1.windows.2
```

# Fundamentos de Git

## Aprendiendo a movernos con la terminal

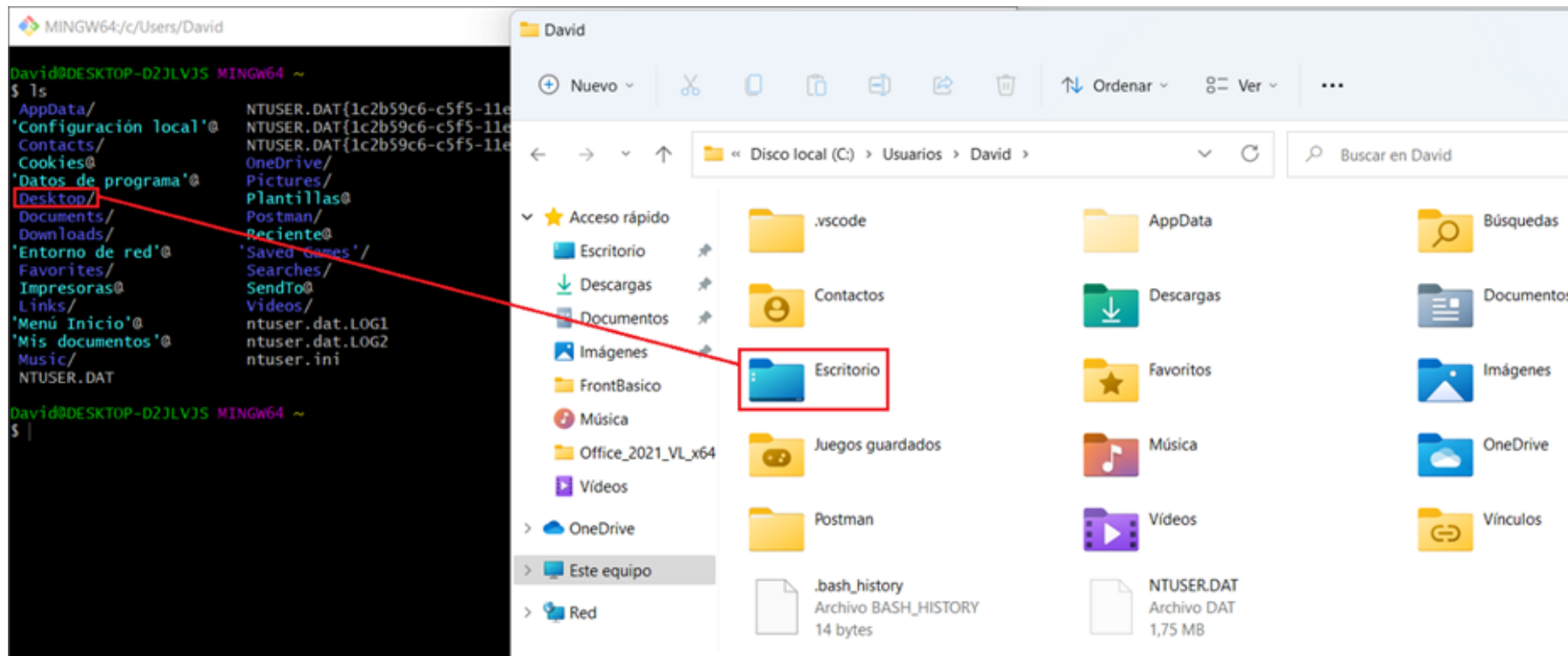
pwd: nos permite ver la ruta en la que estamos situados actualmente:



# Fundamentos de Git

## Aprendiendo a movernos con la terminal

ls: Para listar los elementos de la ruta actual:



# Fundamentos de Git

## Aprendiendo a movernos con la terminal

cd: lo utilizamos para cambiar de directorio, de hecho, si ejecutamos pwd, podemos ver que hemos cambiado de directorio:

```
MINGW64:/c/Users/David/desktop

David@DESKTOP-D2JLVJS MINGW64 ~
$ pwd
/c/Users/David

David@DESKTOP-D2JLVJS MINGW64 ~
$ cd desktop

David@DESKTOP-D2JLVJS MINGW64 ~/desktop
$ pwd
/c/Users/David/desktop
```

```
MINGW64:/c/Users/David/Desktop

David@DESKTOP-D2JLVJS MINGW64 ~
$ cd Desktop/

David@DESKTOP-D2JLVJS MINGW64 ~/Desktop
$ |
```

# Fundamentos de Git

## Aprendiendo a movernos con la terminal

MINGW64:/c/Users/David

David@DESKTOP-D2JLVJS MINGW64 ~

\$ **cd --help**

cd: cd [-L|[-P [-e]] [-@]] [dir]

Change the shell working directory.

Change the current directory to DIR. The default DIR is the value of the HOME shell variable.

The variable CDPATH defines the search path for the directory containing DIR. Alternative directory names in CDPATH are separated by a colon (:). A null directory name is the same as the current directory. If DIR begins with a slash (/), then CDPATH is not used.

If the directory is not found, and the shell option 'cdable\_vars' is set, the word is assumed to be a variable name. If that variable has a value, its value is used for DIR.

Options:

- L force symbolic links to be followed: resolve symbolic links in DIR after processing instances of '..'
- P use the physical directory structure without following symbolic links: resolve symbolic links in DIR before processing instances of '..'
- e if the -P option is supplied, and the current working directory cannot be determined successfully, exit with a non-zero status
- @ on systems that support it, present a file with extended attributes as a directory containing the file attributes

The default is to follow symbolic links, as if '-L' were specified.

**'..' is processed by removing the immediately previous pathname component back to a slash or the beginning of DIR.**

Exit Status:

Returns 0 if the directory is changed, and if \$PWD is set successfully when -P is used; non-zero otherwise.

MINGW64:/c/Users/David

David@DESKTOP-D2JLVJS MINGW64 ~/Desktop

\$ pwd

/c/Users/David/Desktop

David@DESKTOP-D2JLVJS MINGW64 ~/Desktop

\$ **cd ..**

David@DESKTOP-D2JLVJS MINGW64 ~

\$ pwd

/c/Users/David



# Fundamentos de Git

## Aprendiendo a movernos con la terminal

MINGW64:/c/Users/David

```
David@DESKTOP-D2JLVJS MINGW64 ~/desktop
$ pwd
/c/Users/David/desktop

David@DESKTOP-D2JLVJS MINGW64 ~/desktop
$ cd ..

David@DESKTOP-D2JLVJS MINGW64 ~
$ pwd
/c/Users/David
```



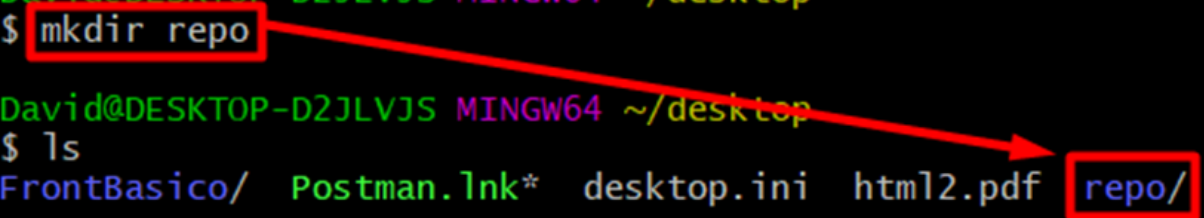
mkdir: nos permite crear un directorio:

MINGW64:/c/Users/David/desktop

```
David@DESKTOP-D2JLVJS MINGW64 ~/desktop
$ ls
FrontBasico/  Postman.lnk*  desktop.ini  html2.pdf

David@DESKTOP-D2JLVJS MINGW64 ~/desktop
$ mkdir repo

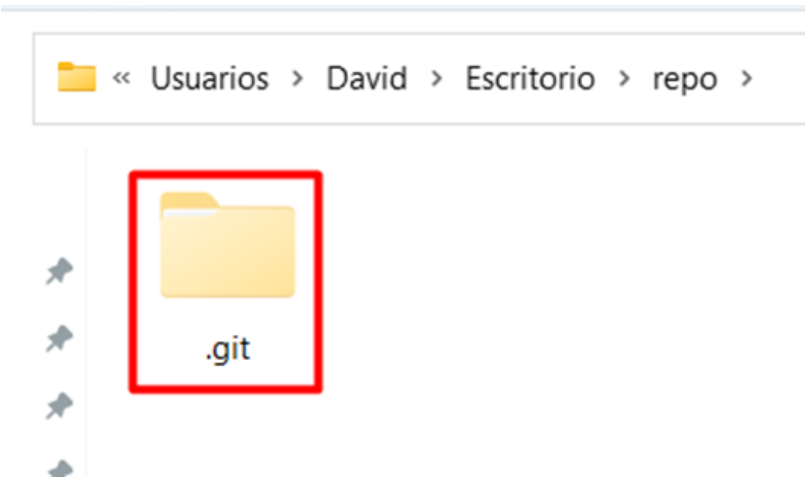
David@DESKTOP-D2JLVJS MINGW64 ~/desktop
$ ls
FrontBasico/  Postman.lnk*  desktop.ini  html2.pdf  repo/
```





# Fundamentos de Git

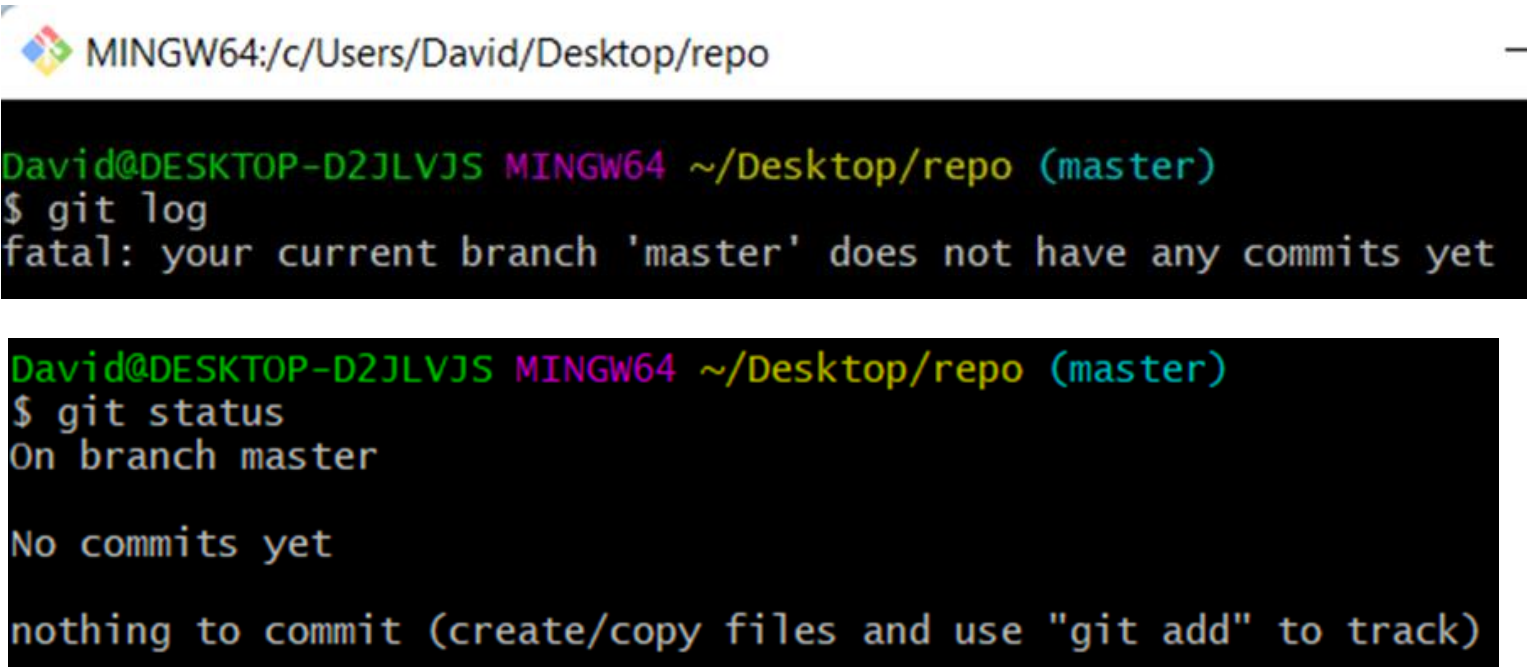
## Iniciando un nuevo repositorio de Git en local



Supongo que las cosas se están poniendo bastante serias

# Fundamentos de Git

## Iniciando un nuevo repositorio de Git en local



A terminal window with a title bar showing the path 'MINGW64:/c/Users/David/Desktop/repo'. The terminal displays two commands and their outputs. The first command is 'git log', which results in a fatal error because the 'master' branch has no commits. The second command is 'git status', which shows the current state of the repository as 'On branch master' with 'No commits yet' and a suggestion to use 'git add' to track new files.

```
MINGW64:/c/Users/David/Desktop/repo

David@DESKTOP-D2JLVJS MINGW64 ~/Desktop/repo (master)
$ git log
fatal: your current branch 'master' does not have any commits yet

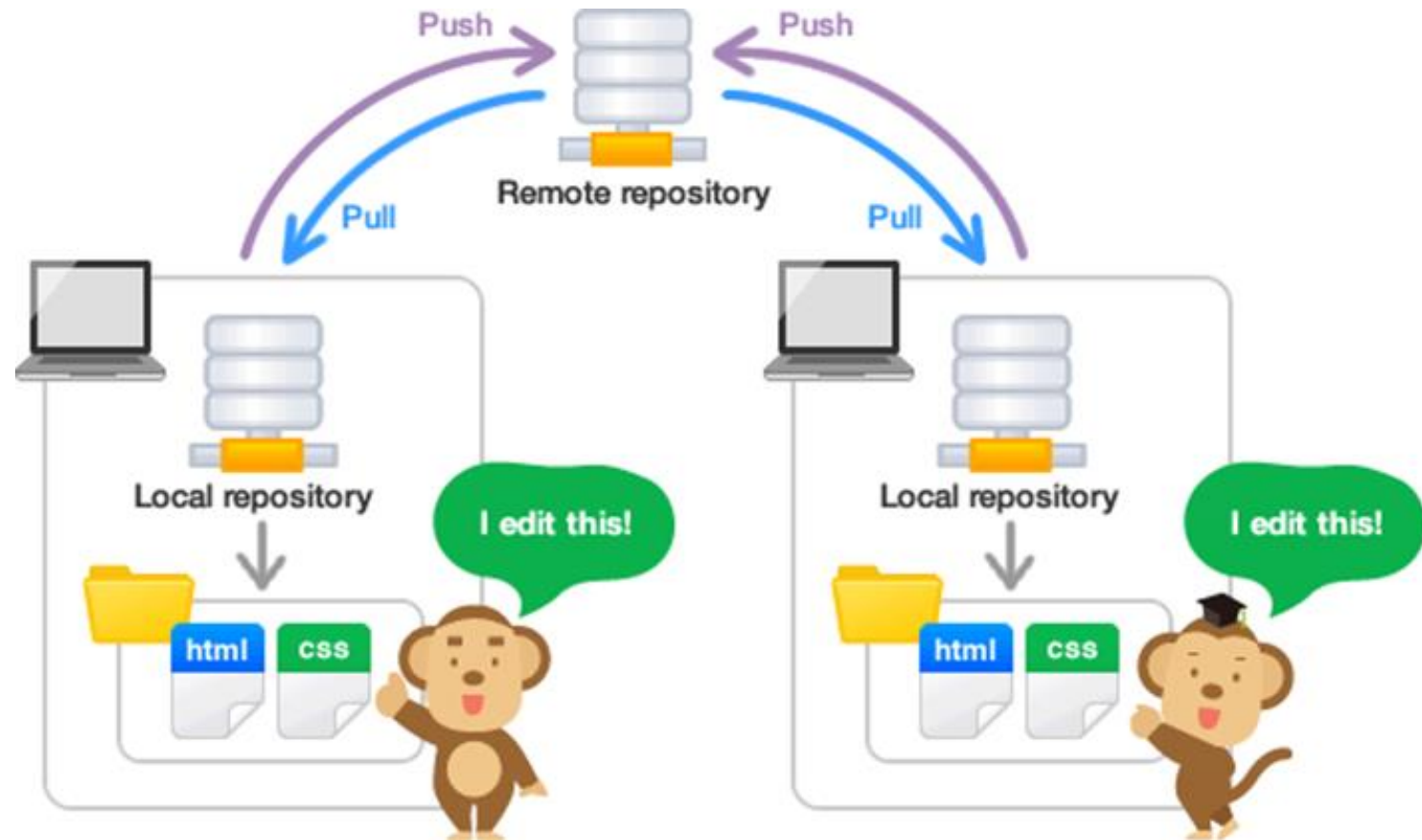
David@DESKTOP-D2JLVJS MINGW64 ~/Desktop/repo (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

# Fundamentos de Git

## Tipos de repositorio en GIT



# Fundamentos de Git

Repositorios en la nube GitHub, GitLab, BitBucket...

[GitHub](#)  
[GitLab](#)  
[BitBucket](#)



# Fundamentos de Git

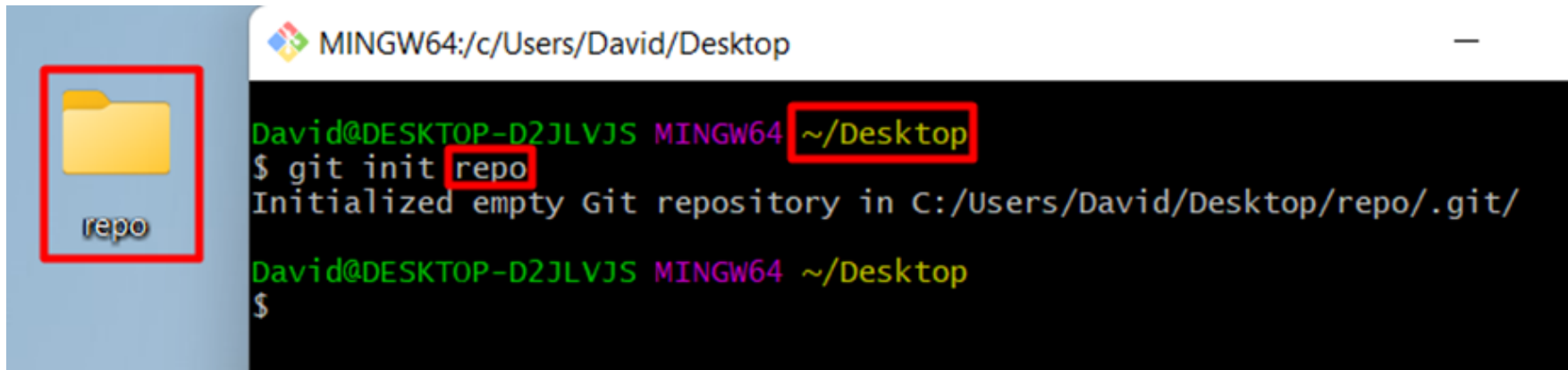
**Octocat, la mascota de GitHub**



# Fundamentos de Git

## Creando un repositorio de GIT (local repository)

git init



The screenshot shows a Windows File Explorer window on the left with a folder named 'repo' highlighted by a red rectangle. To the right is a Windows Command Prompt window titled 'MINGW64:/c/Users/David/Desktop'. The prompt shows the following commands and output:

```
David@DESKTOP-D2JLVJS MINGW64 ~/Desktop
$ git init repo
Initialized empty Git repository in C:/Users/David/Desktop/repo/.git/

David@DESKTOP-D2JLVJS MINGW64 ~/Desktop
$
```

Red rectangles highlight the 'repo' folder in the File Explorer, the 'repo' argument in the command prompt, and the '~/Desktop' path in the prompt's title bar.

# Fundamentos de Git

## Creando un repositorio de GIT (local repository)

git help init



The image shows a web browser window displaying the 'git-init(1)' documentation page. The page title is 'git-init(1)' and the URL is 'C:/Program%20Files/Git/mingw64/share/doc/git-doc/git-init.html'. The main heading is 'DESCRIPTION'. The text describes the 'git init' command, which creates an empty Git repository with subdirectories for objects, refs/heads, refs/tags, and template files. It also mentions the '--initial-branch' option. A red arrow points from the text 'open this tab in a browser' to the browser window. In the background, a terminal window is visible, showing the command 'git help init' being executed.

DESCRIPTION

This command creates an empty Git repository - basically a `.git` directory with subdirectories for `objects`, `refs/heads`, `refs/tags`, and template files. An initial branch without any commits will be created (see the `--initial-branch` option below for its name).

If the `$GIT_DIR` environment variable is set then it specifies a path to use instead of `./.git` for the base of the repository.

If the object storage directory is specified via the `$GIT_OBJECT_DIRECTORY` environment variable then the sha1 directories are created underneath - otherwise the default `$GIT_DIR/objects` directory is used.

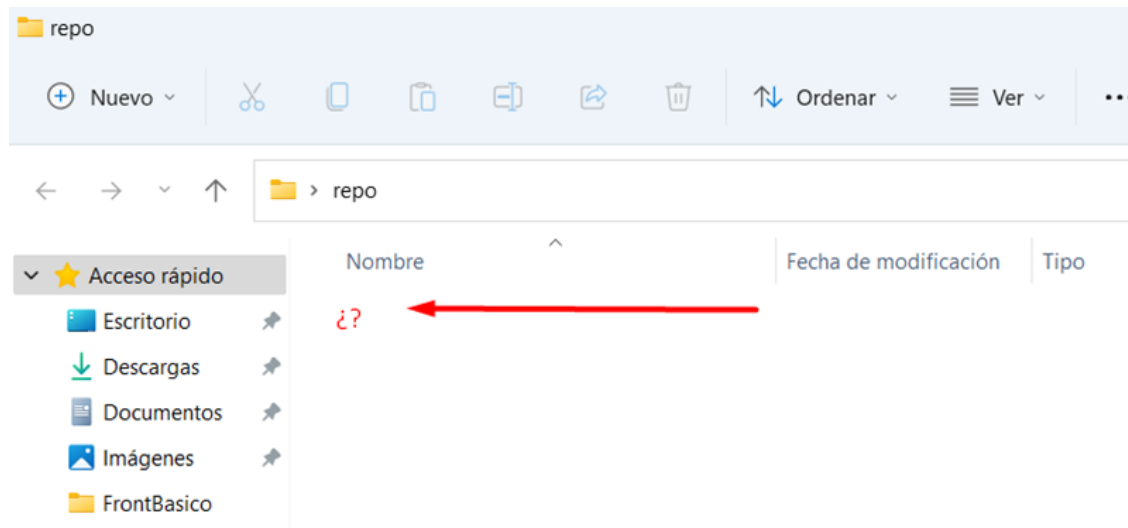
Running `git init` in an existing repository is safe. It will not overwrite things that are already there. The primary reason for rerunning `git init` is to pick up newly added templates (or to move the repository to another place if `--separate-git-dir` is given).

open this tab in a browser

MINGW64/c/Users/David/Desktop  
David@DESKTOP-02JLV25 MINGW64 ~/Desktop  
\$ git help init  
\$

# Fundamentos de Git

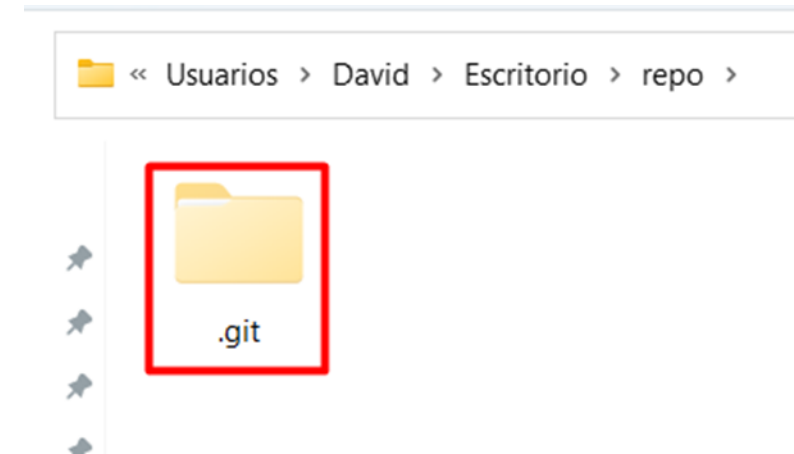
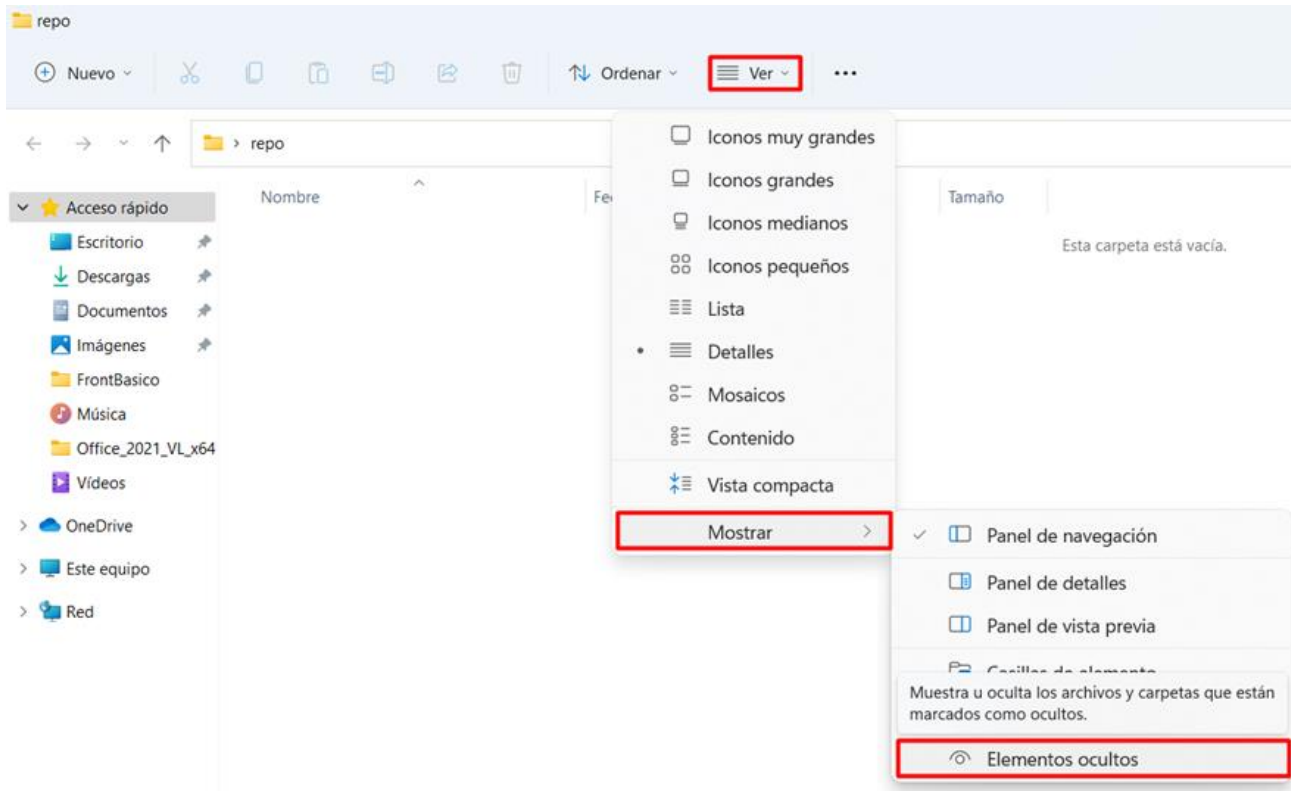
## Creando un repositorio de GIT (local repository)





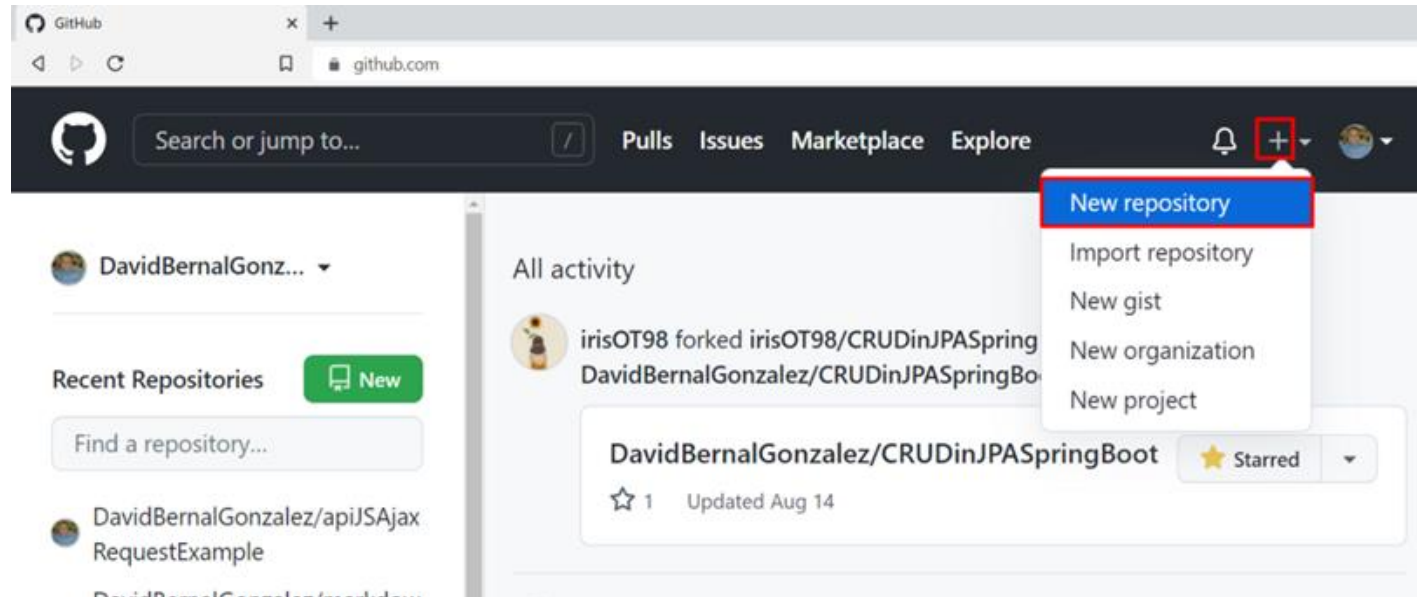
# Fundamentos de Git

## Creando un repositorio de GIT (local repository)



# Fundamentos de Git

## Creando un repositorio de GIT (local repository)



# Fundamentos de Git


## Creando un repositorio de GIT (local repository)

### Create a new repository

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

[Import a repository.](#)

Owner \*


 DavidBernalGonzalez ▾

Repository name \*

miPrimerRepoEnGitHub ✓

Great repository names are short and memorable. Need inspiration? How about [expert-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.

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.](#)

This will set  **main** as the default branch. Change the default name in your [settings](#).

**Create repository**

# Fundamentos de Git

## Creando un repositorio de GIT (local repository)

The image shows a GitHub repository page for 'DavidBernalGonzalez / miPrimerRepoEnGitHub' and a terminal window demonstrating the cloning process.

**GitHub Repository Page:**

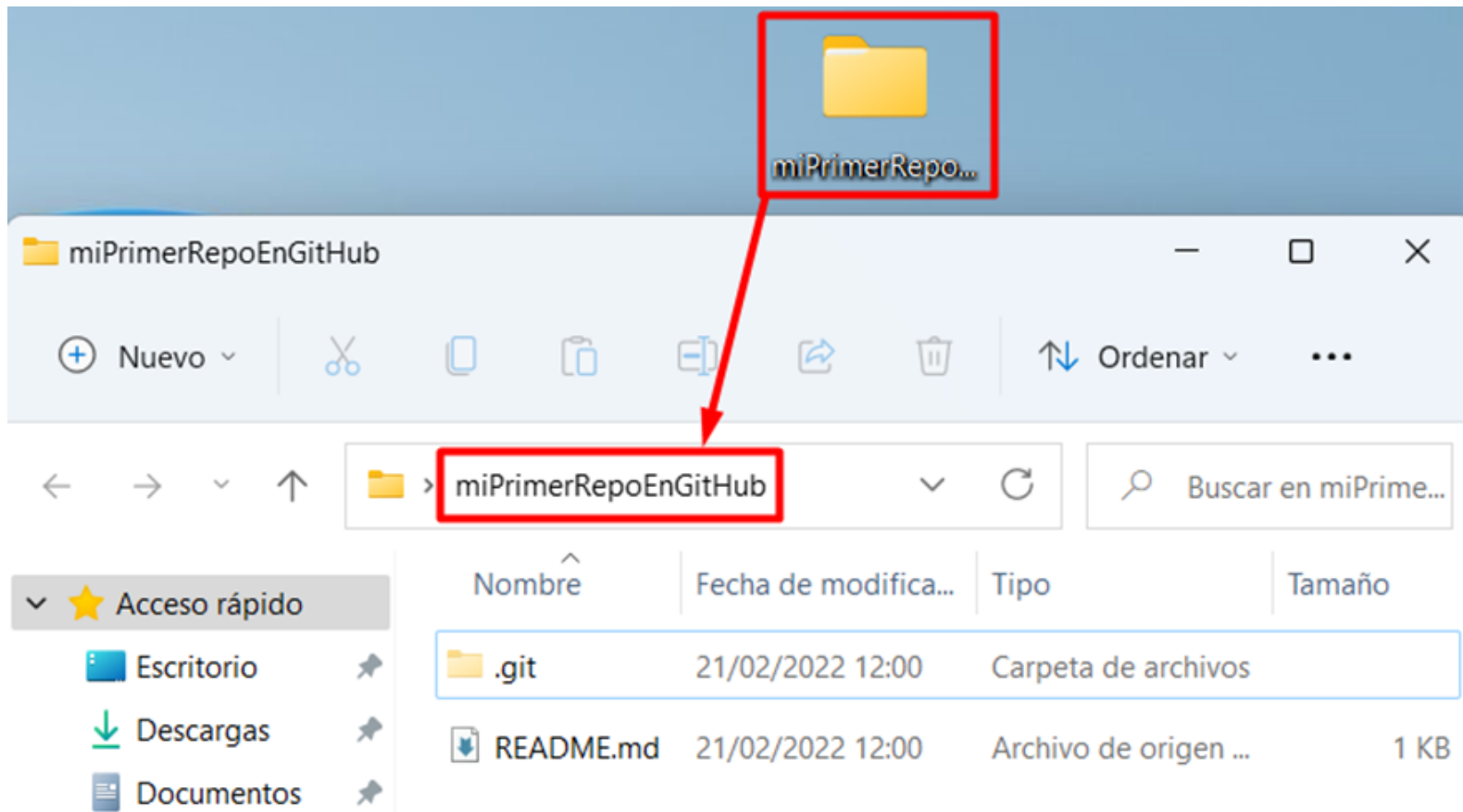
- Repository name: `miPrimerRepoEnGitHub` (Public)
- Navigation tabs: `<> Code`, `Issues`, `Pull requests`, `Actions`, `Projects`, `Wiki`, `Security`, `Insights`, `Settings`
- Branches: `main` (1 branch), `tags` (0 tags)
- Initial commit by DavidBernalGonzalez
- Files: `README.md`
- Buttons: `Go to file`, `Add file`, `Code` (highlighted with a red box)
- Clone options: `HTTPS` (highlighted with a red box), `SSH`, `GitHub CLI`
- Web URL: `https://github.com/DavidBernalGonzalez/miPrimerRepoEnGitHub.git` (highlighted with a red box)

**Terminal Window (MINGW64):**

```
David@DESKTOP-D2JLVJ5 MINGW64 ~/Desktop
$ git clone https://github.com/DavidBernalGonzalez/miPrimerRepoEnGitHub.git
Cloning into 'miPrimerRepoEnGitHub' ...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

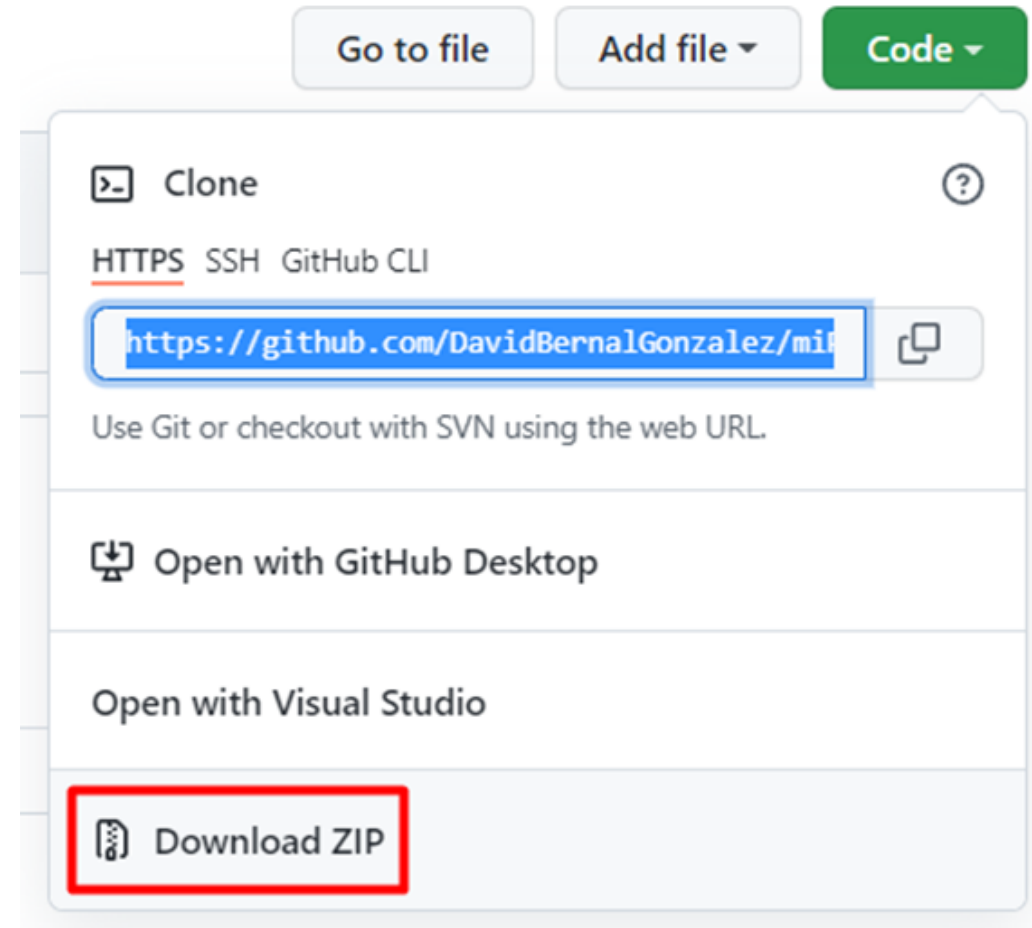
# Fundamentos de Git

## Creando un repositorio de GIT (local repository)



# Fundamentos de Git

## Creando un repositorio de GIT (local repository)



# Fundamentos de Git

## Creando un repositorio de GIT (local repository)

