

Ayelen Etchegoyen

Práctico 2: Git y GitHub

Actividades

1) Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas) :

- ¿Qué es GitHub?

GitHub es una plataforma para alojar repositorios y controlar sus versiones utilizando Git.

- ¿Cómo crear un repositorio en GitHub?

- Empleando el comando `git init`
- Crear el archivo que quieres subir al repositorio.
- Ejecutar `git add .` y `git commit -m "mensaje"` (completar mensaje descriptivo).
- Crear y loguearse en tu cuenta de GitHub.
- Seleccionar en el botón "+" y luego "New repository".
- Colocar un nombre y si es público o privado.
- Click en "create repository"
- Ejecutar el comando proveniente de GitHub:

```
git remote add origin https://github.com/...
git push -u origin main
```

- ¿Cómo crear una rama en Git?

Ejecutando el comando `git branch nombre-rama`

- ¿Cómo cambiar a una rama en Git?

Ejecutando el comando `git checkout nombre-rama`

- ¿Cómo fusionar ramas en Git?

Dirigirse a la rama en la que quieres fusionar los cambios: `git checkout main`

Ejecutar el comando: `git merge nombre-rama`

- ¿Cómo crear un commit en Git?

```
git commit -m "mensaje"
```

- ¿Cómo enviar un commit a GitHub?

```
git push origin nombre-rama
```

- ¿Qué es un repositorio remoto?

Es una carpeta con archivos de un proyecto que está alojada en GitHub (o similar).

- ¿Cómo agregar un repositorio remoto a Git?

- Empleando el comando `git init`
- Crear el archivo que quieres subir al repositorio.

- Ejecutar `git add .` y `git commit -m "mensaje"` (completar mensaje descriptivo).

- ¿Cómo empujar cambios a un repositorio remoto?

Utilizando `git push origin nombre-rama`

- ¿Cómo tirar de cambios de un repositorio remoto?

Utilizando `git pull`

- ¿Qué es un fork de repositorio?

Realizar una copia de un repositorio externo a tu cuenta.

- ¿Cómo crear un fork de un repositorio?

Dirigirse al proyecto deseado.

Hacer click en Fork.

- ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?

Se debe crear una rama en el repositorio, subir los cambios y hacer push.

```
git branch nombre-rama
```

```
git checkout nombre-rama
```

```
git add .
```

```
git commit -m "mi pull request"
```

```
git push origin nombre-rama
```

- ¿Cómo aceptar una solicitud de extracción?

Haciendo merge de la solicitud:

```
git merge nombre-rama
```

- ¿Qué es una etiqueta en Git?

Es una forma de nombrar un commit, generalmente utilizada para marcar un cambio importante en el proyecto.

- ¿Cómo crear una etiqueta en Git?

```
git tag nombre-etiqueta
```

- ¿Cómo enviar una etiqueta a GitHub?

```
git push origin nombre-etiqueta
```

- ¿Qué es un historial de Git?

Es el registro de los cambios que se hicieron y toda la información relacionada.

- ¿Cómo ver el historial de Git?

```
git log
```

- ¿Cómo buscar en el historial de Git?

```
git log --grep="palabra clave"
```

- ¿Cómo borrar el historial de Git?

```
git reset
```

- ¿Qué es un repositorio privado en GitHub?

Es un repositorio al que solo pueden acceder los usuarios invitados.

- ¿Cómo crear un repositorio privado en GitHub?

En el momento de crear el repositorio se debe elegir la opción de privado. Esto se encuentra debajo del nombre del repositorio.

- ¿Cómo invitar a alguien a un repositorio privado en GitHub?

Dirigirse a la configuración del repositorio y añadir colaboradores (Collaborators) con el botón: 'Add people'.

- ¿Qué es un repositorio público en GitHub?

Es un repositorio que puede ser visitado por cualquier usuario.

- ¿Cómo crear un repositorio público en GitHub?

En el momento de crear el repositorio se debe elegir la opción de público. Esto se encuentra debajo del nombre del repositorio.

- ¿Cómo compartir un repositorio público en GitHub?

Compartir el link del repositorio.

2) Realizar la siguiente actividad:

- Crear un repositorio.
 - o Dale un nombre al repositorio.
 - o Elige el repositorio sea público.
 - o Inicializa el repositorio con un archivo.

<https://github.com/ayeetche/TP2-nuevo-repo>

- Creando Branchs
 - o Crear una Branch
 - o Realizar cambios o agregar un archivo
 - o Subir la Branch

<https://github.com/ayeetche/TP2-nuevo-repo/tree/rama>

3) Realizar la siguiente actividad:

Paso 1: Crear un repositorio en GitHub

- Ve a GitHub e inicia sesión en tu cuenta.
- Haz clic en el botón "New" o "Create repository" para crear un nuevo repositorio.
- Asigna un nombre al repositorio, por ejemplo, conflict-exercise.
- Opcionalmente, añade una descripción.
- Marca la opción "Initialize this repository with a README".
- Haz clic en "Create repository".

Paso 2: Clonar el repositorio a tu máquina local

- Copia la URL del repositorio (usualmente algo como `https://github.com/tuusuario/conflict-exercise.git`).
- Abre la terminal o línea de comandos en tu máquina.
- Clona el repositorio usando el comando:

```
git clone https://github.com/tuusuario/conflict-exercise.git
```

- Entra en el directorio del repositorio:

```
cd conflict-exercise
```

Paso 3: Crear una nueva rama y editar un archivo

- Crea una nueva rama llamada feature-branch: `git checkout -b feature-branch`
- Abre el archivo README.md en un editor de texto y añade una línea nueva, por ejemplo: Este es un cambio en la feature branch.

- Guarda los cambios y haz un commit:

```
git add README.md
```

```
git commit -m "Added a line in feature-branch"
```

<https://github.com/ayeetche/TP2-nuevo-repo2/commits/feature-branch>

Paso 4: Volver a la rama principal y editar el mismo archivo

- Cambia de vuelta a la rama principal (main):

```
git checkout main
```

- Edita el archivo README.md de nuevo, añadiendo una línea diferente: Este es un cambio en la main branch.

- Guarda los cambios y haz un commit:

```
git add README.md
```

```
git commit -m "Added a line in main branch"
```

Paso 5: Hacer un merge y generar un conflicto

- Intenta hacer un merge de la feature-branch en la rama main:

```
git merge feature-branch
```

- Se generará un conflicto porque ambos cambios afectan la misma línea del archivo README.md.

Paso 6: Resolver el conflicto

- Abre el archivo README.md en tu editor de texto. Verás algo similar a esto:

```
<<<<<<< HEAD
```

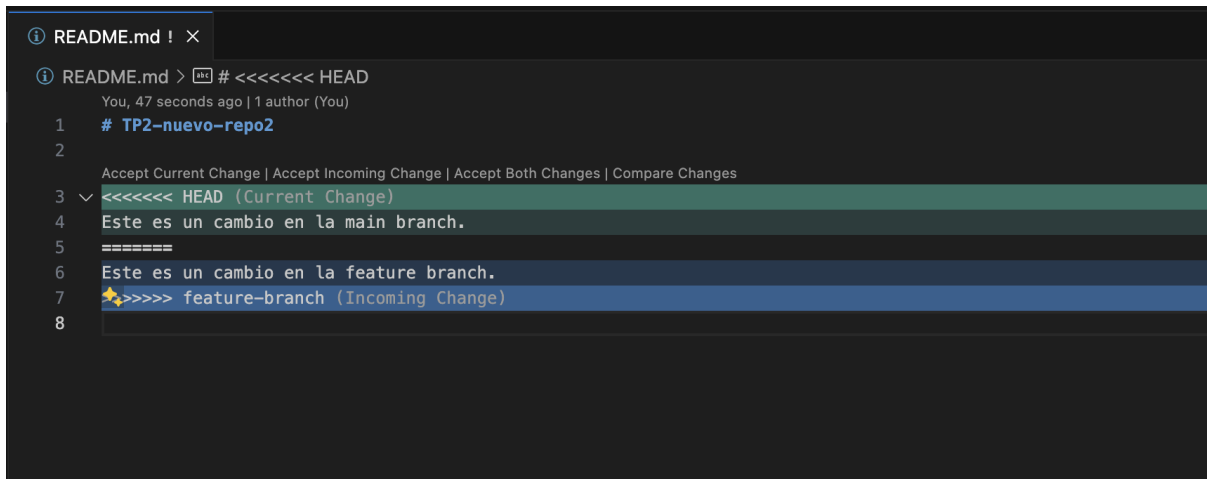
Este es un cambio en la main branch

```
=====
```

Este es un cambio en la feature branch.

```
>>>>>>> feature-branch
```

- Decide cómo resolver el conflicto. Puedes mantener ambos cambios, elegir uno de ellos, o fusionar los contenidos de alguna manera.



```
1 # TP2-nuevo-repo2
2
3 <<<<<< HEAD (Current Change)
4 Este es un cambio en la main branch.
5 =====
6 Este es un cambio en la feature branch.
7 >>>>> feature-branch (Incoming Change)
```

- Edita el archivo para resolver el conflicto y guarda los cambios(Se debe borrar lo marcado en verde en el archivo donde estes solucionando el conflicto. Y se debe borrar la parte del texto que no se quiera dejar).

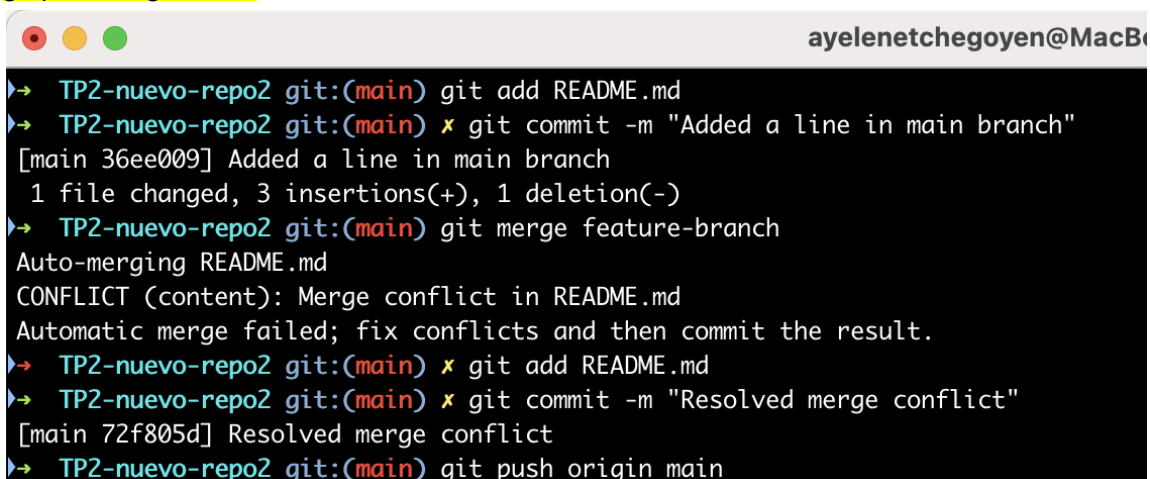
- Añade el archivo resuelto y completa el merge:

```
git add README.md
```

```
git commit -m "Resolved merge conflict"
```

Paso 7: Subir los cambios a GitHub • Sube los cambios de la rama main al repositorio remoto en GitHub:

```
git push origin main
```



```
TP2-nuevo-repo2 git:(main) git add README.md
TP2-nuevo-repo2 git:(main) x git commit -m "Added a line in main branch"
[main 36ee009] Added a line in main branch
1 file changed, 3 insertions(+), 1 deletion(-)
TP2-nuevo-repo2 git:(main) git merge feature-branch
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
TP2-nuevo-repo2 git:(main) x git add README.md
TP2-nuevo-repo2 git:(main) x git commit -m "Resolved merge conflict"
[main 72f805d] Resolved merge conflict
TP2-nuevo-repo2 git:(main) git push origin main
```

- También sube la feature-branch si deseas:

```
git push origin feature-branch
```

Paso 8: Verificar en GitHub

- Ve a tu repositorio en GitHub y revisa el archivo README.md para confirmar que los cambios se han subido correctamente.
- Puedes revisar el historial de commits para ver el conflicto y su resolución.

<https://github.com/ayeetche/TP2-nuevo-repo2/commits/main>