

TRABAJO PRACTICO 2: GIT Y GIT HUB

Alumno: Cristian Ayala. Comision 24.

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

- ¿Qué es GitHub?

Git hub es una plataforma diseñada para crear repositorios de código mediante los cuales se pueden guardar proyectos en la nube, acceder a ellos, sobrescribirlos, y trabajar en equipo. También se pueden gestionar las diferentes versiones de tu proyecto con Git.

- ¿Cómo crear un repositorio en GitHub?

Debes clickear en nuevo repositorio, y configurar el nombre del proyecto y si quieres que sea público o privado. Luego debes conectarlo con tu proyecto local, mediante el link de conexión remota.

- ¿Cómo crear una rama en Git?

Mediante el comando "Git branch nombre_rama".

- ¿Cómo cambiar a una rama en Git?

Mediante el comando "git checkout

- ¿Cómo fusionar ramas en Git?

Mediante el comando "git merge nombre_rama" posicionado en la rama que quieres que adopte los nuevos cambios. El nombre de la rama que se especifica en el comando es la que se fusionara trayendo los nuevos cambios.

- ¿Cómo crear un commit en Git?

Git commit -m "nombre del commit".

- ¿Cómo enviar un commit a GitHub?

Con el comando git push origin "nombre de la rama", en el mayor de los casos es "main" o "master".

- ¿Qué es un repositorio remoto? Es el lugar que albergara nuestro proyecto en la nube. Es el que podemos crear mediante git hub.

- ¿Cómo agregar un repositorio remoto a Git?

Con el comando "Git remote add origin <link del repositorio que nos aporta github>"

- ¿Cómo empujar cambios a un repositorio remoto?

Mediante el comando git Add . Luego commit y luego push.

- ¿Cómo tirar de cambios de un repositorio remoto?

Mediante el comando "Git pull origin main" para obtener las ultimas actualizaciones.

- ¿Qué es un fork de repositorio?

Es hacer una copia exacta de un repositorio ajeno en git hub. Aparecerá copiado en tus repositorios.

- ¿Cómo crear un fork de un repositorio?

Primero se debe ir al repositorio que quieres forquear para presionar "Fork". Luego el repositorio aparecerá copiado en tu cuenta. Luego clonas el repositorio de github a un proyecto local.

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

Ir a Git hub, en tu fork click en "Contribute" y "Open Pull Request". Hay que escribir un titulo y descripción explicando los cambios y luego click en "Create Pull Request".

- ¿Cómo aceptar una solicitud de extracción?

Ir a la pestaña "Pull Requests" en tu repositorio. Abrir la solicitud que voy a revisar mediante "Files changed". Si todo va bien click en "Merge Pull Request" y "Confirm merge".

- ¿Qué es un etiqueta en Git?

Es un marcador en el historial de Git. Se usa generalmente para indicar versiones importantes.

- ¿Cómo crear una etiqueta en Git?

```
"" Git tag -a nombre_etiqueta -m "Versión 1.0" ""
```

Se usa -a para crear la etiqueta, y -m para agregarle un mensaje.

Ejemplo de tag en un commit específico:

```
"" git tag -a v1.0 hash-del-commit -m "Versión 1.0" ""
```

- ¿Cómo enviar una etiqueta a GitHub?

Mediante el comando git push origin "nombre_etiqueta".

- ¿Qué es un historial de Git?

Es el registro histórico de todos los cambios y commits realizados.

- ¿Cómo ver el historial de Git?

Mediante el comando git log.

- ¿Cómo buscar en el historial de Git?

Mediante el comando `git log --grep="texto a buscar"`

- ¿Cómo borrar el historial de Git?

`rm -rf .git`

`git init`

- ¿Qué es un repositorio privado en GitHub?

Un repositorio privado es un repo que solo uno y los colaboradores invitados pueden ver.

- ¿Cómo crear un repositorio privado en GitHub?

De la misma manera que uno publico solo que pones la opción para que sea privado. Esta pregunta la conteste en la segunda pregunta.

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

Ir al repositorio en cuestión y hacer click en settings. Luego Manage Access. Luego Invite a collaborator. Escribir los datos del colaborador y enviar la invitación.

- ¿Qué es un repositorio público en GitHub?

Un repositorio público en GitHub es un proyecto de código que cualquier persona puede ver, clonar y descargar. Sin embargo, solo los colaboradores autorizados pueden hacer cambios directos.

- ¿Cómo crear un repositorio público en GitHub?

Misma respuesta que en pregunta Dos.

- ¿Cómo compartir un repositorio público en GitHub?

Mediante el link que aporta github al crear el repositorio.

2) Realizar la siguiente actividad:

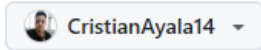
Cree el repositorio remoto.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *



Repository name *

/ tp2

✔ tp2 is available.

Great repository names are short and memorable. Need inspiration? How about [scaling-octo-guide](#) ?

Description (optional)

Creacion de repositorio para ejercicio 2 TP2



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

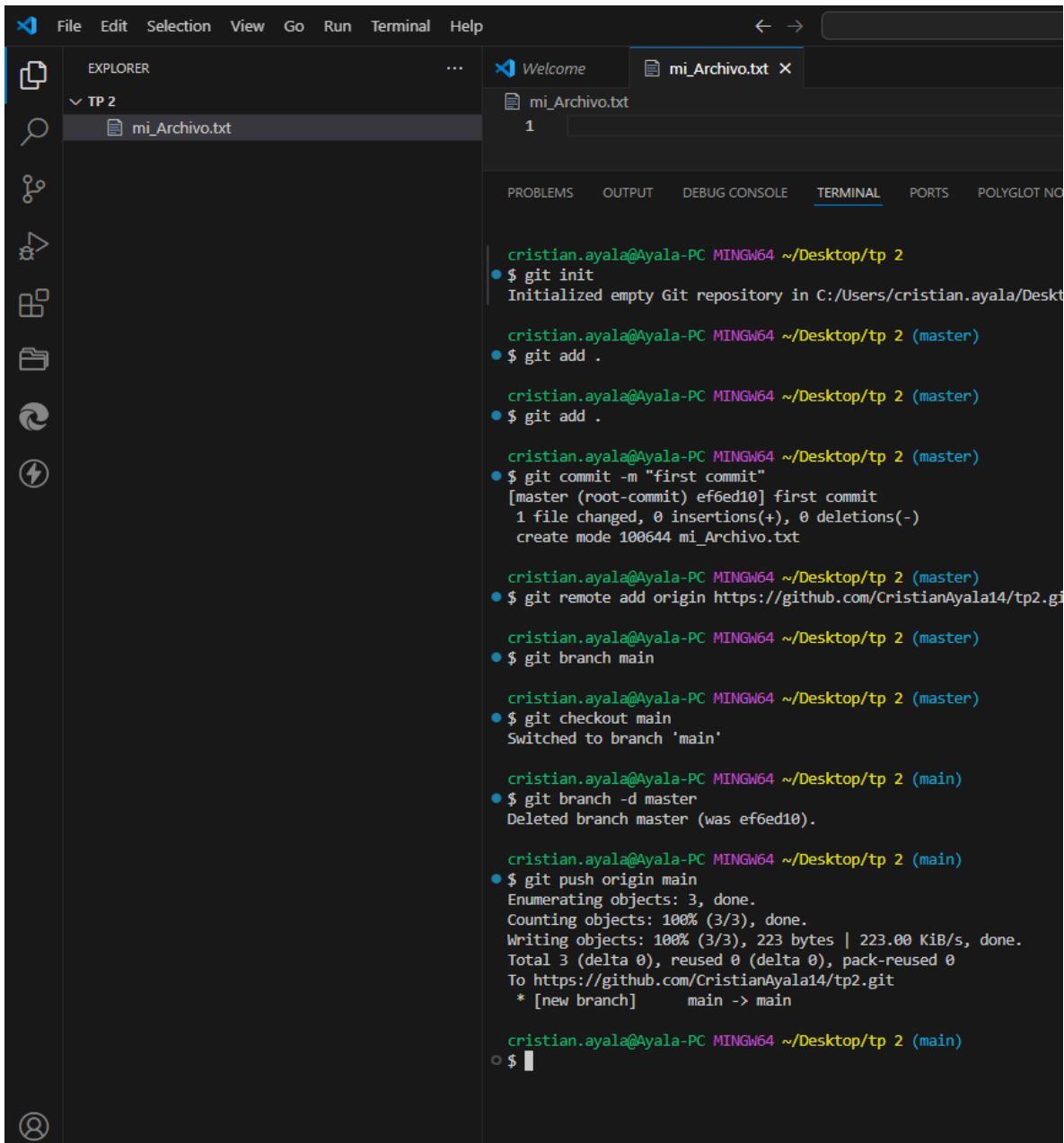
You choose who can see and commit to this repository.

Initialize this repository with:



Add a README file

Mediante el link aportado por git hub luego de la creación del repositorio, conecte con un proyecto local en visual studio code, y con comandos git cree una branch, agregue actualización, hice commit y agregue mi_Archivo.txt al repositorio remoto.



The image shows a screenshot of the Visual Studio Code interface. On the left, the Explorer view shows a file named `mi_Archivo.txt` under a folder named `TP 2`. The main editor area displays the content of `mi_Archivo.txt`, which contains the number `1`. Below the editor, the Terminal panel is open, showing a series of git commands and their output. The commands are executed in a Windows command prompt (MINGW64) at the path `~/Desktop/tp 2`. The output shows the initialization of a git repository, adding the file, committing it with the message "first commit", adding a remote repository, creating a new branch named `main`, checking out `main`, deleting the `master` branch, and pushing the `main` branch to the remote repository.

```
cristian.ayala@Ayala-PC MINGW64 ~/Desktop/tp 2
• $ git init
Initialized empty Git repository in C:/Users/cristian.ayala/Desktop/tp 2/.git/

cristian.ayala@Ayala-PC MINGW64 ~/Desktop/tp 2 (master)
• $ git add .

cristian.ayala@Ayala-PC MINGW64 ~/Desktop/tp 2 (master)
• $ git add .

cristian.ayala@Ayala-PC MINGW64 ~/Desktop/tp 2 (master)
• $ git commit -m "first commit"
[master (root-commit) ef6ed10] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 mi_Archivo.txt

cristian.ayala@Ayala-PC MINGW64 ~/Desktop/tp 2 (master)
• $ git remote add origin https://github.com/CristianAyala14/tp2.git

cristian.ayala@Ayala-PC MINGW64 ~/Desktop/tp 2 (master)
• $ git branch main

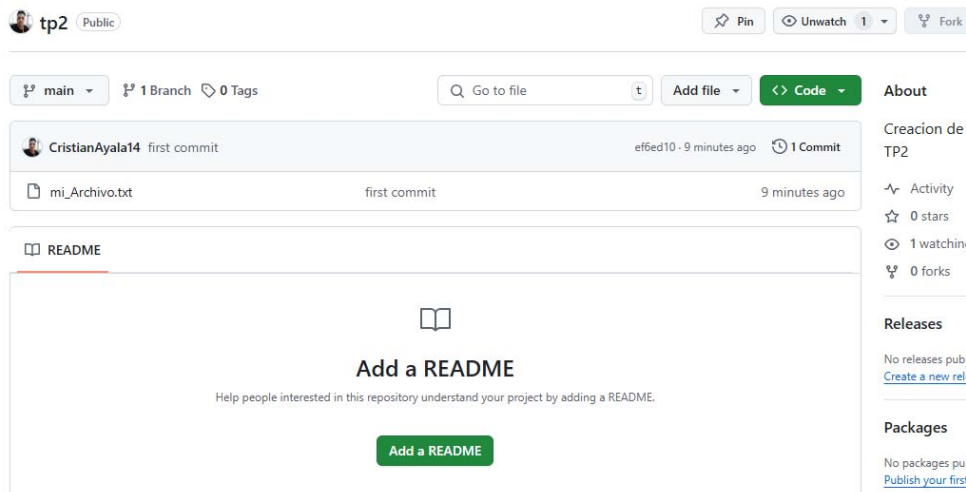
cristian.ayala@Ayala-PC MINGW64 ~/Desktop/tp 2 (master)
• $ git checkout main
Switched to branch 'main'

cristian.ayala@Ayala-PC MINGW64 ~/Desktop/tp 2 (main)
• $ git branch -d master
Deleted branch master (was ef6ed10).

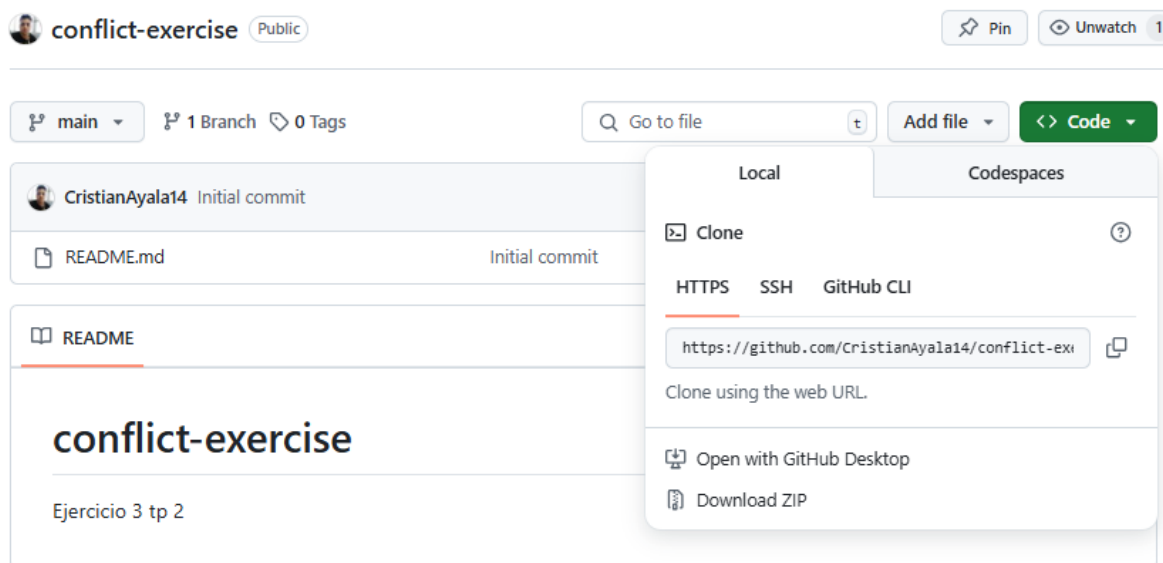
cristian.ayala@Ayala-PC MINGW64 ~/Desktop/tp 2 (main)
• $ git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 223 bytes | 223.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/CristianAyala14/tp2.git
 * [new branch]      main -> main

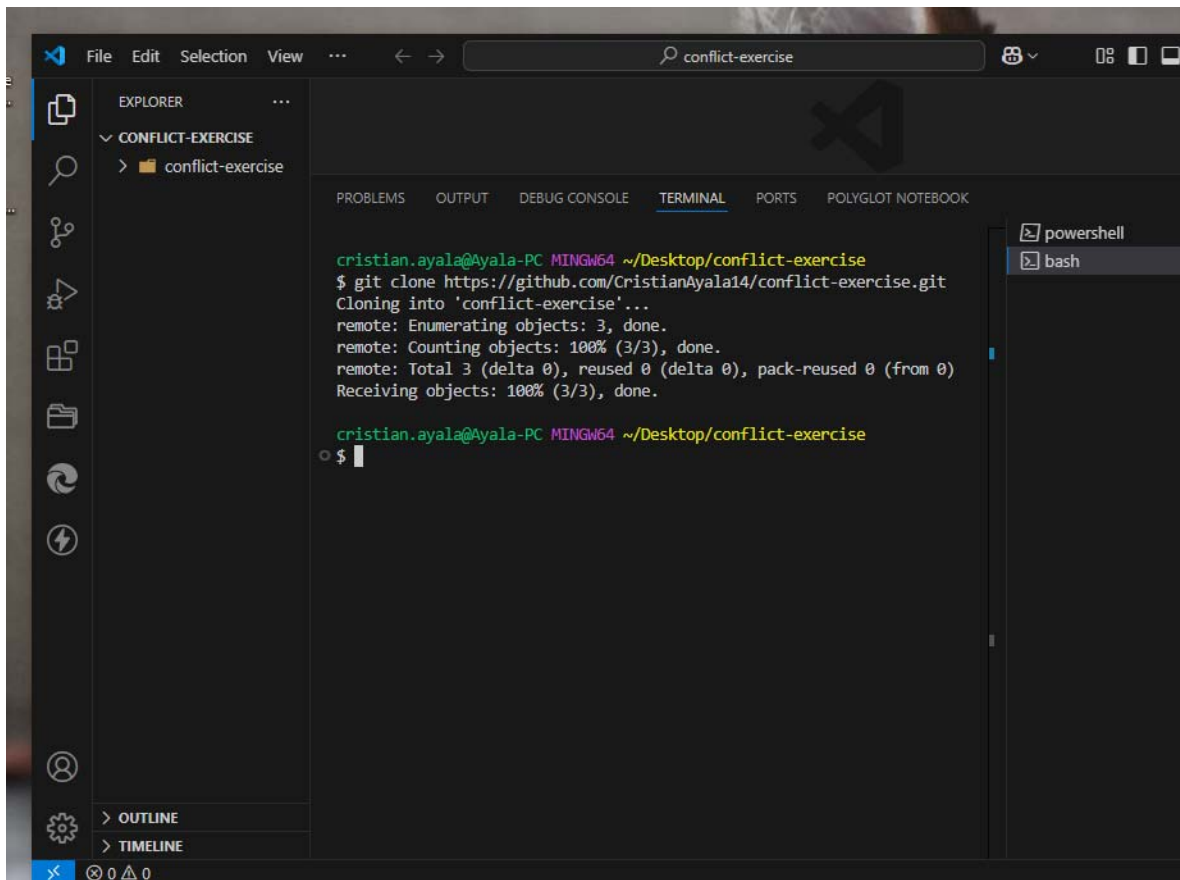
cristian.ayala@Ayala-PC MINGW64 ~/Desktop/tp 2 (main)
• $
```

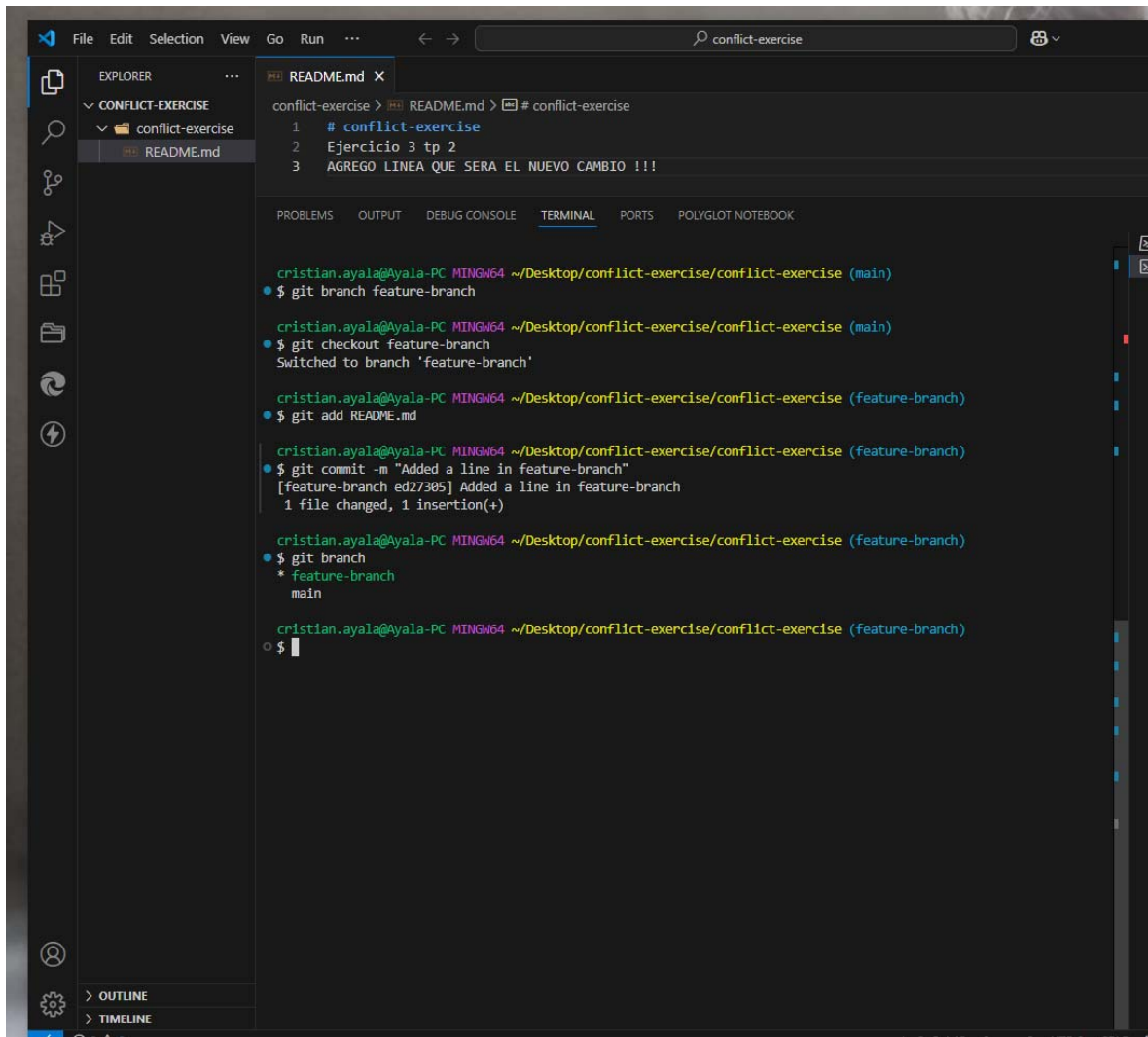
Acá se puede ver como se subió correctamente la actualización al repositorio.

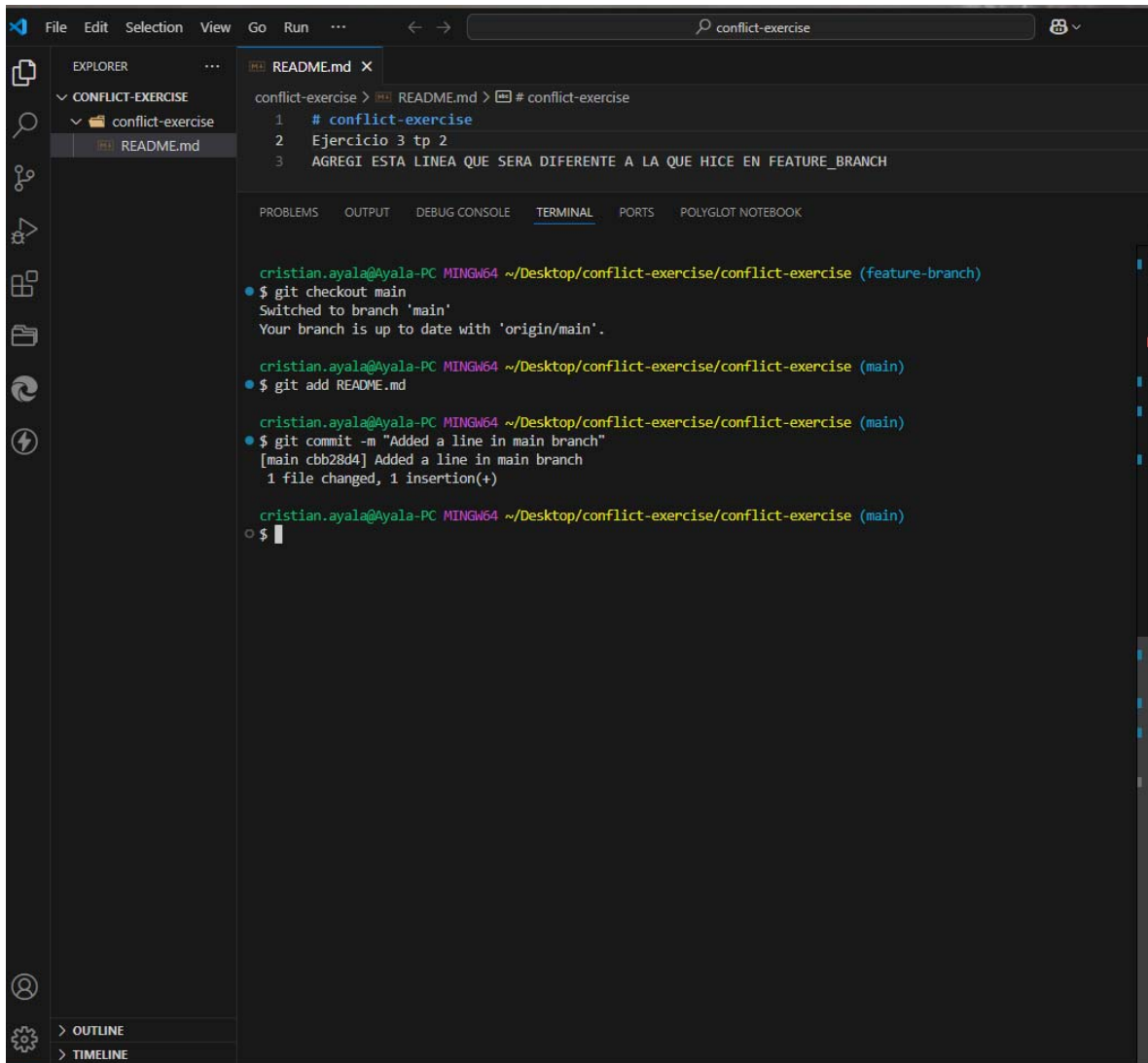


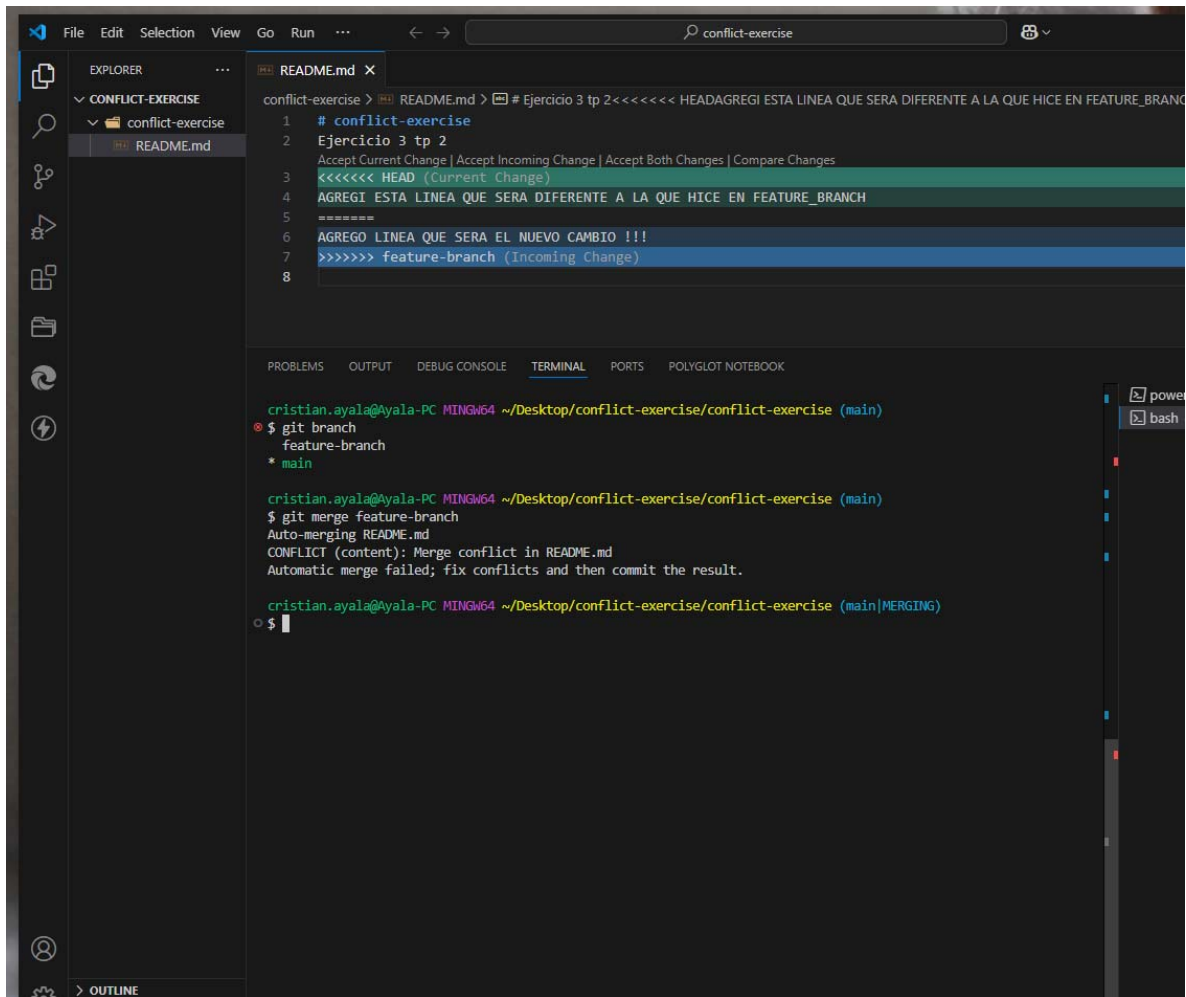
3) Creo repositorio “Conflict-exercise” y lo clono en un repositorio local.

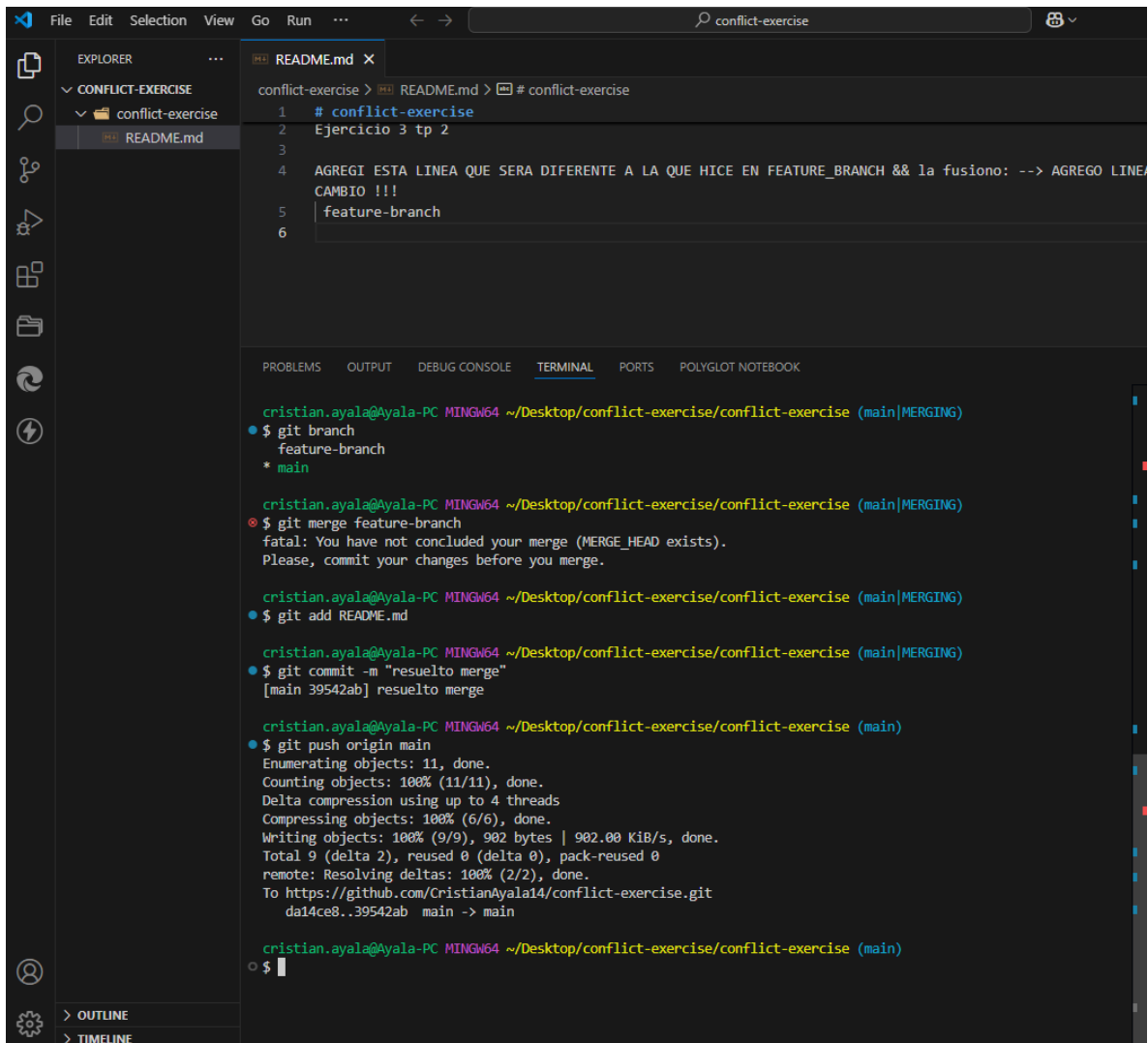


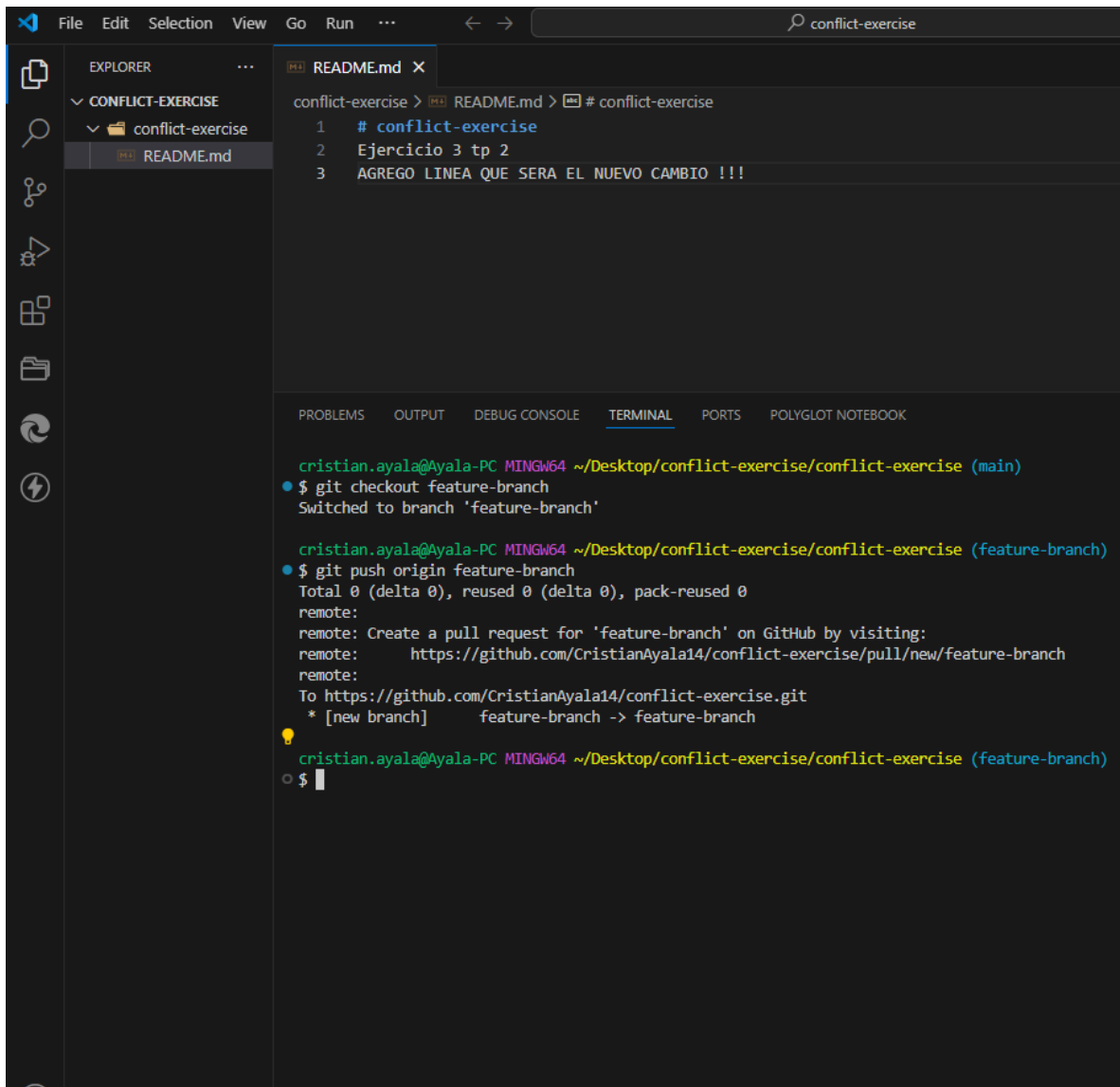
















 main ▾


 2 Branches  0 Tags

 Go to file




Add file ▾

 Code ▾


 **CristianAyala14** resuelto merge

39542ab · 6 minutes ago  4 Commits

 README.md

resuelto merge

6 minutes ago

 README



conflict-exercise

Ejercicio 3 tp 2

AGREGI ESTA LINEA QUE SERA DIFERENTE A LA QUE HICE EN FEATURE_BRANCH && la fusiono: --> AGREGO LINEA QUE SERA EL NUEVO CAMBIO !!! feature-branch