

Materia: PROGRAMACION 1

Tema: Trabajo colaborativo Git y GitHub

Nombre del Estudiante: Holsbach Andrés Maximiliano

Actividades

Contestar las siguientes preguntas utilizando las guías y documentación proporcionada

- 1- ¿Qué es GitHub?
 - a. GitHub es una plataforma para alojar repositorios de forma remota.
- 2- ¿Cómo crear un repositorio en GitHub?
 - a. Para crear un repositorio en GitHub, primero, debemos crearnos una cuenta, una vez realiza esta acción, podemos utilizar el atajo de New Repository, y asignar un nombre a nuestro repositorio, que debe ser de nombre único entre todos los demás repositorios que ya hubiésemos creado.
- 3- ¿Cómo crear una rama en Git?
 - a. Para crear una rama en Git debemos seguir los siguientes pasos:
 - i. Conocer las ramas existentes con el comando git Branch.
 - ii. Con el comando git Branch y a continuación el nombre de la rama, crearemos una rama nueva, ejemplo: git Branch develop.
 - iii. Acotación personal: suelo utilizar el comando git switch -c y a continuación el nombre de la rama, este comando crea la rama, y cambia automáticamente el Head a la rama recién creada.
- 4- ¿Cómo cambiar a una rama en Git?
 - a. Para cambiar de rama en Git podemos utilizar los comando git checkout seguido del nombre de la rama, o git switch seguido del nombre de la rama.
- 5- ¿Cómo fusionar ramas en Git?
 - a. Para fusionar o unir ramas en git. Luego de realizar los cambios que necesitemos en la rama de trabajo, nos posicionaremos sobre la rama donde se acoplaran los cambios utilizando el comando git checkout seguido del nombre de la rama, una vez realizado esto escribiremos el comando git merge seguido del nombre de la rama donde estuvimos trabajando, por ejemplo: si tenemos dos ramas, main y develop, y mi rama de trabajo es develop, para unir los cambios de develop a main debería escribir el comando de la siguiente forma:
 - i. Git merge develop.

- 6- ¿Cómo crear un commit en Git?
 - a. Para crear un commit en Git debemos seguir los siguientes pasos:
 - i. `git status`: conocer el estado del repositorio
 - ii. `git add .`: agrega todos los archivos al Stage
 - iii. `git commit -m "mensaje"`: confirma los cambios, el mensaje es opcional, pero necesario.
- 7- ¿Cómo enviar un commit a GitHub?
 - a. Para enviar un commit a Github previamente debimos haber enlazado nuestro repositorio local, con uno remoto de Github. Si lo anterior fue realizado podemos aplicar el siguiente comando:
 - i. `Git push -u origin main` (si lo enviaremos directamente al main de nuestro repositorio remoto)
- 8- ¿Qué es un repositorio remoto?
 - a. Un repositorio remoto es un espacio en la nube donde se almacenan y gestionan proyectos de desarrollo.
- 9- ¿Cómo agregar un repositorio remoto a Git?
 - a. Para agregar un repositorio remoto a git debemos previamente crear el repositorio remoto en Github. Luego dentro de un repositorio local clonamos el repositorio remoto con el siguiente comando:
 - i. `git clone url`
- 10- ¿Cómo empujar cambios a un repositorio remoto?
 - a. Para empujar un cambio a un repositorio remoto debemos aplicar el siguiente comando: `Git push -u origin main`
- 11- ¿Cómo tirar de cambios de un repositorio remoto?
 - a. Para traer cambios desde un repositorio remoto debemos aplicar el siguiente comando: `Git pull origin main`
- 12- ¿Qué es un fork de repositorio?
 - a. Es una bifurcación, que copia exactamente ese repositorio remoto de ese usuario en tu cuenta de forma totalmente independiente.
- 13- ¿Cómo crear un fork de un repositorio?
 - a. Identificamos el repositorio de nuestro interés, hacemos clic en FORK ubicado en la esquina superior derecha, esto nos redirigirá a una página donde estará el siguiente título: **Create a new fork**, estará preseleccionado nuestro perfil, podremos asignar un nombre a este repositorio, y por último damos click a **create Fork** en la esquina inferior derecha.
- 14- ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?
 - a. Crear una rama en tu repositorio local para trabajar.
 - b. Realizar los cambios necesarios.
 - c. Subir los cambios al fork creado.

d. Ir al repositorio original y hacer clic en Pull Request.

15- ¿Cómo aceptar una solicitud de extracción?

a. Ir a la pestaña Pull Request, seleccionar la solicitud que deseamos aceptar, revisamos los cambios, si esta todo en orden hacemos clic en Merge pull request, y por ultimo confirmamos el merge.

16- ¿Qué es un etiqueta en Git?

a. Es una referencia estática que apunta a un punto específico en el historial de confirmaciones del repositorio.

17- ¿Cómo crear una etiqueta en Git?

a. Para crearla seguimos los siguientes pasos:
i. `git tag nombre_etiqueta` : esto crea la etiqueta.
ii. `git tag -a nombre_etiqueta -m "Mensaje descriptivo"`:
confirma la creación y la carga de metadatos.

18- ¿Cómo enviar una etiqueta a GitHub?

a. Una vez creada la etiqueta de forma local, subimos al repositorio remoto con el siguiente comando:
i. `git push origin nombre_etiqueta`: sube la etiqueta.

19- ¿Qué es un historial de Git?

a. Es el registro de todos los cambios realizados en un repositorio.

20- ¿Cómo ver el historial de Git?

a. Para ver el historial, aplicamos el siguiente comando:
i. `git feflog`

21- ¿Cómo buscar en el historial de Git?

a. Para buscar en el historial, aplicamos el siguiente comando:
i. `git log -- nombre_del_archivo`

22- ¿Cómo borrar el historial de Git?

a. Para borrar el historial de git debemos aplicar el siguiente comando
`git reset --hard ID_del_commit`

23- ¿Qué es un repositorio privado en GitHub?

a. Un repositorio privado en Github, es un repositorio remoto que solo podrán ver las personar a las que yo invite.

24- ¿Cómo crear un repositorio privado en GitHub?

a. Al momento de crear un repositorio nuevo tendremos dos opciones, para asignar al repositorio, de forma pública o privada, elegimos privada.

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

a. Para invitar a un repositorio privado, ingresamos al repositorio en cuestión, ingresamos en configuración, en la parte superior encontraremos la pestaña colaboradores, agregamos

colaboradores, mediante su email, o nombre de usuario, y enviamos la invitación.

26- ¿Qué es un repositorio público en GitHub?

- a. Es un repositorio donde se almacena remoto, y está accesible para cualquier usuario en la plataforma.

27- ¿Cómo crear un repositorio público en GitHub?

- a. Al momento de crear un repositorio nuevo tendremos dos opciones, para asignar al repositorio, de forma pública o privada, elegimos publica.

28- ¿Cómo compartir un repositorio público en GitHub?

- a. Para compartir un repositorio público, ingresamos al repositorio, copiamos el enlace que se muestra luego de darle clic a code, y luego enviamos a las personas con las que deseamos trabajar.

Realizar la siguiente actividad:

1. Crear un repositorio:

- a. Dale un nombre al repositorio: TP2-ACTIVIDAD2.git
- b. Elige el repositorio sea público:
<https://github.com/MaximilianoHolsbach/TP2-ACTIVIDAD2.git>
- c. Inicializa el repositorio con un archivo: realizado

2. Agregando un Archivo:

- a. Crea un archivo simple, por ejemplo, "miNombre.py".
- b. Realiza los comandos git add . y git commit -m "Agregando miNombre.py" : realizado
- c. Sube los cambios al repositorio en GitHub con git push origin main:
SHA 9314a1cdb22d351625de86888ad424d79d0028ae

3. Creando Branchs:

- a. Crear una Branch: develop
- b. Realizar cambios o agregar un archivo
- c. Subir la Branch: 498f0e2e540160fb60bae82beb019cdce70c98f9

Realizar la siguiente actividad:

1. Crear un repositorio en GitHub:

- a. Ve a GitHub e inicia sesión en tu cuenta: realizado
- b. Haz clic en el botón "New" o "Create repository" para crear un nuevo repositorio: realizado.
- c. Asigna un nombre al repositorio: TP2-ACTIVIDAD3

- d. Opcionalmente, añade una descripción: realizado.
 - e. Marca la opción "Initialize this repository with a README": realizado.
 - f. Haz clic en "Create repository": realizado.
2. Clonar el repositorio a tu máquina local:
 - a. Copia la URL del repositorio: realizado.
 - b. Abre la terminal o línea de comandos en tu máquina: realizado.
 - c. Clona el repositorio usando el comando: realizado.
 - d. Entra en el directorio del repositorio: realizado.
3. Crear una nueva rama y editar un archivo:
 - a. Abre el archivo README.md en un editor de texto y añade una línea nueva, por ejemplo:
 - i. Este es un cambio en la feature branch.
4. Volver a la rama principal y editar el mismo archivo:
 - a. Cambia de vuelta a la rama principal (main): realizado.
 - b. Edita el archivo README.md de nuevo, añadiendo una línea diferente: Este es un cambio en la main branch.
 - c. Guarda los cambios y haz un commit:
5. Hacer un merge y generar un conflicto:
 - a. Intenta hacer un merge de la feature-branch en la rama main: realizado.
 - b. Se generará un conflicto porque ambos cambios afectan la misma línea del archivo README.md.

```
i. # TP2-ACTIVIDAD3
    TP2-ACTIVIDAD3

    Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<<< HEAD (Current Change)

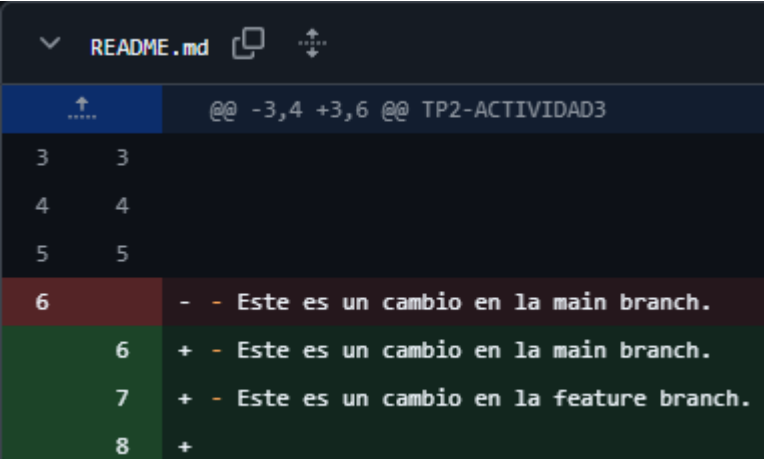
- Este es un cambio en la main branch.
=====
- Este es un cambio en la feature branch.

>>>>>> feature-branch (Incoming Change)
|
```

6. Resolver el conflicto:
 - a. Abre el archivo README.md en tu editor de texto. Verás algo similar a esto: realizado.
 - b. Decide cómo resolver el conflicto. Puedes mantener ambos cambios, elegir uno de ellos, o fusionar los contenidos de alguna manera.

- c. 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: realizado.
 - d. Añade el archivo resuelto y completa el merge: realizado.
- 7. Subir los cambios a GitHub:
 - a. Sube los cambios de la rama main al repositorio remoto en GitHub: Realizado.
 - b. También sube la feature-branch si deseas: realizado.
- 8. Verificar en GitHub:
 - a. Ve a tu repositorio en GitHub y revisa el archivo README.md para confirmar que los cambios se han subido correctamente: realizado.
 - b. Puedes revisar el historial de commits para ver el conflicto y su resolución.

i.



```
▼ README.md  [icon]  [icon]
↑
@@ -3,4 +3,6 @@ TP2-ACTIVIDAD3
3      3
4      4
5      5
6      - Este es un cambio en la main branch.
6      + Este es un cambio en la main branch.
7      + Este es un cambio en la feature branch.
8      +
```