

Práctico 2: Git y GitHub

Alumno: Arturo Kaadú

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?
- ¿Cómo crear un repositorio en GitHub?
- ¿Cómo crear una rama en Git?
- ¿Cómo cambiar a una rama en Git?
- ¿Cómo fusionar ramas en Git?
- ¿Cómo crear un commit en Git?
- ¿Cómo enviar un commit a GitHub?
- ¿Qué es un repositorio remoto?
- ¿Cómo agregar un repositorio remoto a Git?
- ¿Cómo empujar cambios a un repositorio remoto?
- ¿Cómo tirar de cambios de un repositorio remoto?
- ¿Qué es un fork de repositorio?
- ¿Cómo crear un fork de un repositorio?



**TECNICATURA
UNIVERSITARIA EN
PROGRAMACIÓN
A DISTANCIA**

- ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio? •
- ¿Cómo aceptar una solicitud de extracción?
- ¿Qué es un etiqueta en Git?
- ¿Cómo crear una etiqueta en Git?
- ¿Cómo enviar una etiqueta a GitHub?
- ¿Qué es un historial de Git?
- ¿Cómo ver el historial de Git?
- ¿Cómo buscar en el historial de Git?
 - ¿Cómo borrar el historial de Git?
- ¿Qué es un repositorio privado en GitHub?
- ¿Cómo crear un repositorio privado en GitHub?
- ¿Cómo invitar a alguien a un repositorio privado en GitHub?
- ¿Qué es un repositorio público en GitHub?
- ¿Cómo crear un repositorio público en GitHub?
- ¿Cómo compartir un repositorio público en GitHub?

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).



**TECNICATURA
UNIVERSITARIA EN
PROGRAMACIÓN
A DISTANCIA**

- 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

**TECNICATURA
UNIVERSITARIA EN
PROGRAMACIÓN
A DISTANCIA**

Programación I



3

- 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
```

4

Programación I

**TECNICATURA
UNIVERSITARIA EN
PROGRAMACIÓN
A DISTANCIA**



- 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.

RESPUESTAS

¿Qué es GitHub?

Github es una herramienta que permite guardar y organizar proyectos de programacion y usa Git. Se puede guardar modificar, eliminar cambios realizados.

- ¿Cómo crear un repositorio en GitHub?

Desde github se puede directamente crear un repositorio desde el simbolo + en la parte superior derecha de nuestro github. Se hace click en New y te brinda las opciones de elegir si el repositorio será publico o privado.

- ¿Cómo crear una rama en Git?

Para crear una rama usamos desde la terminal el comando `git branch nombrederama`

- ¿Cómo cambiar a una rama en Git?

para cambiar una rama utilizamos desde la terminal el comando `git checkout nombrederama`

- ¿Cómo fusionar ramas en Git?

para fusionar ramas se utiliza `git merge` que fusiona latam en rama actual.

- ¿Cómo crear un commit en Git?

luego de hacer cambios agregamos `git add .` y luego `git commit -m "descripcion de cambios realizados"`

- ¿Cómo enviar un commit a GitHub?

una vez realizado el commit usamos `git push origin nombrederama`

- ¿Qué es un repositorio remoto?

Un repositorio remoto es el repositorio almacenado en nube como github. Permite tener otra copia y que varias personas trabajen en el mismo proyecto

- ¿Cómo agregar un repositorio remoto a Git?

se realiza con `git remote add origin URL del repositorio`

- ¿Cómo empujar cambios a un repositorio remoto?

utilizamos `git push nombre de rama`

- ¿Cómo tirar de cambios de un repositorio remoto?

utilizamos `git pull` para traer los cambios mas recientes del repositorio remoto.

- ¿Qué es un fork de repositorio?

Un fork es una copia de un repositorio que queda en nuestra cuenta de Github. Allí podemos modificarlo sin realizar cambios en el original

- ¿Cómo crear un fork de un repositorio?

Para crear un fork localizamos la opción fork en github, hacemos click y se creará la copia en nuestra cuenta.

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

Desde el repositorio vamos a la pestaña pull requests, y hacemos click en new pull request

- ¿Cómo aceptar una solicitud de extracción?

si se es dueño del repositorio se va a la pestaña de pull requests se abre la solicitud, se revisan los cambios y se hace click en merge pull request

- ¿Qué es una etiqueta en Git?

en github, una etiqueta es un marcador en un punto del proyecto, puede usarse para señalar alguna versión de importancia

- ¿Cómo crear una etiqueta en Git?

para crear una etiqueta usamos `git tag nombre de tag`

- ¿Cómo enviar una etiqueta a GitHub?

para enviar tags usamos `git push --tags`

- ¿Qué es un historial de Git?

el historial es el registro de los cambios, commits y acciones realizadas

- ¿Cómo ver el historial de Git?

con `git log` podemos ver el historial

- ¿Cómo buscar en el historial de Git?

`git log --oneline` sirve para buscar en el historial

- ¿Cómo borrar el historial de Git?

para borrar el historial puede ser util git rebase -i HEAD~n

- ¿Qué es un repositorio privado en GitHub?

Un repositorio privado está restringido al creador y quienes esa persona quiere que vean el proyecto.

- ¿Cómo crear un repositorio privado en GitHub?

para crear un repositorio privado se elige la opcion private al crear el repositorio

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

para invitar a alguien vamos a la pestaña settings, access, collaborators y ahi agregamos.

- ¿Qué es un repositorio público en GitHub?

un repositorio público en github es visible para cualquier persona.

- ¿Cómo crear un repositorio público en GitHub?

para crear un repositorio publico solo basta con crear un repositorio en github y seleccionar la opcion public.

- ¿Cómo compartir un repositorio público en GitHub?

podemos compartir un repositorio publico en github con su URL

Ejercicio 2

<https://github.com/arturokaadu/Arturo-Kaadu-conflict-exercise.git>

Ejercicio 3

<https://github.com/arturokaadu/conflict-exercise-parte-2.git>

Programación I