

Práctico 2: Git y GitHub Agustina Aguilera

Actividades

1) Contestar las siguientes preguntas utilizando las guías y documentación proporcionada

(Desarrollar las respuestas) :

- ¿Qué es GitHub?

GitHub es una web donde podemos subir nuestros repositorios, pueden ser públicos o privados.

- ¿Cómo crear un repositorio en GitHub?

Iniciamos Git con el comando `git init` y luego `git add` . Despues de eso guardamos el commit con `git commit`

- ¿Cómo crear una rama en Git?

Con la función `git Branch` y nombre de la rama

- ¿Cómo cambiar a una rama en Git?

Con la función `git checkout` y el nombre de la rama

- ¿Cómo fusionar ramas en Git?

Con la función `git merge` y el nombre de la rama a fusionar

- ¿Cómo crear un commit en Git?

Con la función `git commit`

- ¿Cómo enviar un commit a GitHub?

Con la función `git push`

- ¿Qué es un repositorio remoto?

Es un repositorio que no se aloja en la pc sino en internet

- ¿Cómo agregar un repositorio remoto a Git?

Con la función `git clone`

- ¿Cómo empujar cambios a un repositorio remoto?

Con la función `git push`

- ¿Cómo tirar cambios de un repositorio remoto?

Con la función git pull

- ¿Qué es un fork de repositorio?

Es una copia de un repositorio para modificarlo sin afectar al original en la cuenta del usuario

- ¿Cómo crear un fork de un repositorio?

Clickeando en el botón fork

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

Con el botón new pull request

- ¿Cómo aceptar una solicitud de extracción?

Se va al apartado de revisar cambios y desde allí se puede aceptar

- ¿Qué es un etiqueta en Git?

Es una marca para los puntos específicos del historial como importantes, usualmente usado para marcar versiones

- ¿Cómo crear una etiqueta en Git?

Utilizando la función git tag

- ¿Cómo enviar una etiqueta a GitHub?

Con la función git push origin [etiqueta]

- ¿Qué es un historial de Git?

Es un registro de los cambios realizados en un repositorio

- ¿Cómo ver el historial de Git?

Con la función git reflog

- ¿Cómo buscar en el historial de Git?

Con la función git log --pretty y el criterio de búsqueda

- ¿Cómo borrar el historial de Git?

Con la función git rebase -i

- ¿Qué es un repositorio privado en GitHub?

Es un repositorio que solo el creador y personas autorizadas pueden ver

- ¿Cómo crear un repositorio privado en GitHub?

En nuevo repositorio y luego seleccionando la visibilidad como privada

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

Administrando el acceso desde la barra lateral “Colaboradores y equipos”

- ¿Qué es un repositorio público en GitHub?

Es un repositorio que puede ver cualquier persona

- ¿Cómo crear un repositorio público en GitHub?

Haciendo click en nuevo repositorio

- ¿Cómo compartir un repositorio público en GitHub?

Compartiendo el enlace al repositorio

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

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.

[Agustina-aguilera/tp_2: Repositorio para el tp 2](#)