

Práctico 2: Git y GitHub.

Actividades

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

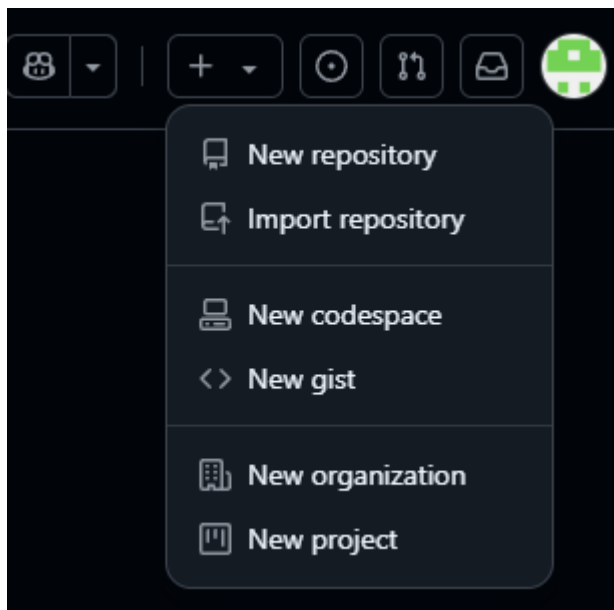
- ¿Qué es GitHub?

Github es una comunidad en donde podemos compartir repositorios, los mismos pueden ser públicos o privados.

- ¿Cómo crear un repositorio en GitHub?

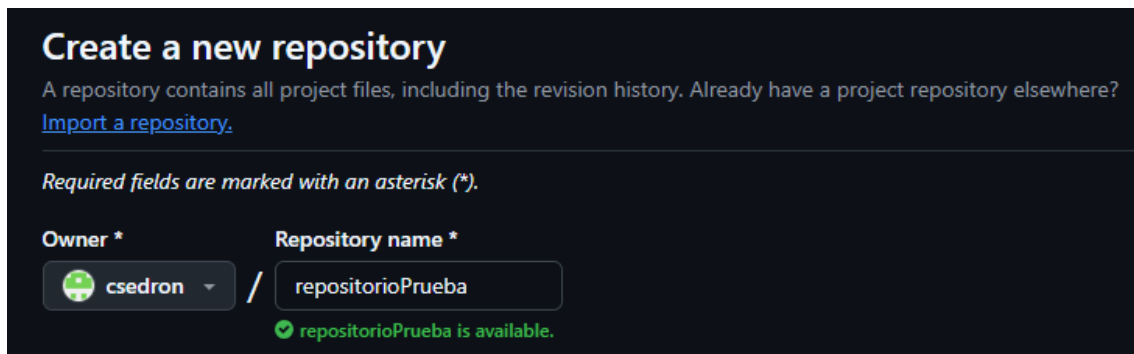
Como primera medida debemos crear una cuenta en GitHub y una vez realizado esto hacemos lo siguiente:

a - En la esquina superior derecha hacemos clic sobre el signo +:



Luego hacemos clic sobre **“New repository”**

b – Proporcionas un nombre para el repositorio.




Create a new repository

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

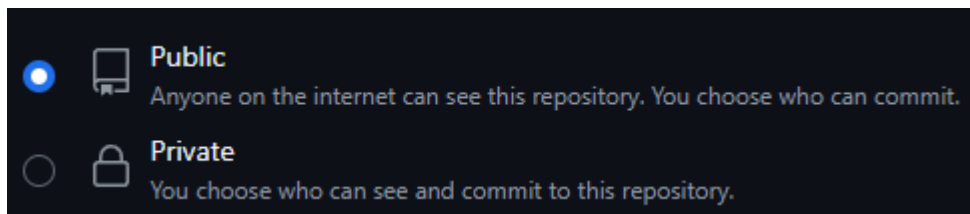
Required fields are marked with an asterisk ().*


Owner * Repository name *


 csedron /

✔ repositorioPrueba is available.

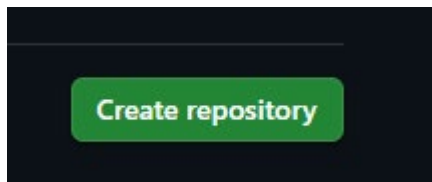
c – Elegir la visibilidad del repositorio. **Público o Privado.**



☒  **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.

d – Hacer clic en “Create repository”



[Create repository](#)

- ¿Cómo crear una rama en Git?

Para la creación de ramas utilizamos el siguiente comando:

git branch **nombre-rama**

- ¿Cómo cambiar a una rama en Git?

Para cambiar a una rama utilizamos el siguiente comando:

git checkout **nombre-rama**

- ¿Cómo fusionar ramas en Git?

Para fusionar una rama utilizamos el siguiente comando:

`git merge nombre-rama` Esto fusiona con **nombre-rama** con la rama en la cual nos encontramos posicionados.

- ¿Cómo crear un commit en Git?

Para realizar commit utilizamos el siguiente comando:

`git commit -m "mensaje que queremos ponerle al mismo"`

- ¿Cómo enviar un commit a GitHub?

Para enviar commit utilizamos el siguiente comando:

`git push -u origin main`

- ¿Qué es un repositorio remoto?

Un repositorio remoto, es aquel que alojado en un servidor fuera de nuestra infraestructura local, permite compartir y sincronizar cambios entre diferentes colaboradores como así también de diferentes dispositivos.

- ¿Cómo agregar un repositorio remoto a Git?

Para agregar un repositorio remoto utilizamos el siguiente comando:

`git remote add origin https://github.com/OWNER/REPOSITORY.git`

- ¿Cómo empujar cambios a un repositorio remoto?

Para empujar cambios utilizamos `git push` con dos argumentos:

- ✓ Un nombre remoto, por ejemplo, origin
- ✓ Un nombre de rama, por ejemplo, main

`git push REMOTE-NAME BRANCH-NAMEi`

- ¿Cómo tirar de cambios de un repositorio remoto?

Para recuperar cambios del repositorio remoto podemos usar:

`git clone https://github.com/USERNAME/REPOSITORY.git`

- ¿Qué es un fork de repositorio?

Un fork de repositorio es una copia de un repositorio, que podemos utilizar de un proyecto sin tener permisos de escritura sobre el mismo.

- ¿Cómo crear un fork de un repositorio?

En el repositorio al cual queremos realizar el fork, debemos hacer clic en la esquina superior derecha. Luego, se creará una copia del repositorio en nuestra cuenta.

- ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?
- ¿Cómo aceptar una solicitud de extracción?
- ¿Qué es una etiqueta en Git?

Una etiqueta es una instantánea que utilizamos para marcación de puntos específicos en el historial de un repositorio.

- ¿Cómo crear una etiqueta en Git?

Para crear la etiqueta usamos:

`Git tag <nombre de etiqueta>`

- ¿Cómo enviar una etiqueta a GitHub?

Se puede realizar utilizando:

`git tag -a v1.4 -m 'my version 1.4'`

- ¿Qué es un historial de Git?

El historial es el registro que me permite ver los cambios realizados en un repositorio, mostrando commits.

- ¿Cómo ver el historial de Git?

El comando para realizar esto es: `git log`

- ¿Cómo buscar en el historial de Git?

Se pueden combinar el uso de log con palabras clave de la siguiente manera:

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

- ¿Cómo borrar el historial de Git?

El borrado del historial se puede realizar de varias maneras. Aquí menciono para eliminar commits recientes.

`git reset --hard <commit>` esto para revertir el repositorio al commit especificado y eliminar los commits posteriores.

- ¿Qué es un repositorio privado en GitHub?

Este tipo de repositorio es el que permite trabajar de restringida solo a los propietarios y colaboradores autorizados. No es visible públicamente.

- ¿Cómo crear un repositorio privado en GitHub?

Hacer clic en el botón "+" en la esquina superior derecha y seleccionar "New repository".

Asignar un nombre a tu repositorio.

En la sección "Public or private", seleccionar "Private".

Configurar las opciones adicionales (como agregar un archivo README o un archivo .gitignore) según las necesidades.

Haz clic en "Create repository".

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

Ir al repositorio en GitHub, hacer clic en Settings (arriba a la derecha), seleccionar Collaborators and teams. Hacer clic en Add people y escribir el nombre o correo del usuario, elegir su nivel de acceso y enviar la invitación.

- ¿Qué es un repositorio público en GitHub?

Un repositorio publico es aquel que puede ser accesible para cualquier persona. Las mismas pueden ver, clonar el repositorio.

- ¿Cómo crear un repositorio público en GitHub?

Hacer clic en New repository en GitHub, ingresar un nombre para el repositorio, seleccionar la opción Public, que viene por defecto, hacer clic en Create repository.

- ¿Cómo compartir un repositorio público en GitHub?

Para compartir el repositorio solamente necesitamos copiar la URL del mismo y enviarla para que pueda acceder otro usuario.

2) Realizar la siguiente actividad:

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

El link para acceder al repositorio solicitado es:

<https://github.com/csedron/actividad2.git>

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"
```

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.

El link para acceder al repositorio solicitado es:

https://github.com/csedron/ejercicio_conflicto.git

claudio@RYZEN-CAS MINGW64 ~/git/ejercicio_conflicto (main)

\$ git log

commit 4020e3dca9787c958e3c68540b906894cbf61c16 (HEAD -> main, origin/main, origin/HEAD)

Merge: 0beff51 42eb1f5

Author: Claudio Andres Sedron <claudiosedron@gmail.com>

Date: Thu Mar 27 19:39:30 2025 -0300

Conflicto de merge resuelto

commit 0beff510c39422f984b09473a79328a913e82b47

Author: Claudio Andres Sedron <claudiosedron@gmail.com>

Date: Thu Mar 27 19:37:29 2025 -0300

Agrego una linea en rama main

commit 42eb1f51bbaf6c71f9b5515b25a8442a1509cabd (origin/feature-branch, feature-branch)

Author: Claudio Andres Sedron <claudiosedron@gmail.com>

Date: Thu Mar 27 19:36:13 2025 -0300

Agrego una linea en archivo en feature-brach

commit e7cdc4119406b37769ef6aa1bea5731d7dbacecc

Author: csedron <74806643+csedron@users.noreply.github.com>

Date: Thu Mar 27 19:32:09 2025 -0300

Initial commit

claudio@RYZEN-CAS MINGW64 ~/git/ejercicio_conflicto (main)

ⁱ <https://docs.github.com/es/get-started/using-git/pushing-commits-to-a-remote-repository>