


Tarea 1: Mi primer repositorio

En primer lugar, vamos a crear un repositorio vacío en nuestro Github.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk ().*

Owner *	Repository name *
 Danimartinez-1997 ▾	/ mi-primer-repo
	✓ mi-primer-repo is available.

Clonamos este nuevo repositorio en el terminal mediante la clave SSH, y accedemos a el para poder empezar a trabajar.

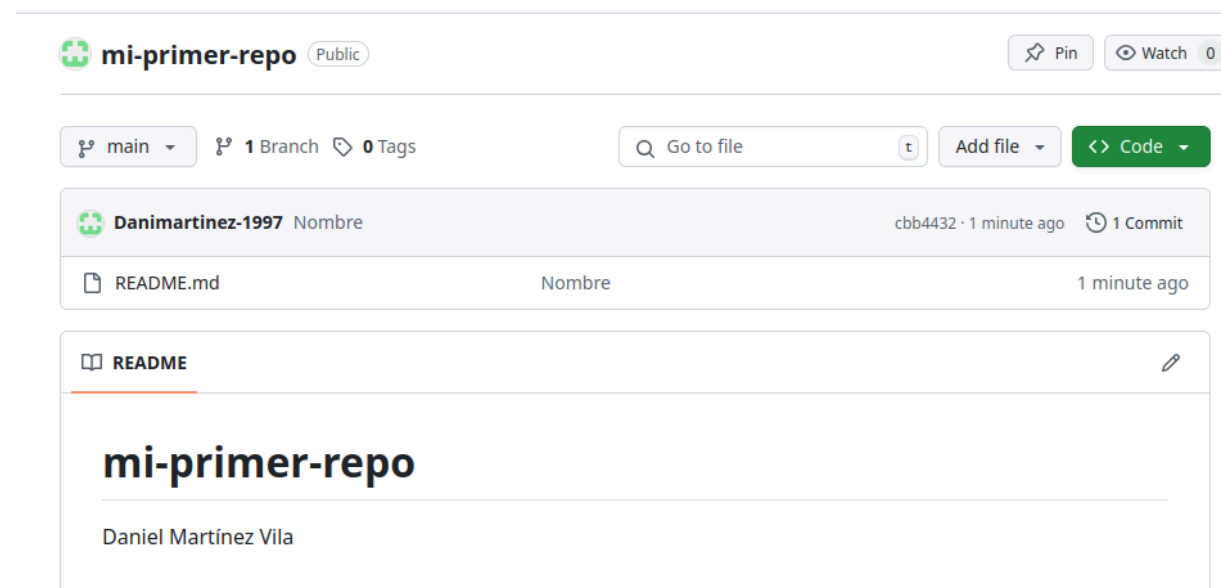
```
root@daniel-MS-7E28:/home# git clone git@github.com:Danimartinez-1997/mi-primer-repo.git
Clonando en 'mi-primer-repo'...
Enter passphrase for key '/root/.ssh/id_ed25519':
warning: Pareces haber clonado un repositorio sin contenido.
root@daniel-MS-7E28:/home# ls
daniel mi-primer-repo
root@daniel-MS-7E28:/home# cd mi-primer-repo/
root@daniel-MS-7E28:/home/mi-primer-repo#
```

Ahora vamos a crear el archivo README con nuestro nombre, FP cursada, y una breve descripción sobre nosotros (haciendo un commit por cada una de ellas. EN primer lugar añadimos nuestro nombre y commiteamos el cambio

```
root@daniel-MS-7E28:/home/mi-primer-repo# echo "# mi-primer-repo" >> README.md
root@daniel-MS-7E28:/home/mi-primer-repo# echo "Daniel Martínez Vila" >> README.md
root@daniel-MS-7E28:/home/mi-primer-repo# git init
Reinicializado el repositorio Git existente en /home/mi-primer-repo/.git/
root@daniel-MS-7E28:/home/mi-primer-repo# git add README.md
root@daniel-MS-7E28:/home/mi-primer-repo# git commit -m "Nombre"
[main (commit-raíz) cbb4432] Nombre
1 file changed, 2 insertions(+)
create mode 100644 README.md
root@daniel-MS-7E28:/home/mi-primer-repo# git branch -M main
```

Ahora hacemos un push y comprobamos que se ha añadido el archivo [readme.MD](#) correctamente.

```
root@daniel-MS-7E28:/home/mi-primer-repo# git push -u origin main
Enter passphrase for key '/root/.ssh/id_ed25519':
Enumerando objetos: 3, listo.
Contando objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 260 bytes | 260.00 KiB/s, listo.
Total 3 (delta 0), reusados 0 (delta 0), pack-reusados 0
To github.com:Danimartinez-1997/mi-primer-repo.git
 * [new branch]      main -> main
rama 'main' configurada para rastrear 'origin/main'.
root@daniel-MS-7E28:/home/mi-primer-repo#
```



The screenshot shows the GitHub interface for a repository named 'mi-primer-repo', which is public. The repository is owned by 'Danimartinez-1997'. It has 1 branch (main) and 0 tags. The 'README.md' file is listed as the only file in the repository, committed 1 minute ago. The README content is visible, showing the repository name 'mi-primer-repo' and the owner's name 'Daniel Martínez Vila'.

Vamos con los otros dos echos, uno con el ciclo formativo y otro con la descripción personal.

```
root@daniel-MS-7E28:/home/mi-primer-repo# echo "Ciclo formativo: ASIR" >> README.md
root@daniel-MS-7E28:/home/mi-primer-repo# git add README.md
root@daniel-MS-7E28:/home/mi-primer-repo# git commit -m "Segundo commit: Ciclo formativo"
[main 32e70ca] Segundo commit: Ciclo formativo
1 file changed, 1 insertion(+)
root@daniel-MS-7E28:/home/mi-primer-repo#
```

```
root@daniel-MS-7E28:/home/mi-primer-repo# echo "Soy Daniel, vivo en un pueblo de A Coruña y me gustan las artes marciales, leer libros y pasear" >> README.md
root@daniel-MS-7E28:/home/mi-primer-repo# git add README.md
root@daniel-MS-7E28:/home/mi-primer-repo# git commit -m "Tercer commit: Mi presentacion"
[main 6a187a5] Tercer commit: Mi presentacion
1 file changed, 1 insertion(+)
root@daniel-MS-7E28:/home/mi-primer-repo#
```

Y por último, hacemos un push al repositorio

```
root@daniel-MS-7E28:/home/mi-primer-repo# git push
Enter passphrase for key '/root/.ssh/id_ed25519':
Enter passphrase for key '/root/.ssh/id_ed25519':
Enumerando objetos: 8, listo.
Contando objetos: 100% (8/8), listo.
Compresión delta usando hasta 12 hilos
Comprimiendo objetos: 100% (4/4), listo.
Escribiendo objetos: 100% (6/6), 643 bytes | 643.00 KiB/s, listo.
Total 6 (delta 1), reusados 0 (delta 0), pack-reusados 0
remote: Resolving deltas: 100% (1/1), done.
To github.com:Danimartinez-1997/mi-primer-repo.git
   cbb4432..6a187a5  main -> main
root@daniel-MS-7E28:/home/mi-primer-repo#
```

Comprobamos que el [readme.MD](#) ha añadido los cambios.

The screenshot shows the GitHub interface for the repository 'mi-primer-repo'. At the top, it indicates the repository is 'Public'. Below the repository name, there are buttons for 'Pin', 'Watch' (0), and a dropdown menu. The main section shows the 'main' branch with 1 branch and 0 tags. A search bar and 'Add file' button are present. The commit history shows a recent commit by 'Danimartinez-1997' titled 'Tercer commit: Mi presentacion' with commit hash '6a187a5' and a timestamp of '2 minutes ago'. Below the commit list, the 'README' file is displayed, showing the title 'mi-primer-repo' and the content: 'Daniel Martínez Vila Ciclo formativo: ASIR Soy Daniel, vivo en un pueblo de A Coruña y me gustan las artes marciales, leer libros y pasear'.

Y que se han realizado los tres commits de forma correcta.

Commits

The screenshot shows the 'Commits' page for the repository. It displays a list of commits on the 'main' branch. The commits are ordered chronologically, with the most recent at the top. Each commit entry includes the commit title, the author's name, the commit hash, and a timestamp. The commits are: 'Tercer commit: Mi presentacion' by 'Danimartinez-1997' (hash '6a187a5', 2 minutes ago), 'Segundo commit: Ciclo formativo' by 'Danimartinez-1997' (hash '32e70ca', 5 minutes ago), and 'Nombre' by 'Danimartinez-1997' (hash 'cbb4432', 10 minutes ago).

Tarea 2: .gitignore práctico

Creamos un nuevo repositorio vacío en nuestro GitHub.

The screenshot shows the GitHub interface for a new repository named 'proyecto-pruebas'. At the top, there are buttons for 'Pin', 'Watch' (0), 'Fork' (0), and 'Star' (0). Below this, there are two main cards: 'Start coding with Codespaces' with a 'Create a codespace' button, and 'Add collaborators to this repository' with an 'Invite collaborators' button. A 'Quick setup' section follows, showing the repository URL 'git@github.com:Danimartinez-1997/proyecto-pruebas.git' and a note about including a README, LICENSE, and .gitignore. Below this, there are two sections for command-line setup: '...or create a new repository on the command line' and '...or push an existing repository from the command line', each with a copy icon. The first section contains the following commands:

```
echo "# proyecto-pruebas" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:Danimartinez-1997/proyecto-pruebas.git
git push -u origin main
```

 The second section contains:

```
git remote add origin git@github.com:Danimartinez-1997/proyecto-pruebas.git
git branch -M main
git push -u origin main
```

 At the bottom, a note says: 'ProTip! Use the URL for this page when adding GitHub as a remote.'

y lo clonamos en la terminal

```
root@daniel-MS-7E28:/home# mkdir proyecto-pruebas
root@daniel-MS-7E28:/home# cd proyecto-pruebas
root@daniel-MS-7E28:/home/proyecto-pruebas# git init
ayuda: Usando 'master' como el nombre de la rama inicial. Este nombre de rama predeterminado
ayuda: está sujeto a cambios. Para configurar el nombre de la rama inicial para usar en todos
ayuda: de sus nuevos repositorios, reprimiendo esta advertencia, llama a:
ayuda: git config --global init.defaultBranch <nombre>
ayuda:
ayuda: Los nombres comúnmente elegidos en lugar de 'master' son 'main', 'trunk' y
ayuda: 'development'. Se puede cambiar el nombre de la rama recién creada mediante este comando:
ayuda: git branch -m <nombre>
Inicializado repositorio Git vacío en /home/proyecto-pruebas/.git/
root@daniel-MS-7E28:/home/proyecto-pruebas# git remote add origin https://github.com/TU_USUARIO/proyecto-pruebas.git
root@daniel-MS-7E28:/home/proyecto-pruebas#
```

Creamos varios archivos de prueba para comprobar que no se subirán, además de un index.txt que si se subirá al repositorio.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  AZURE

root@daniel-MS-7E28:/home/proyecto-pruebas# ls
index.html  node_modules  registro.log
root@daniel-MS-7E28:/home/proyecto-pruebas#
```

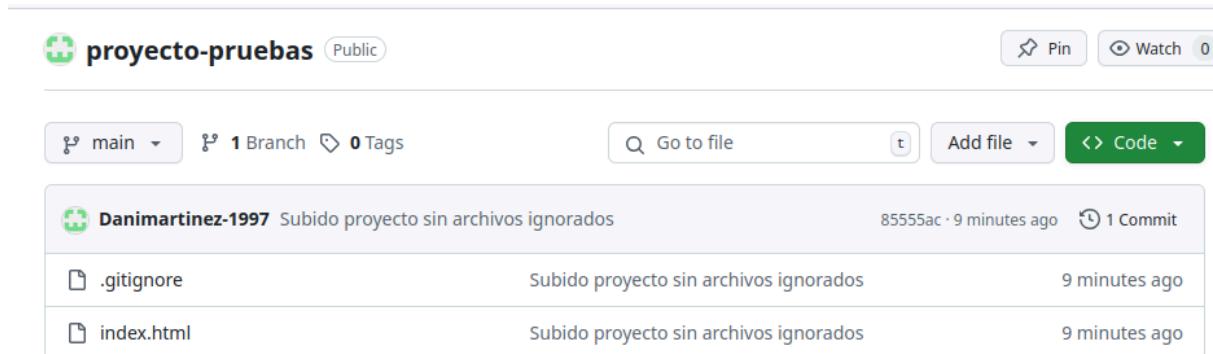
Creamos un archivo .gitignore que ignore todos los archivos con extensión .log, .tmp y la carpeta node_modules/

```
root@daniel-MS-7E28:/home/proyecto-pruebas# touch .gitignore
root@daniel-MS-7E28:/home/proyecto-pruebas# nano .gitignore
root@daniel-MS-7E28:/home/proyecto-pruebas# cat .gitignore
*.log
*.tmp
node_modules/
root@daniel-MS-7E28:/home/proyecto-pruebas#
```

Realizamos el commit y pusheamos al repositorio.

```
root@daniel-MS-7E28:/home/proyecto-pruebas# git add .
root@daniel-MS-7E28:/home/proyecto-pruebas# git commit -m "Subido proyecto sin archivos ignorados"
En la rama main
nada para hacer commit, el árbol de trabajo está limpio
root@daniel-MS-7E28:/home/proyecto-pruebas# git branch -M main
root@daniel-MS-7E28:/home/proyecto-pruebas# git push -u origin main
Enter passphrase for key '/root/.ssh/id_ed25519':
Enumerando objetos: 4, listo.
Contando objetos: 100% (4/4), listo.
Compresión delta usando hasta 12 hilos
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (4/4), 343 bytes | 171.00 KiB/s, listo.
Total 4 (delta 0), reusados 0 (delta 0), pack-reusados 0
To github.com:Danimartinez-1997/proyecto-pruebas.git
 * [new branch]      main -> main
rama 'main' configurada para rastrear 'origin/main'.
root@daniel-MS-7E28:/home/proyecto-pruebas#
```

Y comprobamos que solo se ha subido el .gitignore y el index.html al repositorio.



The screenshot shows the GitHub interface for a repository named 'proyecto-pruebas'. At the top, there's a header with the repository name, a 'Public' badge, and buttons for 'Pin' and 'Watch' (0). Below this, a navigation bar shows 'main' as the selected branch, '1 Branch', and '0 Tags'. A search bar 'Go to file' and buttons 'Add file' and 'Code' are also present. The main content area displays a commit by 'Danimartinez-1997' titled 'Subido proyecto sin archivos ignorados', made 9 minutes ago with 1 commit. Below the commit message, a table lists the files included in the commit:

File	Commit Message	Time
.gitignore	Subido proyecto sin archivos ignorados	9 minutes ago
index.html	Subido proyecto sin archivos ignorados	9 minutes ago

Tarea 4: Simulación de errores

Primero creamos y subimos el archivo [errores.sh](#) que utilizaremos para simular errores.

```
root@daniel-MS-7E28:/home/simulacion-errores# echo '#!/bin/bash' > errores.sh
root@daniel-MS-7E28:/home/simulacion-errores# echo "echo 'Este archivo está para simular errores'" >> errores.sh
root@daniel-MS-7E28:/home/simulacion-errores# chmod +x errores.sh
root@daniel-MS-7E28:/home/simulacion-errores# git add errores.sh
root@daniel-MS-7E28:/home/simulacion-errores# git commit -m "Añadido archivo errores.sh para simulación"
[master (commit-raíz) fb63620] Añadido archivo errores.sh para simulación
1 file changed, 2 insertions(+)
create mode 100755 errores.sh
root@daniel-MS-7E28:/home/simulacion-errores#
```

Ahora vamos a simular el error, borrando el archivo y haciendo commit a nuestro repositorio.

```
root@daniel-MS-7E28:/home/simulacion-errores# rm errores.sh
root@daniel-MS-7E28:/home/simulacion-errores# git add .
root@daniel-MS-7E28:/home/simulacion-errores# git commit -m "ERROR: borrado accidental del archivo errores.sh"
[master b15f2e6] ERROR: borrado accidental del archivo errores.sh
1 file changed, 2 deletions(-)
delete mode 100755 errores.sh
root@daniel-MS-7E28:/home/simulacion-errores#
```

Vamos a recuperar el archivo antes del error. Primero utilizamos git reflog para ver todos los cambios.

Y posteriormente, utilizamos git reset con el id del commit al que nos interese volver.

```
root@daniel-MS-7E28:/home/simulacion-errores# git reflog
b15f2e6 (HEAD -> master) HEAD@{0}: commit: ERROR: borrado accidental del archivo errores.sh
fb63620 HEAD@{1}: commit (initial): Añadido archivo errores.sh para simulación
root@daniel-MS-7E28:/home/simulacion-errores# git reset --hard fb63620
HEAD está ahora en fb63620 Añadido archivo errores.sh para simulación
root@daniel-MS-7E28:/home/simulacion-errores#
```

Por último, creamos un archivo “[RECUPERACION.md](#)” en el que explicamos los pasos realizados.

```
root@daniel-MS-7E28:/home/simulacion-errores# echo "# Recuperación de errores en Git" > RECUPERACION.md
root@daniel-MS-7E28:/home/simulacion-errores# echo "" >> RECUPERACION.md
root@daniel-MS-7E28:/home/simulacion-errores# echo "## Error simulado" >> RECUPERACION.md
root@daniel-MS-7E28:/home/simulacion-errores# echo "Borré accidentalmente el archivo errores.sh y confirmé el cambio con un commit." >> RECUPERACION.md
root@daniel-MS-7E28:/home/simulacion-errores# echo "" >> RECUPERACION.md
root@daniel-MS-7E28:/home/simulacion-errores# echo "## Comandos usados para recuperarlo" >> RECUPERACION.md
root@daniel-MS-7E28:/home/simulacion-errores# echo "1. git reflog" >> RECUPERACION.md
root@daniel-MS-7E28:/home/simulacion-errores# echo "2. git reset --hard 'id_commit' " >> RECUPERACION.md
root@daniel-MS-7E28:/home/simulacion-errores# echo "" >> RECUPERACION.md
root@daniel-MS-7E28:/home/simulacion-errores# echo "## Lecciones aprendidas" >> RECUPERACION.md
root@daniel-MS-7E28:/home/simulacion-errores# echo "Aprendí que Git tiene formas de recuperar los errores subidos y los archivos perdidos o mal configurados si se actúa a tiempo." >> RECUPERACION.md
root@daniel-MS-7E28:/home/simulacion-errores# git add .
root@daniel-MS-7E28:/home/simulacion-errores# git commit -m "Archivo de recuperacion"
[master a65cd88] Archivo de recuperacion
1 file changed, 11 insertions(+)
create mode 100644 RECUPERACION.md
root@daniel-MS-7E28:/home/simulacion-errores#
```

Y pusheamos todo a nuestro repositorio.

```
root@daniel-MS-7E28:/home/simulacion-errores# git remote add origin git@github.com:Danimartinez-1997/simulacion-errores.git
root@daniel-MS-7E28:/home/simulacion-errores# git branch -M main
root@daniel-MS-7E28:/home/simulacion-errores# git push -u origin main
Enter passphrase for key '/root/.ssh/id_ed25519':
Enumerando objetos: 6, listo.
Contando objetos: 100% (6/6), listo.
Compresión delta usando hasta 12 hilos
Comprimiendo objetos: 100% (5/5), listo.
Escribiendo objetos: 100% (6/6), 823 bytes | 823.00 KiB/s, listo.
Total 6 (delta 0), reusados 0 (delta 0), pack-reusados 0
To github.com:Danimartinez-1997/simulacion-errores.git
 * [new branch]      main -> main
rama 'main' configurada para rastrear 'origin/main'.
root@daniel-MS-7E28:/home/simulacion-errores#
```

Conclusión

Para terminar, comprobamos en nuestro repositorio que todo se ha subido correctamente y que la tarea está bien realizada.

simulacion-errores Public

main 1 Branch 0 Tags

Go to file Add file Code

Danimartinez-1997 Archivo de recuperacion a65cd88 · 5 minutes ago 2 Commits

RECUPERACION.md	Archivo de recuperacion	5 minutes ago
errores.sh	Añadido archivo errores.sh para simulación	19 minutes ago

Y entramos en el archivo [RECUPERACION.md](#) para ver su contenido

Recuperación de errores en Git

Error simulado

Borré accidentalmente el archivo errores.sh y confirmé el cambio con un commit.

Comandos usados para recuperarlo

1. git reflog
2. git reset --hard 'id_commit'

Lecciones aprendidas

Aprendí que Git tiene formas de recuperar los errores subidos y los archivos perdidos o mal configurados si se actúa a tiempo.

Recuperación de errores en Git

Error simulado

Borré accidentalmente el archivo errores.sh y confirmé el cambio con un commit.

Comandos usados para recuperarlo

1. git reflog
2. git reset --hard 'id_commit'

Lecciones aprendidas

Aprendí que Git tiene formas de recuperar los errores subidos y los archivos perdidos o mal configurados si se actúa a tiempo.

