

Práctico 2: Git y GitHub

Objetivo:

El estudiante desarrollará competencias para trabajar con Git y GitHub, aplicando conceptos fundamentales de control de versiones, colaboración en proyectos y resolución de conflictos, en un entorno simulado y guiado.

Resultados de aprendizaje:

1. Comprender los conceptos básicos de Git y GitHub: Identificar y explicar los principales términos y procesos asociados con Git y GitHub, como repositorios, ramas, commits, forks, etiquetas y repositorios remotos.
2. Manejar comandos esenciales de Git: Ejecutar comandos básicos para crear, modificar, fusionar y gestionar ramas, commits y repositorios, tanto en local como en remoto.
3. Aplicar técnicas de colaboración en GitHub: Configurar y utilizar repositorios remotos, realizar forks, y gestionar pull requests para facilitar el trabajo colaborativo.
4. Resolver conflictos en un entorno de control de versiones: Identificar, analizar y solucionar conflictos de merge generados en un flujo de trabajo con múltiples ramas.

Actividades

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

- ¿Qué es GitHub?

Git es un sistema de control de versiones distribuido y de código abierto, creado por Linus Torvalds, que permite a los desarrolladores rastrear y gestionar los cambios en el código fuente de un proyecto a lo largo del tiempo, facilitando la colaboración y el desarrollo de software

- ¿Cómo crear un repositorio en GitHub?

Se crea con el comando `git init`

- ¿Cómo crear una rama en Git?

`git branch nombre-rama`

- ¿Cómo cambiar a una rama en Git?

`Git checkout rama`

- ¿Cómo fusionar ramas en Git?

Git merge rama (Fusionar con rama, en rama actual)

- ¿Cómo crear un commit en Git?

Git commit -m "Mensaje"

- ¿Cómo enviar un commit a GitHub?

git add .

git add -A

git add --all Agregar todos los archivos al Stage (Los tres comandos son equivalentes)

¿Qué es un repositorio remoto?

Un repositorio remoto de Git es una copia de un proyecto que se encuentra en un servidor remoto, como internet, y que se comparte entre varios usuarios.

- ¿Cómo agregar un repositorio remoto a Git?

Git remote add origin url

- ¿Cómo empujar cambios a un repositorio remoto?

git push -u origin master

git push (Luego de la primera vez) Subir cambios al repositorio remoto

- ¿Cómo tirar de cambios de un repositorio remoto?

git pull origin master

git pull (Si usamos -u en el push)

- ¿Qué es un fork de repositorio?

Un fork de un repositorio es una copia independiente de un proyecto que se crea en tu cuenta para modificarlo o trabajar en él sin afectar el original. Se utiliza comúnmente en colaboración, permitiéndote proponer cambios al proyecto principal mediante pull requests.

- ¿Cómo crear un fork de un repositorio?

Accede al repositorio: Ve al repositorio original en la plataforma (como GitHub).

Haz clic en "Fork": En la parte superior derecha, haz clic en el botón "Fork".

Selecciona tu cuenta: Elige tu perfil o cuenta donde quieres crear el fork.

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

Crea un fork, realiza cambios, y luego desde GitHub haz clic en "Pull Request" para proponer los cambios al repositorio original.

- ¿Cómo aceptar una solicitud de extracción?

En el repositorio, ve a la pestaña "Pull Requests", revisa el código y haz clic en "Merge".

- ¿Qué es un etiqueta en Git?

Es un marcador usado para señalar versiones específicas de un proyecto (como lanzamientos).

- ¿Cómo crear una etiqueta en Git?

Usa: `git tag nombre`

`git tag -a v1.0 -m "Versión 1.0".` (Con mensaje)

- ¿Cómo enviar una etiqueta a GitHub?

Ejecuta: `git push origin v1.0.`

- ¿Qué es un historial de Git?

Es el registro de todos los commits realizados en un repositorio.

- ¿Cómo ver el historial de Git?

`git log`

- ¿Cómo buscar en el historial de Git?

`git log --grep="término" o git log -S "código"`

- ¿Cómo borrar el historial de Git?

No es recomendable, pero puedes reiniciar el historial eliminando el `.git` y corriendo `git init`

- ¿Qué es un repositorio privado en GitHub?

Es un repositorio solo accesible por sus propietarios o invitados.

- ¿Cómo crear un repositorio privado en GitHub?

Al crear el repositorio selecciona la opción "PRIVATE"

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

En "Settings" > "Collaborators", añade a la persona

- ¿Qué es un repositorio público en GitHub?

Es accesible para cualquiera en internet

- ¿Cómo crear un repositorio público en GitHub?

Al crear el repositorio selecciona la opción "PUBLIC"

- ¿Cómo compartir un repositorio público en GitHub?

Comparte la URL del repositorio (por ejemplo,
<https://github.com/usuario/repositorio>)

2) 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

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.