

Nombre: Valeria Lucila Arellano
GitHub: <https://github.com/ValeriaArellano>

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? Es una plataforma en línea que permite a los usuarios almacenar, compartir y editar código
- ¿Cómo crear un repositorio en GitHub? En la esquina superior seleccionar + y luego nuevo repositorio. Escribir un nombre corto y fácil para el repositorio. Opcionalmente se puede agregar una descripción. Elegir la visibilidad del repositorio. También se puede crear un archivo README en el repositorio. Y finalmente hacer click en crear repositorio.
- ¿Cómo crear una rama en Git? Inicialmente se crea automáticamente una rama main / master. Para crear otra se usa el comando `git branch nombre-de-rama`
- ¿Cómo cambiar a una rama en Git? Para cambiar de rama en Git, primero usa `git branch` para ver el listado de ramas y con el comando `git checkout nombre-de-rama` se cambia a esa otra rama
- ¿Cómo fusionar ramas en Git? Si quieres fusionar una rama secundaria a la master, posicionado en la rama master inicialmente, se hace con el comando `git merge nombre-de-rama`. Esto fusiona lo que hay en “nombre-de-rama” con la rama master
- ¿Cómo crear un commit en Git? Antes, se debe hacer un `git init` para inicializar, luego “`git add .`” para añadir los cambios y luego el `git commit -m “comentario del commit”`
- ¿Cómo enviar un commit a GitHub? Con el comando `git push -u origin`
- ¿Qué es un repositorio remoto? Es una copia de un proyecto que se encuentra en un servidor

remoto

- ¿Cómo agregar un repositorio remoto a Git? Con el comando `git remote add` que toma dos argumentos, un nombre remoto y una dirección URL remota. Por ejemplo, `git remote add origin https://github.com/.....`
- ¿Cómo empujar cambios a un repositorio remoto? Con el comando `git push -u origin nombre-de-rama`
- ¿Cómo tirar de cambios de un repositorio remoto? Con el comando `git pull nombre-remoto nombre-de-rama`
- ¿Qué es un fork de repositorio? Una copia del repositorio, donde se puede escribir, haciendo públicos los propios cambios, como una manera abierta de participación
- ¿Cómo crear un fork de un repositorio?

1

Programación I

**TECNICATURA UNIVERSITARIA
EN PROGRAMACIÓN
A DISTANCIA**



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

Para eso se debe tener acceso de escritura a la rama principal activa o la rama de origen. Navega al repositorio original. Encima de la lista de archivos, hacer clic en Comparar y solicitud de incorporación de cambios para crear una solicitud de incorporación de cambios para la rama asociada. En la página para crear una nueva solicitud de incorporación de cambios, hacer clic en comparar entre bifurcaciones. En el menú desplegable "rama base", seleccionar la rama del repositorio ascendente donde quieras combinar los cambios. En el menú desplegable "bifurcación principal", seleccionar tu bifurcación. Usar el menú desplegable "comparar rama" para seleccionar aquella en la que realizaste los cambios. Escribir un título y una descripción para tu solicitud de extracción. Para crear una solicitud de incorporación de cambios que esté lista para revisión, hacer clic en Crear solicitud de incorporación de cambios. Para crear un borrador de una solicitud de incorporación de cambios, usar el menú desplegable y seleccionar Crear solicitud de incorporación de cambios de borrador, y después hacer clic en Solicitud de incorporación de cambios de borrador.

- ¿Cómo aceptar una solicitud de extracción? En el nombre del repositorio, hacer clic en Solicitudes de incorporación de cambios. En la lista de solicitudes de incorporación de cambios, hacer clic en la que quieras revisar.

En la solicitud de incorporación de cambios, hacer clic en Archivos cambiados.

Revisar los cambios en la solicitud de cambios y, opcionalmente, comentar líneas o archivos

específicos. Encima del código cambiado, hacer clic en Revisar cambios.

Teclear un comentario que resuma tu retroalimentación sobre los cambios propuestos.

Seleccionar Aprobar para aprobar la combinación de los cambios propuestos en la solicitud de incorporación de cambios.

Hacer clic en Enviar revisión.

- ¿Qué es un etiqueta en Git? Es una referencia que marca un punto específico en el historial de un repositorio

- ¿Cómo crear una etiqueta en Git?

Para crear etiquetas anotadas se usa:

```
git tag -a v1.0 -m "Mensaje descriptivo"
```

donde -a v1.0 crea una etiqueta llamada v1.0 y -m "mensaje descriptivo" agrega el mensaje

- ¿Cómo enviar una etiqueta a GitHub? Con el comando git push con la opción --tags

- ¿Qué es un historial de Git? Es el registro de todos los cambios realizados en un repositorio a lo largo del tiempo. Cada vez que realizas un commit, Git guarda una instantánea del código, incluyendo el autor, la fecha y un mensaje descriptivo.

- ¿Cómo ver el historial de Git? Con el comando git log

- ¿Cómo buscar en el historial de Git? Con git log --grep="palabra clave" busca un commit con esa palabra clave. Con git log -S "texto a buscar" busca el texto dentro del código. Con git log -- archivo.txt busca el historial de un archivo específico. Con git blame archivo.txt, se puede buscar quien cambió una línea de un archivo

- ¿Cómo borrar el historial de Git? Para eliminar el repositorio Git localmente usar rm -rf .git

- ¿Qué es un repositorio privado en GitHub? Es un repositorio al que solo las personas con permiso pueden acceder

- ¿Cómo crear un repositorio privado en GitHub? Al crearlo se puede elegir la visibilidad

- ¿Cómo invitar a alguien a un repositorio privado en GitHub? Si tienes un repositorio privado en GitHub y quieres compartirlo con alguien, debes agregarlo como colaborador o miembro del equipo en una organización.

- ¿Qué es un repositorio público en GitHub? Un repositorio público en GitHub es un repositorio accesible para cualquier persona en internet. Esto significa que cualquiera puede verlo y clonarlo, pero solo los colaboradores autorizados pueden modificarlo.

- ¿Cómo crear un repositorio público en GitHub? En la esquina superior seleccionar + y luego nuevo repositorio. Escribir un nombre corto y fácil para el repositorio. Opcionalmente se puede agregar una descripción. Elegir la visibilidad del repositorio como público.

- ¿Cómo compartir un repositorio público en GitHub? La forma más sencilla es enviar el enlace del repositorio. Ir a tu repositorio en GitHub. Copiar la URL desde la barra de direcciones. Compartir el enlace. Cualquiera con el enlace podrá ver y clonar el 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).

2

Programación I

**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

3

Programación I

**TECNICATURA UNIVERSITARIA
EN PROGRAMACIÓN
A DISTANCIA**



- 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.