

Práctico 2: Git y GitHub

Actividades

¿Qué es GitHub?

GitHub es una página/comunidad donde nosotros podemos compartir nuestro repositorio con el resto de la gente. Pueden ser público o privado.

¿Cómo crear un repositorio en GitHub?

Debemos dirigirnos a la pagina de GitHub, presionar en el botón "+", "New repository" después debemos colocarle un nombre, una descripción y hacerlo público o privado según sea nuestra necesidad.

¿Cómo crear una rama en Git?

Para crear una rama debemos ejecutar el siguiente comando en la consola/terminal

```
git branch "NombreDeLaRama"
```

¿Cómo cambiar a una rama en Git?

Para cambiar a una rama debemos ejecutar el siguiente comando en la consola/terminal

```
git checkout "nombreDeLaRamaACambiar"
```

¿Cómo fusionar ramas en Git?

Para fusionar ramas debemos ejecutar el siguiente comando en la consola/terminal

```
git merge "NombreDeLaRamaAFusionar"
```

¿Cómo crear un commit en Git?

```
git commit -m "texto que deseemos poner"
```

¿Cómo enviar un commit a GitHub?

```
git push -u origen main/master (si es la primera vez que hacemos un envío)
```

```
git push (si ya lo hicimos)
```

¿Qué es un repositorio remoto?

Es una versión de nuestro repositorio local pero alojada en una plataforma/página como GitHub

¿Cómo agregar un repositorio remoto a Git?

Podemos hacerlo con el siguiente comando usado en la consola/terminal

```
git remote add origin url
```

¿Cómo empujar cambios a un repositorio remoto?

Lo podemos realizar de la siguiente manera:

git add .

git commit -m "ejemploCambios"

git push

¿Cómo tirar de cambios de un repositorio remoto?

git pull origin main/master

¿Qué es un fork de repositorio?

En git un fork es la copia de un repositorio creada en una cuenta diferente permitiendo desarrollar diferentes cambios, pero sin afectar al original

¿Cómo crear un fork de un repositorio?

Debemos dirigirnos al repositorio en cuestión y pulsar donde dice "Fork" y lo siguiente es opcional, podemos cambiar el nombre y la descripción del mismo

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

Para hacer esto primero debemos hacer un fork del repositorio original, clonar el fork para llevarlo a mi máquina, crear una rama para los cambios, realizarlos y por último pulsar “Pull Request”

¿Cómo aceptar una solicitud de extracción?

Dirigirnos al repositorio que quieren extraer, revisar cambios, hacer click en “Merge pull request” y pulsar el botón “Confirm merge”

¿Qué es una etiqueta en Git?

Es un cartel o etiqueta que podemos poner a un proyecto

¿Cómo crear una etiqueta en Git?

```
git tag "nombre de la etiqueta"
```

¿Cómo enviar una etiqueta a GitHub?

```
git push --tags
```

¿Qué es un historial de Git?

Es un registro de todos los cambios que realizamos en un repositorio

¿Cómo ver el historial de Git?

```
git log
```

¿Cómo buscar en el historial de Git?

```
git log -s
```

¿Cómo borrar el historial de Git?

```
git reset
```

¿Qué es un repositorio privado en GitHub?

Es un tipo de repositorio que tiene acceso restringido, es decir solamente las personas con acceso a él pueden verlo y contribuir con el código.

¿Cómo crear un repositorio privado en GitHub?

Debemos dirigirnos a la página de GitHub, presionar en el botón “+”, “New repository” después debemos colocarle un nombre, una descripción y presionar el botón “**privado**”

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

Debemos dirigirnos al repositorio, configuración, colaboradores, agregar personas y hacer click en invitar

¿Qué es un repositorio público en GitHub?

Es un espacio virtual para almacenar archivos y códigos, al ser publico las personas tienen acceso a el pudiendo clonarlo, hacer fork y contribuir con el mismo

¿Cómo crear un repositorio público en GitHub?

Debemos dirigirnos a la página de GitHub, presionar en el botón "+", "New repository" después debemos colocarle un nombre, una descripción y hacerlo público.

¿Cómo compartir un repositorio público en GitHub?

Se comparte por medio de la URL del mismo.

1) Realizar la siguiente actividad:

- Crear un repositorio.
 - Dale un nombre al repositorio.
 - Elije 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).

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

<https://github.com/LucianoGallac/UTN-personal.git>

2) 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.

<https://github.com/LucianoGallac/EjercicioConflic.git>