Actividades Clase n°6:

Actividad n°1:

Responder Cuestionario para asistencia (Aula del Campus)

Actividad n°2:(Trabajo Grupal) Ver consignas al final

A) Realizar la siguiente actividad utilizando comandos vistos en la clases anteriores en la terminal Git Bach

1- Crear un nuevo repositorio Git con el nombre Clase6 desde la terminal Git Bach

2-iniciar repositorio

3-configurar repositorio

```
Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6
$ git init
Initialized empty Git repository in C:/Users/Matias/Tecnicatura en programacion/SPD/Cl
ase6/.git/
```

4- crear 2 carpetas de nombre :

-Teoría

- Código

```
Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6 (master)
$ mkdir Teoria

Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6 (master)
$ mkdir Codigo
```

5-abrimos carpeta Teoría y creamos una subcarpeta:

- Trabajo

luego abrimos la carpeta Código y creamos una subcarpeta:

-Evaluación.

```
Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6 (master)
$ cd Teoria/

Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6/Teoria (master)
$ mkdir Trabajo

Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6/Teoria (master)
$ cd ..

Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6 (master)
$ cd Codigo/

Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6/Codigo (master)
$ mkdir Evaluacion
```

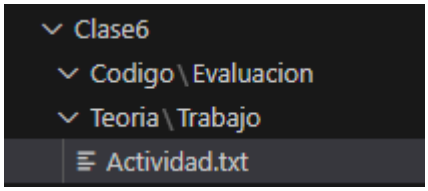
B)Realizar la siguiente actividad utilizando los comandos vistos en la terminal Git Bash y en el editor de texto que utilicen.

1- Abrimos el editor de texto, creamos un nuevo documento con el nombre :

-Actividad

Guardamos documento en nuestra carpeta trabajo.

Trabajo <-Actividad



2- Realizamos cuatro modificaciones en el archivo y commiteamos cada cambio realizado utilizando comandos vistos.

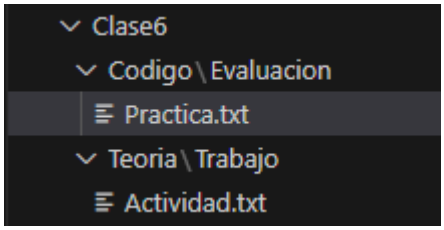
```
Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6 (master)
$ git log --oneline --graph
* 01fe9fa (HEAD -> master) cambio4
* a6d7472 cambio3
* 5bd5743 cambio2
* 00e2530 first commit
```

3- Creamos otro documento con el nombre en nuestro editor de texto con el nombre:

-Practica

Guardamos el documento Practica en la carpeta Evaluación

Evaluación <- Practica



4- realizamos 4 modificaciones en el documento Práctica, añadimos cambios y commit por separado

```
Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6 (master)
$ git add .

Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6 (master)
$ git commit -m 'cambio 5'
[master 4e54f18] cambio 5
 1 file changed, 1 insertion(+)
 create mode 100644 Codigo/Evaluacion/Practica.txt

Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6 (master)
$ git commit -am 'cambio 6'
[master f877c2e] cambio 6
 1 file changed, 2 insertions(+), 1 deletion(-)

Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6 (master)
$ git commit -am 'cambio 7'
[master 1e9e428] cambio 7
 1 file changed, 2 insertions(+), 1 deletion(-)

Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6 (master)
$ git commit -am 'cambio 8'
[master e928607] cambio 8
 1 file changed, 2 insertions(+), 1 deletion(-)
```

C)Realizar la siguiente actividad utilizando los comandos vistos las clases anteriores:

1- Creamos una rama auxiliar con el nombre :

-Prueba

```
Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6 (master)
$ git checkout -b prueba
Switched to a new branch 'prueba'
```

2- Creamos una nueva carpeta con el nombre :

-Integrador

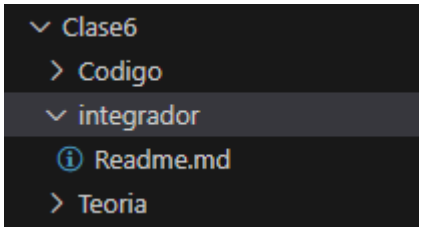
```
Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6 (prueba)
$ mkdir integrador
```

3- Abrimos un documento en nuestro editor de texto con el nombre :

-Readme

Guardamos documento en nuestra carpeta de nombre Integrador

Integrador <- Readme



4- Realizamos 4 modificaciones en nuestro documento Readme, añadimos cada modificación y commit por separado.

```
Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6 (prueba)
$ git add .

Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6 (prueba)
$ git commit -m 'cambio 1 rama prueba'
[prueba ce8fd67] cambio 1 rama prueba
 1 file changed, 1 insertion(+)
 create mode 100644 integrador/Readme.md

Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6 (prueba)
$ git commit -am 'cambio2 rama prueba'
[prueba 3febc23] cambio2 rama prueba
 1 file changed, 2 insertions(+), 1 deletion(-)

Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6 (prueba)
$ git commit -am 'cambio3 rama prueba'
[prueba 1939700] cambio3 rama prueba
 1 file changed, 2 insertions(+), 1 deletion(-)

Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6 (prueba)
$ git commit -am 'cambio4 rama prueba'
[prueba 9bcd649] cambio4 rama prueba
 1 file changed, 2 insertions(+), 1 deletion(-)
```

5 - Fusionamos Rama Prueba con nuestra Rama Clase6 (Master o Main)

6- Tomar captura de la fusión de las ramas.

```
Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6 (prueba)
$ git checkout master
Switched to branch 'master'

Matias@ASUSMatias MINGW64 ~/Tecnicatura en programacion/SPD/clase6 (master)
$ git merge prueba
Updating e928607..9bcd649
Fast-forward
 integrador/Readme.md | 4 ++++
 1 file changed, 4 insertions(+)
 create mode 100644 integrador/Readme.md
```

D) Responder las siguientes preguntas : desarrollar y utilizar comandos vistos (se puede acompañar de capturas)

1- ¿Cómo creamos una carpeta desde Git Bash?

Para crear una carpeta desde Git Bash, puedes utilizar el comando mkdir seguido del nombre de la carpeta que deseas crear. Por ejemplo, si deseas crear una carpeta llamada "mi_carpeta", puedes ejecutar el siguiente comando:

```
mkdir mi_carpeta
```

2- ¿Cómo iniciamos nuestro Repositorio Git?

Para iniciar un repositorio Git, debes navegar hasta la ubicación de tu proyecto en Git Bash utilizando el comando cd (change directory) y luego ejecutar el comando git init.

```
cd ruta/del/proyecto
git init
```

3- ¿Qué comando utilizamos al configurar usuario y email ?

El comando utilizado para configurar el nombre de usuario y el correo electrónico en Git es: Debes reemplazar "Tu Nombre" con tu nombre y "tu@email.com" con tu dirección de correo electrónico.

```
git config --global user.name "Tu Nombre"
git config --global user.email "tu@email.com"
```

4- ¿Cuál es el comando que me permite viajar en el tiempo de los diferentes commit?

El comando que te permite viajar en el tiempo y revisar diferentes commits es git checkout. Puedes utilizarlo para cambiar entre ramas o para desplazarte a commits específicos. Por ejemplo:

```
git checkout <nombre_de_la_rama>
git checkout <hash_del_commit>
```

5- ¿A qué llamamos Rama Master o Main?

La rama "Master" o "Main" se refiere a la rama principal de un repositorio Git. Es la rama predeterminada creada cuando se inicializa un repositorio y suele ser utilizada como la rama principal donde se encuentran las versiones estables del código.

6- ¿Cuál es la diferencia entre Rama Auxiliar y Rama Master o Main?

La diferencia entre una rama auxiliar y la rama "Master" o "Main" es que la rama "Master" o "Main" es la rama principal del repositorio y suele contener las versiones estables del código. Las ramas auxiliares se crean a partir de la rama principal y se utilizan para desarrollar nuevas características o solucionar problemas sin afectar directamente la rama principal. Una vez que las modificaciones en una rama auxiliar son probadas y consideradas estables, pueden fusionarse con la rama principal.

7- ¿Qué comando utilizamos para cambiar de una rama a otra?

Para cambiar de una rama a otra en Git, se utiliza el comando git checkout seguido del nombre de la rama a la que deseas cambiar. Por ejemplo:

```
git checkout <nombre_de_la_rama>
```

8- ¿Se puede modificar la Rama Master?

En Git, la rama "Master" o "Main" es la rama principal y por defecto no se puede modificar directamente. Sin embargo, puedes crear nuevas ramas a partir de la rama principal y realizar modificaciones en esas ramas. Luego, si deseas incorporar esas modificaciones a la rama principal, puedes fusionar las ramas utilizando el comando git merge.

9- ¿Cuál es el comando para crear una nueva rama?

El comando para crear una nueva rama en Git es git branch seguido del nombre de la nueva rama que deseas crear. Por ejemplo:

```
git branch nueva_rama
```

Esto creará una nueva rama llamada "nueva_ rama" basada en la rama actual en la que te encuentras.

10- Mencionar 3 funcionalidades que tiene el comando Git Checkout

- Cambiar de rama: Puedes utilizar git checkout <nombre_de_la_rama> para cambiar de una rama a otra.
- Deshacer cambios locales: Si has realizado modificaciones en tus archivos pero aún no has confirmado los cambios, puedes utilizar git checkout -- <nombre_del_archivo> para descartar los cambios y restaurar el archivo a su estado anterior.

- Viajar en el tiempo: Como se mencionó anteriormente, puedes utilizar git checkout <hash_del_commit> para desplazarte a un commit específico en el historial del repositorio.

11- ¿Cuál es el comando que me permite ver de manera gráfica los commit y las ramas creadas?

El comando que te permite ver de manera gráfica los commits y las ramas creadas es git log con la opción --graph. Puedes ejecutar el siguiente comando

```
git log --graph
```

O puede ser también de la siguiente manera:

```
git log --graph --oneline
```

Esto mostrará un gráfico con las ramas y los commits en tu repositorio.

12- Realizar una captura del último punto realizado, se puede utilizar captura de la actividad n° 4.