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?

GitHub es una plataforma basada en la web que permite gestionar repositorios Git. Facilita la colaboración entre desarrolladores al proporcionar control de versiones, seguimiento de cambios, revisiones de código y alojamiento de proyectos.

¿Cómo crear un repositorio en GitHub?

1. Inicia sesión en GitHub.
2. Haz clic en el ícono “+” en la esquina superior derecha y selecciona 'New repository'.
3. Escribe un nombre para el repositorio.
4. Selecciona público o privado y configura opciones adicionales.
5. Haz clic en 'Create repository'.

¿Cómo crear una rama en Git?

Ejecuta el siguiente comando en la terminal:

```bash
git branch nombre-de-la-rama
```

¿Cómo cambiar a una rama en Git?

Para cambiar de rama, usa:

```bash
git checkout nombre-de-la-rama
```

Desde Git 2.23, se recomienda usar:

```bash
git switch nombre-de-la-rama
```

¿Cómo fusionar ramas en Git?

1. Cambia a la rama principal:

```bash
git checkout main
```

2. Fusiona la otra rama:

```bash
git merge nombre-de-la-rama
```

¿Cómo crear un commit en Git?

1. Añadir cambios al área de preparación:

```bash
git add .
```

2. Crear un commit con un mensaje:

```bash
git commit -m "Descripción del cambio"
```

¿Cómo enviar un commit a GitHub?

Para enviar un commit a GitHub:

```bash
git push origin nombre-de-la-rama
```

Si es la primera vez que subes la rama:

```bash
git push -u origin nombre-de-la-rama
```

¿Qué es un repositorio remoto?

Es un repositorio alojado en un servidor externo (como GitHub, GitLab o Bitbucket) al que puedes empujar o extraer cambios.

¿Cómo agregar un repositorio remoto a Git?

Para agregar un repositorio remoto:

```bash
git remote add origin URL
```

¿Cómo empujar cambios a un repositorio remoto?

Para enviar cambios al repositorio remoto:

```bash
git push origin nombre-de-la-rama
```

¿Cómo tirar de cambios de un repositorio remoto?

Para obtener los últimos cambios del repositorio remoto:

```bash
git pull origin nombre-de-la-rama
```

¿Qué es un fork de repositorio?

Un fork es una copia de un repositorio en tu cuenta de GitHub. Permite experimentar y modificar el código sin afectar el original.

¿Cómo crear un fork de un repositorio?

1. Ve al repositorio en GitHub.
2. Haz clic en 'Fork' en la parte superior derecha.
3. Se creará una copia en tu cuenta.

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

1. Sube tus cambios a GitHub.
2. Ve al repositorio y haz clic en 'Pull requests'.
3. Crea una nueva solicitud indicando la rama de origen y destino.
4. Agrega un mensaje y envía la solicitud.

¿Cómo aceptar una solicitud de extracción?

El propietario del repositorio puede aceptar una pull request revisando los cambios y haciendo clic en 'Merge pull request'.

¿Qué es una etiqueta en Git?

Las etiquetas (tags) son referencias a commits específicos, utilizadas para marcar versiones en el historial del repositorio.

¿Cómo crear una etiqueta en Git?

Ejecuta el siguiente comando:
```bash
git tag -a v1.0 -m "Versión 1.0"
```

¿Cómo enviar una etiqueta a GitHub?

Para subir una etiqueta:
```bash
git push origin v1.0
```

¿Qué es un historial de Git?

El historial de Git registra todos los cambios realizados en un repositorio, incluyendo commits, merges y ramas.

¿Cómo ver el historial de Git?

Ejecuta:
```bash
git log
```

¿Cómo buscar en el historial de Git?

Puedes buscar palabras clave en los commits usando:
```bash
git log --grep="palabra clave"
```

¿Cómo borrar el historial de Git?

No se recomienda borrar el historial, pero puedes reiniciar un repositorio con:
```bash
git reset --hard HEAD~N
``` donde `N` es el número de commits a borrar.

## ¿Qué es un repositorio privado en GitHub?

Un repositorio privado es uno al que solo tienen acceso usuarios autorizados. No es visible públicamente.

## ¿Cómo crear un repositorio privado en GitHub?

Al crear un repositorio en GitHub, selecciona la opción 'Private'.

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

1. Ve a 'Settings' del repositorio.
2. En 'Manage access', haz clic en 'Invite a collaborator'.
3. Ingresa su nombre de usuario o correo y envía la invitación.

### ¿Qué es un repositorio público en GitHub?

Es un repositorio visible para cualquier usuario de GitHub.

### ¿Cómo crear un repositorio público en GitHub?

Selecciona 'Public' al crear un repositorio en GitHub.

### ¿Cómo compartir un repositorio público en GitHub?

Comparte la URL del repositorio o usa la opción 'Share' 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).
- 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.

LINK DE GITHUB: https://github.com/Luchiolm76/Programacion-I