

Nombre: Alejandro Farfán

Materia: Programación I

Comisión: 13

Actividades

1) Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas) :

- ¿Qué es GitHub?

Github es una plataforma que permite guardar nuestro código en la nube, fomentando el trabajo colaborativo y que utiliza el sistema de control de versiones Git.

- ¿Cómo crear un repositorio en GitHub?

Para crear un repositorio en Github se debe crear previamente una cuenta, acceder al menu Repositories y hacer click en New. Luego elegir un nombre y si será un repositorio público o privado y clickear en Create repository.

- ¿Cómo crear una rama en Git?

Para crear una rama en Git se debe utilizar el comando “git branch nombre-rama”

- ¿Cómo cambiar a una rama en Git?

Para cambiar a otra rama se puede utilizar el comando “git checkout nombre-rama”.

- ¿Cómo fusionar ramas en Git?

Para fusionar ramas con Git, una vez que nos encontramos en la rama a la cual queremos fusionar, usaríamos el comando “git merge nombre-rama”

- ¿Cómo crear un commit en Git?

Para crear un commit se utiliza el comando “git commit -m “comentario””. El comentario se utiliza para poder recordarnos a nosotros y a otros colaboradores lo que se actualizo en esta nueva versión. Tambien se pueden utilizar tags para eventos mas importantes con “git tag nombre”

- ¿Cómo enviar un commit a GitHub?

Para enviar un commit a Github se debe primero agregar la url de nuestros repository, usando “git remote add origin url”, donde la url seria la que nos brinda Github en nuestro perfil. Una vez indicada la url del repositorio en la nube, se podría subir el commit con “git push”

- ¿Qué es un repositorio remoto?

Un repositorio remoto es una copia de nuestro código en la nube, lo que nos permite descargarlo desde otros dispositivos y compartir nuestro código con otros desarrolladores.

- ¿Cómo agregar un repositorio remoto a Git?

Usando “git remote add origin url”, donde la url seria la que nos brinda Github en nuestro perfil.

- ¿Cómo empujar cambios a un repositorio remoto?

Una vez indicada la url del repositorio a Git, se podría subir el commit con “git push”

- ¿Cómo tirar de cambios de un repositorio remoto?

Se pueden verificar si hay cambios de un repositorio remoto con “git fetch”

- ¿Qué es un fork de repositorio?

Un fork es una copia de un repositorio, normalmente se realiza cuando queremos tener disponible una copia en nuestra cuenta de Github y realizar modificaciones al código.

- ¿Cómo crear un fork de un repositorio?

Una vez ubicados en el repositorio que queremos realizar el fork, arriba a la derecha clicleamos en fork y luego elegimos el nombre que vamos a darle y hacemos click en Create fork.

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

Se realiza con el comando “git pull”.

- ¿Cómo aceptar una solicitud de extracción?

En el caso de que hubiera conflictos en el código que Git no puede resolver automáticamente, se deberán resolver manualmente indicando que código se desea mantener.

- ¿Qué es un etiqueta en Git?

Las etiquetas son referencias que utilizamos para nombrar commits que fueran importantes como el lanzamiento de nuevas versiones o cuando se agrega una nueva funcionalidad, estos nombres permiten reconocer mas rápido los commits.

- ¿Cómo crear una etiqueta en Git?

Se crean con el comando “git tag nombre”.

- ¿Cómo enviar una etiqueta a GitHub?

Una vez creado el tag, se utilizaría “git push” para subir el código al repositorio.

- ¿Qué es un historial de Git?

El historial de git es un listado de todos los cambios de estado guardados, de todos los commits realizados.

- ¿Cómo ver el historial de Git?

Se utiliza el comando “git log”.

- ¿Cómo buscar en el historial de Git?

Si tenemos el hash del commit se puede utilizar “git show hash-commit”. Para verificar el listado de commits se puede utilizar “git log –oneline” que permite ver los commits en una línea cada uno.

- ¿Cómo borrar el historial de Git?

Para borrar todo el historial de Git, se podría borrar la carpeta oculta .git en el repositorio local.

- ¿Qué es un repositorio privado en GitHub?

Es un repositorio al que en principio solo el titular tiene acceso, si desea compartirlo debe dar permisos a los usuarios específicos.

- ¿Cómo crear un repositorio privado en GitHub?

Cuando se crea el repositorio se selecciona la casilla privado.

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

Se debe hacer click en invite collaborators y luego buscarlos e invitarlos.

- ¿Qué es un repositorio público en GitHub?

Es un repositorio el cual todos pueden ver.

- ¿Cómo crear un repositorio público en GitHub?

Para crear un repositorio público se debe acceder al menú Repositories y hacer click en New. Luego elegir un nombre, seleccionar la casilla de público y clicar en Create repository.

- ¿Cómo compartir un repositorio público en GitHub?

Se puede compartir la URL del repositorio.

2) Realizar la siguiente actividad:

- Crear un repositorio.

-Dale un nombre al repositorio.

-Elija el repositorio sea público.

-Inicializa el repositorio con un archivo

Owner* / Repository name*

AlejandroF

UTN-TUPaD-P1

Description (optional)

Este es mi repositorio de Organización Empresarial



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:



Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

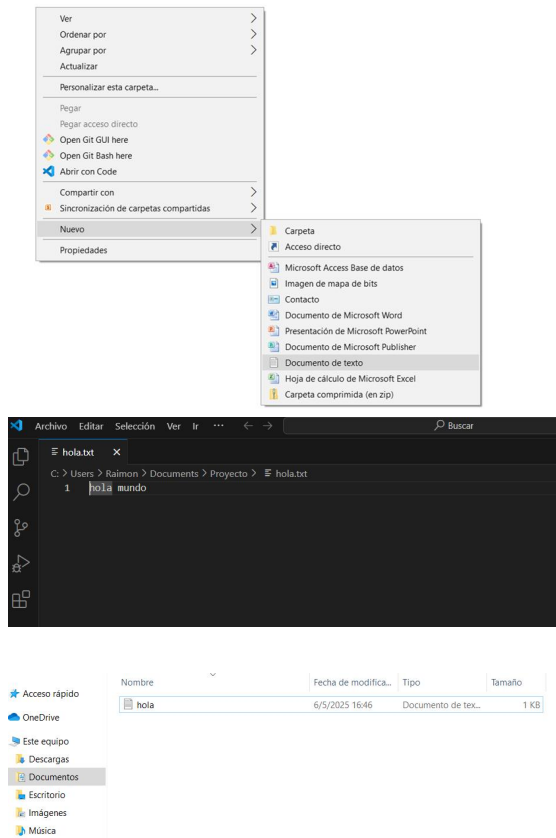
.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

.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).

Toco en la carpeta (Proyecto)



>git status (me dice el estado en que esta el directorio de trabajo)

hola.txt (aparece en rojo txt que todavía no fue agregado stage o que todavía no fue comiteado)

```
>git add hola.txt
```

```
>git commit -m "Agregando este comit"
```

```
>git push
```

.Creando Branchs

- Crear una Branch

-Realizar cambios o agregar un archivo

-Subir la Branch

```
>git branch <branch-name>:Crea una nueva rama.
```

```
>git checkout <branch-name>: Cambia a la rama especificada.
```

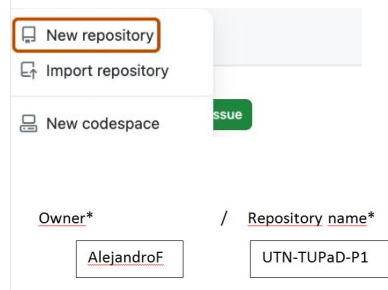
```
>git checkout -b <branch-name>:Crea una nueva rama y cambia a ella.
```

```
>git checkout main: Cambia a la rama principal.
```

3) Realizar la siguiente actividad:

Paso 1:

Crear un repositorio en GitHub



.Opcionalmente, puede agregar una descripción del repositorio. Por ejemplo, "Mi primer repositorio en GitHub".

.Elige la visibilidad del repositorio.

.Seleccione **Initialize this repository with a README** (Inicializar este repositorio con un archivo Léame).

.Haga clic en **Create repository** (Crear repositorio).

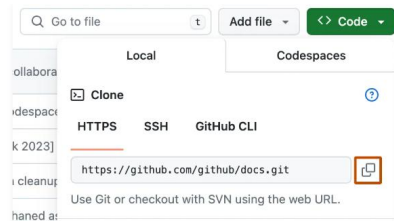
Paso 2:

clonar el repositorio a tu máquina local

.En GitHub, navegue hasta la página principal del repositorio.

.Encima de la lista de archivos, haz clic en **Código**.

.Copia la dirección URL del repositorio.



.Cambia el directorio de trabajo actual a la ubicación en donde quieres clonar el directorio.

.Escriba git clone y pegue la dirección URL que ha copiado antes. Enter

Ya tengo mi repositorio en mi área local.

Paso 3:

Crear una nueva rama y editar un archivo

Crear una nueva rama en Git y luego editar un archivo

git checkout -b <nombre_de_la_rama>.

Esto crea la rama y te cambia a ella.

Luego, puedes editar el archivo como de costumbre, usando tu editor de texto favorito.

Una vez terminado, utiliza git add <nombre_del_archivo>, git commit -m "mensaje descriptivo"

y git push para subir los cambios a la rama remot

-git checkout -b feature/login

-Edita app.js en tu editor de texto.

-git add app.js

-git commit -m "Implementación del login"

-git push (o git push -u origin feature/login si es la primera vez).

Paso 4: Volver a la rama principal y editar el mismo archivo

-git checkout es un comando de Git que te permite cambiar de rama o volver a una versión anterior del archivo.

-main o master (según el nombre de tu rama principal) son las ramas principales en la mayoría de los repositorios.

Una vez que estás en la rama principal, los cambios que hagas al archivo se aplicarán a esa rama.

Paso 5:

Hacer un merge y generar un conflicto

Crear dos ramas: Crea dos ramas en tu repositorio Git (por ejemplo, master y feature).

.Hacer cambios en diferentes ramas:

En feature, modifica un archivo (por ejemplo, README.md) en una línea específica.

En master, modifica el mismo archivo en la misma línea.

Realiza los cambios y haz commit en ambas ramas.

.Intenta fusionar las ramas:

Cambia a la rama master.

Ejecuta el comando `git merge feature` para intentar fusionar la rama feature en master.

Verás que Git detecta un conflicto y no puede fusionar las ramas automáticamente.

Paso 6: Resolver el conflicto

1. Abrir el archivo conflictivo:

Verás que el archivo conflictivo tiene marcas especiales en Git (por ejemplo, <<<<<<, =====, >>>>>>) que indican donde está el conflicto.

2. Seleccionar la versión deseada:

Decidirás qué versión del cambio quieres mantener (la de feature, la de master o una combinación de ambas).

3. Eliminar las marcas de conflicto:

Una vez que hayas seleccionado la versión deseada, borra las marcas de conflicto.

4. Hacer commit:

Agrega el archivo con el conflicto resuelto a la fase de almacenamiento y haz commit para finalizar la fusión.

Paso 7: Subir los cambios a GitHub

Ejecuta el comando `git push origin <nombre_de_la_rama>` para subir los cambios al repositorio remoto en GitHub.

Si estás usando la rama principal (por defecto suele ser main o master), puedes usar `git push origin main`.

Si estás usando una rama diferente, reemplaza <nombre_de_la_rama> con el nombre de tu rama.

Paso 8: Verificar en GitHub

Verifico en mi repositorio remoto que los cambios se han subido correctamente.

