

Solución Laboratorio Git

Nombre: Miguel Angel Tinoco Arroyo

1. Crea un proyecto de repositorio en Github: github.com
 - a. Nombre del proyecto: sesión_github
 - b. Añade archivo README

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 * MiguelAngelTinocoArroyo / **Repository name *** sesión_github
✓ sesión_github is available.

Great repository names are short and memorable. Need inspiration? How about [bug-free-guacamole](#) ?

Description (optional)
Laboratorio de la clase git

☐ **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:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

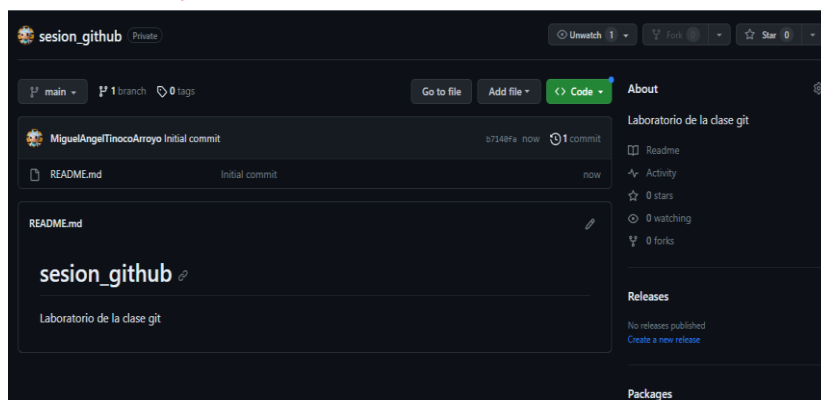
Add .gitignore
.gitignore template: **None**
Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license
License: **None**
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

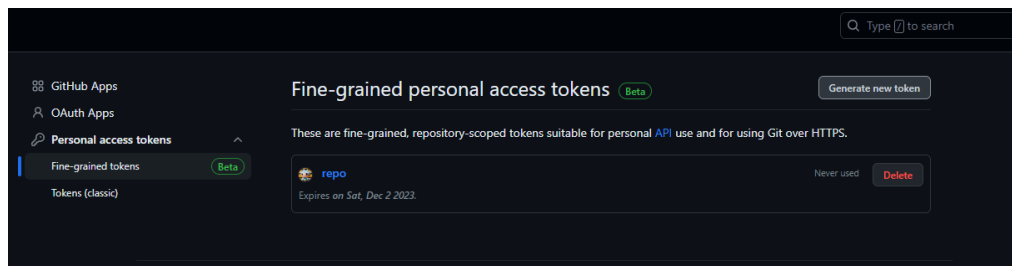
This will set `main` as the default branch. Change the default name in your [settings](#).

ⓘ You are creating a private repository in your personal account.

Create repository



2. Genera token como clave de acceso a repositorios privados



3. Verifica que git esté instalado, ya sea por cualquier medio de instalación.

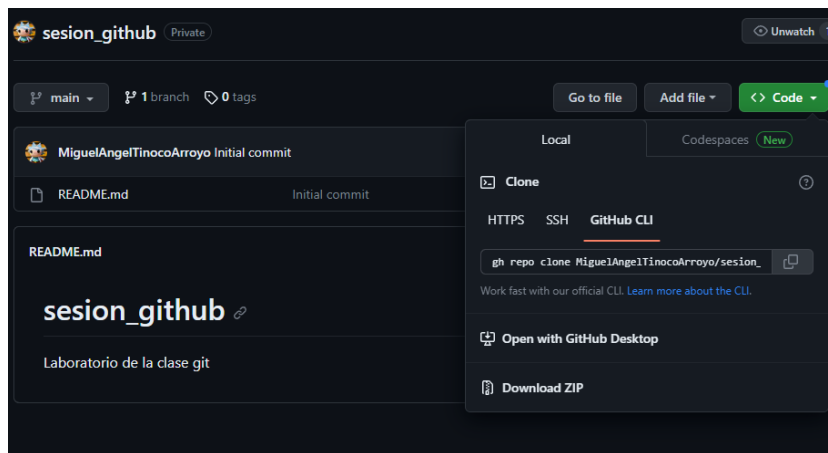
```
Miguel Angel@Miguel MINGW64 ~/Desktop
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.email=tinocoarroyomiguel@gmail.com
credential.helper=store --file ~/.git-credentials
```

4. Ahora dentro de la terminal crea una carpeta llamada laboratorio_git

```
Miguel Angel@Miguel MINGW64 ~/Desktop
$ mkdir laboratorio_git

Miguel Angel@Miguel MINGW64 ~/Desktop
$ ls
'Captura de pantalla 2023-10-03 214134.png' 'Laboratorio Git.docx'
'Captura de pantalla 2023-10-03 214308.png' 'Microsoft Teams (work or school).lnk'*
'DBeaver.lnk*' 'MySQL Workbench 8.0 CE.lnk'*
'Datapath Linux Practica calificada/' 'MySQL/'
'Datos repitencias 2023-1/' 'OGPL/'
'EXAMEN FINAL.pdf' 'Tarea 2'/'
'Examen Final'/' 'Tarea 2.docx'
'Exámen Final.docx' 'Visual Studio Code.lnk'*
'Exámen Final.pdf' 'desktop.ini'
'Exámen Final.sql' 'laboratorio_git/'
'Gráficas Informe 2023-1/' '~$boratorio Git.docx'
```

5. Ahora vamos a clonar el proyecto github creado



```
Miguel Angel@Miguel MINGW64 ~/Desktop
$ git clone https://github.com/MiguelAngelTinocoArroyo/sesion_github.git
Cloning into 'sesion_github'...
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.
```

6. Creamos tres ramas:

- a. Dev_git_branch
- b. qa_git_branch
- c. prod_git_branch

```
MINGW64; c:/Users/Miguel Angel/Desktop/sesion_github

Miguel Angel@Miguel MINGW64 ~/Desktop/sesion_github (main)
$ git branch dev_git_branch

Miguel Angel@Miguel MINGW64 ~/Desktop/sesion_github (main)
$ git branch qa_git_branch

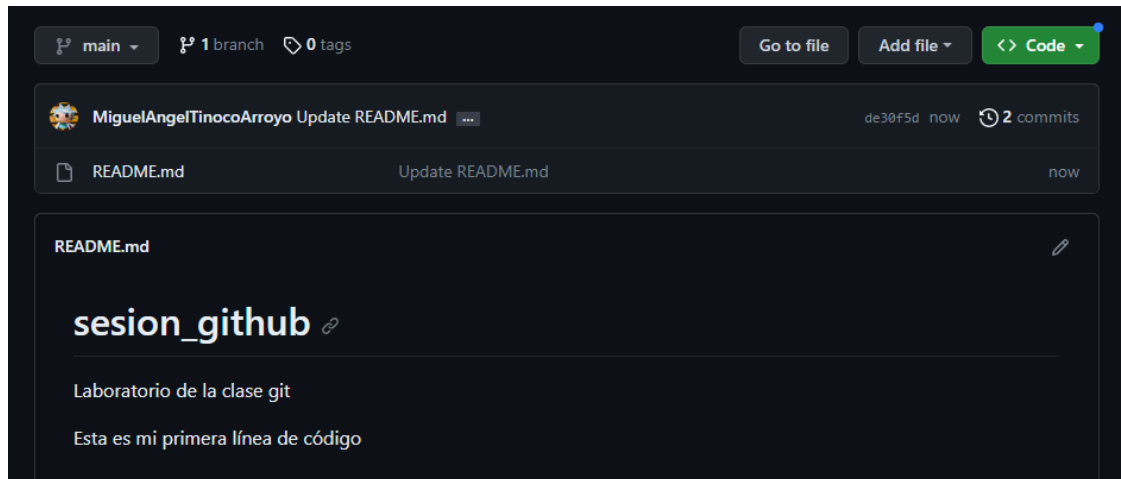
Miguel Angel@Miguel MINGW64 ~/Desktop/sesion_github (main)
$ git branch prod_git_branch
```

7. Enumere todos los Branch existentes

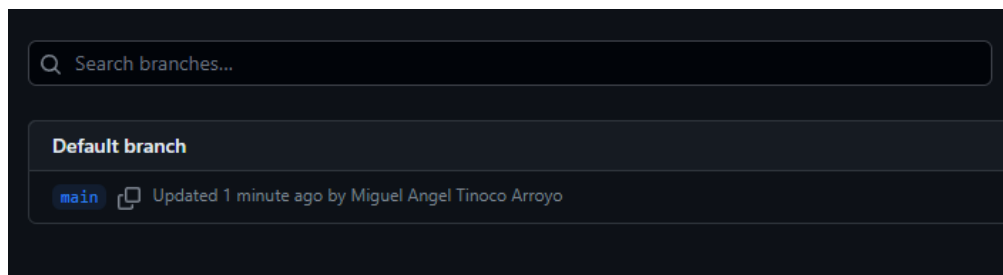
```
Miguel Angel@Miguel MINGW64 ~/Desktop/sesion_github (main)
$ git branch --list
dev_git_branch
* main
prod_git_branch
qa_git_branch

Miguel Angel@Miguel MINGW64 ~/Desktop/sesion_github (main)
$
```

8. Con cualquier editor de texto, editamos el archivo README.MD añadimos y guardamos: “Esta es mi primera línea de código”



9. Y checamos el estado del repositorio



```
Miguel Angel@Miguel MINGW64 ~/Desktop/sesion_github (main)
$ git log
commit b7140facb0fffa1345c2d6448fdd8b6c42ca045e (HEAD -> main, origin/main, origin/HEAD, qa_git_branch, prod_git_branch, dev_git_branch)
Author: Miguel Angel Tinoco Arroyo <101363440+MiguelAngelTinocoArroyo@users.noreply.github.com>
Date: Tue Oct 3 21:37:35 2023 -0500

Initial commit
```

10. Agregamos el archivo editado al staging área

```
MINGW64:/c/Users/Miguel Angel/Desktop/sesion_github

Miguel Angel@Miguel MINGW64 ~/Desktop/sesion_github (main)
$ git add archivo_editado.txt

Miguel Angel@Miguel MINGW64 ~/Desktop/sesion_github (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   archivo_editado.txt
```

11. Y checamos el estado del repositorio

```
commit b7140facb0fffa1345c2d6448fdd8b6c42ca045e (origin/main, origin/HEAD, qa_gi
t_branch, prod_git_branch, dev_git_branch)
Author: Miguel Angel Tinoco Arroyo <101363440+MiguelAngelTinocoArroyo@users.nore
ply.github.com>
Date:   Tue Oct 3 21:37:35 2023 -0500

    Initial commit

Miguel Angel@Miguel MINGW64 ~/Desktop/sesion_github (main)
$
```

12. Confirmamos la edición del archivo con un comit

```
Miguel Angel@Miguel MINGW64 ~/Desktop/sesion_github (main)
$ git commit -m "Agregado archivo_editado.txt al repositorio"
[main 4f8054b] Agregado archivo_editado.txt al repositorio
 1 file changed, 1 insertion(+)
 create mode 100644 archivo_editado.txt

Miguel Angel@Miguel MINGW64 ~/Desktop/sesion_github (main)
$
```

13. Checamos el estado

```

Miguel Angel@Miguel MINGW64 ~/Desktop/sesion_github (main)
$ git show
commit 4f8054b4c21eb3f5056babec05a0a60d85058f8f (HEAD -> main)
Author: Miguel Angel Tinoco Arroyo <tinocoarroyomiguel@gmail.com>
Date: Wed Oct 4 20:32:02 2023 -0500

    Agregado archivo_editado.txt al repositorio

diff --git a/archivo_editado.txt b/archivo_editado.txt
new file mode 100644
index 0000000..ac0211d
--- /dev/null
+++ b/archivo_editado.txt
@@ -0,0 +1 @@
+Archito de prueba modificado
Miguel Angel@Miguel MINGW64 ~/Desktop/sesion_github (main)
$

```

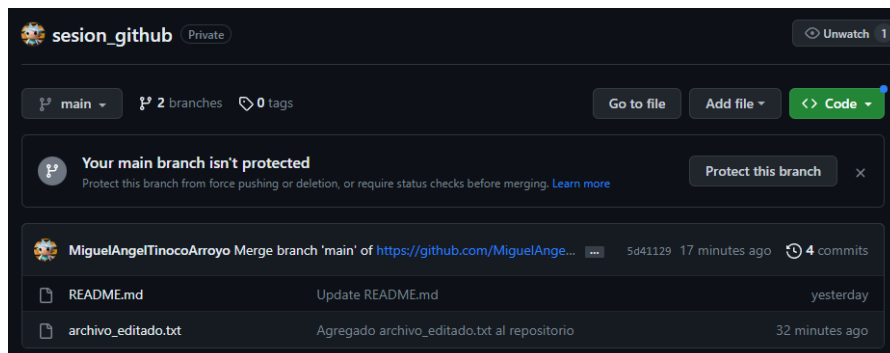
14. Publique sus cambios a la instancia de GitHub en la rama creada

```

Miguel Angel@Miguel MINGW64 ~/Desktop/sesion_github (dev_git_branch)
$ git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 683 bytes | 683.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MiguelAngelTinocoArroyo/sesion_github.git
   de30f5d..5d41129  main -> main
branch 'main' set up to track 'origin/main'.

```

15. Confirmamos los cambios del nuevo commit en GitHub en la rama creada.



16. Dentro de la plataforma modificamos README.md y añadimos una nueva línea y hacemos commit: “Segunda línea de código”



17. Como se modificó los cambios extraeremos los últimos cambios con pull del repositorio remoto.

```
Miguel Angel@Miguel MINGW64 ~/Desktop/sesion_github (dev_git_branch)
$ git pull origin main
From https://github.com/MiguelAngelTinocoArroyo/sesion_github
 * branch          main       -> FETCH_HEAD
Updating b7140fa..ecfaf86
Fast-forward
 README.md         | 2 ++
 archivo_editado.txt | 2 ++
 2 files changed, 4 insertions(+)
 create mode 100644 archivo_editado.txt

Miguel Angel@Miguel MINGW64 ~/Desktop/sesion_github (dev_git_branch)
$ |
```

18. Ahora fusionamos los cambios de la rama: dev_git_branch en una rama llamada qa_git_branch localmente con merge y lo subimos a GitHub con push.

