

Actividades

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

- ¿Qué es GitHub?

GitHub es una plataforma basada en la nube donde puedes almacenar, compartir y trabajar junto con otros usuarios para escribir código. Almacenar tu código en un "repositorio" en GitHub te permite lo siguiente: Presentar o compartir el trabajo. Seguir y administrar los cambios en el código a lo largo del tiempo. Estos "repositorios" se crean de manera local en Git. También podemos ver repositorios de otras personas o equipos, guardarlos para posteriormente modificarlos.

- ¿Cómo crear un repositorio en GitHub?

Para crear un repositorio en Git Hub:

1. Debes seleccionar "create new"
2. Elegir la opción "repository"
3. Elegir un nombre y si es público o privado
4. Seleccionar "Create repository"

- ¿Cómo crear una rama en Git?

Para crear una rama en Git se debe usar el comando: `git branch` "el nombre de la rama que deseas crear"

- ¿Cómo cambiar a una rama en Git?

Para cambiar de rama debo ejecutar el comando `git checkout` "el nombre de la rama a la que deseas cambiar"

- ¿Cómo fusionar ramas en Git?

Para funcionar dos ramas de Git debes estar "parado" o trasladarte con el comando `git checkout` a la rama donde quieras que la otra rama se fusione. Luego utilizas el comando: `git merge` "el nombre de la rama que deseas fusionar". Si me coloque en "master" y quiero que se le fusione la rama `ejercicio1`; debo colocar `git merge` "ejercicio1"

- ¿Cómo crear un commit en Git?

Para crear un commit en git debo colocar: `git commit -m` "Descripción del commit hecho"

- ¿Cómo enviar un commit a GitHub?

Se utiliza el comando git push origin “nombre de la rama”

- ¿Qué es un repositorio remoto?

Un repositorio remoto es un repositorio externo a nuestros repositorios locales, que se guarda en la nube, a través de sitios web, como GitHub, sirven para: tener al alcance repositorios a pesar de que cambiemos de lugares de trabajos o equipos, además otorgan la posibilidad de que varias personas desde distintos lugares pueden trabajar en conjunto sobre el mismo repositorio.

- ¿Cómo agregar un repositorio remoto a Git?

Se utiliza el comando git clone “Y el URL del repositorio remoto”

- ¿Cómo empujar cambios a un repositorio remoto?

Se utiliza el comando git push u- origin “nombre de la rama”

- ¿Cómo tirar de cambios de un repositorio remoto?

Git pull origin “nombre de la rama”

- ¿Qué es un fork de repositorio?

Un fork es una herramienta que GitHub nos ofrece para, copiar un repositorio y que se nos guarde en nuestra cuenta de GitHub.

- ¿Cómo crear un fork de un repositorio?

Crear un fork:

1. Dirigirse al repositorio de interés.
2. Seleccionar el botón “fork”
3. Configurar parámetros
4. Seleccionar “Crear fork” para terminar el proceso.

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

1. Ve a tu fork en GitHub.
2. Aparecerá un mensaje para crear un Pull Request (PR).
3. Si no aparece, ve a la pestaña "Pull requests" y haz clic en "New Pull Request".
4. Selecciona las ramas correctas:
5. Base branch: La rama a la que quieres enviar los cambios (ejemplo: main o develop).
6. Compare branch: Tu rama con los cambios (fix-bug-login).
7. Escribe un título y una descripción del PR.
8. Haz clic en "Create Pull Request".

- ¿Cómo aceptar una solicitud de extracción?

- a) Ir al repositorio
- b) Ir a pull requests
- c) Tocar en merge pull requests
- d) Seleccionar la opción que se considere más conveniente para la fusión.

- ¿Qué es una etiqueta en Git?

Una etiqueta en git es como una especie de papel dentro del repositorio donde se pueden anotar cosas que se consideren importantes: cambios fundamentales, bugs, versiones estables, etc. Estas etiquetas no afectan al código, son una buena práctica que nos ayuda a marcar cosas y que estas sean sencillas de buscar luego.

- ¿Cómo crear una etiqueta en Git?

Para crear etiquetas se utiliza git tag "nombre de la etiqueta"

Para dejar un mensaje en la etiqueta git tag -a "nombre de la etiqueta" -m "mensaje"

- ¿Cómo enviar una etiqueta a GitHub?

Para enviar la etiqueta se usa git push "nombre de la etiqueta"

- ¿Qué es un historial de Git?

El historial de Git, es una línea cronológica de todos los cambios realizados en el repositorio, organizado por commits.

- ¿Cómo ver el historial de Git?

Para ver el historial de Git, se utiliza Git log

- ¿Cómo buscar en el historial de Git?

Para ver los commits graficando: `Git log --decorate --all --graph --oneline`

Para buscar por palabra: `git log --grep="palabra clave"`

Para buscar por fecha: `git log --since="fecha estimada"`

- ¿Cómo borrar el historial de Git?

Se utiliza: `rm -rf .git`

- ¿Qué es un repositorio privado en GitHub?

Un repositorio privado en GitHub es un repositorio cuyo acceso está restringido a un grupo específico de personas. Solo los usuarios a quienes se les haya otorgado permiso pueden ver, clonar o colaborar en el proyecto. Esto es útil para proyectos privados, sensibles o en desarrollo.

- ¿Cómo crear un repositorio privado en GitHub?

- a) En la esquina superior derecha, haz clic en el botón "+" y selecciona "New repository".
- b) Rellena los campos de nombre, descripción.
- c) Visibility: Selecciona "Private" para que el repositorio sea privado.
- d) Haz clic en "Create repository".

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

1. Ve a tu repositorio privado en GitHub.
2. Haz clic en la pestaña "Settings" (Configuración) del repositorio.
3. En el menú de la izquierda, selecciona "Manage access" (Gestionar acceso).
4. Haz clic en "Invite a collaborator" (Invitar a un colaborador).
5. En el campo "Search by username, full name or email", ingresa el nombre de usuario de GitHub, nombre completo o correo electrónico de la persona que deseas invitar.
6. Selecciona el usuario correcto de los resultados de búsqueda.
7. Haz clic en "Add" (Agregar) para enviar la invitación.

- ¿Qué es un repositorio público en GitHub?

Un repositorio público en GitHub es un repositorio cuyo código y archivos están disponibles para que cualquier persona en Internet los vea, descargue y contribuya. Es la opción ideal para proyectos de código abierto, donde cualquier persona puede colaborar o revisar el código.

- ¿Cómo crear un repositorio público en GitHub?

1. Haz clic en el botón "+" en la esquina superior derecha y selecciona "New repository".

2. Rellena los siguientes campos: nombre, descripción
3. Visibility: Selecciona "Public" para hacer el repositorio público.
4. Haz clic en "Create repository".

- ¿Cómo compartir un repositorio público en GitHub?

1. Abre el repositorio.
2. Copia el link, tocando code, luego copia el que esta en https
3. Compartelo.

2) Realizar la siguiente actividad:

- Crear un repositorio.
 - Dale un nombre al repositorio.
 - Elige el repositorio sea público.
 - Inicializa el repositorio con un archivo.
- Agregando un Archivo
 - Crea un archivo simple, por ejemplo, "mi-archivo.txt".
 - Realiza los comandos `git add .` y `git commit -m "Agregando mi-archivo.txt"` en la línea de comandos.
 - Sube los cambios al repositorio en GitHub con `git push origin main` (o el nombre de la rama correspondiente).

C: > Users > facun > OneDrive > Documentos > Ejercicios > PHYTON > Tp2 > archivo.txt

1

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON\Ejercicios-git> cd "C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON"
PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON> git clone https://github.com/FacuuArrieta10/TrabajoN2.git
Cloning into 'TrabajoN2'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON>
```

C: > Users > facun > OneDrive > Documentos > Ejercicios > PHYTON > Ejercicios de Phytton > impor.py

1

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON> cd "C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON\TrabajoN2"
PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON\TrabajoN2> echo "Este es un archivo de prueba" > mi-archivo.txt
PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON\TrabajoN2> git add .
PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON\TrabajoN2> git commit -m "Agregando mi-archivo.txt"
[main d9943e9] Agregando mi-archivo.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 mi-archivo.txt
PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON\TrabajoN2>
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
PS C:\Users\facu\OneDrive\Documentos\Ejercicios\PHYTON\TrabajoN2> git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 348 bytes | 348.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/FacuuArrieta10/TrabajoN2.git
1886c24..d9943e9  main -> main
PS C:\Users\facu\OneDrive\Documentos\Ejercicios\PHYTON\TrabajoN2> |
```

FacuuArrieta10 / TrabajoN2

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

TrabajoN2 Public

main 1 Branch 0 Tags

Go to file Add file Code

FacuuArrieta10 Agregando mi-archivo.txt d9943e9 4 minutes ago 2 Commits

README.md Initial commit 15 minutes ago

mi-archivo.txt Agregando mi-archivo.txt 4 minutes ago

README

TrabajoN2

About

No description, website, or topics provided.

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

https://github.com/FacuuArrieta10/TrabajoN2/pulls

github.com/FacuuArrieta10/TrabajoN2

FacuuArrieta10 / TrabajoN2

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

TrabajoN2 Public

main 1 Branch 0 Tags

Go to file Add file Code

FacuuArrieta10 Initial commit 1886c24 6 minutes ago 1 Commit

README.md Initial commit 6 minutes ago

README

TrabajoN2

About

No description, website, or topics provided.

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

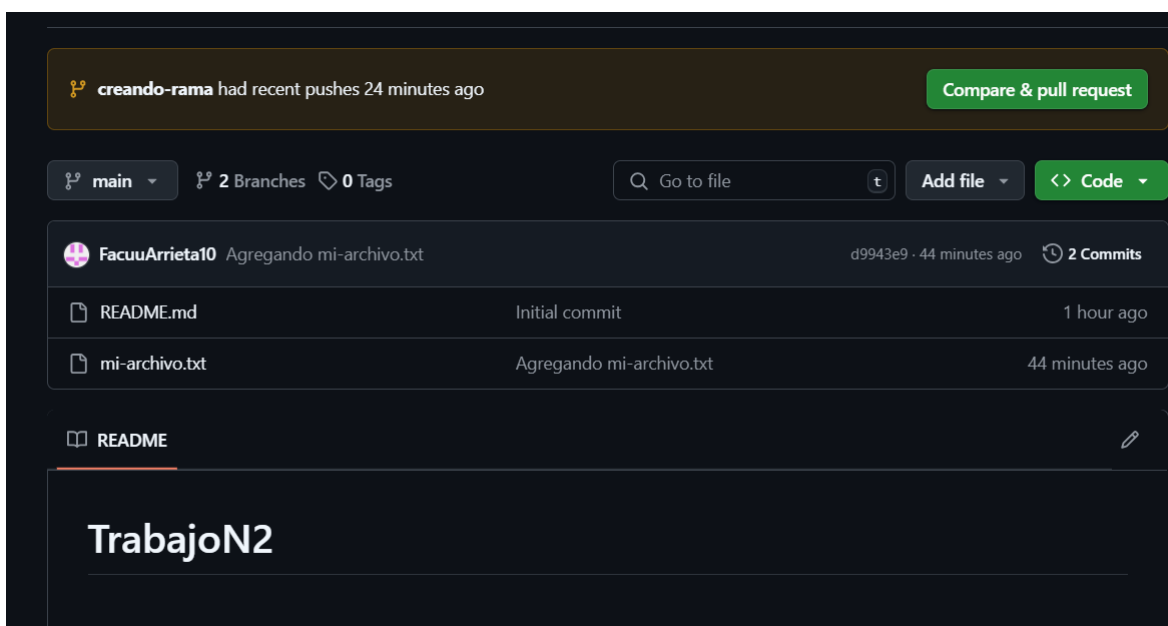
- Creando Branchs
 - Crear una Branch
 - Realizar cambios o agregar un archivo
 - Subir la Branch

```
PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON\TrabajoN2> git branch creando-rama
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON\TrabajoN2> git add .
PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON\TrabajoN2> git commit -m "se modifiko mi-archivo"
[creando-rama f2034a2] se modifiko mi-archivo
1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON\TrabajoN2> |
```

```
PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON\TrabajoN2> git push -u origin creando-rama
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 336 bytes | 168.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'creando-rama' on GitHub by visiting:
remote:   https://github.com/FacuuArrieta10/TrabajoN2/pull/new/creando-rama
remote:
To https://github.com/FacuuArrieta10/TrabajoN2.git
 * [new branch]      creando-rama -> creando-rama
branch 'creando-rama' set up to track 'origin/creando-rama'.
PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON\TrabajoN2> |
```



The screenshot shows the GitHub web interface for a repository named 'TrabajoN2'. At the top, a notification bar indicates that the 'creando-rama' branch has recent pushes 24 minutes ago, with a 'Compare & pull request' button. Below this, the repository overview shows the 'main' branch selected, with 2 other branches and 0 tags. A search bar and buttons for 'Add file' and 'Code' are visible. The commit history table lists two commits: an 'Initial commit' for 'README.md' made 1 hour ago, and a more recent commit 'Agregando mi-archivo.txt' made 44 minutes ago. Below the commit history, the 'README' file is displayed with the title 'TrabajoN2'.

- 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"
```

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.
- 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:

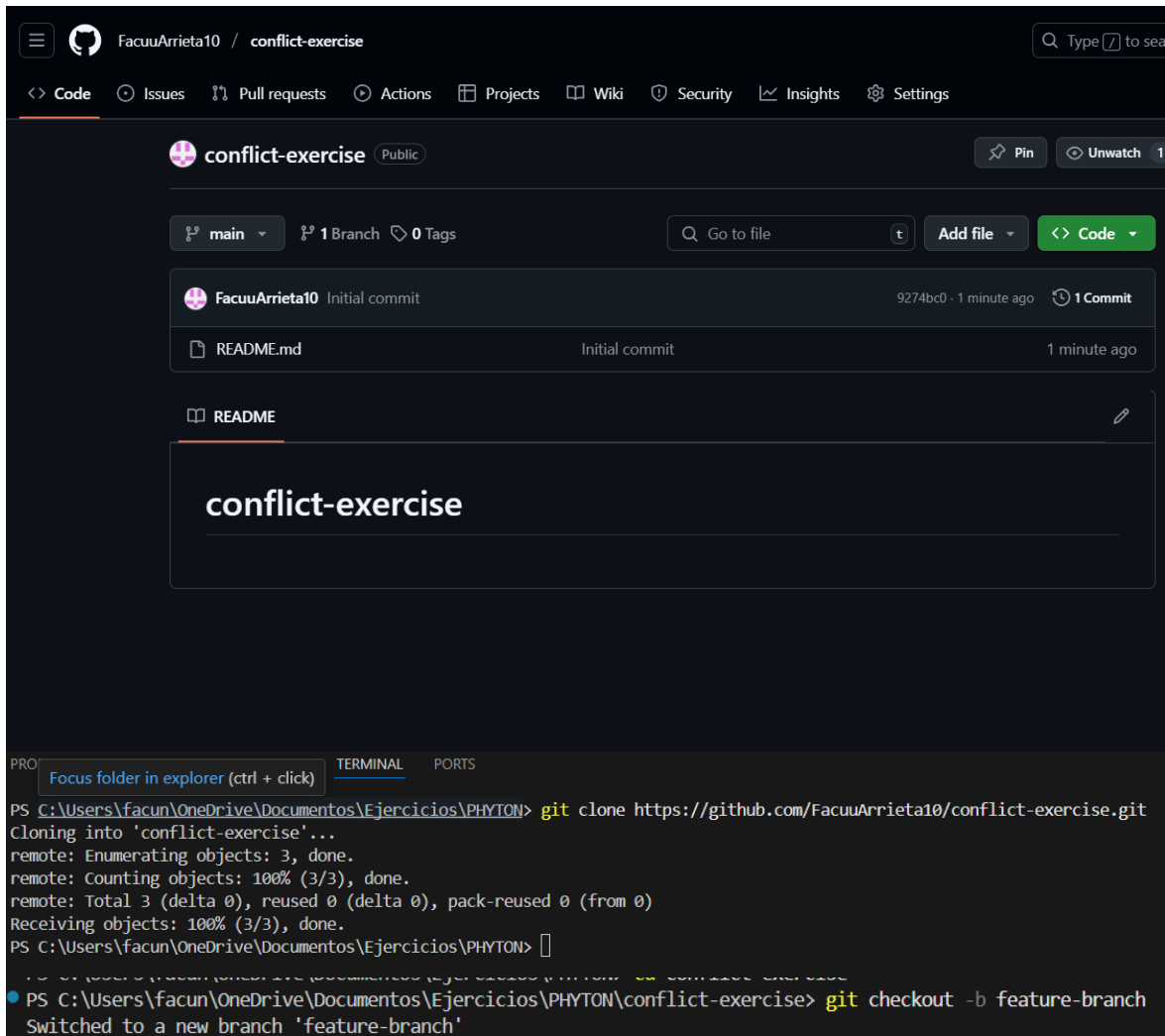
```
git push origin main
```

- 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.



The screenshot displays the GitHub interface for a repository named "conflict-exercise" owned by "FacuuArrieta10". The repository is public and shows an initial commit with a README.md file. Below the repository view, a terminal window shows the commands used to clone the repository and create a new branch named "feature-branch".

```
PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON> git clone https://github.com/FacuuArrieta10/conflict-exercise.git
Cloning into 'conflict-exercise'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON> 
PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON> git checkout -b feature-branch
Switched to a new branch 'feature-branch'
```

```
conflict-exercise > ① README.md > # <<<<<<< HEADprint ("Aguanto Boca")
1  # conflict-exercise
   Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
2  <<<<<<< HEAD (Current Change)
   print ("Aguanto Boca")
3  =====
   print ("hola mundo")
4  >>>>>>> feature-branch (Incoming Change)
5
6
7
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
● PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PYTHON\conflict-exercise> git commit -m "Added a line in main branch"
[main 1af4efe] Added a line in main branch
1 file changed, 2 insertions(+), 1 deletion(-)
● PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PYTHON\conflict-exercise> 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.
○ PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PYTHON\conflict-exercise>
```

① README.md M X

conflict-exercise > ① README.md > # conflict-exercise

1 # conflict-exercise

2 print ("Aguanto Boca")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Automatic merge failed; fix conflicts and then commit the result.

- PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON\conflict-exercise> git add README.md
- PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON\conflict-exercise> git add README.md
- PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON\conflict-exercise> git commit -m "Resolved merge conflict"

[main 65a7c48] Resolved merge conflict

PS C:\Users\facun\OneDrive\Documentos\Ejercicios\PHYTON\conflict-exercise> git push origin main

conflict-exercise / README.md

FacuuArrieta10 Se resolvió el conflicto ae00fff · now History

Preview Code Blame 2 lines (2 loc) · 42 Bytes Code 55% faster with GitHub Copilot Raw

1 # conflict-exercise

2 print ("Aguanto Boca")