

Práctico 2: Git y GitHub

Actividades

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

- ¿Qué es GitHub?
- Es una plataforma de desarrollo que permite guardar proyectos y utilizar el sistema de control de versiones Git.

Es un software creado por Linus Torvalds que sirve para el control del ciclo de vida del proyecto. De esta forma, permite llevar de forma eficaz el registro de los cambios del proyecto de forma local o de forma remota, cuando es un trabajo colaborativo donde varias personas comparten archivo en un repositorio.

- ¿Cómo crear un repositorio en GitHub?
En la esquina superior derecha seleccionar + y luego Nuevo repositorio. Escribir el nombre y, opcionalmente, una descripción. Elegir si es privado o público. Hacer clic en Crear repositorio.
- ¿Cómo crear una rama en Git?
`git branch "ramanueva"`
- ¿Cómo cambiar a una rama en Git?
`git checkout "rama"`
- ¿Cómo fusionar ramas en Git?
`git checkout "rama1"`
`git merge "rama2" → vuelvo los cambios de la rama 2 en la rama 1`
- ¿Cómo crear un commit en Git?
`git commit -m "descripción del commit"`
- ¿Cómo enviar un commit a GitHub?
`git add .`
`git commit -m`
`git push "nombre de la rama remota" "nombre de la rama" / git push origin main`
- ¿Qué es un repositorio remoto?
Es un almacenamiento virtual de un proyecto. Permite guardar versiones del código y acceder a ellas.
- ¿Cómo agregar un repositorio remoto a Git?
`git init`
- ¿Cómo empujar cambios a un repositorio remoto?
`git push`

- ¿Cómo tirar de cambios de un repositorio remoto?
git pull request
- ¿Qué es un fork de repositorio?
Es una copia de un repositorio en mi cuenta de GitHub de forma independiente del original.
- ¿Cómo crear un fork de un repositorio?
En la página del repositorio, clic en "Fork". Agregar un comentario para el repositorio que ahora está en nuestra cuenta y hacer clic en "Crear Fork". El repositorio se encuentra en la cuenta personal de GitHub.
- ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?
Abrir el repositorio desde GitHub
Ir a Pull Requests >> New Pull Request >> base: main compare: nuevaRama
Escribir título y descripción
Clic en **"Create Pull Request"**
- ¿Cómo aceptar una solicitud de extracción?
Abrir el repositorio desde GitHub
Ir Pull Request y buscar el Pull Request
Hacer clic en Merge pull request
Confirm merge
- ¿Qué es una etiqueta en Git?
Es un marcador que indica un punto específico del historial de commits de un proyecto. Es usual su uso en versiones importantes.
- ¿Cómo crear una etiqueta en Git?
git tag "nombre"
- ¿Cómo enviar una etiqueta a GitHub?
git push origin "nombre"
- ¿Qué es un historial de Git?
Es el registro de todos los cambios realizados en un repositorio.
- ¿Cómo ver el historial de Git?
git log
- ¿Cómo buscar en el historial de Git?
git log --grep="palabra clave"
- ¿Cómo borrar el historial de Git?
git rebase -i HEAD~(cantidad de commits)

Seleccionar los commits

`git push --force`

- ¿Qué es un repositorio privado en GitHub?
Un repositorio al que solo pueden acceder personas autorizadas con diferentes permisos.
- ¿Cómo crear un repositorio privado en GitHub?
Crear Repositorio en GitHub y en la visibilidad seleccionar Private
- ¿Cómo invitar a alguien a un repositorio privado en GitHub?
Accede al repositorio privado en GitHub.
Ir a "Settings" (Configuración).
A la izquierda elegir "Manage".
Elegir "Manage access" y Luego Collaborators.
Clic en "Invite a collaborator".
Escribe el nombre de usuario o email de la persona para darle acceso seleccionando el tipo de permiso.
- ¿Qué es un repositorio público en GitHub?
Un repositorio que es accesible para cualquier persona en internet.
- ¿Cómo crear un repositorio público en GitHub?
Crear Repositorio en GitHub y en la visibilidad seleccionar Public
- ¿Cómo compartir un repositorio público en GitHub?
Compartir la URL del 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

<https://github.com/veronicamfalconi/falconiv.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.

<https://github.com/veronicamfalconi/conflict-exercise>