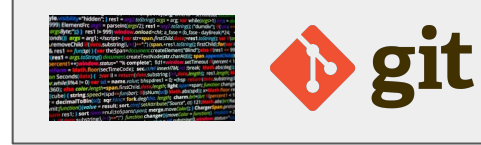


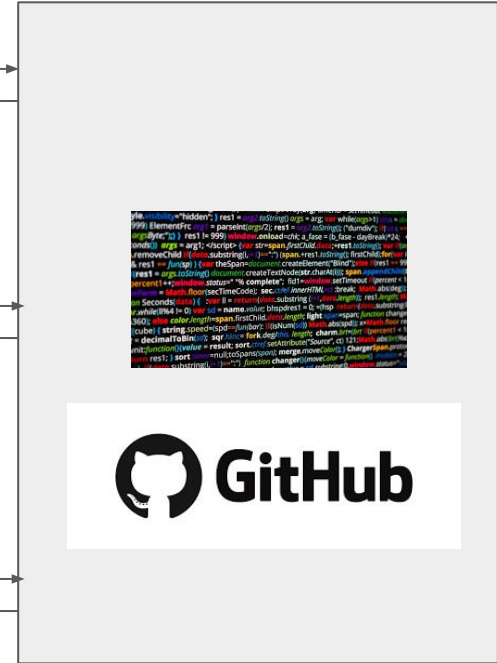
# GIT

By: Guillermo Andres De Mendoza Corrales





CLOUD

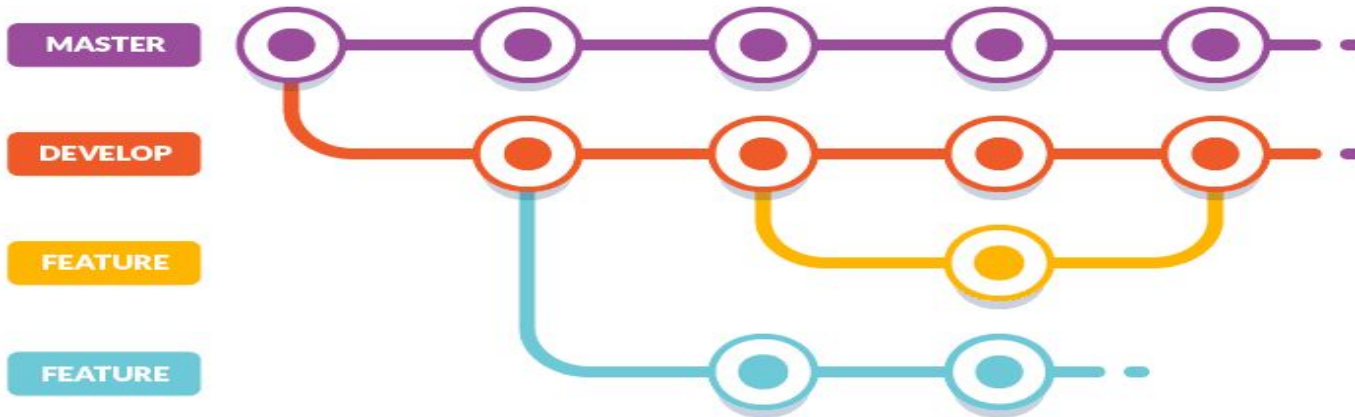


# GIT

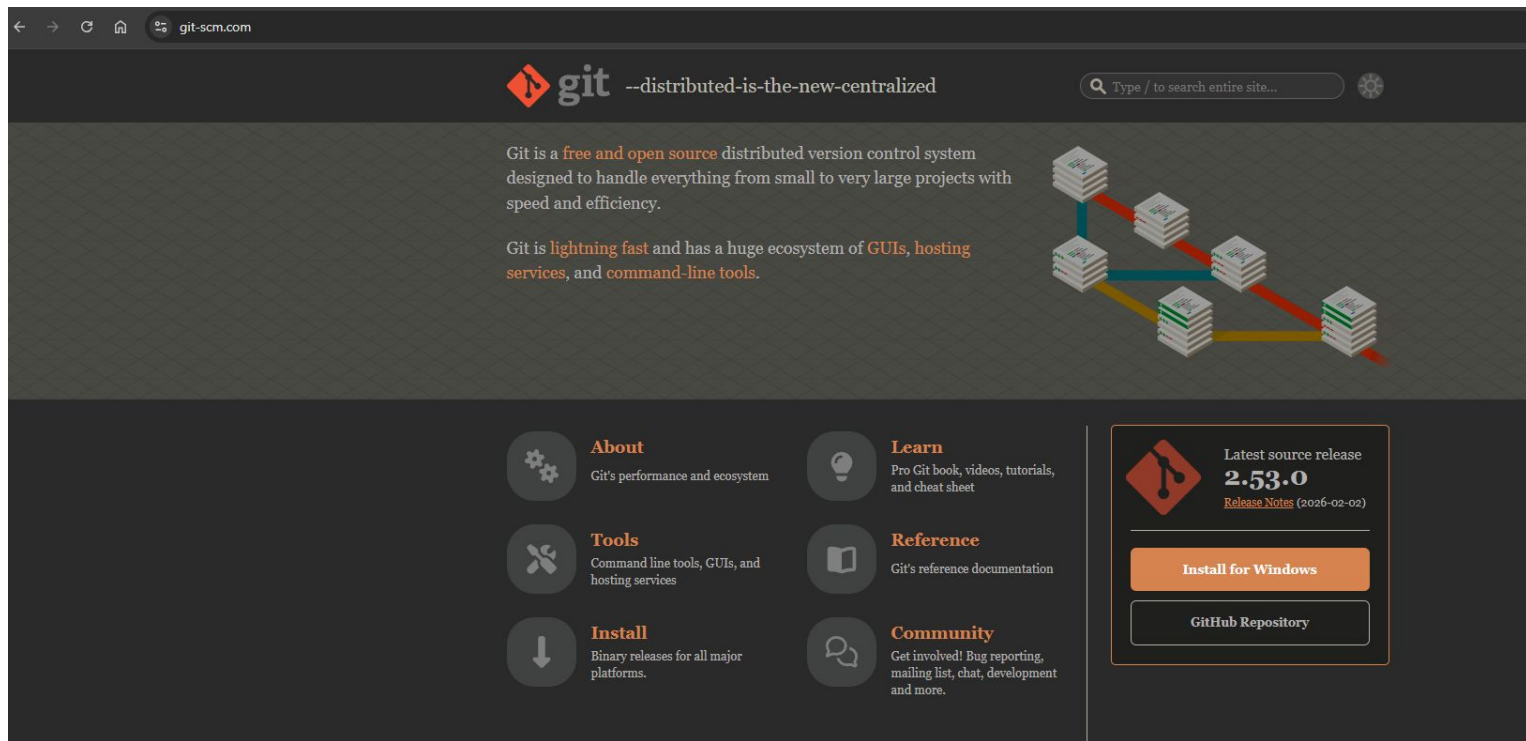
# Theory



**Git** is a free and open source **distributed version control system** designed to handle everything from small to very large projects with speed and efficiency.



# Install



The screenshot shows the Git website (git-scm.com) with a dark theme. The header features the Git logo and the tagline "--distributed-is-the-new-centralized". A search bar is located in the top right corner. The main content area describes Git as a "free and open source" distributed version control system and mentions its speed and efficiency. It also highlights that Git is "lightning fast" and has a huge ecosystem of GUIs, hosting services, and command-line tools. To the right of the text is a 3D illustration of a branching model with stacks of papers representing commits and colored lines representing branches. Below the main text, there are six circular icons with corresponding text: "About" (performance and ecosystem), "Learn" (Pro Git book, videos, tutorials, and cheat sheet), "Tools" (command line tools, GUIs, and hosting services), "Reference" (Git's reference documentation), "Install" (binary releases for all major platforms), and "Community" (get involved! Bug reporting, mailing list, chat, development and more). On the right side, there is a section for the "Latest source release 2.53.0" with a "Release Notes" link and two buttons: "Install for Windows" and "GitHub Repository".

git --distributed-is-the-new-centralized

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **lightning fast** and has a huge ecosystem of **GUIs**, **hosting services**, and **command-line tools**.

**About**  
Git's performance and ecosystem

**Learn**  
Pro Git book, videos, tutorials, and cheat sheet

**Tools**  
Command line tools, GUIs, and hosting services

**Reference**  
Git's reference documentation

**Install**  
Binary releases for all major platforms

**Community**  
Get involved! Bug reporting, mailing list, chat, development and more.

Latest source release  
**2.53.0**  
[Release Notes](#) (2026-02-02)

**Install for Windows**

**GitHub Repository**



# Install

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.26100.7623]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\memo>git
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
        [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
        [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--no-lazy-fetch]
        [--no-optional-locks] [--no-advice] [--bare] [--git-dir=<path>]
        [--work-tree=<path>] [--namespace=<name>] [--config-env=<name>=<envvar>]
        <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  restore    Restore working tree files
  rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect    Use binary search to find the commit that introduced a bug
  diff      Show changes between commits, commit and working tree, etc
  grep      Print lines matching a pattern
  log       Show commit logs
  show      Show various types of objects
  status    Show the working tree status

grow, mark and tweak your common history
  backfill  Download missing objects in a partial clone
  branch    List, create, or delete branches
  commit    Record changes to the repository
  merge     Join two or more development histories together
  rebase    Reapply commits on top of another base tip
  reset     Reset current HEAD to the specified state
  switch    Switch branches
  tag       Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
  fetch     Download objects and refs from another repository
  pull      Fetch from and integrate with another repository or a local branch
```



GITHUB

# GITHUB

Is a cloud-based platform that helps developers store, manage, and share their code using Git version control. It enables collaborative, real-time editing of projects, making it essential for tracking changes, reviewing code, and managing software development tasks. Owned by Microsoft, it is widely used for both open-source and professional, team-based software projects.





# GITHUB

1

Create a new account

2

Authenticate my PC with github

3

Create a new repository

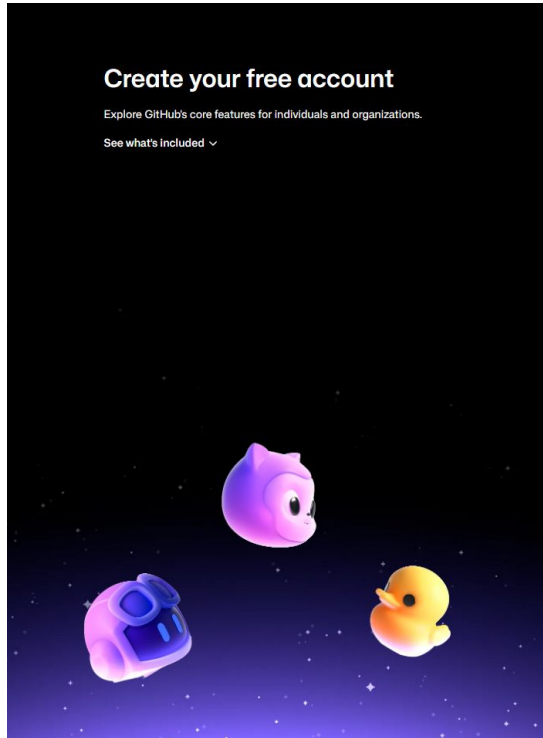
---

4

Send/Receive files





# 1. Create a new account



Already have an account? [Sign in](#)

### Sign up for GitHub

 Continue with Google

 Continue with Apple

or

Email\*

Password\*

Password should be at least 15 characters OR at least 8 characters including a number and a lowercase letter.

Username\*

Username may only contain alphanumeric characters or single hyphens, and cannot begin or end with a hyphen.

Your Country/Region\*

Colombia ▼

For compliance reasons, we're required to collect country information to send you occasional updates and announcements.

Email preferences

☐ Receive occasional product updates and announcements

Create account >

By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll occasionally send you account-related emails.



### 3. Authenticate my PC with github



```
C:\Users\memoo\OneDrive\Desktop\git\test>git config --global user.email "github email"  
C:\Users\memoo\OneDrive\Desktop\git\test>git config --global user.name alias
```



## 2. Access token

The screenshot shows the GitHub profile page for user 'memoodm'. The profile includes a custom avatar of a cat wearing sunglasses, the name 'memoodm', and the bio 'memoodm · he/him'. The user has 23 followers and 0 following. The 'Overview' tab is selected, showing a list of popular repositories: 'AI-8Puzzle-SearchAlgorithm', 'python-libs', 'python-multithreading', 'simulation-prey-predator', 'usa-computacion-cientifica-1', and 'AI-Game-2048-MinMax'. A dropdown menu is open on the right side of the profile, listing various settings and options. The 'Settings' option is highlighted with a red box.

memoodm  
memoodm · he/him  
23 followers · 0 following  
Colombia  
memoodm@gmail.com

Popular repositories

- [AI-8Puzzle-SearchAlgorithm](#) (Public)  
Solving 8 puzzle game with algorithms: BFS, DFS, A\*  
Python 20 stars 15 forks
- [python-libs](#) (Public)  
Jupyter Notebook 1 star
- [python-multithreading](#) (Public)  
Jupyter Notebook 1 star
- [simulation-prey-predator](#)  
python prey predator simulation  
Jupyter Notebook 1 star
- [usa-computacion-cientifica-1](#)  
Jupyter Notebook 1 star 3 forks
- [AI-Game-2048-MinMax](#)  
Python

64 contributions in the last year

Contribution settings

	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb
Mon													

memoodm  
memoodm

- Set status
- Profile
- Repositories
- Stars
- Gists
- Organizations
- Enterprises
- Sponsors
- Settings**
- Copilot settings
- Feature preview
- Appearance
- Accessibility
- Try Enterprise (Free)
- Sign out



## 2. Access token

The screenshot shows the GitHub Settings interface for a user named 'memoodm'. The left sidebar contains a list of settings categories. The 'Developer settings' option at the bottom is highlighted with a red rectangle. The main content area displays the 'Public profile' settings, including fields for Name, Public email, Bio, Pronouns, URL, Social accounts, Company, Location, and ORCID ID. A 'Profile picture' section on the right shows a cat avatar with sunglasses. At the bottom of the page, there is a disclaimer about optional fields and a link to the privacy statement.

Settings

memoodm (memoodm)  
Your personal account

Go to your personal profile

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and licensing

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Enterprises

Moderation

Code, planning, and automation

Repositories

Codespaces

Models

Packages

Copilot

Pages

Saved replies

Security

Code security

Integrations

Applications

Scheduled reminders

Archives

Security log

Sponsorship log

Developer settings

Public profile

Name

memoodm

Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.

Public email

memoodm@gmail.com

X Remove

You can manage verified email addresses in your [email settings](#).

Bio

Tell us a little bit about yourself

You can @mention other users and organizations to link to them.

Pronouns

he/him

URL

https://www.linkedin.com/in/guillermo-andres-de-mendoza-corral

Social accounts

Link to social profile 1

Link to social profile 2

Link to social profile 3

Link to social profile 4

Company

You can @mention your company's GitHub organization to link it.

Location

Colombia

Display current local time

Other users will see the time difference from their local time.

ORCID ID

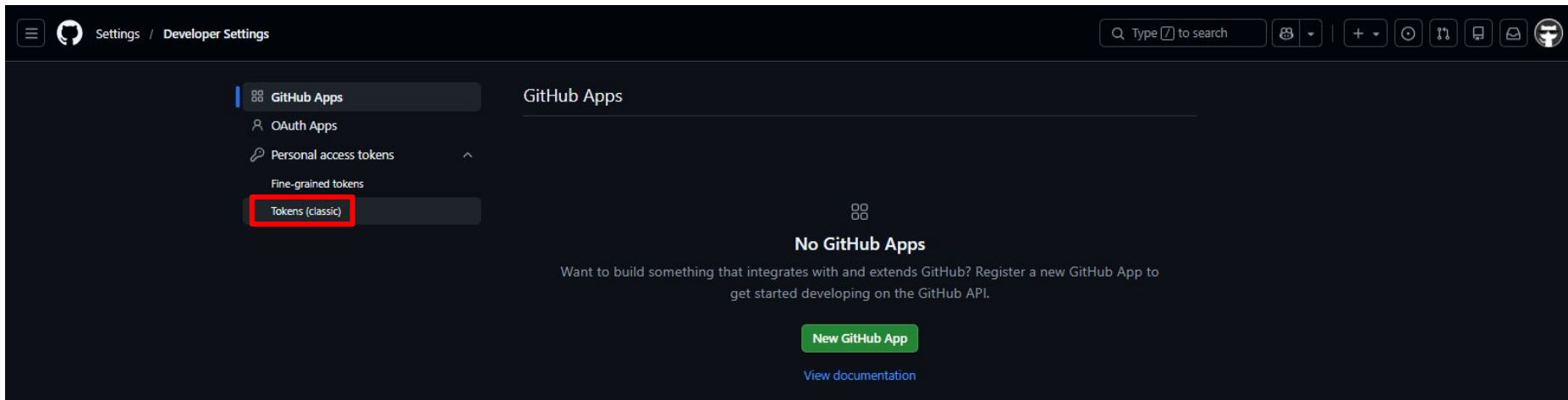
ORCID provides a persistent Identifier - an ORCID ID - that distinguishes you from other researchers. Learn more at [ORCID.org](#).

Connect your ORCID ID

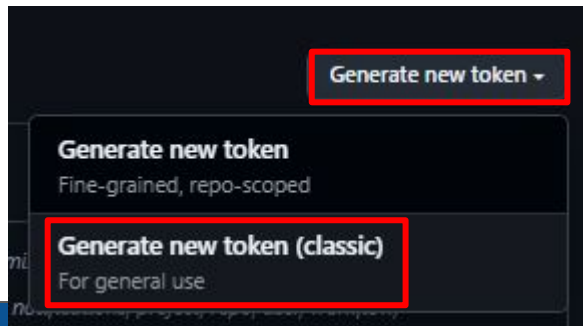
All of the fields on this page are optional and can be deleted at any time, and by filling them out, you're giving us consent to share this data wherever your user profile appears. Please see our [privacy statement](#) to learn more about how we use this information.



## 2. Access token



The screenshot shows the GitHub Developer Settings page. On the left sidebar, the navigation menu includes 'GitHub Apps', 'OAuth Apps', 'Personal access tokens', 'Fine-grained tokens', and 'Tokens (classic)'. The 'Tokens (classic)' option is highlighted with a red rectangle. The main content area is titled 'GitHub Apps' and displays a message: 'No GitHub Apps'. Below this message is a green button labeled 'New GitHub App' and a link 'View documentation'. The top of the page features a search bar and various utility icons.



This image is a close-up of the 'Generate new token' dropdown menu. The dropdown is open, showing two options: 'Generate new token' (Fine-grained, repo-scoped) and 'Generate new token (classic)' (For general use). The 'Generate new token (classic)' option is highlighted with a red rectangle. The 'Generate new token' button itself is also highlighted with a red rectangle.



## 2. Access token

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens

Tokens (classic)

### New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

What's this token for?

Expiration

No expiration

⚠ GitHub strongly recommends that you set an expiration date for your token to help keep your information secure. [Learn more](#)

Select scopes

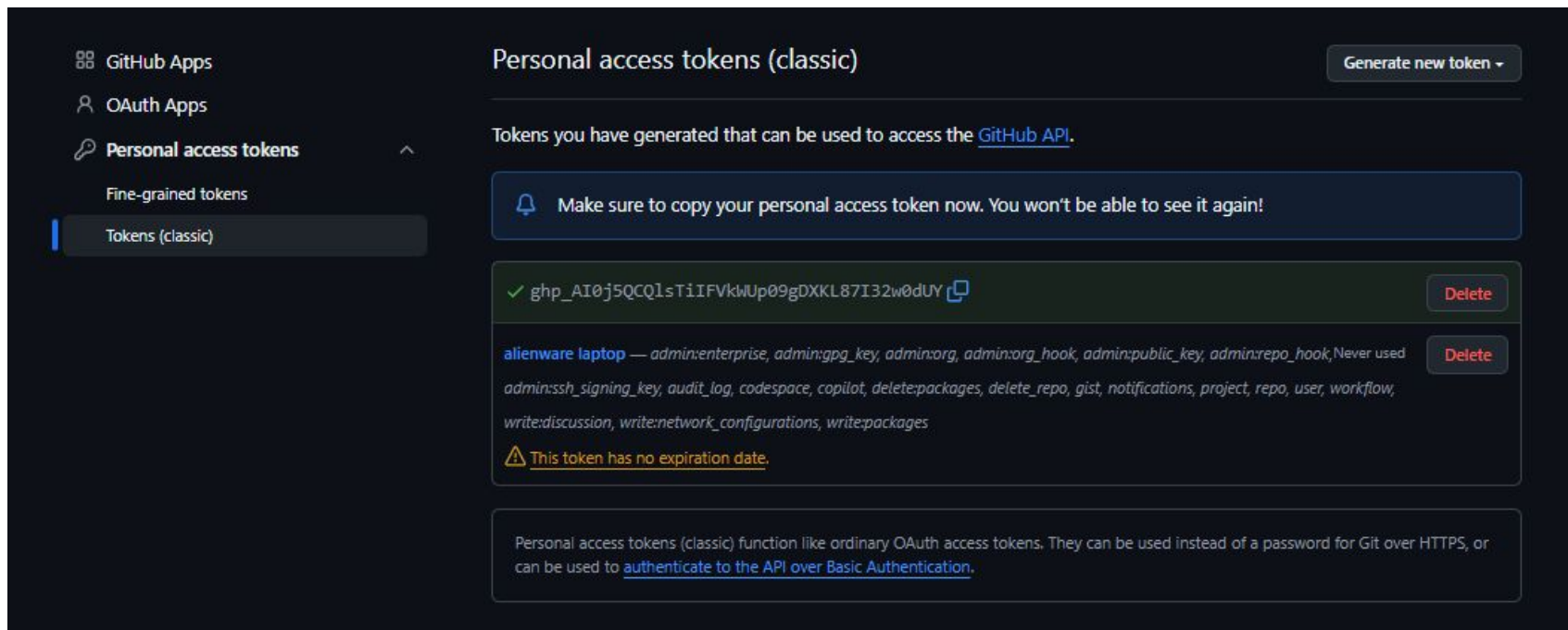
Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repostatus	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repoinvite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> workflow	Update GitHub Action workflows
<input checked="" type="checkbox"/> write_packages	Upload packages to GitHub Package Registry
<input checked="" type="checkbox"/> read_packages	Download packages from GitHub Package Registry
<input checked="" type="checkbox"/> delete_packages	Delete packages from GitHub Package Registry
<input checked="" type="checkbox"/> admin_org	Full control of orgs and teams, read and write org projects
<input checked="" type="checkbox"/> write_org	Read and write org and team membership, read and write org projects
<input checked="" type="checkbox"/> read_org	Read org and team membership, read org projects
<input checked="" type="checkbox"/> manage_runners_org	Manage org runners and runner groups
<input checked="" type="checkbox"/> admin_public_key	Full control of user public keys
<input checked="" type="checkbox"/> write_public_key	Write user public keys
<input checked="" type="checkbox"/> read_public_key	Read user public keys
<input checked="" type="checkbox"/> admin_repo_hook	Full control of repository hooks
<input checked="" type="checkbox"/> write_repo_hook	Write repository hooks
<input checked="" type="checkbox"/> read_repo_hook	Read repository hooks
<input checked="" type="checkbox"/> admin_org_hook	Full control of organization hooks
<input checked="" type="checkbox"/> gist	Create gists
<input checked="" type="checkbox"/> notifications	Access notifications
<input checked="" type="checkbox"/> user	Update ALL user data
<input checked="" type="checkbox"/> read_user	Read ALL user profile data
<input checked="" type="checkbox"/> user_email	Access user email addresses (read-only)
<input checked="" type="checkbox"/> user_follow	Follow and unfollow users
<input checked="" type="checkbox"/> delete_repo	Delete repositories
<input checked="" type="checkbox"/> write_discussion	Read and write team discussions
<input checked="" type="checkbox"/> read_discussion	Read team discussions

token name



## 2. Access token



The screenshot shows the GitHub interface for managing personal access tokens. On the left sidebar, the navigation menu includes 'GitHub Apps', 'OAuth Apps', 'Personal access tokens' (which is expanded to show 'Fine-grained tokens' and 'Tokens (classic)' with the latter selected), and 'Tokens (classic)'. The main content area is titled 'Personal access tokens (classic)' and features a 'Generate new token' button. Below the title, a message states: 'Tokens you have generated that can be used to access the [GitHub API](#).' A warning box with a bell icon says: 'Make sure to copy your personal access token now. You won't be able to see it again!'. A list of generated tokens is shown, with the first token highlighted: 'ghp\_AI0j5QCQ1sTiIFVklWUp09gDXKL87I32w0dUY' with a copy icon and a 'Delete' button. Below the token, the permissions are listed: 'alienware laptop' — admin:enterprise, admin:gpg\_key, admin:org, admin:org\_hook, admin:public\_key, admin:repo\_hook, Never used, admin:ssh\_signing\_key, audit\_log, codespace, copilot, delete:packages, delete\_repo, gist, notifications, project, repo, user, workflow, write:discussion, write:network\_configurations, write:packages. A warning icon and text state: 'This token has no expiration date.' At the bottom, a note explains: 'Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).'

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens


Tokens (classic)

### Personal access tokens (classic)

Generate new token ▾

Tokens you have generated that can be used to access the [GitHub API](#).

⚠ Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp\_AI0j5QCQ1sTiIFVklWUp09gDXKL87I32w0dUY  Delete

**alienware laptop** — admin:enterprise, admin:gpg\_key, admin:org, admin:org\_hook, admin:public\_key, admin:repo\_hook, Never used, admin:ssh\_signing\_key, audit\_log, codespace, copilot, delete:packages, delete\_repo, gist, notifications, project, repo, user, workflow, write:discussion, write:network\_configurations, write:packages

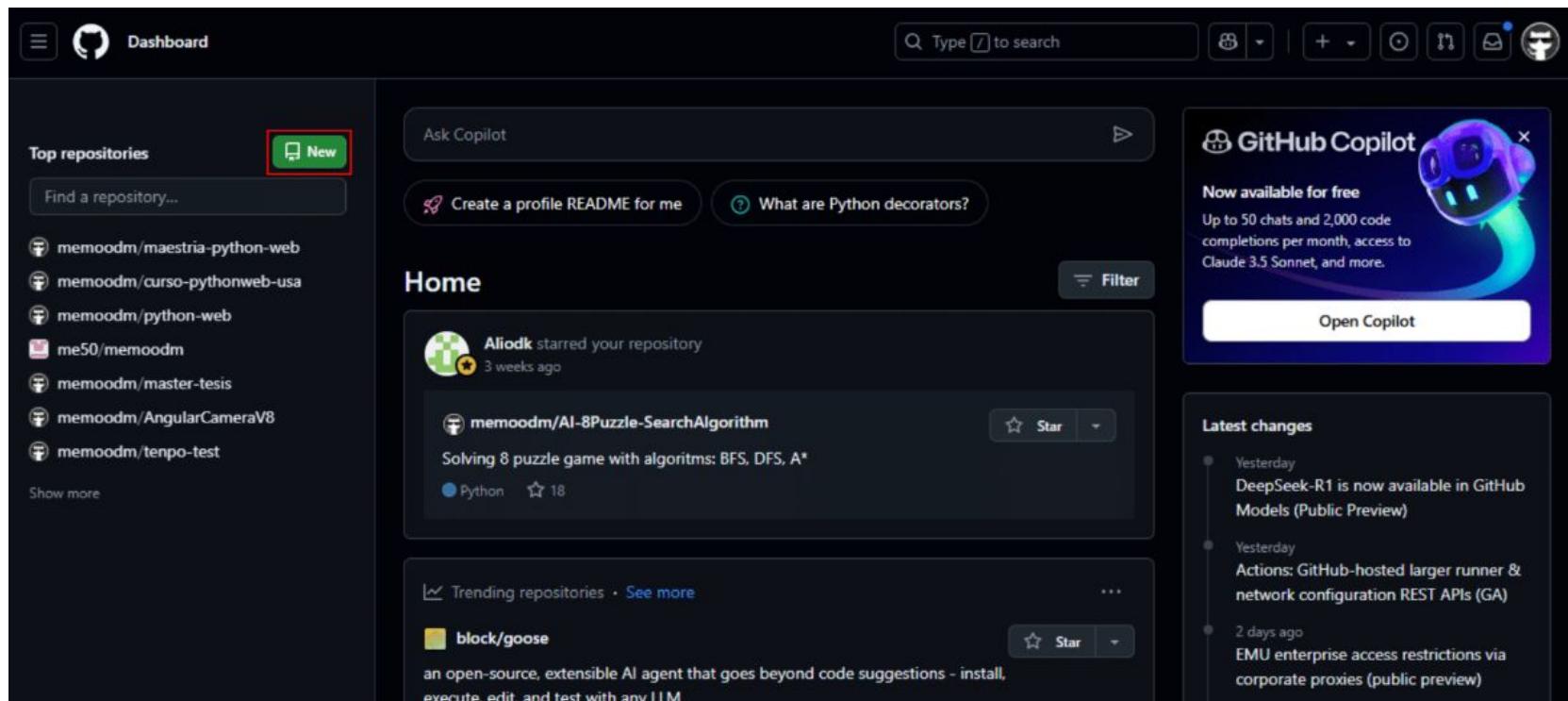
⚠ This token has no expiration date.

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).





### 3. Create a new repository



The screenshot shows the GitHub Dashboard interface. In the top-left sidebar, under the 'Top repositories' section, a green 'New' button with a repository icon is highlighted with a red box. Below this is a search bar labeled 'Find a repository...'. A list of repositories follows, including 'memoodm/maestria-python-web', 'memoodm/curso-pythonweb-usa', 'memoodm/python-web', 'me50/memoodm', 'memoodm/master-tesis', 'memoodm/AngularCameraV8', and 'memoodm/tenpo-test'. A 'Show more' link is at the bottom of the list.

The main content area is titled 'Home' and features a 'Filter' button. It displays a notification: 'Aliodk starred your repository 3 weeks ago' for the repository 'memoodm/AI-8Puzzle-SearchAlgorithm', which is described as 'Solving 8 puzzle game with algorithms: BFS, DFS, A\*' and has 18 stars. Below this, there is a section for 'Trending repositories' featuring 'block/goose', an open-source AI agent.

On the right side, there is a 'GitHub Copilot' advertisement with the text 'Now available for free' and 'Up to 50 chats and 2,000 code completions per month, access to Claude 3.5 Sonnet, and more.' with an 'Open Copilot' button. Below that is a 'Latest changes' section listing updates like 'DeepSeek-R1 is now available in GitHub Models (Public Preview)' and 'Actions: GitHub-hosted larger runner & network configuration REST APIs (GA)'.




### 3. Create a new repository

**Create a new repository**

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).  
Required fields are marked with an asterisk (\*).

**1 General**

Owner \* Repository name \*

 memoodm /

Great repository names are short and memorable. How about [turbo-waffle](#)?

**Description**

0 / 350 characters

**2 Configuration**

**Choose visibility \***

Choose who can see and commit to this repository

**Add README** ☒ On

READMEs can be used as longer descriptions. [About READMEs](#)

**Add .gitignore**

.gitignore tells git which files not to track. [About ignoring files](#)

**Add license**

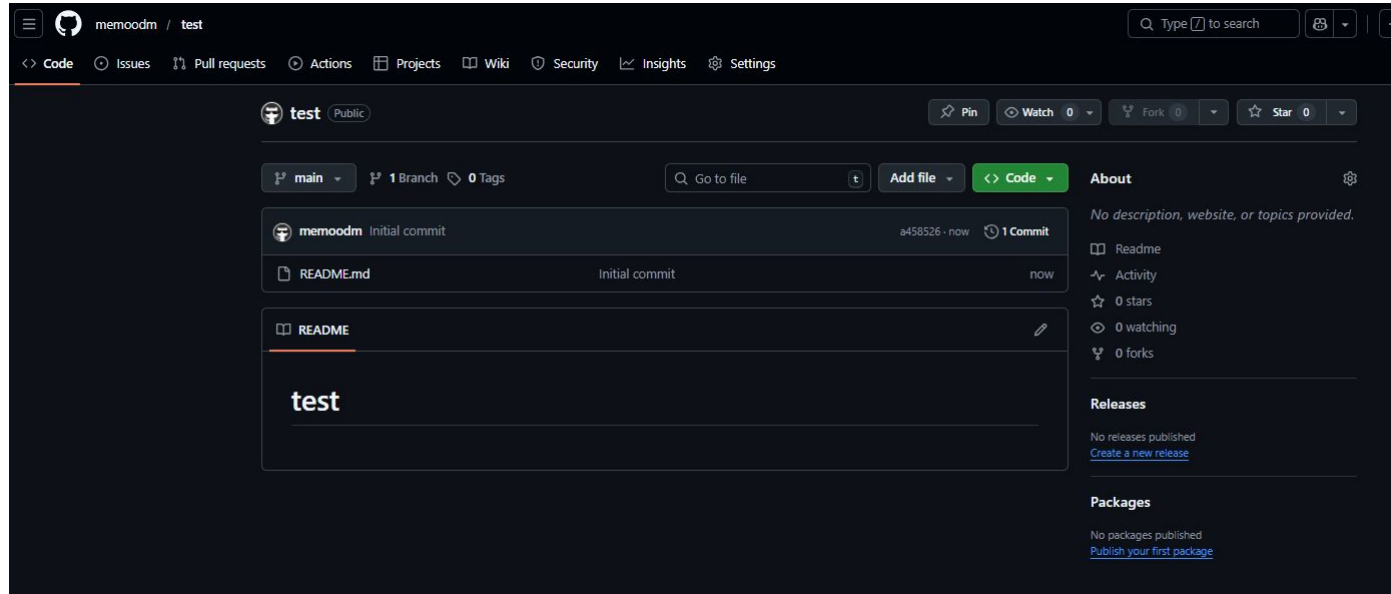
Licenses explain how others can use your code. [About licenses](#)

unique name

public access

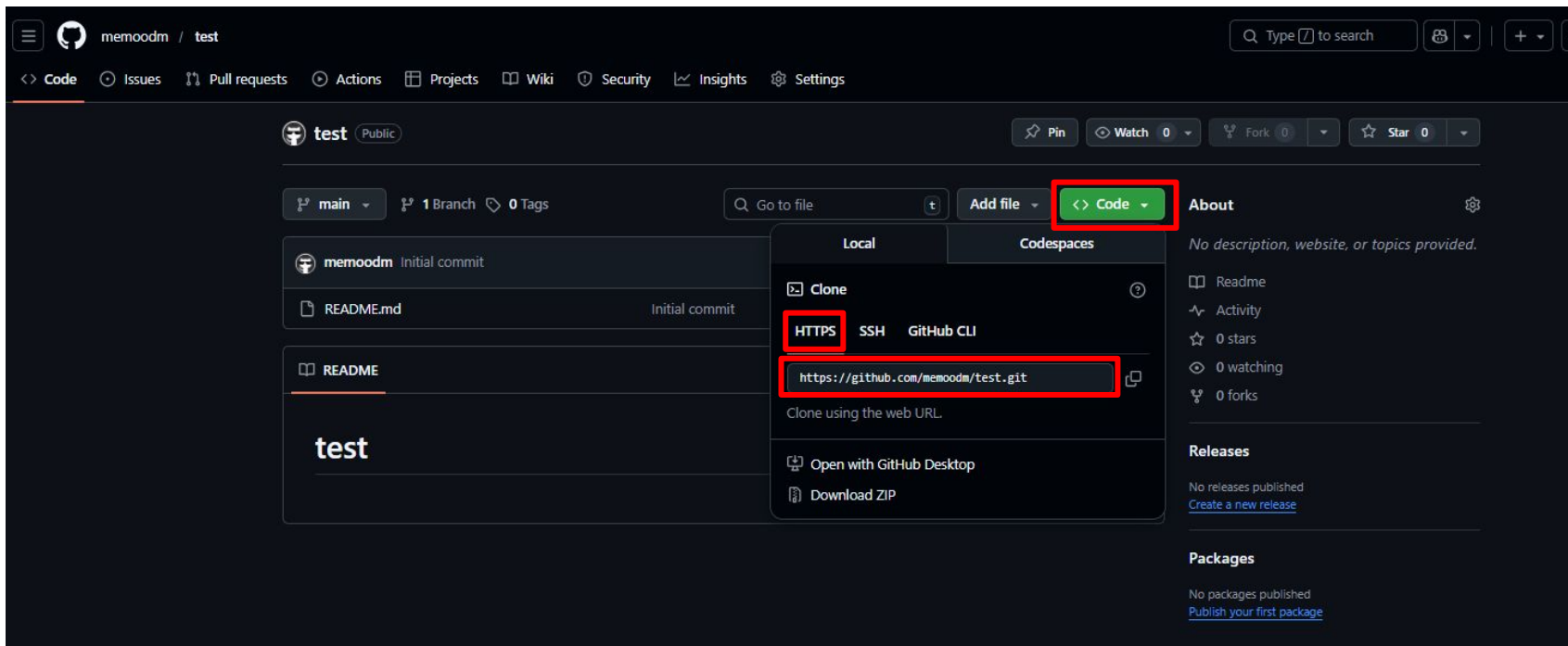


### 3. Create a new repository



# GIT COMMANDS

# 4. CLONE



The screenshot shows the GitHub interface for a repository named 'test' by user 'memoodm'. The repository is public and has 1 branch and 0 tags. The 'Code' dropdown menu is open, showing options for cloning the repository. The 'HTTPS' option is selected, and the URL 'https://github.com/memoodm/test.git' is highlighted. The 'About' tab is also visible, showing no description, website, or topics provided. The 'Releases' and 'Packages' sections are also visible, both showing no published items.

memoodm / test

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

test Public

main 1 Branch 0 Tags

Go to file Add file <> Code

Local Codespaces

Clone

HTTPS SSH GitHub CLI

https://github.com/memoodm/test.git

Clone using the web URL

Open with GitHub Desktop

Download ZIP

About

No description, website, or topics provided.

Readme

Activity

0 stars

0 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

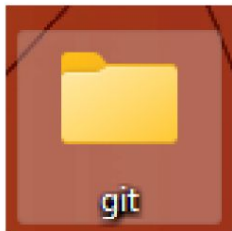
No packages published

[Publish your first package](#)



## 4. CLONE

Creo una carpeta en mi pc



Comando en CMD:

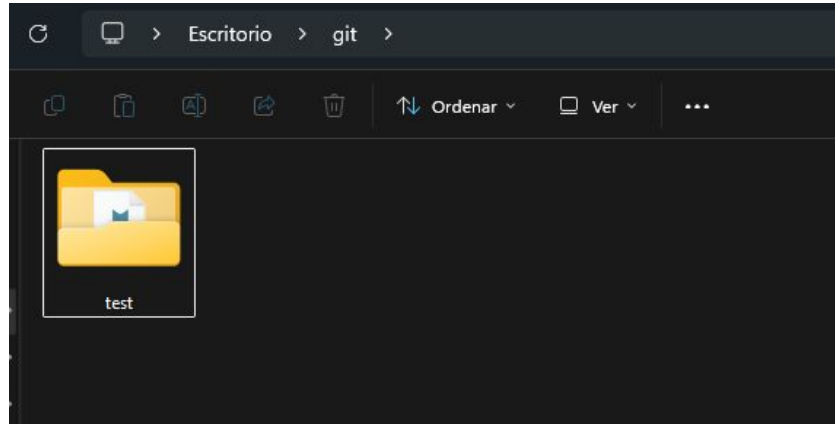
`git clone URL_REPO`


```
C:\Windows\System32\cmd.e  X  +  v
Microsoft Windows [Versión 10.0.26100.7623]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\memoo\Desktop\git>git clone https://github.com/memoodm/test.git|
```



## 4. CLONE

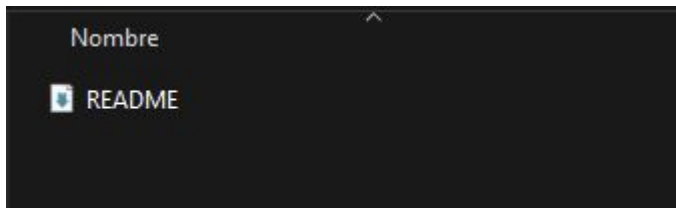


Nombre	Fecha de modificación	Tipo	Tamaño
 README	9/02/2026 9:59 a. m.	Markdown Source File	1 KB

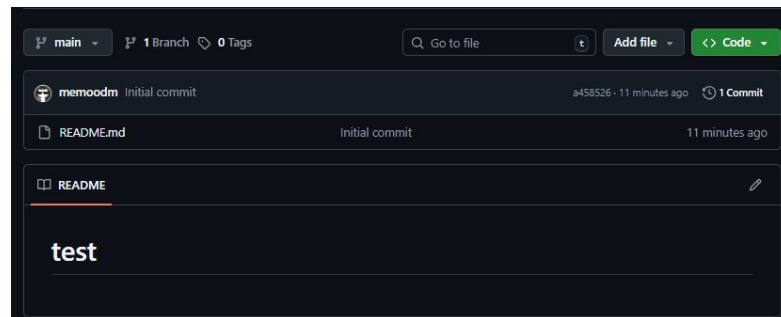


## 4. Compare

my pc



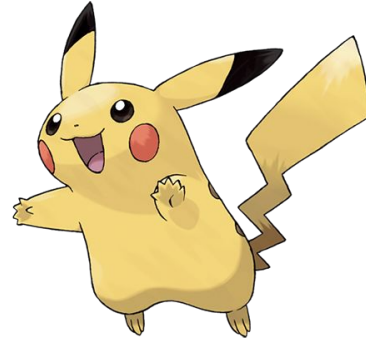
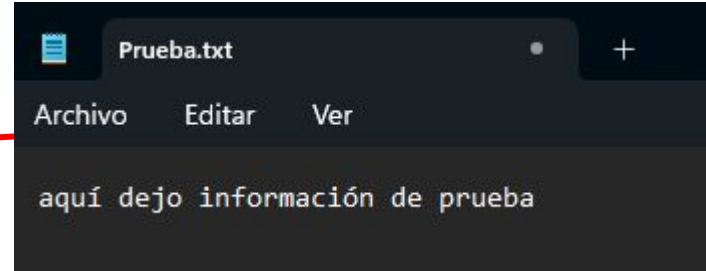
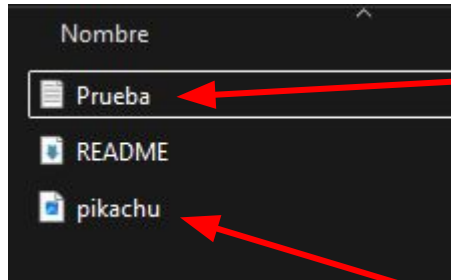
github





## 4. Add content to pc repository

my pc



## 4. Add content to pc repository

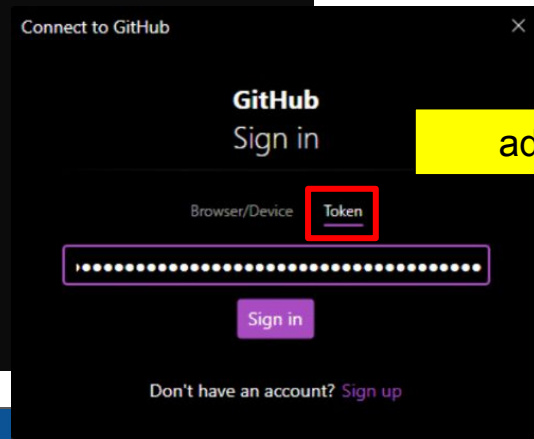
```
C:\Windows\System32\cmd.e X + v

Microsoft Windows [Versión 10.0.26100.7623]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\memoo\Desktop\git\test>git add .

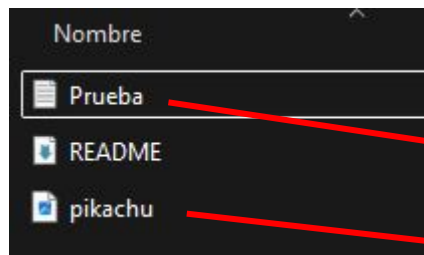
C:\Users\memoo\Desktop\git\test>git commit -m "subo un archivo y una imagen como prueba"
[main 6986b63] subo un archivo y una imagen como prueba
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Prueba.txt
create mode 100644 pikachu.png

C:\Users\memoo\Desktop\git\test>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 114.86 KiB | 38.29 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/memoodm/test.git
a458526..6986b63 main -> main
```

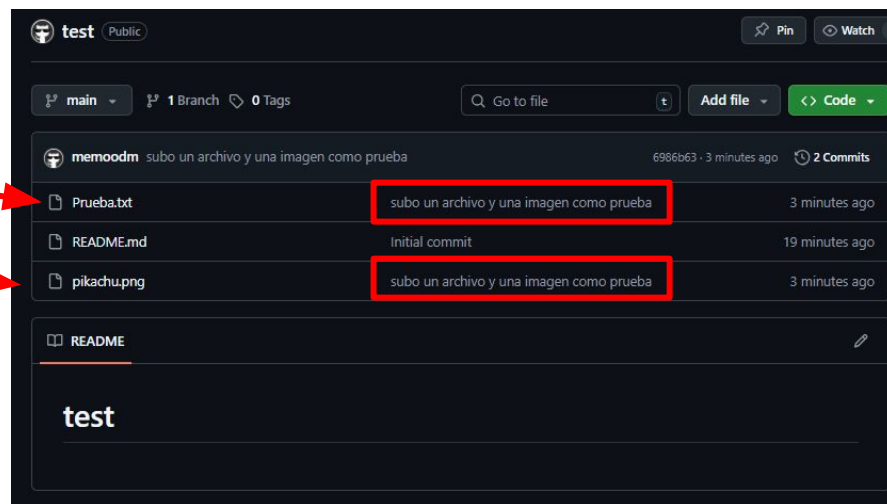


## 4. Compare

my pc












github



## 4. Compare

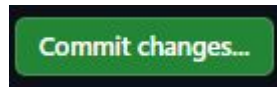
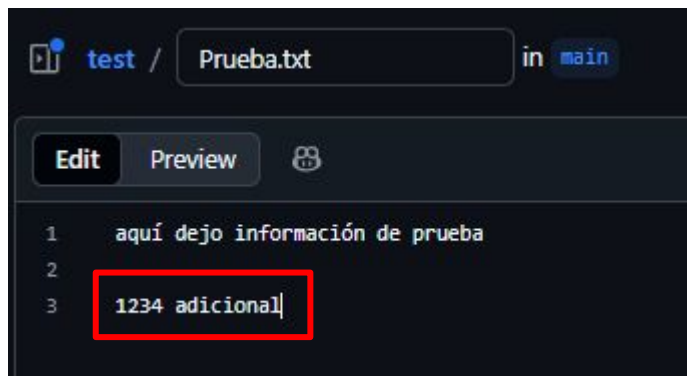
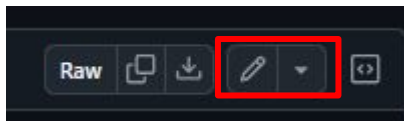
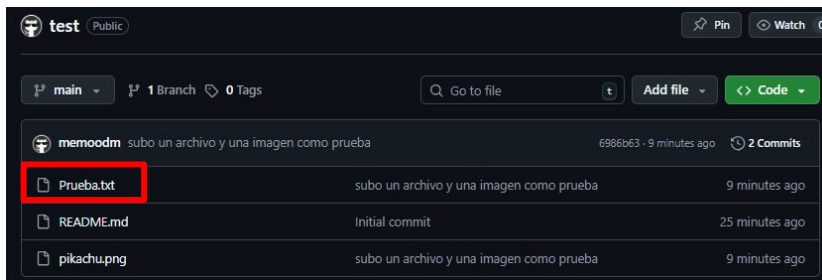
 <b>memoodm</b>	subo un archivo y una imagen como prueba	6986b63 · 5 minutes ago	 <b>2 Commits</b>
 <b>Prueba.txt</b>	subo un archivo y una imagen como prueba	5 minutes ago	
 <b>README.md</b>	Initial commit	21 minutes ago	
 <b>pikachu.png</b>	subo un archivo y una imagen como prueba	5 minutes ago	

### Commits

 <b>main</b> ▾	 <b>All users</b> ▾	 <b>All time</b> ▾
Commits on Feb 9, 2026		
<b>subo un archivo y una imagen como prueba</b>		6986b63  
 memoodm committed 5 minutes ago		
<b>Initial commit</b>		<b>Verified</b> a458526  
 memoodm authored 21 minutes ago		



## 4. Changes in git



## 4. Changes in git

Commit changes

Commit message

Update Prueba.txt

Extended description

Add an optional extended description...

☒ Commit directly to the main branch

☐ Create a new branch for this commit and start a pull request [Learn more about pull requests](#)

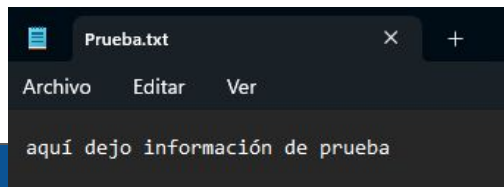
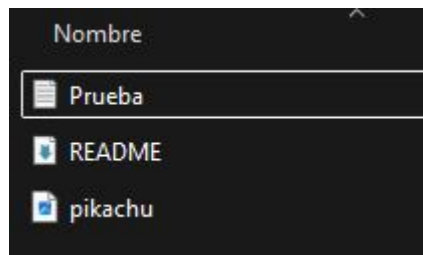
Cancel

Commit changes

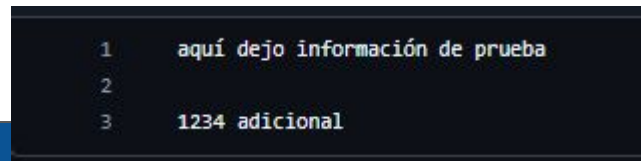
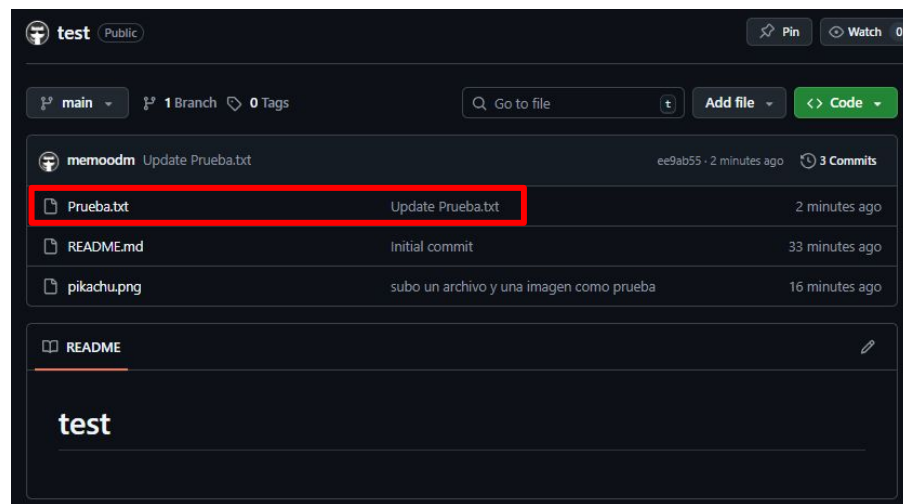


## 4. Compare

my pc



github



## 4. Pull

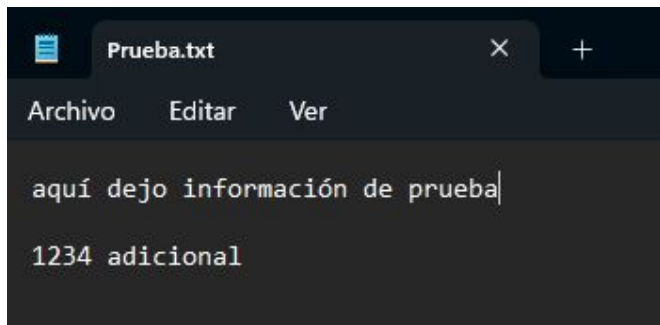
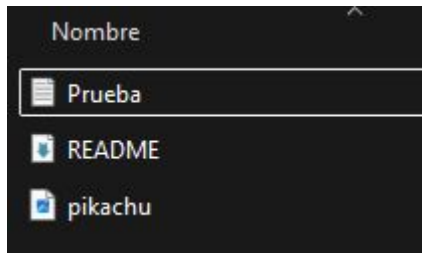
```
C:\Windows\System32\cmd.e X + v
C:\Users\memoo\Desktop\git\test>git pull
Updating 6986b63..ee9ab55
Fast-forward
 Prueba.txt | 3 +++
1 file changed, 3 insertions(+)
```



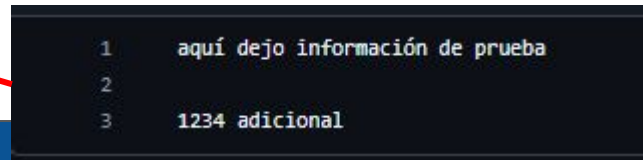
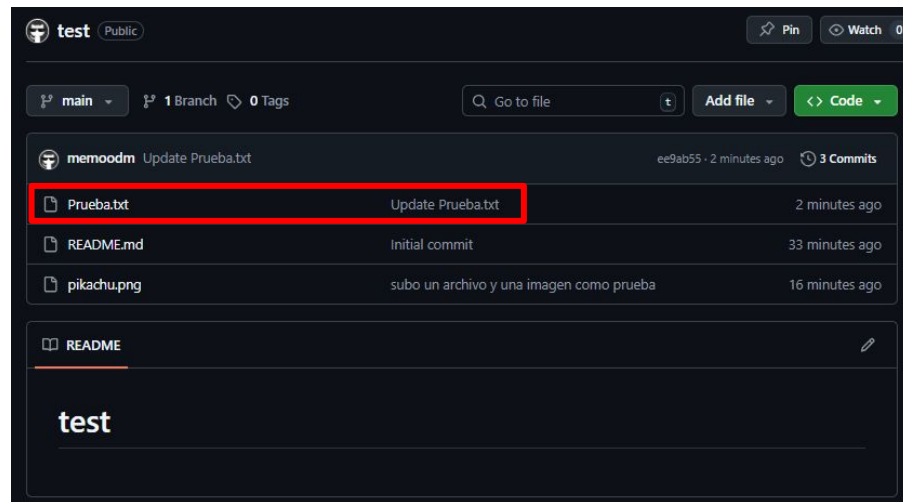


## 4. Compare

my pc



github



## 4. Resume

\*only in master branch

