Configuración inicial del proyecto

Realice los siguientes pasos para configurar el entorno de trabajo para esta actividad:

Cree y acceda a la carpeta del proyecto:

Elija una ubicación en su sistema y cree una nueva carpeta para este proyecto

Ingrese a esa carpeta desde su consola (Ej: cd ruta\a\<nombre\_de\_carpeta>). Todos los comandos siguientes se ejecutan dentro de esta carpeta.

Inicialice el repositorio Git: git init

Cree y confirme archivos iniciales:

Cree un archivo de texto <nombre\_de\_archivo> con cualquier contenido para el proyecto

git add <nombre\_de\_archivo>

git commit -m "Commit inicial: Agrega <nombre\_de\_archivo>"

Cree el repositorio remoto:

Acceda al sitio web de GitHub

Cree un nuevo repositorio como vimos en clase

Copie la URL (HTTPS) del repositorio remoto

Conecte el repositorio local al remoto:

Reemplace <url\_del\_repositorio\_remoto> con la URL copiada:

git remote add origin <url\_del\_repositorio\_remoto>

Verifique la conexión remota: git remote -v

Pregunta: ¿Qué información proporcionan las URLs listadas para el remoto origin?

Proporciona la siguiente informacion: origin <https://github.com/Mauricio4534/Primer-Repositorio.git> (fetch)

origin <https://github.com/Mauricio4534/Primer-Repositorio.git> (push)

Suba el estado inicial al remoto:

git push -u origin main

# (Use 'master' si su rama principal se llama así)

Verifique en la interfaz web del repositorio remoto que los archivos y commits iniciales ahora están presentes.

Ejercicio 1: Introducción a las ramas (Branches)

Liste y verifique las ramas: git branch

Pregunta: ¿Qué indica el asterisco (\*) al lado del nombre de las ramas?

El asterisco (\*) al lado de una rama en GitHub indica que esa rama es la rama actual que estás trabajando en el momento.

Cree una nueva rama:

Elija un nombre descriptivo para una nueva característica o tarea (ej: desarrollo-login, fix-bug-123, mejora-ui). Ejecute: git branch <nombre\_de\_su\_rama>

Ejecute git branch nuevamente.

Pregunta: ¿Se lista la nueva rama que creó? ¿Cuál sigue siendo la rama activa según el indicador? Se lista la nueva rama, y la rama master sigue siendo la activa porque no eh cambiado de rama.

Cámbiese a la nueva rama: git checkout <nombre\_de\_su\_rama>

Ejecute: git branch

Pregunta: ¿Qué rama figura como activa ahora? ¿Qué referencia interna importante de Git cree que se actualizó para reflejar este cambio? Ahora la rama desarrollo-login es la activa porque esta en verde y lleva el \* al lado del nombre que se usa como referencia para indicar en que rama estas.

Realice cambios en la nueva rama:

Modifique uno de los archivos existentes o cree uno nuevo.

Prepare y confirme este cambio en esta rama, utilizando un mensaje de commit descriptivo:

git add <archivo\_modificado\_o\_nuevo>

git commit -m "Su mensaje descriptivo del cambio"

Pregunta: ¿Cuál es la relación entre la rama activa (la que creó), el nuevo commit creado y la referencia HEAD? Que ese commit fue hecho en la rama desarrollo-login por mi ya que head indica mi posicion en la rama.

Compare con la rama principal:

Vuelva a la rama principal: git checkout main

Pregunta: ¿Cuál es ahora la rama activa indicada por HEAD?

La Rama Master es la indicada.

Revise el contenido del archivo que modificó en la otra rama.

Pregunta: ¿Está presente la modificación que realizó en la rama anterior? ¿Qué demuestra esto sobre el aislamiento entre ramas? No esta presente.

Demuestra que si quieres ver los cambios realizados en otras ramas tienes que estar en la rama donde quieres ver esos cambios.

Visualice las ramas: git log --oneline --graph --all

Pregunta: ¿Cómo representa el gráfico la separación de la historia entre main y la rama que creó? Asi las Representa:\* 635dac1 (**desarrollo-login**) Su mensaje descriptivo del cambio

\* 3894fa5 (**HEAD** -> **master**, **origin/master**) Commit inicial:Agrega texto.txt ¿Cómo identifica la posición actual (HEAD) en este gráfico? Indica que se encuentra en la rama master que es la rama donde estoy aplicando los cambios.

Ejercicio 2: Merge sin conflictos (Fast-Forward Merge)

Asegúrese de estar en la rama receptora (main o master).

Fusione la rama de la característica:

Fusione la rama que creó en el ejercicio anterior: git merge <nombre\_de\_su\_rama>

Pregunta: ¿La salida del comando git merge indica qué tipo de fusión se realizó?Indica que ocurrio una fusion Fast-Forward ¿Qué implica ese tipo de fusión para el historial? que ahora puedo ver los cambios ocurridos en la rama desarrollo-login sin tener que cambiar de rama.(o que los cambios pasaron ahora a la rama principal master)

Verifique el resultado:

Revise el contenido del archivo que modificó en la rama fusionada.

Pregunta: ¿Contiene ahora las modificaciones realizadas en la rama fusionada? Si , ahora puedo ver los cambios de ambas ramas en la rama principal.

Ejecute: git log --oneline --graph

Pregunta: ¿Cómo se refleja en el historial que main ahora incluye los cambios? se refleja asi: \* 635dac1 (**HEAD** -> **master**, **desarrollo-login**) Su mensaje descriptivo del cambio

\* 3894fa5 (**origin/master**) Commit inicial:Agrega texto.txt

¿Qué indica la posición de la etiqueta main en el grafo?

Indica que ambas ramas (o sus cambios) se encuentran el el mismo espacio (rama)

Elimine la rama fusionada:

Una vez fusionada, puede eliminar la rama de característica: git branch -d <nombre\_de\_su\_rama>

Ejecute: git branch

Pregunta: ¿Qué efecto tuvo el comando git branch -d en la lista de ramas?

Borro la rama nueva y solo quedo la rama master.

Ejercicio 3: Merge con conflictos

Cree y cámbiese a una nueva rama, rama donde se provocará el conflicto: git checkout -b <nombre\_rama\_conflicto>

Modifique un archivo en la nueva rama:

Elija un archivo y modifique una línea específica.

git add <archivo\_modificado>

git commit -m "Su mensaje para el primer cambio conflictivo"

Modifique la MISMA línea en la rama principal:

Vuelva a main: git checkout main

Modifique la misma línea en el mismo archivo pero de forma diferente.

git add <archivo\_modificado>

git commit -m "Su mensaje para el segundo cambio conflictivo en main"

Intente la fusión:

Asegúrese de estar en main y ejecute: git merge <nombre\_rama\_conflicto>

Pregunta: ¿Qué mensaje específico de Git le alerta sobre la existencia de un conflicto y en qué archivo? Dice que hay un contenido que causa conflicto en el archivo "texto.txt".

Inspeccione el estado y el archivo:

Ejecute: git status

Pregunta: ¿Qué indica la sección "Unmerged paths" sobre el estado del archivo durante un conflicto? Indica que todavia no fue integrado a la lista de cambios de git.

Abra el archivo en conflicto con su editor.

Pregunta: ¿Qué versión del código representa el bloque delimitado por <<<<<<< HEAD y =======?El bloque de código delimitado por <<<<<<< HEAD y ======= representa la versión de la rama actual (en la que se está trabajando) en un conflicto de fusión (merge conflict) dentro del sistema de control de versiones ¿Y el bloque entre ======= y >>>>>>> <nombre\_rama\_conflicto>? Representa el conflicto que esta ocurriendo en la otra rama que impide que la fusion se concrete.

Resuelva el conflicto manualmente:

Edite el archivo, decida qué contenido final debe tener la sección conflictiva y elimine completamente las marcas de conflicto (<<<<<<<, =======, >>>>>>>).

Marque el conflicto como resuelto: git add <archivo\_modificado\_y\_resuelto>

Complete el commit de merge:

Ejecute: git status

Pregunta: ¿Cómo cambió el estado del archivo después de ejecutar git add en el contexto de un conflicto resuelto? Que ahora los cambios fueron agregados a la lista de cambios de git.

Ejecute: git commit y confirme el mensaje en el editor.

Verifique:

Ejecute: git log --oneline --graph

Pregunta: ¿Cómo se visualiza en el historial la fusión completada después de resolver el conflicto?Que ahora en la rama master se encuentran los cambios de la rama "nombre\_rama\_conflicto" de la siguiente manera:

\* 1533b3b (**HEAD** -> **master**) Merge branch 'nombre\_rama\_conflicto':wq

|\

| \* d89a1db (**nombre\_rama\_conflicto**) Su mensaje para el primer cambio conflictivo

\* | 5c7384f Su mensaje para el segundo cambio conflictivo en main

|/

\* 635dac1 Su mensaje descriptivo del cambio

\* 3894fa5 (**origin/master**) Commit inicial:Agrega texto.txt

¿Dónde se ubica HEAD tras este commit de merge?

Se sigue ubicando en la rama master.

(Opcional) Elimine la rama: git branch -d <nombre\_rama\_conflicto>

Ejercicio 4: Subir cambios al repositorio remoto (push)

Suba la rama principal (main):

Asegúrese de estar en main.

git push origin main

# O simplemente 'git push'

Verifique en el remoto:

Visite la página web de su repositorio remoto.

Pregunta: ¿Se sincronizó el historial remoto con los últimos cambios de su rama main local?

Si

Suba otra rama cualquiera:

Si tuviera otra rama local con commits que desea compartir o respaldar, ¿cómo la subiría al remoto origin? la subiria usando el comando git push origin "rama\_cualquiera".

Ejercicio 5: Simular colaboración y bajar cambios (pull)

Para practicar el flujo de trabajo colaborativo, simularemos que otra persona (o usted mismo desde otra ubicación) realiza cambios en el repositorio remoto.

Navegue fuera del directorio original:

En su consola, asegúrese de estar dentro de otra carpeta, diferente de la que estuvo usando hasta el momento.

Clone el repositorio remoto en dicha carpeta:

Utilice git clone seguido de la URL de su repositorio remoto para clonarlo: git clone <url\_del\_repositorio\_remoteo>

Pregunta: ¿Observa mensajes indicando que se está clonando el repositorio? si ¿Se creó una nueva carpeta llamada de la misma forma en la cual nombró al repositorio? si se creo una carpeta con el mismo nombre del repositorio anterior.

Ingrese al repositorio clonado y realice cambios:

Acceda a la nueva carpeta creada debido a la clonación.

Ejecute: git status para verificar que se encuentra en la carpeta correcta.

Abra y modifique un archivo cualquiera en este clon, o agregue algún archivo.

Confirme y suba los cambios desde el clon:

Prepare y confirme el cambio realizado en el clon con un mensaje descriptivo:

git add <archivo\_modificado\_en\_clon>

git commit -m "Su mensaje descriptivo del cambio desde el clon"

Suba este commit desde el clon al repositorio remoto origin:

git push origin main

# (O la rama principal que esté usando)

Pregunta: ¿El comando push se ejecutó correctamente, indicando que los cambios se subieron al remoto? si, el archivo se subio al remoto.

Regrese al repositorio original:

Salga de la carpeta del repositorio clonado y vuelva a ingresar a su carpeta de trabajo original.

Verifique que se encuentra en el directorio correcto con los comandos ya dados.

Obtenga y fusione los cambios remotos en el repositorio original:

Ahora, desde su repositorio original, obtenga los cambios que "otra persona" (usted, desde el clon) subió al remoto. Asegúrese de estar en la rama main.

Ejecute:

git pull origin main

# O simplemente 'git pull' si el seguimiento está configurado

Verifique los cambios localmente:

Pregunta: ¿Qué información relevante le proporciona la salida del comando git pull acerca de los cambios bajados y la estrategia de integración utilizada (merge, fast-forward, etc.)? Que todos los cambios y commits realizados en el clon ahora estan en el original.

Revise el contenido del archivo que modificó desde el clon.

Pregunta: ¿Contiene ahora la modificación realizada desde el repositorio clonado?

si

Ejecute: git log --oneline --graph.

Pregunta: ¿Cómo se representa en el historial local el commit realizado "externamente"? Se Representa como\* f085fad (**HEAD** -> **master**, **origin/master**) Su mensaje descriptivo del cambio desde el clon:

¿Cómo se integró este commit en relación a la rama main y a la referencia HEAD? Porque en el clon yo ejecute ese cambio en la rama master y se transfirio a la rama master del original.

Ejercicio 6: Merge con remotos

Sin instrucción previa y según lo visto en clase, intente crear un conflicto con cambios hechos en el remoto, que usted aún no tenga en su rama local. Debe usar los conocimientos de este trabajo práctico y los métodos vistos en clase para recrear este escenario común en la práctica laboral.

Ejercicio 7: Push de las respuestas

Copie y pegue el archivo de las respuestas solicitado al principio de esta actividad en la carpeta del repositorio creado. Cree una versión del repositorio con este archivo y haga push de esta nueva versión con las respuestas al repositorio remoto.