

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 de alojamiento de código basada en Git que permite a los desarrolladores colaborar en proyectos, gestionar repositorios, realizar seguimiento de issues y revisar código mediante pull requests.

- ¿Cómo crear un repositorio en GitHub?
 1. Inicia sesión en GitHub.
 2. Hacer clic en "**Create new**" en la esquina superior derecha.
 3. Asignarle un nombre al repositorio.
 4. Elegir si es público o privado
 5. Hacer clic en "**Create repository**".

- ¿Cómo crear una rama en Git?

Mediante el comando **git checkout -b nombre-rama** en la consola de git ubicada en el directorio del repositorio

- ¿Cómo cambiar a una rama en Git?

git checkout nombre-rama

- ¿Cómo fusionar ramas en Git?

Una vez que se está en la rama main utilizar el comando **git merge nombre-rama**

- ¿Cómo crear un commit en Git?

git commit -m "Mensaje descriptivo"

- ¿Cómo enviar un commit a GitHub?

git push origin nombre-rama

- ¿Qué es un repositorio remoto?

Los repositorios remotos son versiones de tu proyecto que están hospedadas en Internet o en cualquier otra red.

- ¿Cómo agregar un repositorio remoto a Git?

Para agregar un repositorio remoto nuevo, use el comando `git remote add` en el terminal, dentro del directorio donde está almacenado su repositorio.

`git remote add nombre-remoto url-repositorio`

- ¿Cómo empujar cambios a un repositorio remoto?

`git push -u origin nombre-rama`

- ¿Cómo tirar de cambios de un repositorio remoto?

`git pull origin nombre-rama`

- ¿Qué es un fork de repositorio?

Es una copia personal de un repositorio ajeno en tu cuenta de GitHub, para modificarlo sin afectar el original.

- ¿Cómo crear un fork de un repositorio?

1. Ir al repositorio que se desea forkear
2. Seleccionar la opción de fork el repositorio
3. Darle un nombre y descripción al fork
4. Presionar el botón de crear fork

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

Estar en el repositorio en github, ir a la pestaña "Pull requests" y haz clic en "New pull request".

- ¿Cómo aceptar una solicitud de extracción?

Vas al pull request y si los cambios son correctos, hacer click en "Merge pull request"

- ¿Qué es un etiqueta en Git?

Las etiquetas son referencias que apuntan a puntos concretos en el historial de Git.

- ¿Cómo crear una etiqueta en Git?

git tag -a mi-etiqueta -m "Versión mi-etiqueta"

- ¿Cómo enviar una etiqueta a GitHub?

git push origin mi-etiqueta

- ¿Qué es un historial de Git?

Es el registro de todos los commits realizados en un repositorio, mostrando autores, fechas y cambios.

- ¿Cómo ver el historial de Git?

git log

- ¿Cómo buscar en el historial de Git?

Para buscar en el historial de Git, puedes utilizar los comandos git log, git grep, git blame

- ¿Cómo borrar el historial de Git?

Localmente se puede borrar el historial de Git eliminando la carpeta oculta .git

- ¿Qué es un repositorio privado en GitHub?

Es un repositorio solo accesible para colaboradores autorizados.

- ¿Cómo crear un repositorio privado en GitHub?

Al momento de crear el repositorio, seleccionar la opción "Private" en lugar de "Public".

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

Necesitas agregarlo como colaborador mediante su usuario de Github.

- ¿Qué es un repositorio público en GitHub?

Un repositorio visible para todos, pero solo editable por colaboradores.

- ¿Cómo crear un repositorio público en GitHub?

Al momento de crear el repositorio, dejar seleccionada la opción "Public".

- ¿Cómo compartir un repositorio público en GitHub?

A través de la url ya se puede compartir un repositorio público.

2) Realizar la siguiente actividad:

- Crear un repositorio
- Agregando un Archivo
- Creando Branchs

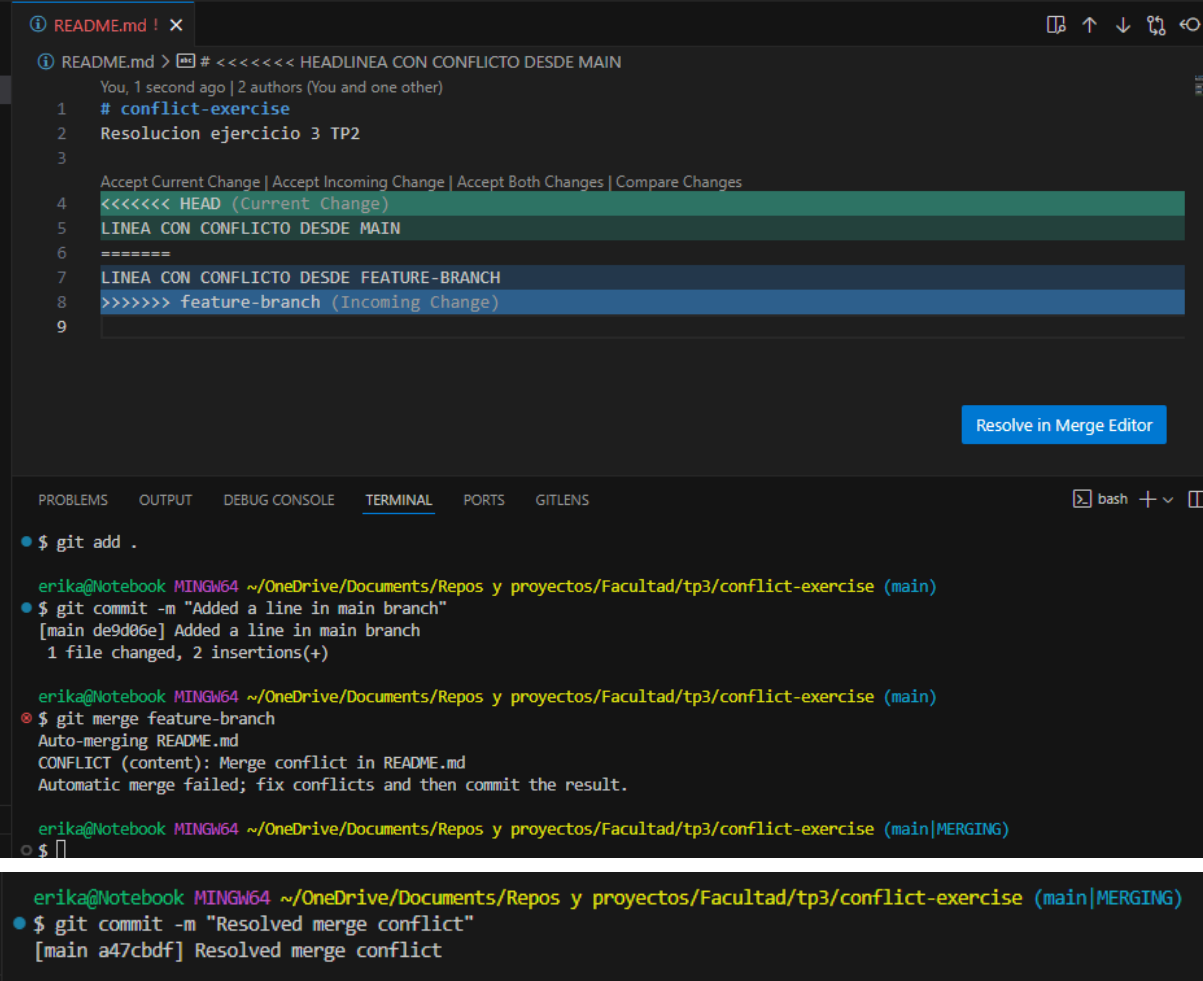
<https://github.com/samgnz/TP2EJ2>

3) Realizar la siguiente actividad

Paso 1: Crear un repositorio en GitHub

<https://github.com/samgnz/conflict-exercise>

Paso 5: Hacer un merge y generar un conflicto



The image shows a Visual Studio Code editor window with a file named `README.md` open. The file content is as follows:

```
1 # conflict-exercise
2 Resolución ejercicio 3 TP2
3
4 Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
5 <<<<<< HEAD (Current Change)
6 LINEA CON CONFLICTO DESDE MAIN
7 =====
8 LINEA CON CONFLICTO DESDE FEATURE-BRANCH
9 >>>>>> feature-branch (Incoming Change)
```

Below the editor, the `TERMINAL` tab is active, showing the following commands and output:

```
$ git add .
erika@Notebook MINGW64 ~/OneDrive/Documents/Repos y proyectos/Facultad/tp3/conflict-exercise (main)
$ git commit -m "Added a line in main branch"
[main de9d06e] Added a line in main branch
1 file changed, 2 insertions(+)

erika@Notebook MINGW64 ~/OneDrive/Documents/Repos y proyectos/Facultad/tp3/conflict-exercise (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.

erika@Notebook MINGW64 ~/OneDrive/Documents/Repos y proyectos/Facultad/tp3/conflict-exercise (main|MERGING)
$
```

Below the terminal, a new code block shows the final steps of the merge process:

```
erika@Notebook MINGW64 ~/OneDrive/Documents/Repos y proyectos/Facultad/tp3/conflict-exercise (main|MERGING)
$ git commit -m "Resolved merge conflict"
[main a47cbdf] Resolved merge conflict
```