

ACTIVIDAD 5

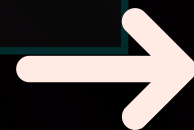
DESARROLLO DE SOFTWARE

Aarón Flores Alberca
Leonardo Chacón Roque

PREGUNTA 1

Clona un repositorio Git con múltiples ramas.

Para esto crearemos un repositorio de “./repo” con dos ramas “main” y “feature”, para luego clonarlo en “./repo2”



```
C:\Users\Public\Downloads>mkdir repo
```

```
C:\Users\Public\Downloads>cd repo
```

```
C:\Users\Public\Downloads\repo>git init -b main  
Initialized empty Git repository in C:/Users/Public/Downloads/repo/.git/
```

```
C:\Users\Public\Downloads\repo>echo "Texto 1 en main" > archivo.txt
```

```
C:\Users\Public\Downloads\repo>git add archivo.txt
```

```
C:\Users\Public\Downloads\repo>git commit -m "Commit 1 en main"  
[main (root-commit) 2a43211] Commit 1 en main  
1 file changed, 1 insertion(+)  
create mode 100644 archivo.txt
```

```
C:\Users\Public\Downloads\repo>git checkout -b feature  
Switched to a new branch 'feature'
```

```
C:\Users\Public\Downloads\repo>echo "Texto 1 en feature" >> archivo.txt
```

```
C:\Users\Public\Downloads\repo>git add archivo.txt
```

```
C:\Users\Public\Downloads\repo>git commit -m "Commit 1 en feature"  
[feature aa18a04] Commit 1 en feature  
1 file changed, 1 insertion(+)
```

```
C:\Users\Public\Downloads\repo>git checkout main  
Switched to branch 'main'
```

```
C:\Users\Public\Downloads\repo>cd ..
```

```
C:\Users\Public\Downloads>git clone ./repo repo2  
Cloning into 'repo2'...  
done.
```

```
C:\Users\Public\Downloads>cd repo2
```

Identifica dos ramas que
puedas fusionar

Haz el proceso de fusión
utilizando git merge --ff.

Verifica el historial con git
log --graph --oneline.

```
C:\Users\Public\Downloads>cd repo2
```

```
C:\Users\Public\Downloads\repo2>git branch  
* main
```

```
C:\Users\Public\Downloads\repo2>git fetch origin
```

```
C:\Users\Public\Downloads\repo2>git checkout feature  
branch 'feature' set up to track 'origin/feature'.  
Switched to a new branch 'feature'
```

```
C:\Users\Public\Downloads\repo2>git branch  
* feature  
main
```

```
C:\Users\Public\Downloads\repo2>git checkout main  
Switched to branch 'main'  
Your branch is up to date with 'origin/main'.
```

```
C:\Users\Public\Downloads\repo2>git merge --ff feature  
Updating b4ee8e1..75d6ced  
Fast-forward  
 archivo.txt | 1 +  
 1 file changed, 1 insertion(+)
```

```
C:\Users\Public\Downloads\repo2>git log --graph --oneline --all  
* 75d6ced (HEAD -> main, origin/feature, feature) Commit 1 en feature  
* b4ee8e1 (origin/main, origin/HEAD) Commit 1 en main
```

Pregunta: ¿En qué situaciones recomendarías evitar el uso de git merge --ff?
Reflexiona sobre las desventajas de este método.

Al usarlo no se crea un commit de fusión o merge, por lo que no queda evidencia ni rastro de una rama que fue editada e integrada, dificultando la trazabilidad.

Otro problema es que no se pueden revertir funcionalidades porque no hay un merge al que retroceder.

PREGUNTA 2

Simula un flujo de trabajo de equipo

Para este ejercicio contaremos con un repositorio de dos ramas independientes que contengan diferentes commits en cada una

```
bxco@bxco-NBD-WXX9:~/ejercicio-2$ git init
Inicializado repositorio Git vacío en /home/bxco/ejercicio-2/.git/
bxco@bxco-NBD-WXX9:~/ejercicio-2$ echo "# Iniciación del ejercicio 2 - Actividad 5" > README.md
bxco@bxco-NBD-WXX9:~/ejercicio-2$ echo "#include<iostream>

using namespace std;

int main(){
    cout << "Actividad 4 - Desarrollo de Software" << endl;
    return 0;
}" > main.cpp
bxco@bxco-NBD-WXX9:~/ejercicio-2$ git add README.md
bxco@bxco-NBD-WXX9:~/ejercicio-2$ git commit -m 'Add README file'
[main (commit-raíz) d792d8f] Add README file
1 file changed, 1 insertion(+)
create mode 100644 README.md
bxco@bxco-NBD-WXX9:~/ejercicio-2$ git add main.cpp
bxco@bxco-NBD-WXX9:~/ejercicio-2$ git commit -m 'Add main.cpp to greet'
[main 19263a8] Add main.cpp to greet
1 file changed, 8 insertions(+)
create mode 100644 main.cpp
bxco@bxco-NBD-WXX9:~/ejercicio-2$ git branch feature
bxco@bxco-NBD-WXX9:~/ejercicio-2$ git checkout feature
Cambiado a rama 'feature'
bxco@bxco-NBD-WXX9:~/ejercicio-2$ echo "public class Test{
    public static void main(String[] args){
        System.out.println("Actividad 4 - Desarrollo de Software");
    }
}" > Test.java
bxco@bxco-NBD-WXX9:~/ejercicio-2$ git add Test.java
bxco@bxco-NBD-WXX9:~/ejercicio-2$ git commit -m 'Add java file to greet through a different language'
[feature 482f84b] Add java file to greet through a different language
1 file changed, 5 insertions(+)
create mode 100644 Test.java
```


Basado en la construcción de dichas ramas, supongamos que deseamos combinar nuestros commits en nuestra rama principal.

Dentro de la rama main, haremos uso del comando:
`$ git merge --no-ff feature`

Cuando se ejecute se nos pedirá insertar un mensaje para el commit merge.

Finalmente haciendo uso de:
`$ git log --graph --oneline --all`
podremos visualizar la estructura de dicho merge.

```
bxco@bxco-NBD-WXX9:~/ejercicio-2$ git checkout main
Cambiado a rama 'main'
bxco@bxco-NBD-WXX9:~/ejercicio-2$ git merge --no-ff feature
Merge made by the 'ort' strategy.
Test.java | 5 +++++
1 file changed, 5 insertions(+)
create mode 100644 Test.java
```

```
GNU nano 6.2 /home/bxco/ejercicio-2/.git/MERGE_MSG *
Commit Merge con feature para registrar los nuevos archivos.
# Por favor ingresa un mensaje de commit que explique por qué es necesaria esta fusión,
# especialmente si esto fusiona un upstream actualizado en una rama de tópico.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
```

```
bxco@bxco-NBD-WXX9:~/ejercicio-2$ git log --graph --oneline --all
* 2650c09 (HEAD -> main) Commit Merge con feature para registrar los nuevos archivos.
|
| * 482f84b (feature) Add java file to greet through a different language
|/
* 19263a8 Add main.cpp to greet
* d792d8f Add README file
```

Pregunta: ¿Cuáles son las principales ventajas de utilizar git merge --no-ff en un proyecto de equipo? ¿Qué problemas podrían surgir al depender excesivamente de commits de fusión?

Entre las principales ventajas se pueden destacar la de la creación de commits merge que permitan una mejor interpretación y legibilidad del historial de commits de su respectivo repositorio sin combinar ramas; además, permite un rollback mejor entendido gracias a dichos commits merge.

Sin embargo, un historial de commits completamente repleto de estos commits merge pueden resultar ser contraproductivos y entorpecer los rebases que se deseen realizar e incluso interferir con los pipelines CI/CD.

PREGUNTA 3

Crea múltiples commits en una rama.

```
C:\Users\Public\Downloads>mkdir repo
```

```
C:\Users\Public\Downloads>cd repo
```

```
C:\Users\Public\Downloads\repo>git init -b main  
Initialized empty Git repository in C:/Users/Public/Downloads/repo/.git/
```

```
C:\Users\Public\Downloads\repo>echo "Primera escritura en arch.txt (main)" > arch.txt
```

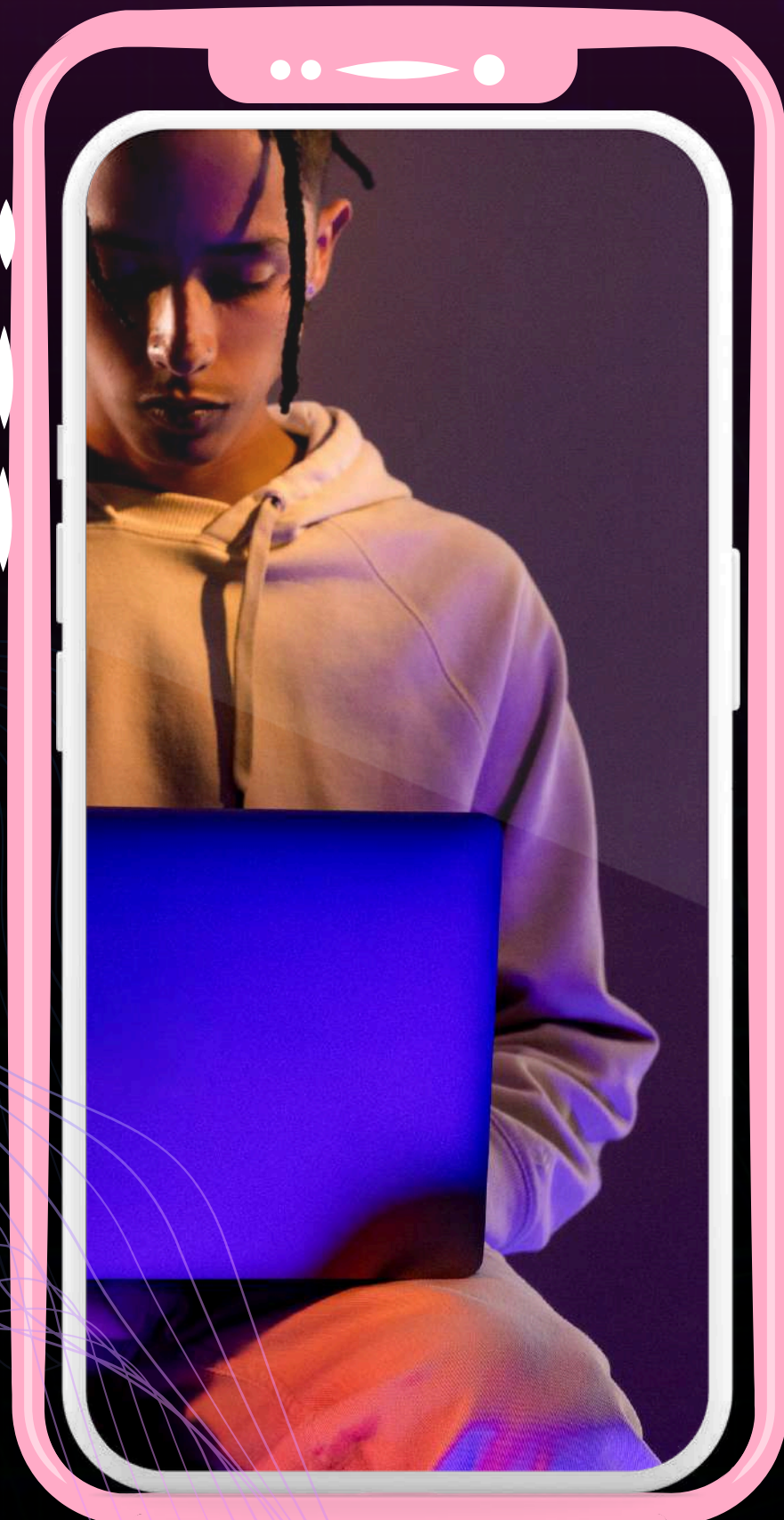
```
C:\Users\Public\Downloads\repo>git add arch.txt
```

```
C:\Users\Public\Downloads\repo>git commit -m "Primer commit en main: escritura 1 en arch.txt"  
[main (root-commit) 068dd16] Primer commit en main: escritura 1 en arch.txt  
1 file changed, 1 insertion(+)  
create mode 100644 arch.txt
```

```
C:\Users\Public\Downloads\repo>echo "Segunda escritura en arch.txt (main)" >> arch.txt
```

```
C:\Users\Public\Downloads\repo>git add arch.txt
```

```
C:\Users\Public\Downloads\repo>git commit -m "Segundo commit en main: escritura 2 en arch.txt"  
[main d173f84] Segundo commit en main: escritura 2 en arch.txt  
1 file changed, 1 insertion(+)
```



Haz varios cambios y commits en una rama feature

```
C:\Users\Public\Downloads\repo>git checkout -b feature
Switched to a new branch 'feature'

C:\Users\Public\Downloads\repo>echo "Cambio 1 en feature" > cambio1.txt

C:\Users\Public\Downloads\repo>git add cambio1.txt

C:\Users\Public\Downloads\repo>git commit -m "Commit 1 en feature"
[feature 1e0bfac] Commit 1 en feature
1 file changed, 1 insertion(+)
create mode 100644 cambio1.txt

C:\Users\Public\Downloads\repo>echo "Cambio 2 en feature" > cambio2.txt

C:\Users\Public\Downloads\repo>git add cambio2.txt

C:\Users\Public\Downloads\repo>git commit -m "Commit 2 en feature"
[feature 1993648] Commit 2 en feature
1 file changed, 1 insertion(+)
create mode 100644 cambio2.txt

C:\Users\Public\Downloads\repo>git log --graph --oneline --all
* 1993648 (HEAD -> feature) Commit 2 en feature
* 1e0bfac Commit 1 en feature
* 6a551b7 (main) Commit 2 en main
* 34843e4 Commit 1 en main
```

Fusiona la rama con git merge --squash para aplanar todos los commits en uno solo.

```
C:\Users\Public\Downloads\repo>git checkout main
Already on 'main'

C:\Users\Public\Downloads\repo>git merge --squash feature
Updating 58024ae..1be24e3
Fast-forward
Squash commit -- not updating HEAD
 cambio1.txt | 1 +
 cambio2.txt | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 cambio1.txt
 create mode 100644 cambio2.txt
```

Verifica el historial de commits antes y después de la fusión para ver la diferencia.

```
C:\Users\Public\Downloads\repo>git log --graph --oneline --all
* b71cebd (HEAD -> main) Fusión con squash de feature: agrega arch1.txt y arch2.txt
| * 1be24e3 (feature) Commit 2 en feature
| * 2846784 Commit 1 en feature
|/
* 58024ae Segundo commit en main: escritura 2 en arch.txt
* c455434 Primer commit en main: escritura 1 en arch.txt
```


Pregunta: ¿Cuándo es recomendable utilizar una fusión squash? ¿Qué ventajas ofrece para proyectos grandes en comparación con fusiones estándar?

03

Cuando los commits no se afectan entre ellos, por ejemplo, crear nuevos archivos en cada commit. Cuando queremos limpiar el historial una vez que la funcionalidad fue finalizada.

Nos ofrece un historial más limpio, podemos hacer una mejor trazabilidad de la funcionalidad y facilita la revisión del código.

PREGUNTA 4

Resolver conflictos en una fusión non-fast-forward



PREGUNTA 5

Comparar los historiales con git log después de diferentes fusiones



PREGUNTA 6

Usando fusiones automáticas y revertir fusiones

Veremos como fusionar automáticamente cambios cuando no hay conflictos y cómo revertir una fusión si hay algún error.

Para esto creamos un repositorio dentro de 'prueba-auto-merge' y con ella creamos y modificamos un archivo file.txt.

```
bxco@bxco-NBD-WXX9:~$ mkdir prueba-auto-merge
bxco@bxco-NBD-WXX9:~$ cd prueba-auto-merge/
bxco@bxco-NBD-WXX9:~/prueba-auto-merge$ git init
Iniciado repositorio Git vacío en /home/bxco/prueba-auto-merge/.git/
bxco@bxco-NBD-WXX9:~/prueba-auto-merge$ echo "Linea 1" > file.txt
bxco@bxco-NBD-WXX9:~/prueba-auto-merge$ git add file.txt
bxco@bxco-NBD-WXX9:~/prueba-auto-merge$ git commit -m 'Agrega linea 1'
[main (commit-raíz) 29e41fb] Agrega linea 1
1 file changed, 1 insertion(+)
create mode 100644 file.txt
bxco@bxco-NBD-WXX9:~/prueba-auto-merge$ echo "Linea 2" >> file.txt
bxco@bxco-NBD-WXX9:~/prueba-auto-merge$ git add file.txt
bxco@bxco-NBD-WXX9:~/prueba-auto-merge$ git commit -m "...linea 2"
[main 9804ee2] ...linea 2
1 file changed, 1 insertion(+)
```


Continuamos creando una nueva rama auto-merge y realizamos un nuevo commit dentro de file.txt

```
bxco@bxco-NBD-WXX9:~/prueba-auto-merge$ git checkout -b auto-merge
Cambiado a nueva rama 'auto-merge'
bxco@bxco-NBD-WXX9:~/prueba-auto-merge$ echo "Linea 3" >> file.txt
bxco@bxco-NBD-WXX9:~/prueba-auto-merge$ git add file.txt
bxco@bxco-NBD-WXX9:~/prueba-auto-merge$ git commit -m "... linea 3"
[auto-merge c2efe67] ... linea 3
1 file changed, 1 insertion(+)
```

Ahora volvemos a main y realizamos cambios no conflictivos en una nueva parte del archivo

```
bxco@bxco-NBD-WXX9:~/prueba-auto-merge$ git checkout main
Cambiado a rama 'main'
bxco@bxco-NBD-WXX9:~/prueba-auto-merge$ echo "" >> file.txt
bxco@bxco-NBD-WXX9:~/prueba-auto-merge$ echo "Footer: Fin del archivo" >> file.txt
bxco@bxco-NBD-WXX9:~/prueba-auto-merge$ git add file.txt
bxco@bxco-NBD-WXX9:~/prueba-auto-merge$ git commit -m "Add footer al archivo file.txt"
[main 05635f9] Add footer al archivo file.txt
1 file changed, 2 insertions(+)
```

Ahora realizaremos una fusión de la rama auto-merge con la main y veremos que se fusionaran sin ningún conflicto causado.

PREGUNTA 7

Fusión remota en un repositorio colaborativo

```
C:\Users\Public\Downloads>git clone https://github.com/lealcharro/nombre-del-repositorio.git
Cloning into 'nombre-del-repositorio'...
warning: You appear to have cloned an empty repository.

C:\Users\Public\Downloads>cd nombre-del-repositorio

C:\Users\Public\Downloads\nombre-del-repositorio>git checkout -b colaboracion
Switched to a new branch 'colaboracion'

C:\Users\Public\Downloads\nombre-del-repositorio>echo "Colaboración remota" > colaboracion.txt

C:\Users\Public\Downloads\nombre-del-repositorio>git add colaboracion.txt

C:\Users\Public\Downloads\nombre-del-repositorio>git commit -m "...."
[colaboracion (root-commit) ed8ac50] ....
 1 file changed, 1 insertion(+)
 create mode 100644 colaboracion.txt

C:\Users\Public\Downloads\nombre-del-repositorio>git push origin colaboracion
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 236 bytes | 118.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/lealcharro/nombre-del-repositorio.git
 * [new branch]      colaboracion -> colaboracion
```


Paso 4. Simula una fusión desde la rama colaboracion en la rama main de otro colaborador. (Puedes usar la interfaz de GitHub para crear un Pull Request y realizar la fusión).

Creamos una nueva branch main, donde habia un archivo textoMain.txt, hicimos un pull desde cmd y un pullRequest de merge desde Github, el cual fue exitoso.

Agregar archivo textoMain.txt a colaboracion #1

Open lealcharro wants to merge 1 commit into `colaboracion` from `main`

Conversation 0 Commits 1 Checks 0 Files changed 1

lealcharro commented now

Aquí estamos agregando el archivo textoMain.txt a colaboracion

Agregar archivo textoMain.txt b58a33e

No conflicts with base branch
Merging can be performed automatically.

Merge pull request You can also merge this with the command line. [View command line instructions.](#)

Add a comment

Agregar archivo textoMain.txt a colaboracion

Merged lealcharro merged 1 commit into `colaboracion` from `main` now

Conversation 0 Commits 1 Checks 0 Files changed 1

lealcharro commented now

Aquí estamos agregando el archivo textoMain.txt a colaboracion

Agregar archivo textoMain.txt

lealcharro merged commit 51a04c3 into `colaboracion` now

Pull request successfully merged and closed
You're all set — the `main` branch can be safely deleted.

PREGUNTA 8

Se simulará un mini proyecto colaborativo entre 4 distintas ramas de trabajo: main, feature1, feature2 y feature3. Donde haremos diferentes cambios y probaremos las formas de merging dadas: --ff, --no-ff y squash.

```
bxco@bxco-NBD-WXX9:~$ mkdir proyecto-actividad-5
bxco@bxco-NBD-WXX9:~$ cd proyecto-actividad-5/
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git init
Inicializado repositorio Git vacío en /home/bxco/proyecto-actividad-5/.git/
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ echo "# Proyecto Actividad 5 - Desarrollo de Software" > README.md
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git add README.md
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git commit -m "Commit inicial del proyecto Actividad 5"
[main (commit-raíz) 7fb4db2] Commit inicial del proyecto Actividad 5
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git branch feature1
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git branch feature2
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git branch feature3
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git checkout feature1
Cambiado a rama 'feature1'
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ touch presentacion.md
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ nano presentacion.md
```

```
GNU nano 6.2                                presentacion.md *
# Presentación
## Saludos formales
Mi nombre es Aarón, soy un estudiante de ciencias de la computación de la Universidad Nacional de Ingeniería
## Hobbys
- Escuchar música
- Programar
- Leer papers sobre inteligencia artificial
## Comida favorita
Mi comida favorita es el caldo de gallina
```


Ya hemos añadido un archivo llamado presentacion.md a la rama feature1, ahora añadiremos otro nuevo archivo llamado chistes.md y añadiremos una lista.

```
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ nano chistes.md
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git add chistes.md
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git commit -m "Añadimos chistes.md en nuestro mini proyecto"
[feature1 6d9d0d1] Añadimos chistes.md en nuestro mini proyecto
1 file changed, 6 insertions(+)
create mode 100644 chistes.md
```

```
GNU nano 6.2 chistes.md *
# Chistes
## Lista de chistes
- ¿Qué le dice un pez a otro? Nada.
- ¿Cómo se llama el campeón de buceo japonés? Tokofondo.
- ¿Por qué las focas miran siempre hacia arriba? Porque ahí están los focos.
- ¿Qué hace una abeja en el gimnasio? Zumba.
```

Pasaremos a la rama feature2, donde realizaremos ciertas modificaciones a los archivos que teníamos al crearlos nuevamente con ligeros cambios.

```
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git checkout feature2
Cambiado a rama 'feature2'
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ touch presentacion.md
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ nano presentacion.md
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git add presentacion.md
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git commit -m "Añadir archivo de presentación.md para el nuevo branch feature2"
[feature2 ed05982] Añadir archivo de presentación.md para el nuevo branch feature2
1 file changed, 6 insertions(+)
create mode 100644 presentacion.md
```

```
GNU nano 6.2 presentacion.md *
# Presentación
## Saludos informales
¿Qué tal con todos? Un gustazo verlos aquí, mi nombre es Aarón, conocido como A.F. Espero hablarles en unos minutitos.
## Genero de música preferida
Jazz
```


Además dentro de la rama feature2, también añadiremos un archivo main.cpp que pueda contar los numeros desde 1 hasta 20.

```
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ touch main.cpp
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ nano main.cpp
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git add main.cpp
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git commit -m "Añadir archivo main.cpp para contar números del 1 al 20 y saludar"
[feature2 48b9834] Añadir archivo main.cpp para contar números del 1 al 20 y saludar
1 file changed, 15 insertions(+)
create mode 100644 main.cpp
```

```
GNU nano 6.2 main.cpp *
#include<iostream>

using namespace std;

int main(){
    int len = 20;
    cout << "Se imprimirán los elementos desde el 1 al 20" << endl;

    for(int i = 0; i < len; i++){
        cout << i + 1 << endl;
    }

    cout << "Muchísimas gracias" << endl;
    return 0;
}
```

Por último nos cambiamos a la rama feature3 y añadimos una nueva versión del archivo chistes.md

```
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git checkout feature3
Cambiado a rama 'feature3'
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ touch chistes.md
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ nano chistes.md
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git add chistes.md
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git commit -m "Añadir archivo de chistes.md modificado en nuestro mini proyecto"
[feature3 b32fb38] Añadir archivo de chistes.md modificado en nuestro mini proyecto
1 file changed, 5 insertions(+)
create mode 100644 chistes.md
```

```
GNU nano 6.2 chistes.md *
# Chistes
## Lista de chistes
- ¿Qué hace una vaca en el suelo? Steakdown.
- ¿Sabes cómo se queda un mago despues de comer? Magordito.
- ¿Por qué los esqueletos no pelean? Porque no tienen agallas.
```


Ahora que las ramas de desarrollo fueron concluidas, procederemos a fusionarlas con la rama main, comenzaremos con feature1 y un merge --ff.

```
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git checkout main
Cambiado a rama 'main'
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git merge --ff feature1
Actualizando 7fb4db2..6d9d0d1
Fast-forward
 chistes.md      | 6 ++++++
 presentacion.md | 9 ++++++++
 2 files changed, 15 insertions(+)
 create mode 100644 chistes.md
 create mode 100644 presentacion.md
```

Notamos que no nos encontramos con ningun conflicto, ahora procederemos a realizar un merge --no-ff con la rama feature2.

```
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git merge --no-ff feature2
Auto-fusionando presentacion.md
CONFLICTO (agregar/agregar): Conflicto de fusión en presentacion.md
Fusión automática falló; arregle los conflictos y luego realice un commit con el resultado.
```

Aquí ocurre un pequeño conflicto en el archivo presentación.md para lo cual la abriremos con editor de texto y corregiremos manualmente

```
GNU nano 6.2                                presentacion.md
# Presentación
<<<<<<< HEAD
## Saludos formales
Mi nombre es Aarón, soy un estudiante de ciencias de la computación de la Universidad Nacional de Ingeniería
## Hobbys
- Escuchar música
- Programar
- Leer papers sobre inteligencia artificial
## Comida favorita
Mi comida favorita es el caldo de gallina
=====
## Saludos informales
¿Qué tal con todos? Un gustazo verlos aquí, mi nombre es Aarón, conocido como A.F. Espero hablarles en unos minutos.
## Genero de música preferida
Jazz
>>>>>>> feature2
```


Solucionaremos dicho conflicto de la siguiente forma y tendremos lo siguiente:

```
GNU nano 6.2                                presentacion.md *
# Presentación
## Saludos formales
Mi nombre es Aarón, soy un estudiante de ciencias de la computación de la Universidad Nacional de Ingeniería
## Saludos informales
¿Qué tal con todos? Un gustazo verlos aquí, mi nombre es Aarón, conocido como A.F. Espero hablarles en unos minutitos.
## Hobbys
- Escuchar música
- Programar
- Leer papers sobre inteligencia artificial
## Comida favorita
Mi comida favorita es el caldo de gallina
## Genero de música preferida
Jazz
█
```

Una vez corregido dicho conflicto, lo añadiremos y commitearemos con todos los cambios para la creación del commit merge

```
fatal: Detenido porque existe un conflicto sin resolver.
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git add presentacion.md
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git merge --no-ff feature2
fatal: No has concluido la fusión (existe MERGE_HEAD).
Por favor, realiza un commit con los cambios antes de fusionar.
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git add presentacion.md
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git commit -m 'Resolucion de conflictos de merge --no-ff con feature2'
[main a1cab81] Resolucion de conflictos de merge --no-ff con feature2
1 archivo añadido: presentacion.md
 565 líneas de código (+/- 0 cambios)
```

Ahora por último realizaremos un merge --squash con la rama feature3

```
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git merge --squash feature3
Auto-fusionando chistes.md
CONFLICTO (agregar/agregar): Conflicto de fusión en chistes.md
Commit de aplastamiento -- no actualizando HEAD
Fusión automática falló; arregle los conflictos y luego realice un commit con el resultado.
```


Nuevamente nuestro merge fallo por un conflicto en chistes.md, así que editamos y corregimos manualmente

```
GNU nano 6.2 chistes.md *
# Chistes
## Lista de chistes
- ¿Qué le dice un pez a otro? Nada.
- ¿Cómo se llama el campeón de buceo japonés? Tokofondo.
- ¿Por qué las focas miran siembre hacia arriba? Porque ahí estan los focos.
- ¿Qué hace una abeja en el gimnasio? Zumba.
- ¿Qué hace una vaca en el suelo? Steakdown.
- ¿Sabes cómo se queda un mago despues de comer? Magordito.
- ¿Por qué los esqueletos no pelean? Porque no tienen agallas.
```

Añadimos nuestra corrección y commiteamos.

```
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ nano chistes.md
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git add chistes.md
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git commit -m 'Corrección de conflicto en chistes.md y merge --squash entre main y feature3'
[main faec1ad] Corrección de conflicto en chistes.md y merge --squash entre main y feature3
1 file changed, 4 insertions(+)
```

Podremos visualizar nuestro log de forma interactiva como el siguiente gráfico.

```
bxco@bxco-NBD-WXX9:~/proyecto-actividad-5$ git log --graph --oneline --all
* faec1ad (HEAD -> main) Corrección de conflicto en chistes.md y merge --squash entre main y feature3
* a1cab81 Resolucion de conflictos de merge --no-ff con feature2
|
| * 48b9834 (feature2) Añadir archivo main.cpp para contar números del 1 al 20 y saludar
| * ed05982 Añadir archivo de presentación.md para el nuevo branch feature2
| * 6d9d0d1 (feature1) Añadimos chistes.md en nuestro mini proyecto
| * 5cd21da Añadir archivo de presentacion.md
|/
| * b32fb38 (feature3) Añadir archivo de chistes.md modificado en nuestro mini proyecto
|/
* 7fb4db2 Commit inicial del proyecto Actividad 5
```


**GRACIAS POR
SU ATENCIÓN**