

## Parte II

### **Que es Git**

Git fue creado originalmente como una herramienta para gestionar el desarrollo del kernel de Linux, desde entonces se ha convertido en una herramienta fundamental en todo ámbito de la programación. Git es un sistema de control de versiones distribuido para su uso principal para el desarrollo de software

Git va a permitir a los desarrolladores darle un mejor seguimiento de los cambios de código fuente de un proyecto a lo largo del tiempo. Git proporcionara los registros detallados de cada modificación realizada. Lo cual facilita revertir y fusionar diferentes versiones de forma mas eficiente.

La preferencia de la herramienta de Git es por sus funcionalidades integradas como, clonar repositorios, crear ramas, fusionar cambios, comparar diferencias, gestionar conflictos, entre otros. Lo cual lo hace mucho mas versátil que otras formas de compartir proyectos.

### **Control de Versiones con GIT**

Para el control de Versiones en Git hay que entender algunos conceptos tales como:

- Repositorios: va a ser una estructura que almacena todos los archivos u la historia de cambios del proyecto. El cual puede ser local, o remoto como un servidor o servicio en la nube.
- Commits: Un commit en Git es una instantánea o punto de control que captura los cambios realizados en uno o varios archivos en un momento determinado.
- Ramas: git permite crear ramas paralelas en la que se puede realizar cambios de forma independiente. Las ramas permite trabajar en una solución sin afectar la rama principal. Si esta rama se considera estable se procede con la fusión con el ramal principal

Git almacena el historial completo de cambios, cuál va a permitir ver la evolución del proyecto a lo largo del tiempo. Además, puedes comparar diferencias entre commits, ramas o archivos específicos para comprender los cambios realizados.

### **Estado de un archivo en git**

Estado	Descripción
Untracked (No rastreado)	Cuando un archivo no se encuentra en el área de preparación y Git no está realizando un seguimiento de sus cambios, se considera "no rastreado". Esto significa que Git no está al tanto de las modificaciones realizadas en el archivo.
Unmodified (Sin modificar)	Después de agregar un archivo al repositorio y realizar un commit, el archivo pasa al estado "sin modificar". Esto significa que no se han realizado cambios en el archivo desde el último commit y Git no detecta modificaciones en él.
Modified (Modificado)	Cuando se realizan cambios en un archivo que se encuentra en el repositorio, Git detecta las modificaciones y el archivo pasa al estado "modificado". Estos cambios aún no se han incluido en el próximo commit.
Staged (Preparado)	Para incluir los cambios de un archivo en el próximo commit, debes agregarlo al área de preparación (staging area). El archivo en este estado se considera "preparado" y está listo para ser incluido en el siguiente commit.
Committed (Confirmado)	Después de realizar un commit, los cambios en el archivo se confirman en el repositorio y se guarda una instantánea del estado actual. El archivo vuelve al estado "sin modificar" hasta que se realicen nuevas modificaciones.

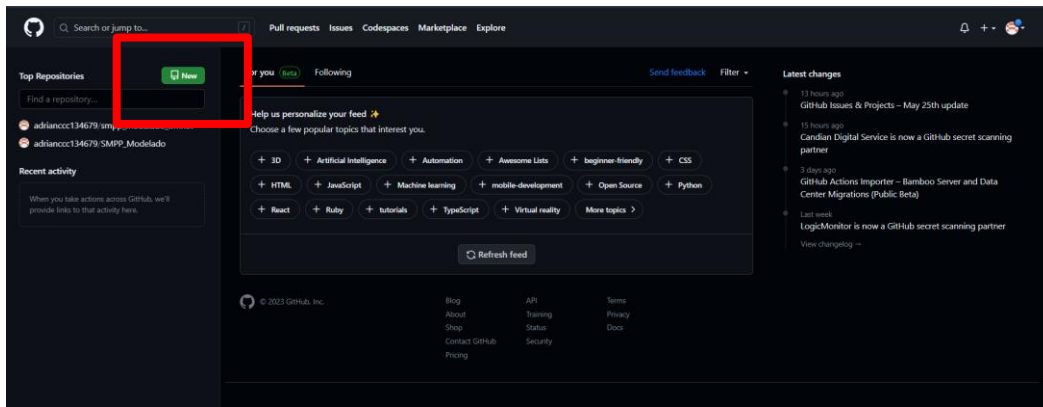
Para averiguar el estado de los archivos del repositorio se utiliza el comando "git status"

## Como se configura un repositorio

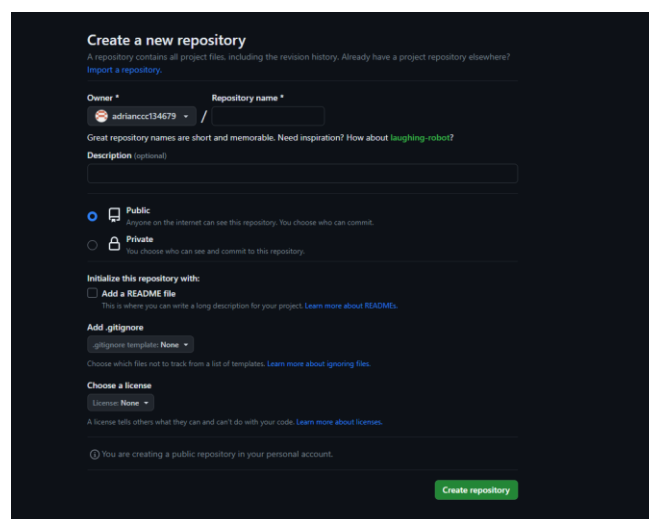
Acceder a tu sesión de git hub



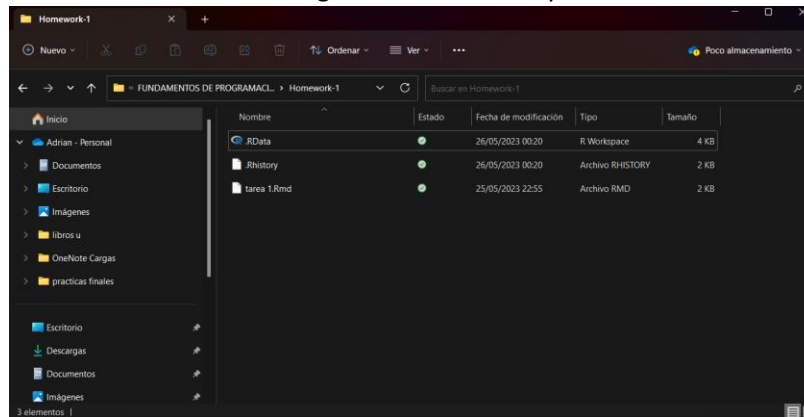
Le damos click en new



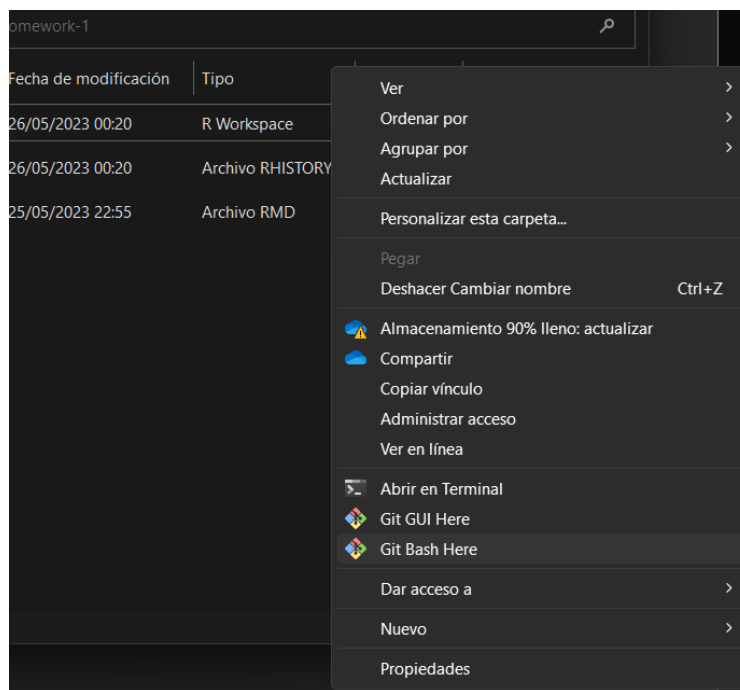
Llenamos la información de nuestro repositorio

A screenshot of the "Create a new repository" form on GitHub. The form includes fields for "Owner" (set to "adrianc134679"), "Repository name", and "Description (optional)". It also has radio buttons for "Public" (selected) and "Private". Below these are sections for "Initialize this repository with:" (including "Add a README file"), "Add .gitignore" (with a dropdown menu), and "Choose a license" (with a dropdown menu). A green "Create repository" button is at the bottom right. A note at the bottom states: "You are creating a public repository in your personal account."

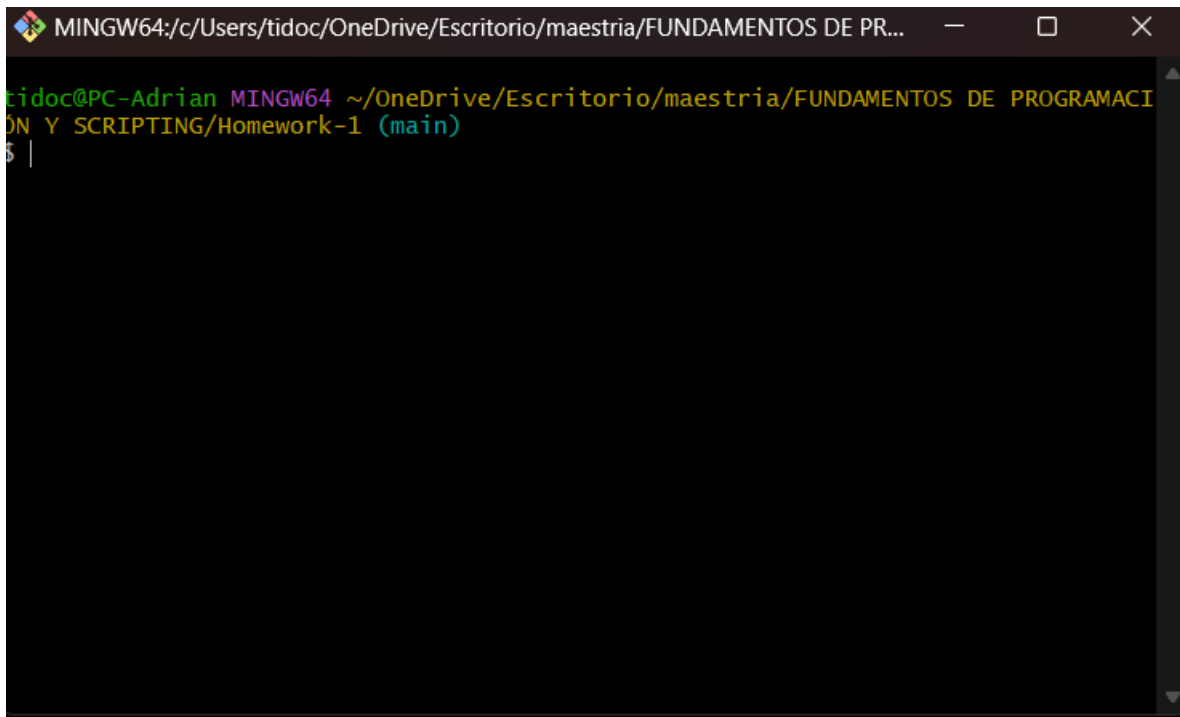
Nos dirigimos a nuestra carpeta



Le damos a la opción “Git Bash Here”



En la terminal emergente utilizamos los siguientes comandos:



```
MINGW64:/c/Users/tidoc/OneDrive/Escritorio/maestria/FUNDAMENTOS DE PR...
tidoc@PC-Adrian MINGW64 ~/OneDrive/Escritorio/maestria/FUNDAMENTOS DE PROGRAMACION Y SCRIPTING/Homework-1 (main)
$ |
```

Sobre la terminal se utilizan los comandos recomendados por la pagina de git en el momento que creamos el repositorio

1. `git init`
2. `git remote add origin https://github.com/adrianccc134679/Homework-R.git`
3. `git status`
4. `git add .`
5. `git commit -m "Primer commit"`
6. `git branch -M main`
7. `git push -u origin main`

### Comando en GIT

Comandos	Función
git init	Crea un nuevo repositorio Git vacío en el directorio actual.
git status	Al arrastrar un archivo a nuestra carpeta, utilizamos este comando para verificar si git ya guardo de manera adecuada nuestros archivos en el repositorio. Apareciendo de color verde si se realizó correctamente
git add + nombre del archivo	Utilizar este comando hay que especificar el .pdf o .m para agregar un archivo a nuestro repositorio. O se puede utilizar "git add ." para agregar todos los archivos que después de su nombre tengan un punto algo.
git rm + nombre del archivo	Nos permite remover un archivo del repositorio, lo cual borrara el archivo. Por lo que si quieres mantener el archivo en la computadora, pero fuera de tu repositorio utilizamos " git rm --cached +nombre del archivo"
git commit -a -m "primer commit"	Con este comando nos va estar ayudando a tener control sobre nuestro historial. Una vez ejecutado el comando nos mostrara los archivos que podremos encontrar en ese commit y a partir de ahora todas las modificaciones se realizaran sobre el siguiente historial. la bandera -a indicar que quieres hacer commit sobre todos los archivos. la bandera -m es para agregar un mensaje a nuestro commit.
git Branch	Con este comando nos va ayudar a mostrar una lista de todas las ramas en el repositorio.
git push	Va ser el comando que nos apoye a subir los commits locales al repositorio remoto
Git pull	Descargara los cambios mas recientes del repositorio remoto y los fusiona con tu rama actual