

Pasos para la creación de un repositorio local y clonarlo a GitHub:

- 1) Una vez descargado e instalado Git debemos abrir la consola y escribir los comandos:
 - `git config --global user.name "nombre y apellido"`
 - `git config --global user.email "mail"`

```
C:\Users\Facu>git version
git version 2.36.0.windows.1

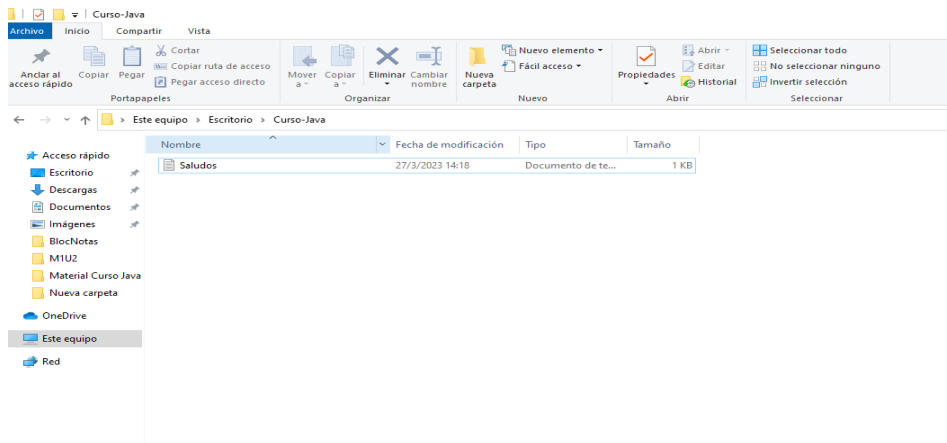
C:\Users\Facu>git config --global user.name "Facundo Irala"

C:\Users\Facu>git config --global user.email "facundoiralam@gmail.com"

C:\Users\Facu>git config --global --list
user.name=Facundo Irala
user.email=facundoiralam@gmail.com
color.ui=auto
```

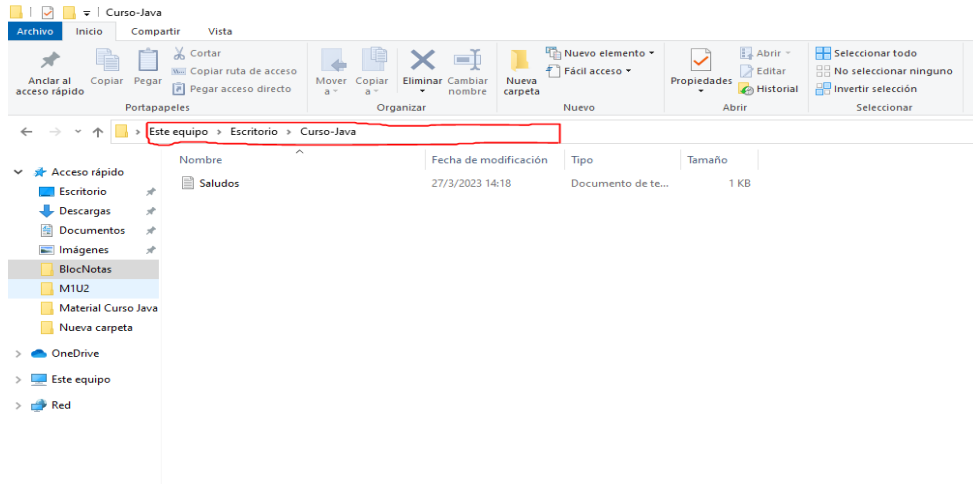
Para corroborar que dichos comandos se hayan efectuado, podremos usar el comando:

- `git config --global --list.`
- 2) Una vez realizado el primer paso, procederemos a crear una carpeta en nuestro sistema operativo. La carpeta deberá tener preferentemente un nombre relacionado al curso o actividad que estemos realizando. En mi casa se llamará "Curso-Java".

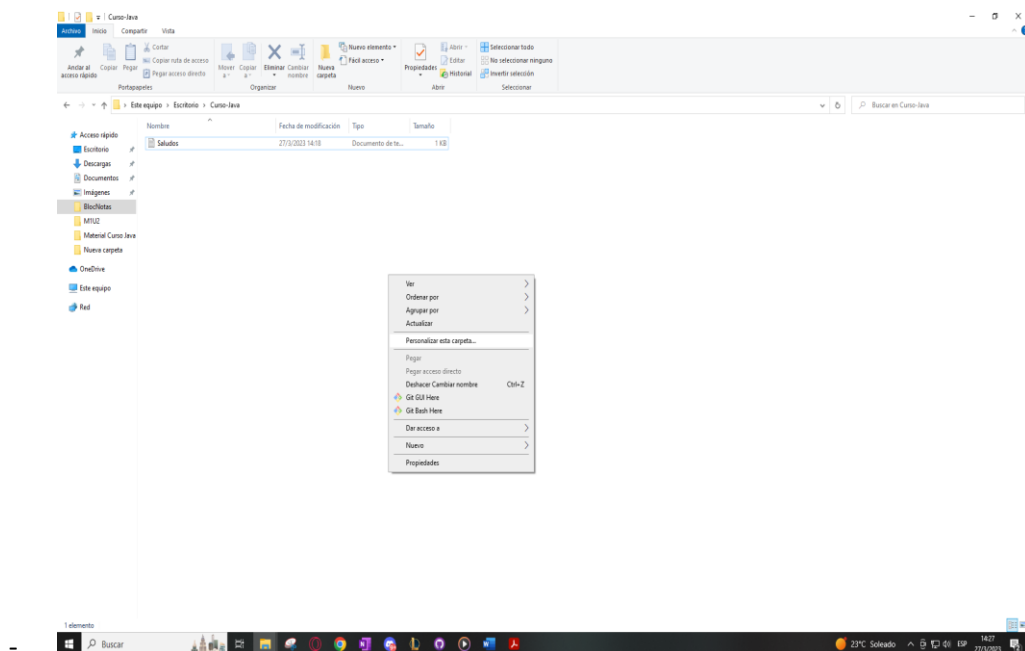


- 3) Crearemos un archivo txt, para así tener algo que subir al repositorio y luego abriremos la consola con CMD. Para abrir la consola CMD tenemos 2 opciones:

- La primera consiste en escribir cmd donde se encuentra la ruta de la carpeta



- O la segunda en hacer click ya derecho dentro de la carpeta y seleccionar la opción “Git bash here”.



- 4) Una vez dentro de la consola deberemos ejecutar el código “git init”, la salida debería ser similar a esto.

C:\Windows\System32\cmd.exe

```
Microsoft Windows [Versión 10.0.19044.2728]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Facu\Desktop\Curso-Java>git init
Initialized empty Git repository in C:/Users/Facu/Desktop/Curso-Java/.git/

C:\Users\Facu\Desktop\Curso-Java>
```

Podemos corroborar que se creó verificando que se haya creado en ese directorio una carpeta con el nombre .git. Es en esta carpeta donde Git almacena todos los cambios y configuraciones de nuestro repositorio.

- 5) Para verificar el estado de nuestro repositorio ejecutamos el comando git status. En este caso la salida debería ser similar a esta:

```
Microsoft Windows [Versión 10.0.19044.2728]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Facu\Desktop\Curso-Java>git init
Initialized empty Git repository in C:/Users/Facu/Desktop/Curso-Java/.git/

C:\Users\Facu\Desktop\Curso-Java>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Saludos.txt

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\Facu\Desktop\Curso-Java>
```

Lo que Git nos dice acá es que tenemos “Untracked files” o archivos sin versionar y nos muestra una lista donde aparece nuestro archivo. También nos indica que para agregarlo en un commit debemos usar el comando git add seguido de la ruta del archivo.

- 6) Procedemos a escribir el comando “git add”. En este caso como tenemos el archivo “Saludos.txt”, vamos a escribir “git add Saludos.txt”. Recordemos siempre respetar las mayúsculas y minúsculas respectivamente. Este comando no devolverá ningún resultado. Lo que el comando add hace es pasar al archivo al estado denominado stage. Si volvemos a ejecutar git status ahora veremos lo siguiente:

```
C:\Users\Facu\Desktop\Curso-Java>git add Saludos.txt

C:\Users\Facu\Desktop\Curso-Java>git status
On branch master

No commits yet

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

Git nos indica que en el próximo commit se incluirá el archivo Saludos.txt que acabamos de crear y agregar al repositorio.

- 7) Confirmamos la operación escribiendo el comando “git commit -m “creado el archivo de saludos”. La salida debería ser similar a:

```
C:\Users\Facu\Desktop\Curso-Java>git commit -m "creado el archivo de saludos"
[master (root-commit) c20a1c8] creado el archivo de saludos
 1 file changed, 1 insertion(+)
 create mode 100644 Saludos.txt
```

- 8) Una vez hecho todo esto, deberemos conectar nuestro repositorio local al GitHub. Para ellos deberemos ir a GitHub y crear un repositorio, haremos click en la solapa “NEW”:



- 9) Lo único que haremos será nombrar al repositorio y tildarlo como público. Procedemos a crear el repositorio.

The screenshot shows the GitHub 'Create a new repository' interface. At the top, it says 'Create a new repository' and provides a brief explanation. Below this, there are two input fields: 'Owner' (set to 'FacundoIrala') and 'Repository name' (set to 'Curso-Java' with a green checkmark). A note suggests great repository names are short and memorable. There is a 'Description (optional)' text area. Under the 'Visibility' section, 'Public' is selected with a radio button, indicating that anyone on the internet can see the repository. Below this, there are sections for 'Initialize this repository with:', 'Add a README file', 'Add .gitignore', and 'Choose a license'. Each of these sections has a 'None' option selected. At the bottom, there is a green 'Create repository' button and a note stating 'You are creating a public repository in your personal account.'

- 10) Una vez creado el repositorio, nos deberá salir esta pantalla. Debemos copiar los tres comandos resaltados y pegarlos en la consola CMD, dentro de nuestro repositorio local.

The screenshot shows the 'Quick setup' page on GitHub, which provides instructions for setting up a repository. It offers three options: 'Set up in Desktop', 'HTTPS', and 'SSH'. The 'SSH' option is selected, and the URL 'https://github.com/FacundoIrala/Curso-Java.git' is displayed. Below this, there is a section titled '...or create a new repository on the command line' which lists a series of git commands. The last three commands are highlighted with red underlines: 'git branch -M main', 'git remote add origin https://github.com/FacundoIrala/Curso-Java.git', and 'git push -u origin main'. Another section titled '...or push an existing repository from the command line' shows a similar set of commands. At the bottom, there is a section titled '...or import code from another repository' with a note about initializing the repository with code from other systems and an 'Import code' button.

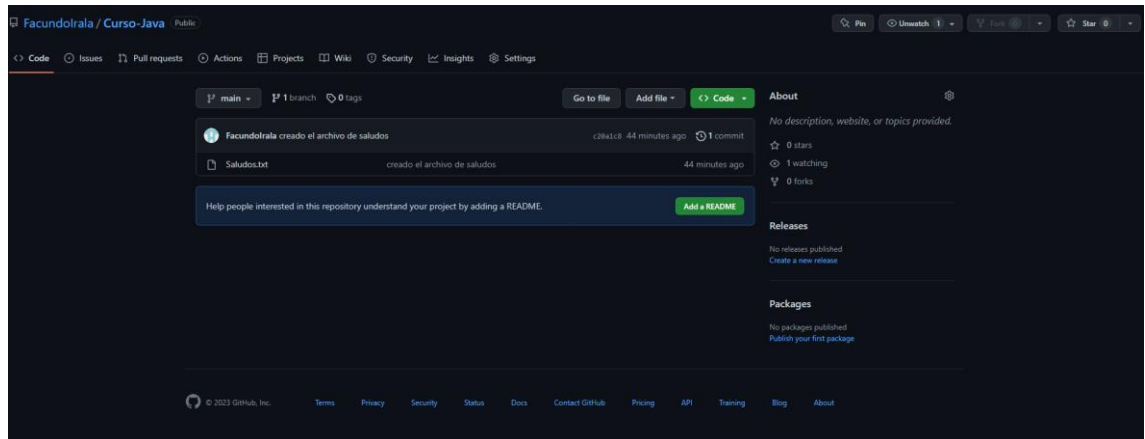
```
echo "# Curso-Java" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/FacundoIrala/Curso-Java.git
git push -u origin main
```

```
git remote add origin https://github.com/FacundoIrala/Curso-Java.git
git branch -M main
git push -u origin main
```

La salida debe ser similar a esta:

```
C:\Users\Facu\Desktop\Curso-Java>git branch -M main
C:\Users\Facu\Desktop\Curso-Java>git remote add origin https://github.com/FacundoIrala/Curso-Java.git
C:\Users\Facu\Desktop\Curso-Java>git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 269 bytes | 269.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/FacundoIrala/Curso-Java.git
 * [new branch]    main -> main
branch 'main' set up to track 'origin/main'.
```

- 11) Una vez realizado todo esto, volveremos al repositorio de GitHub y lo recargaremos.



Como podemos apreciar, nuestro repositorio ha sido creado exitosamente y el archivo que hemos hecho ya se encuentra subido.

- 12) Ahora si creamos otro archivo, en la carpeta de nuestro repositorio local, debemos hacer el mismo procedimiento. En este caso subiremos el archivo tp_2.doc. Para ellos haremos:
- Git add .
 - Git commit -m “nombre descriptivo”
- Y a lo último el comando “git push”, que lo que hará será subir el archivo al repositorio de GitHub.