Gestión de versiones

Git y GitHub

Luis Miguel de la Cruz Salas

2025-05-07

Table of contents

1 Ejercicio 1.A.

Configuración del usuario y del correo electrónico.

- Usuario:

```
git config --global user.name "tu_usuario_github"
```

- Email:

```
git config --global user.email tu_usuario@correo.com
```

- Verificar:

```
git config --global --list
```

2 Ejercicio 1.B.

Crear un nuevo repositorio con un archivo.

- Crear el directorio `new_proy`:

```
mkdir new_proy
```

- Cambiarse al directorio `new_proy`:

```
cd new_proy
```

- Crear el archivo `README.md` con el siguiente contenido:

```
Este proyecto es una prueba de **Git** y **GitHub**.
```

- Listar el contenido del directorio.

```
ls -l
```

- Inicializar el repositorio local `new_proy`:

```
git init
```

- Listar nuevamente el contenido del directorio incluyendo archivos ocultos:

```
ls -la
```

3 Ejercicio 1.C.

Dar seguimiento al archivo y crear una confirmación.

- Revisar el estado del proyecto:

```
git status
```

- Preparar el archivo `README.md`:

```
git add README.md
```

- Revisar el estado del proyecto:

```
git status
```

- Confirmar los cambios:

```
git commit -m "el archivo README.md está listo"
```

- Revisar el estado del proyecto:

```
git status
```

- Revisar la bitácora del proyecto:

```
git log
```

- Revisar la bitácora (una sola línea por confirmación):

```
git log --oneline
```

4 Ejercicio 2.

Creación de las llaves **SSH** pública y privada para configurar la comunicación entre la computadora y la cuenta en GitHub.

1. Cambiarse al directorio home:

```
cd
```

2. Crear las llaves:

```
ssh-keygen -t ed25519
```

3. Ver la llave pública:

```
cat .ssh/id_ed25519.pub
```

4. Agregar la llave pública en GitHub.
5. Verificar que todo esté correcto:

```
ssh -T git@github.com
```

5 Ejercicio 3.

Crear un repositorio en tu cuenta de GitHub con el nombre `new_proy`.

6 Ejercicio 4.

Sincronización del repo local `new_proy` con GitHub.

1. En el repositorio local, agregar la dirección del repositorio remoto en GitHub:

```
git remote add origin git@github.com:tu_usuario/new_proy.git
```

2. Cambiar el nombre de la rama principal:

```
git branch -M main
```

3. Sincronizar los cambios locales con el repo en GitHub:

```
git push -u origin main
```

4. Verificar en el repositorio en GitHub que estén los cambios de la última confirmación.

7 Ejercicio 5.

Clonar el repositorio `new_proy` en el equipo **pc1** desde GitHub.

1. Clonar usando el protocolo SSH.

```
git clone git@github.com:tu_usuario/new_proy.git
```

8 Ejercicio 5.

Clonar el repositorio `new_proy` en el equipo **pc1** desde GitHub.

1. Clonar usando el protocolo SSH.

```
git clone git@github.com:tu_usuario/new_proy.git
```

Este comando creará un directorio de nombre `new_proy`.

2. Listar los archivos:

```
ls -l
```

3. Cambiarse a `new_proy`:

```
cd new_proy
```

4. Checar la dirección del repositorio remoto:

```
git remote -v
```

5. Checar la bitácora (historial de confirmaciones):

```
git log --oneline
```

9 Ejercicio 6.

Modificación de archivos.

1. Modificar el archivo `README.md` en el equipo **pc1** de tal manera que ahora tenga el siguiente contenido:

```
Este proyecto es una prueba de **Git** y **GitHub**.
* introducción
* descripción de git y github
* ejemplos
```

2. Revisar el estado del proyecto:

```
git status
```

3. Preparar el archivo `README.md`:

```
git add README.md
```

4. Confirmar los cambios:

```
git commit -m "Agregamos el índice en el archivo README.md"
```

5. Subir los cambios a GitHub:

```
git push
```

6. Checar los cambios en GitHub.

10 Ejercicio 7.

1. Bajar los cambios del repositorio `new_proy` en el equipo **mac1**:

```
git pull
```

2. Checar el estado del repositorio:

```
git status
```

3. Revisar el contenido del archivo `README.md`:

```
cat README.md
```

4. Revisar la bitácora.

```
git log
```

11 Ejercicio 8.

1. Modificar el archivo `README.md` en el equipo **pc1** de tal manera que ahora tenga el siguiente contenido

```
Este proyecto es una prueba de **Git** y **GitHub**.
- introducción
- descripción de git y github
- ejemplos

Ecuación de calor:

$$
\dfrac{\partial T}{\partial t}-\alpha \nabla^{2} T=0
$$
```

2. Actualizar los cambios y subirlos a GitHub.

```
git add README.md
git commit -m "Agregué la ecuación de calor en el README.md"
git push
```

3. Checar la bitácora:

```
git log --oneline
```

4. Checar los cambios en GitHub.

12 Ejercicio 9.

El repositorio local en el equipo **mac1** contiene solo dos confirmaciones, pues no ha sido actualizado:

```
$ git log --oneline
ce5d5c9 (HEAD -> main, origin/main) Agregamos descripción en README.md
9b0faa0 el archivo README.md está listo
```

Sin actualizar el repositorio en el equipo **mac1**, realizar lo siguiente:

1. Modificar el archivo `README.md` en el equipo **mac1**, de tal manera que ahora contenga lo siguiente:

```
Este proyecto es una prueba de **Git** y **GitHub**.
- introducción
- descripción de git y github
- ejemplos

Diferencias finitas:

$$
\dfrac{\partial^2 T}{\partial x} \approx \dfrac{T_{i-1} - 2 T_i + T_{i+1}}{\Delta x}
$$
```

2. Actualizar los cambios y subirlos a GitHub.

```
git add README.md
git commit -m "Agregué la fórmula de diferencias finitas en el README.md"
git push
```

13 Ejercicio 10.

Para resolver el error obtenido al final del ejercicio 9, primero debemos actualizar el repositorio local con la última versión que se tiene en GitHub, antes de intentar subir los cambios desde **mac1**.

1. Actualiza el repositorio local de **mac1** como sigue:

```
git pull --rebase origin main
```

- Este comando actualiza la rama principal, como está en GitHub.
- Después, desde ese estado de la rama principal, va agregando las confirmaciones locales una por una.
- Si encuentra un conflicto nos avisa para que lo resolvamos.

2. Resolver el conflicto localmente en el equipo **mac1**. Para ello editamos el archivo con el conflicto y lo corregimos:

```
Este proyecto es una prueba de **Git** y **GitHub**.
- introducción
- descripción de git y github
- ejemplos

<<<<<<< HEAD
Ecuación de calor:

$$
\dfrac{\partial T}{\partial t}-\alpha \nabla^{2} T=0
$$
=======
Diferencias finitas:

$$
\dfrac{\partial^2 T}{\partial x} \approx
\dfrac{T_{i-1} - 2 T_i + T_{i+1}}{\Delta x}
$$
>>>>>>> fd92909 (Agregué la fórmula de diferencias finitas en el README.md)
```

- El archivo en conflicto contiene la etiqueta `<<<<<<< HEAD` que indica el inicio del conflicto.
- A partir de esa línea, muestra el contenido de lo que tiene la rama principal, y termina cuando se encuentra con la etiqueta: `=======`
- A partir de esta línea se muestra el contenido de lo que está en conflicto en la confirmación `fd92909`, que son los cambios locales.
- Este contenido termina cuando se encuentra la etiqueta `>>>>>>> fd92909`
- **Lo que debemos hacer es definir que va y que no va en el archivo `README.md` de estas dos secciones. Al final, se deben eliminar las tres líneas con las etiquetas que muestran el conflicto y guardar el archivo.**
- En este caso solo eliminamos las líneas con las etiquetas y guardamos el contenido del archivo. Esto significa que ambos contenidos son correctos y van en el archivo `README.md`

3. Una vez modificado el archivo, se debe preparar con:

```
git add README.md
```

4. Posteriormente, para continuar con el proceso se hace:

```
git rebase --continue
```

Esto abrirá un editor donde se debe poner un texto describiendo cómo se resolvió el problema.

5. Checar el estado del repositorio.

```
git status
```

6. Checar la bitácora.

```
git log --oneline
```

7. Subir los cambios a GitHub.

```
git push
```

14 Ejercicio 11.

1. Actualizar el repositorio en el equipo **pc1**.

```
git pull
```

2. Revisar la bitácora.

```
git log --oneline
```

3. Checar el contenido del archivo README.md.

```
cat README.md
```