

Actividad 2.1: instalación y primeros pasos en Git

Introducción

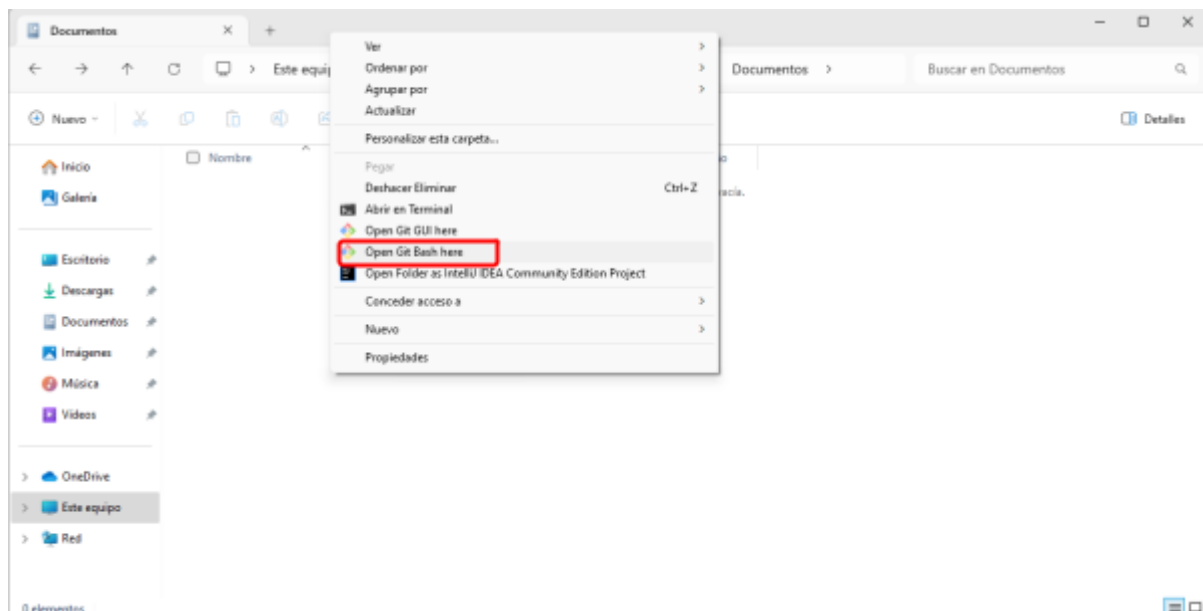
El objetivo de esta actividad es instalar Git para la gestión de repositorios y realizar los primeros pasos a través de un repositorio de prueba que nos permita entender cómo funciona el sistema de control de versiones.

Debes documentar los ejercicios propuesto, haciendo capturas de pantalla de todas las acciones realizadas y los resultados obtenidos, creando un informe en formato pdf.

1. Introducción a Git y configuración básica

Objetivo: Configurar Git y familiarizarse con los comandos básicos.

1. **Instalar Git:** descargar instalador y ejecutar: <https://git-scm.com/downloads>
2. **Configurar Git:** Abrimos una terminal de Git sobre vuestro directorio personal. Para ello, pulsamos botón derecho sobre el directorio y clicamos en “git bash here”



3. En la ventana de comandos que se nos abre, ejecutar los siguientes comandos para configurar su nombre de usuario y correo electrónico:

```
git config --global user.name "Tu Nombre"
```

```
git config --global user.email "tu.email@example.com"
```

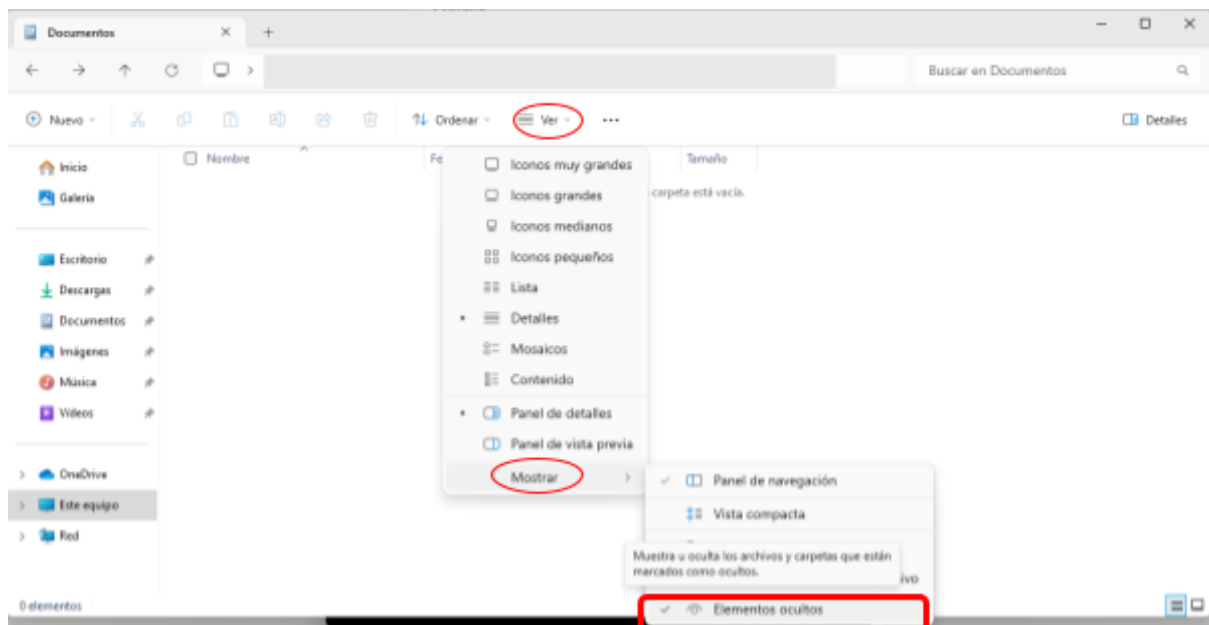
4. Vamos a configurar también el editor de texto por defecto para algunas de las acciones a realizar en git. En este caso, configuraremos notepad++ como editor:

```
git config --global core.editor "notepad++"
```

5. **Comprobar configuración:** Usar `git config --list` para verificar que la configuración es correcta.
6. **Crear un repositorio vacío:** Crear un directorio en su ordenador y ejecutar

```
git init miPrimerRepositorio
```

para iniciar un nuevo repositorio. Sobre el directorio, comprobar qué ha sucedido. Si no veis ningún contenido, configurad Windows para que os muestre los archivos ocultos (menú *ver>Mostrar>Elementos Ocultos*):



7. **Accede al repositorio creado:** en la consola ya abierta, muévete al directorio creado utilizando el comando `cd`:

```
cd miPrimerRepositorio
```

¿Aparece algo nuevo en la consola?

8. **Estado del repositorio:** comprueba el estado del repositorio utilizando el comando

```
git status
```

```
danielmartinan@WN-I9E31 MINGW64 ~/Documents
$ git init miPrimerRepositorio
Initialized empty Git repository in C:/Users/danielmartinan/Documents/miPrimerRepositorio/.git/

danielmartinan@WN-I9E31 MINGW64 ~/Documents
$ cd miPrimerRepositorio/

danielmartinan@WN-I9E31 MINGW64 ~/Documents/miPrimerRepositorio (master)
$ git status
On branch master

no commits yet

nothing to commit (create/copy files and use "git add" to track)

danielmartinan@WN-I9E31 MINGW64 ~/Documents/miPrimerRepositorio (master)
$ |
```

2. Primer commit

Objetivo: Comprender el flujo básico de Git para realizar cambios en un repositorio.

1. Añadir un archivo al repositorio:

- Crear un archivo de texto llamado **README.md** con una breve descripción del proyecto (por ejemplo, "Este es mi primer proyecto con Git").
- Comprueba cómo ha cambiado el estado del repositorio utilizando el comando

```
git status
```

- Añadir el archivo al *staging area* con

```
git add README.md.
```

- Vuelve a comprobar el estado del repositorio (**git status**)

2. Realizar el primer commit:

- Ejecutar el comando

```
git commit -m "Primer commit"
```

para registrar el cambio.

- Usar

```
git status
```

para comprobar el estado del después del commit.

3. Historial y revertir cambios

Objetivo: Consultar el historial de cambios y revertir modificaciones.

- **Ver el historial de commits:**
 - Usar `git log` para ver el historial de commits.
- **Modificar un archivo existente:**
 - Editar el archivo `README.md` y añadir una nueva línea de texto. estoy x aca
 - Ver el estado del repositorio con `git status`.
- **Revertir un archivo modificado:**
 - Usar `git checkout -- README.md` para deshacer los cambios y volver al estado anterior del archivo.
- **Hacer un nuevo commit:** Modificar el archivo de nuevo, añadirlo al staging con `git add` y realizar un nuevo commit.

4. Trabajo con ramas

Objetivo: Crear y trabajar con ramas para desarrollar características de manera aislada.

Ejercicios:

- **Crear una rama nueva:**
 - Crear una rama llamada `nueva-funcionalidad` usando `git branch nueva-funcionalidad`.
 - Cambiar a la nueva rama con `git checkout nueva-funcionalidad`.

Nota: se puede crear una rama y cambiarte a ella con un solo comando:

`git checkout -b nueva-funcionalidad`.
- **Hacer cambios en la nueva rama:**
 - Editar el archivo `README.md` para añadir una nueva sección.
 - Realizar un commit con los cambios.
- **Cambiar de rama:** Volver a la rama principal (`main`) con `git checkout main` y comprobar que los cambios no se ven en esta rama.
- **Unir ramas (merge):**
 - Unir la rama `nueva-funcionalidad` a `main` usando `git merge nueva-funcionalidad`.

5. Resolución de conflictos

Objetivo: Gestionar conflictos al hacer merges entre ramas.

Ejercicios:

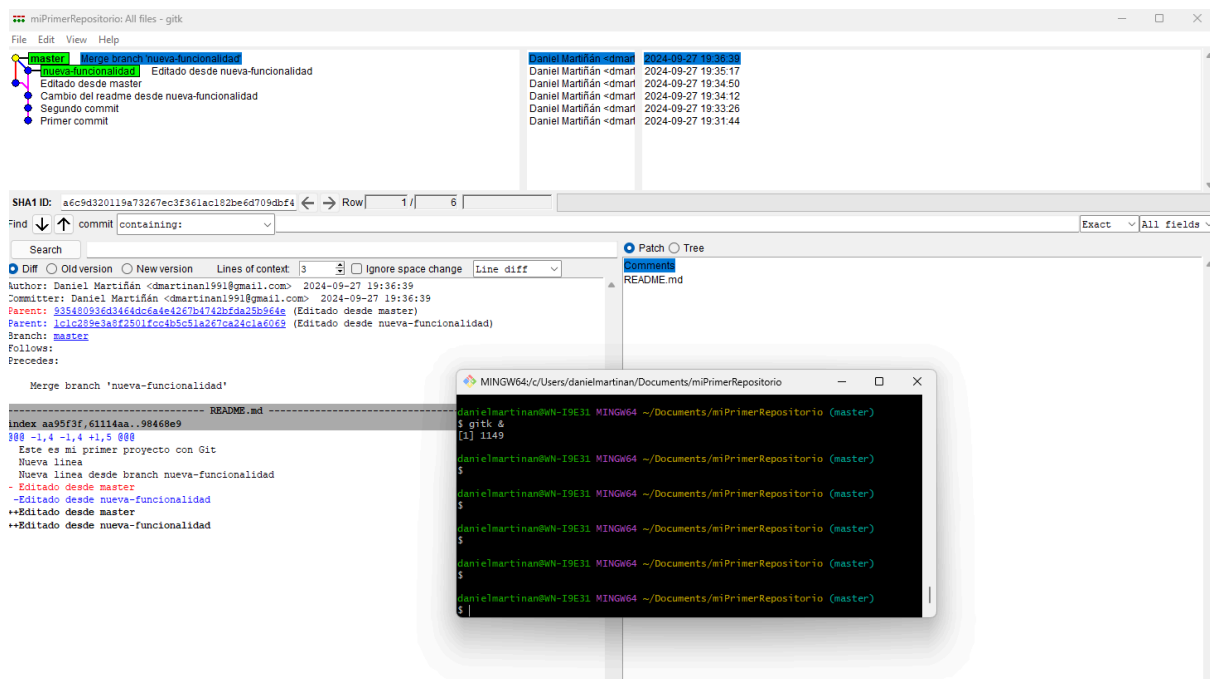
- **Generar un conflicto:**
 - En la rama `main`, modificar una línea en el archivo `README.md` y hacer un commit.
 - Cambiar a la rama `nueva-funcionalidad`, modificar la misma línea de texto y hacer un commit.
- **Intentar fusionar las ramas:**
 - Al intentar fusionar `nueva-funcionalidad` con `main`, surgirá un conflicto.
- **Resolver el conflicto:**
 - Usar el editor de texto para resolver el conflicto manualmente y después hacer el commit de la resolución. En el commit resultante, deben mantenerse las dos modificaciones del archivo `README.md`, primero la realizada en la rama `main` y luego la realizada en la rama `nueva-funcionalidad`

6. Visualización de las ramas

Objetivo: utilizar la utilidad `gitk` para visualizar las ramas.

Ejercicios:

- **Utilizacion de gitk**
 - Ejecuta el comando `gitk &` en la terminal de git. Se abrirá una herramienta visual con el siguiente aspecto



En la esquina superior izquierda podrás ver los commits (cada una de las bolas) junto a su mensaje descriptor; además, en las cajas verdes, verás el commit en el que se sitúa el HEAD de cada rama (en este caso, `master` en lugar de `main` y `nueva-funcionalidad`).

Verás en la parte superior también el autor y la fecha de realización de cada uno de esos commits. Si pinchas en uno de ellos, podrás ver en la parte inferior izquierda los cambios con su versión anterior, y en la parte derecha los archivos que han sido modificados o añadidos. Te aconsejo que navegues por esta interfaz y te familiarices con ella.

Si has realizado todos los pasos descritos en esta práctica, tendrías que ver un resultado muy similar al de la imagen anterior.