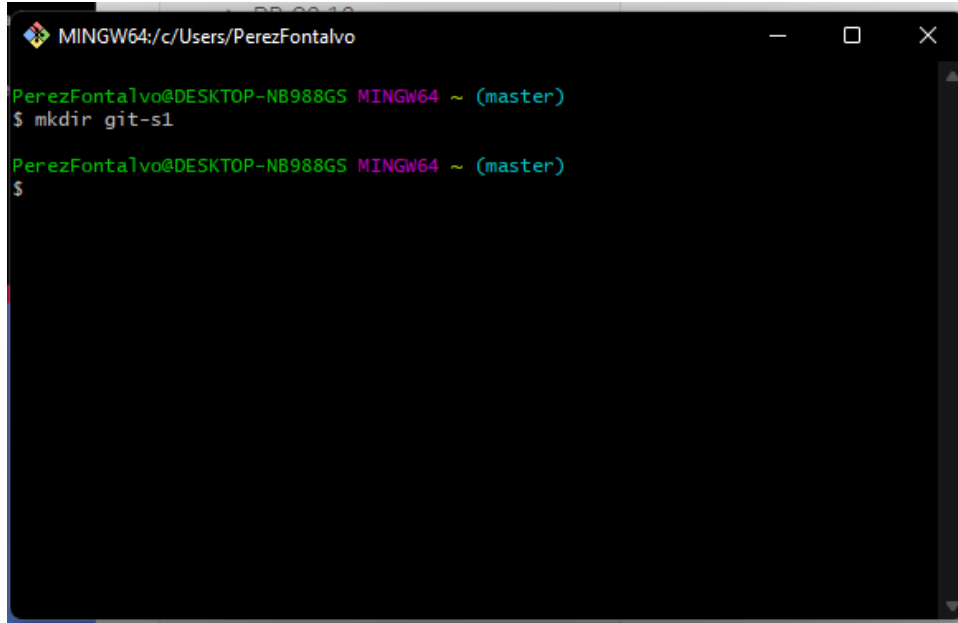


Crea una carpeta llamada git-s1

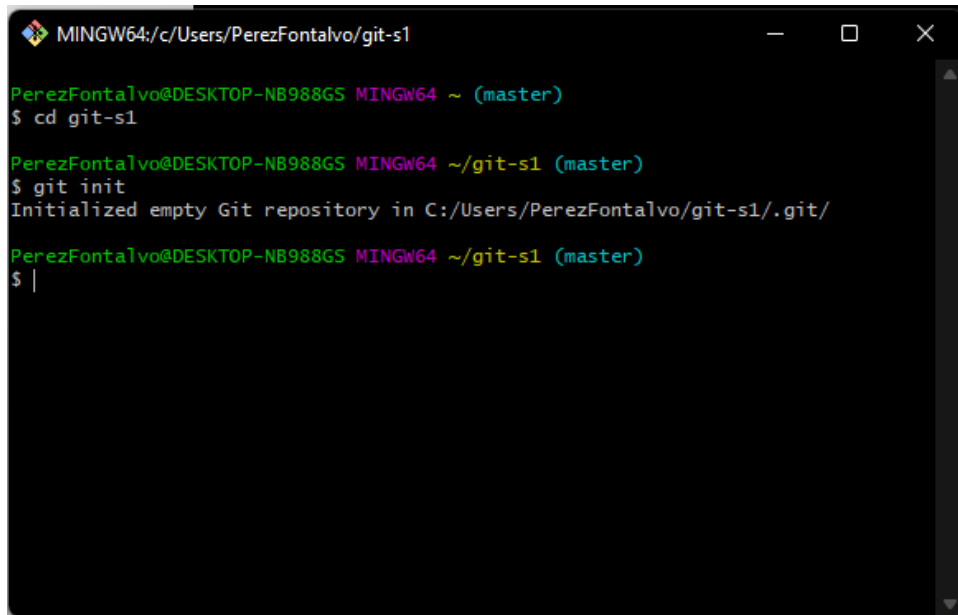
A terminal window titled 'MINGW64:/c/Users/PerezFontalvo' with standard window controls. The prompt is 'PerezFontalvo@DESKTOP-NB988GS MINGW64 ~ (master)'. The command '\$ mkdir git-s1' is entered and executed. The prompt returns to '\$'.

```
MINGW64:/c/Users/PerezFontalvo

PerezFontalvo@DESKTOP-NB988GS MINGW64 ~ (master)
$ mkdir git-s1

PerezFontalvo@DESKTOP-NB988GS MINGW64 ~ (master)
$
```

Ejecuta dentro de esta carpeta el comando “git init”

A terminal window titled 'MINGW64:/c/Users/PerezFontalvo/git-s1' with standard window controls. The prompt is 'PerezFontalvo@DESKTOP-NB988GS MINGW64 ~ (master)'. The command '\$ cd git-s1' is entered and executed. The prompt changes to 'PerezFontalvo@DESKTOP-NB988GS MINGW64 ~/git-s1 (master)'. The command '\$ git init' is entered and executed, resulting in the output 'Initialized empty Git repository in C:/Users/PerezFontalvo/git-s1/.git/'. The prompt returns to '\$ |'.

```
MINGW64:/c/Users/PerezFontalvo/git-s1

PerezFontalvo@DESKTOP-NB988GS MINGW64 ~ (master)
$ cd git-s1

PerezFontalvo@DESKTOP-NB988GS MINGW64 ~/git-s1 (master)
$ git init
Initialized empty Git repository in C:/Users/PerezFontalvo/git-s1/.git/


PerezFontalvo@DESKTOP-NB988GS MINGW64 ~/git-s1 (master)
$ |
```

Desde github crea un repositorio llamado `git_semana1-repositorio`

## Create a new repository

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

---

**Owner \***  `ing-jsperezg` / **Repository name \*** `git-semana1-repositorio` ✓

Great repository names are short and memorable. Need inspiration? How about [shiny-dollop](#)?

**Description** (optional)

`Reto Git-01 semana 1`

---

☒ **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.](#)

---

**Create repository**

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


 Set up in Desktop or **HTTPS** `https://github.com/ing-jsperezg/git-semana1-repositorio.git` 

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

---

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


```
echo "# git-semana1-repositorio" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/ing-jsperezg/git-semana1-repositorio.git
git push -u origin main
```



---

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

```
git remote add origin https://github.com/ing-jsperezg/git-semana1-repositorio.git
git branch -M main
git push -u origin main
```



---

#### ...or import code from another repository

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

**Import code**

Dentro de la carpeta git-s1 crea 2 archivos:

```
PerezFontalvo@DESKTOP-NB988GS MINGW64 ~/MisionTic2022/git-s1 (master)
$ echo Git-s1-doc1.txt
Git-s1-doc1.txt

PerezFontalvo@DESKTOP-NB988GS MINGW64 ~/MisionTic2022/git-s1 (master)
$ echo Git-s1-doc2.txt
Git-s1-doc2.txt
```

Agrega texto diferente en cada uno de ellos

```
PerezFontalvo@DESKTOP-NB988GS MINGW64 ~/MisionTic2022/git-s1 (master)
$ echo Git-s1-doc1.txt
Git-s1-doc1.txt

PerezFontalvo@DESKTOP-NB988GS MINGW64 ~/MisionTic2022/git-s1 (master)
$ echo Git-s1-doc2.txt
Git-s1-doc2.txt
```

Agrega a Git-s1-doc1.txt al área de preparación

```
PerezFontalvo@DESKTOP-NB988GS MINGW64 ~/MisionTic2022/git-s1 (master)
$ ls
Git-s1-doc1.txt  Git-s1-doc2.txt

PerezFontalvo@DESKTOP-NB988GS MINGW64 ~/MisionTic2022/git-s1 (master)
$ git add Git-s1-doc1.txt
warning: LF will be replaced by CRLF in Git-s1-doc1.txt.
The file will have its original line endings in your working directory
```

Revisa el estado de los archivos mediante el comando "git status". ¿Cual es la diferencia entre los dos archivos? ¿a qué se debe?

```
PerezFontalvo@DESKTOP-NB988GS MINGW64 ~/MisionTic2022/git-s1 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   Git-s1-doc1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Git-s1-doc2.txt
```

Confirmar el envío de los archivos preparados con el mensaje específico para ello.

```
PerezFontalvo@DESKTOP-NB988GS MINGW64 ~/MisionTic2022/git-s1 (master)
$ git commit -m "mi primer envio Archivo Git-s1-doc1.txt"
[master (root-commit) 2d72ac3] mi primer envio Archivo Git-s1-doc1.txt
1 file changed, 1 insertion(+)
create mode 100644 Git-s1-doc1.txt
```

Ahora se debe agregar al área de preparación el archivo Git-s1-doc2.txt

```
PerezFontalvo@DESKTOP-NB988GS MINGW64 ~/MisionTic2022/git-s1 (master)
$ git add Git-s1-doc2.txt
warning: LF will be replaced by CRLF in Git-s1-doc2.txt.
The file will have its original line endings in your working directory
```

Al tiempo se modifica el archivo Git-s1-doc1.txt nuevamente.

```
PerezFontalvo@DESKTOP-NB988GS MINGW64 ~/MisionTic2022/git-s1 (master)
$ echo ">Esto es tora linea para el archivo Git-s1-doc1" >> Git-s1-doc1.txt
```

Al mismo tiempo también se agrega un archivo nuevo llamado Git.s1.doc3.txt.

```
PerezFontalvo@DESKTOP-NB988GS MINGW64 ~/MisionTic2022/git-s1 (master)
$ echo "Tercer archivo creado desde git bash" >> Git-s1-doc3.txt
```

Revisar el estado del área de trabajo y describir lo que sucede.

```
PerezFontalvo@DESKTOP-NB988GS MINGW64 ~/MisionTic2022/git-s1 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   Git-s1-doc2.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   Git-s1-doc1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Git-s1-doc3.txt
```

Preparar los cambios y confirmarlos.

```
PerezFontalvo@DESKTOP-NB988GS MINGW64 ~/MisionTic2022/git-s1 (master)
$ git add Git-s1-doc1.txt
warning: LF will be replaced by CRLF in Git-s1-doc1.txt.
The file will have its original line endings in your working directory

PerezFontalvo@DESKTOP-NB988GS MINGW64 ~/MisionTic2022/git-s1 (master)
$ git add Git-s1-doc2.txt

PerezFontalvo@DESKTOP-NB988GS MINGW64 ~/MisionTic2022/git-s1 (master)
$ git add Git-s1-doc3.txt
warning: LF will be replaced by CRLF in Git-s1-doc3.txt.
The file will have its original line endings in your working directory
```

```
PerezFontalvo@DESKTOP-NB988GS MINGW64 ~/MisionTic2022/git-s1 (master)
$ git commit -m "Segunda actualizacion de repositorio"
[master 157e226] Segunda actualizacion de repositorio
3 files changed, 3 insertions(+)
 create mode 100644 Git-s1-doc2.txt
 create mode 100644 Git-s1-doc3.txt
```

Cargar los archivos al repositorio de github.

```
PerezFontalvo@DESKTOP-NB988GS MINGW64 ~/MisionTic2022/git-s1 (master)
$ git push -u origin master
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (8/8), 773 bytes | 257.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ing-jsperezg/git-semana1-repositorio.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

The screenshot shows the GitHub interface for the repository 'ing-jsperezg / git-semana1-repositorio'. The repository is public and has a 'master' branch with 1 branch and 0 tags. The commit history shows three commits, each titled 'Segunda actualizacion de repositorio' and dated '2 minutes ago'. The commit messages are 'Git-s1-doc1.txt', 'Git-s1-doc2.txt', and 'Git-s1-doc3.txt'. The repository has a README file and a 'Add a README' button. The right sidebar shows the repository's about section, releases, and packages.

ing-jsperezg / git-semana1-repositorio Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags Go to file Add file Code About

ing-jsperezg Segunda actualizacion de repositorio 157e226 2 minutes ago 2 commits

Git-s1-doc1.txt	Segunda actualizacion de repositorio	2 minutes ago
Git-s1-doc2.txt	Segunda actualizacion de repositorio	2 minutes ago
Git-s1-doc3.txt	Segunda actualizacion de repositorio	2 minutes ago

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

Reto Git-01 semana 1

Releases

No releases published  
[Create a new release](#)

Packages

No packages published  
[Publish your first package](#)