

1 Clona un repositorio Git con múltiples ramas

- a. Clonar el repositorio de spring projects específicamente spring pet clinic

```
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5 (master)
$ git clone https://github.com/spring-projects/spring-petclinic.git
Cloning into 'spring-petclinic'...
remote: Enumerating objects: 10854, done.
remote: Total 10854 (delta 0), reused 0 (delta 0), pack-reused 10854 (from 1)
Receiving objects: 100% (10854/10854), 10.23 MiB | 2.26 MiB/s, done.
Resolving deltas: 100% (4113/4113), done.
```

- b. Extraer de forma local la rama Springboot3

```
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (main)
$ git checkout -b springboot3 origin/springboot3
branch 'springboot3' set up to track 'origin/springboot3'.
Switched to a new branch 'springboot3'

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (springboot3)
$ ls
LICENSE.txt  build.gradle  docker-compose.yml  gradle/  gradlew*  gradlew.bat  mvnw*  mvnw.cmd  pom.xml  readme.md  settings.gradle  src/

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (springboot3)
$ git branch
* main
* springboot3
```

- c. Uso de git merge --ff

```
cbf80df remove unused import
5ef8190 Move vscode settings up a level
80d03f3 985 remove lastName method param from OwnerController
2a12048 Update javadoc: it incorrectly said to return 'true'
21b9486 Whitespace tidy
dbcd37f Relocate deprecated annotation
abee859 Add devcontainer and gitpod
0c1fa8e Update name of deprecated config property
1db99db Update to Spring Boot 2.7.0
7c91e1b Update gradle build boot version
5ecacef Make postgres data.sql properly idempotent
4381fb6 Use open session in view and populate model attrs
a870b18 Update readme.md
0827bc8 Update readme.md
31e5aa6 Update readme.md
60ddc68 Fixed Typo In PetClinicIntegrationTests
702747f Update jacoco to work with Java 17
604c9c5 Remove unused node.version
28ea39c Upgrade to Maven Checkstyle plugin 3.1.2
d90ac66 Upgrade to spring javaformat 0.0.31
604a9df Upgrade to nohttp 0.0.10
cae097b Upgrade to Spring Boot 2.6.3
9cb8dde Add section on building a container
bdcaa85 Alternative approach with eager fetch
e765e3f Fix lazy loading issue in owner details
8a28801 Add actuators to Gradle build
4725753 Additional tweaks in aggregate model.
b559077 Remove manual id management in child entities
43beff9 Avoid dropping Postgres schema
423a337 Move Visit to owner package (it's aggregate root)

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (springboot3)
$ git checkout -b temp-base 9ecd11
Switched to a new branch 'temp-base'

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (temp-base)
$ git branch
* main
  springboot3
* temp-base

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (temp-base)
$ git switch main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (main)
$ git merge --ff temp-base
Already up to date.
```

- d. Fast Forward no agrega ningun commit del mergeo, dado que main no tuvo mas commits entonces el HEAD de main ahora apunta al ultimo commit de Springboot3

```
main@spring-petclinic:~/Documents/ACTIVIDAD_1/spring-petclinic (main)
$ git log --graph --oneline
* 32abbc (HEAD -> main, origin/main, origin/HEAD) Note about windows shell
* 07c03f Use the java.util.List.of() instead of the ArrayList class
* f0ce27 Associating a <label> with a form control
* 71a1f3 Remove unnecessary @Autowired annotation from the PetTypeFormatter
* 2aa3f9 Fix pet update functionality. - Fixed issue where editing a pet's type or name wasn't persisting - updated processUpdateForm to modify existing pet instead of
* 07c03f Add proper handling of existing pet update
* 32abbc Add DDD to README
* f0ce27 Add dco config
* ea273e Make sure MySQL test app works
* 1f89998 Update Spring Boot to 3.4.2
* 7d78d0 Use Java Streams to sort the Specialty objects by their name
* ea4b97a Remove unused Transactional Import
* 014446d Use the date property of the Visit entity
* ea8f86c Migrate Spring links for issues from Jira to Github
* 2c4a81a feat: add Persian and Turkish localization files for application messages
* 42e2c74 Add a Dockerfile for dev environments other than codespaces
* 00004e Fix occasional stale volume in postgres tests
* 000077 Update timestamps for reproducible builds
* 0aa137 Add new test file -> modify: remove @Transactional - > add: create unit-test file related to PetValidator. -> refactor: move pet objects initialization to @
* 07c03f setup
* 21aa8fb <fix>: rename the ID variable name in constructor.
* 0ba0d5 Further updates for Spring Boot 3.4
* 01f5a4 Versionless webjars (again)
* 01c011 Update to Boot 3.4
* 017562a Implement K8S deployment testing using Kind in Github Actions
* 2c4a81a Refactor code logic -> refactor: remove useless logic cod. -> refactor: delete useless annotation which is provided by Jpa. -> refactor: refactor implement of 'find
* 01c011 Use Jpa to simplify query
* 06889d Refactor OwnerRepository: -> replace: use JpaRepository to replace Repository in OwnerRepository class. -> remove: remove save() method. JpaRepository
* 06889d provides it by default. -> remove: remove @Query because in Owner class the @OneToMany annotation achieved fetch in query. -> refactor: use OptionalOwner. ->
* 06889d receive the result from findById() and if is null, throw IllegalArgumentException. -> refactor: achieve the assert to judge return value in tests. -> add: add name
* 06889d to Owner tag.
* 03028bd Add Kubernetes support
* 01846e Refactor the logic and add unit test -> add: add @NotBlank validation to pet's name. -&to refactor: delete useless code and add unit test to check duplicate Pet
* 01846e new validation logic. -> modify: add id to pet in unit test. -> refactor: classify unit test.
* 01846e Refactor: -> update: delete logic add owner to model because of the comment @ModelAttribute("owner") . - <fix>: add logical judgment in order to avoid 'own
* 01846e ' from form and ownerId from url mismatch.
* 01846e Fix harmless bug. - <fix>: use equals to replace == to compare Integer variable. - <delete>: remove redundant 'toLowerCase()' method and simplify pet look
* 01846e p logic. -> update: rewrite method getName() comments
* 01846e Update Spring Boot release, adding gradle build and cleanup
* 01846e Refactor: fix: temporarily removing accentuation from messages.
* 01846e Update README to refer to test main classes
* 01846e Test: Add support for the Portuguese language.
```

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

Answer: No se recomienda cuando se quiere tener un registro de ramas explícito y detallado, cuando se necesita auditar cambios o cuando se hace integración continua

2 Simula un flujo de trabajo en equipo

- a. Creación de la rama TEAM A para simular un entorno de trabajo

```

$ cd /home/mauricio/Programacion/actividad_5/spring-petclinic (main)
$ git checkout -b feature-team-a
Switched to a new branch 'feature-team-a'

$ cd /home/mauricio/Programacion/actividad_5/spring-petclinic (feature-team-a)
$ echo "Trabajo del equipo A: nueva funcionalidad" > feature.a.txt
$ git add .
warning: in the working copy of 'feature.a.txt', LF will be replaced by CRLF the next time Git touches it
$ git commit -m "añadir funcionalidad del team a en feature-team-a"
[feature-team-a c8cf3] añadir funcionalidad del team a en feature-team-a
1 file changed, 1 insertion(+),
create mode 100644 feature.a.txt

$ cd /home/mauricio/Programacion/actividad_5/spring-petclinic (feature-team-a)

```

- b. Creación de la rama TEAM B para simular un entorno de trabajo

```

$ cd ~/workspace/ACTIVIDAD_5/spring-petclinic (feature-team-a)
$ git switch main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

$ cd ~/workspace/ACTIVIDAD_5/spring-petclinic (main)
$ git checkout -b feature-team-b
Switched to a new branch 'feature-team-b'

$ echo "Trabajo de team B: corrección de estilos" > feature_b.txt

$ cd ~/workspace/ACTIVIDAD_5/spring-petclinic (feature-team-b)
$ git add .
warning: the working copy of 'feature_b.txt', LF will be replaced by CRLF the next time Git touches it

$ git commit -m "Team b añade estilos en feature_b.txt"

$ cd ~/workspace/ACTIVIDAD_5/spring-petclinic (feature-team-b)
$ git merge --no-commit --no-push feature-team-b
[feature-team-b 37aa41d] Team b añade estilos en feature_b.txt
1 file changed, 1 insertion(+), 1 deletion(-)
create mode 100644 feature_b.txt

```

- c. Merge –no–ff de TEAM A y TEAM B y muestra del tree

```

huamam@LAPTOP-159KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (feature-team-B)
$ git switch main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

huamam@LAPTOP-159KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (main)
$ git branch
* main
  feature-team-a
  feature-team-b
  springboot3
  temp-base

huamam@LAPTOP-159KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (main)
$ git merge --no-ff feature-team-a -m "Mergeo de main con el trabajo del equipo A: nueva funcionalidad equipo A"
Merge made by the 'ort' strategy.
 feature_a.txt | 1 +
 1 file changed, 1 insertion(+)
 Create mode 100644 feature_a.txt

huamam@LAPTOP-159KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (main)
$ git merge --no-ff feature-team-b -m "Mergeo de main con trabajo Team B: estilos añadidos por team B"
Merge made by the 'ort' strategy.
 feature_b.txt | 1 +
 1 file changed, 1 insertion(+)
 Create mode 100644 feature_b.txt

huamam@LAPTOP-159KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (main)
$ git tree
2F28434 (HEAD -> main) Mergeo de main con trabajo Team B: estilos añadidos por team B
* c7a441 (feature-team-b) Team b añade estilos en feature_b.txt
*f54afab Mergeo de main con el trabajo del equipo A: nueva funcionalidad equipo A

* c41cf3 (feature-team-a) añadir funcionalidad del team a en feature-team-a

332abc (origin/main, origin/HEAD) Note about windows shell
d7c02f Use the java.util.List.of() instead of the Asset3 class

```

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

ANSWER: Permite tener un historial mas claro y estructurado dando un mensaje en cada merge que se realiza asi como tambien brinda un commit nuevo para dicho merge. Tambien da mayor facilidad en la revision de cambios y nos permite revertir funcionalidades complejas en caso sea necesario.

3 Creación de multiples commits en una rama

- a. Implementación de una nueva rama llamada *feature-squash-demo* donde se haran multiples commit

```
huanabLAPTOP-159KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (feature-squash-demo)
$ ls
IMPLEMENTACION.py  README.md  docker-compose.yml  feature_a.txt  gradlew/  gradlew.bat  mvnw/  pom.xml  src/
LICENSE.txt        build.gradle  feature-demo.txt  feature_b.txt  gradlew.kds/  mvnw.cmd  settings.gradle

huanabLAPTOP-159KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (feature-squash-demo)
$ git status
On branch feature-squash-demo
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        IMPLEMENTACION.py
        feature-demo.txt

no changes added to commit (use "git add" and/or "git commit -a")

huanabLAPTOP-159KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (feature-squash-demo)
$ git add .
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'IMPLEMENTACION.py', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'feature-demo.txt', LF will be replaced by CRLF the next time Git touches it

huanabLAPTOP-159KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (feature-squash-demo)
$ git commit -m "Añadimos IMPLEMENTACION.py README.md y 3 modificaciones a feature-demo"
[feature-squash-demo 64812cf] Añadimos IMPLEMENTACION.py README.md y 3 modificaciones a feature-demo
3 files changed, 5 insertions(+), 165 deletions(-)
create mode 100644 IMPLEMENTACION.py
create mode 100644 feature-demo.txt

huanabLAPTOP-159KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (feature-squash-demo)
$ echo "Cuarta funcionalidad" >> feature-demo

huanabLAPTOP-159KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (feature-squash-demo)
$ git add .
warning: in the working copy of 'feature-demo', LF will be replaced by CRLF the next time Git touches it

huanabLAPTOP-159KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (feature-squash-demo)
$ git commit -m "Creamos nueva rama con la cuarta funcionalidad de feature-demo"
[feature-squash-demo 7e2d323] Creamos nueva rama con la cuarta funcionalidad de feature-demo
1 file changed, 1 insertion(+)
create mode 100644 feature-demo
```

- b. Realizamos el *merge --squash* y dejamos listo en el area de staging

```
huanabLAPTOP-159KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (feature-squash-demo)
$ git switch main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 4 commits.
  (use "git push" to publish your local commits)

huanabLAPTOP-159KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (main)
$ git merge --squash feature-squash-demo
Updating 2f28434..7e2d323
Fast-forward
Squash commit -- not updating HEAD
IMPLEMENTACION.py | 1 +
README.md         | 165 +-----
feature-demo      | 1 +
feature-demo.txt  | 3 +
4 files changed, 6 insertions(+), 165 deletions(-)
create mode 100644 IMPLEMENTACION.py
create mode 100644 feature-demo
create mode 100644 feature-demo.txt

huanabLAPTOP-159KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (main)
$ git tree
7e2d323 (feature-squash-demo) Creamos nueva rama con la cuarta funcionalidad de feature-demo
* 64812cf Añadimos IMPLEMENTACION.py README.md y 3 modificaciones a feature-demo
* 2f28434 (HEAD -> main) Mergeo de main con trabajo Team B: estilos añadidos por team B
|
| c7aa441 (feature-team-b) Team b añade estilos en feature_b.txt
| f54afab Mergeo de main con el trabajo del equipo A: nueva funcionalidad equipo A
|
|
|
* c41cef3 (feature-team-a) añadir funcionalidad del team a en Feature-team-a
232abbc (origin/main, origin/HEAD) Note about windows shell
```

- c. Finalmente realizamos el commit y verificamos el grafo resultante

```
huanan@LAPTOP-159KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 4 commits.
(use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new File:   IMPLEMENTACION.py
        modified:   README.md
        new File:   feature-demo
        new File:   feature-demo.txt

huanan@LAPTOP-159KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (main)
$ git commit -m "Realizamos oficialmente el commit del mergeo squash"
[main d3ca8ed] Realizamos oficialmente el commit del mergeo squash
4 files changed, 6 insertions(+), 165 deletions(-)
create mode 100644 IMPLEMENTACION.py
create mode 100644 feature-demo
create mode 100644 feature-demo.txt

huanan@LAPTOP-159KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/spring-petclinic (main)
$ git tree
* d3ca8ed (HEAD -> main) Realizamos oficialmente el commit del mergeo squash
  * 7e2d323 (feature-squash-demo) Creamos nueva rama con la cuarta funcionalidad de feature-demo
  * 64812cf Añadimos IMPLEMENTACION.py README.md y 3 modificaciones a feature-demo
  * 2f28434 Mergeo de main con trabajo Team B: estilos añadidos por team B
  * c7aa441 (feature-team-b) Team b añade estilos en feature_b.txt
  * f54afab Mergeo de main con el trabajo del equipo A: nueva funcionalidad equipo A
  * c41cef3 (feature-team-a) añadir funcionalidad del team a en feature-team-a
  * 332abbc (origin/main, origin/HEAD) Note about windows shell
```

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

ANSWER: Se recomienda su uso cuando deseamos un historial limpio en general cuando existen cambios o modificaciones pequeñas que no ameritan su propio commit o su vision en el historial, tambien se suele usar en el pull request para un cambio en una rama que se vea como una sola unidad logica

4 Resolver conflictos non-fast-forward

- a. Creacion del repositorio, añadimos main.html en master y creamos la rama feature y sus respectivos commits

```
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5 (master)
$ mkdir prueba-merge-conflict

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5 (master)
$ cd prueba-merge-conflict

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-merge-conflict (master)
$ git init
Initialized empty Git repository in C:/Users/huama/Documents/ACTIVIDAD_5/prueba-merge-conflict/.git/

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-merge-conflict (master)
$ echo "<html><body><h1>Proyecto inicial CC3S2</h1></body></html>" > index.html

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-merge-conflict (master)
$ git add index.html
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-merge-conflict (master)
$ git commit -m "commit inicial del index.html en main"
[master (root-commit) 1dc269e] commit inicial del index.html en main
1 file changed, 1 insertion(+)
create mode 100644 index.html

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-merge-conflict (master)
$ git checkout -b feature-update
Switched to a new branch 'feature-update'

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-merge-conflict (feature-update)
$ echo "<p>.....</p>" >> index.html

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-merge-conflict (feature-update)
$ git add .
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-merge-conflict (feature-update)
$ git commit -m "actualiza ...."
[feature-update 1e6a424] actualiza ....
1 file changed, 1 insertion(+)
```

- b. Volvemos a la rama master e iniciamos el mergeo con feature el cual nos mostrara un conflicto

```
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-merge-conflict (feature-update)
$ git switch master
Switched to branch 'master'

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-merge-conflict (master)
$ echo "<footer>Contacta aqui example@example.com</footer>" >> index.html

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-merge-conflict (master)
$ git add .
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-merge-conflict (master)
$ git commit -m "... index.html"
[master 632bade] ... index.html
1 file changed, 1 insertion(+)

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-merge-conflict (master)
$ git merge --no-ff feature-update
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

c. Solucion del conflicto, commit y finalmente merge correcto

```
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-merge-conflict (master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-merge-conflict (master|MERGING)
$ git add .
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-merge-conflict (master|MERGING)
$ git commit -m "modificacion de index.html"
[master 9599521] modificacion de index.html
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-merge-conflict (master)
$ git tree
* 9599521 (HEAD -> master) modificacion de index.html
|
| * 1e6a424 (feature-update) actualiza ....
| * 632bade .... index.html
|/
* 1dc269e commit inicial del index-htm en main
```

PREGUNTA: ¿Qué pasos adicionales tuviste que tomar para resolver el conflicto?

ANSWER: Fue necesario abrir el archivo en conflicto para solucionar manualmente un error que git no puede solucionar, finalmente el merge fue hecho

5 Comparar los historiales con git log después de diferentes fusiones

- a. Creación de las ramas feature-1 y feature-2 y sus modificaciones

```
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-merge-conflict (master)
$ cd ..
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5 (master)
$ mkdir prueba-compare-merge
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5 (master)
$ cd prueba-compare-merge
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (master)
$ git init
Initialized empty Git repository in C:/Users/huama/Documents/ACTIVIDAD_5/prueba-compare-merge/.git/
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (master)
$ echo "version 1.0" > version.txt
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (master)
$ git add .
warning: in the working copy of 'version.txt', LF will be replaced by CRLF the next time Git touches it
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (master)
$ git commit -m "..."
[master (root-commit) 388c9b0] ....
1 file changed, 1 insertion(+)
create mode 100644 version.txt
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (master)
$ git checkout -b feature-1
Switched to a new branch 'feature-1'
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (feature-1)
$ echo "caracteristica 1 agregada" >> version.txt
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (feature-1)
$ git add .
warning: in the working copy of 'version.txt', LF will be replaced by CRLF the next time Git touches it
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (feature-1)
$ git commit -m "agregar caracteristica 1"
[feature-1 315702c] agregar caracteristica 1
1 file changed, 1 insertion(+)
```

```
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (feature-1)
$ git switch master
Switched to branch 'master'
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (master)
$ git checkout -b feature-2
Switched to a new branch 'feature-2'
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (feature-2)
$ echo "Agrega caracteristica 2" >> version.txt
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (feature-2)
$ git add .
warning: in the working copy of 'version.txt', LF will be replaced by CRLF the next time Git touches it
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (feature-2)
$ git commit -m "Agregacion de caracteristica 2"
[feature-2 42ec2d8] Agregacion de caracteristica 2
1 file changed, 1 insertion(+)
```


- b. Mergeo de las ramas feature-1 y feature-2 incluye solucion del conflicto

```
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (master)
$ git merge feature-1 --ff
Updating 388c9b0..315702c
Fast-forward
 version.txt | 1 +
 1 file changed, 1 insertion(+)
```

```
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (master)
$ git merge feature-2 --no-ff
Auto-merging version.txt
CONFLICT (content): Merge conflict in version.txt
Automatic merge failed; fix conflicts and then commit the result.

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (master|MERGING)
$ git add .

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (master|MERGING)
$ git commit -m "solucion conflicto"
[master 0f8d8f2] solucion conflicto
```

```
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (master)
$ git merge --squash feature-3
Updating 0f8d8f2..701ea62
Fast-forward
Squash commit -- not updating HEAD
 version.txt | 2 ++
 1 file changed, 2 insertions(+)
```

- c. Muestra grafica de todos los tipos de log

```
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (master)
$ git log --graph --oneline --merges --first-parent --branches
* 0f8d8f2 (HEAD -> master) solucion conflicto

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (master)
$ git log --graph --oneline --merges
* 0f8d8f2 (HEAD -> master) solucion conflicto

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (master)
$ git log --graph --oneline --merges --all
fatal: unrecognized argument: --online

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (master)
$ git log --graph --oneline --merges --all
* 0f8d8f2 (HEAD -> master) solucion conflicto

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-compare-merge (master)
$ git tree
* 701ea62 (feature-3) Caracteristica 3 paso 2
* 9d51f35 Caracteristica 3 paso 1
* 0f8d8f2 (HEAD -> master) solucion conflicto
| \
| * 42ec2d8 (feature-2) Agregacion de caracteristica 2
* | 315702c (feature-1) agregar caracteristica 1
| /
* 388c9b0 ....
```

6 Usando fusiones automáticas y revertir fusiones

- a. Agregacion de la linea 1 y 2 en el main

```
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5 (master)
$ mkdir prueba-auto-merge

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5 (master)
$ cd prueba-auto-merge

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-auto-merge (master)
$ git init
Initialized empty Git repository in C:/Users/huama/Documents/ACTIVIDAD_5/prueba-auto-merge/.git/

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-auto-merge (master)
$ echo "linea 1" > file.txt

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-auto-merge (master)
$ git add .
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-auto-merge (master)
$ git commit -m "agregar linea 1"
[master (root-commit) de57ee0] agregar linea 1
1 file changed, 1 insertion(+)
create mode 100644 file.txt

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-auto-merge (master)
$ echo "linea 2" >> file.txt

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-auto-merge (master)
$ git add .
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-auto-merge (master)
$ git commit -m "... linea 2"
[master d26f596] ... linea 2
1 file changed, 1 insertion(+)
```

- b. Se crea la rama auto-merge y se agrega la linea 3

```
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-auto-merge (master)
$ git checkout -b auto-merge
Switched to a new branch 'auto-merge'

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-auto-merge (auto-merge)
$ echo "Linea 3" >> file.txt

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-auto-merge (auto-merge)
$ git add .
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it

huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-auto-merge (auto-merge)
$ git commit -m "...linea 3"
[auto-merge fd6b0ac] ...linea 3
1 file changed, 1 insertion(+)
```

- c. Aplicamos auto-merge en el main y finalmente mostramos el resultado de las ramas

```
huama@LAPTOP-1S9KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-auto-merge (master)
$ git tree
* 2328292 (HEAD -> master) last modification
|
| * fd6b0ac (auto-merge) ...linea 3
| * 2336dcb Agregamos el final del archivo file.txt
|/
* d26f596 ... linea 2
* de57ee0 agregar linea 1
```

- d. Usamos git revert para revertir el ultimo commit que es la fusion

```
huama@LAPTOP-159KBL6U MINGW64 ~/Documents/ACTIVIDAD_5/prueba-auto-merge (master)
$ git log --graph --oneline
* a1ed1c5 (HEAD -> master) Reapply "last modification"
* d5f9870 Revert "last modification"
* 2328292 last modification
\
| * fd6b0ac (auto-merge) ...linea 3
* | 2336dcb Agregamos el final del archivo file.txt
|/
≈ d26f596 ... linea 2
≈ de57ee0 agregar linea 1
```

PREGUNTAS: ¿Cuándo usarías un comando como git revert para deshacer una fusión? ¿Qué tan útil es la función de fusión automática en Git?

ANSWERS: Usarías git revert para deshacer una fusión cuando quieras deshacer los efectos de una fusión específica sin alterar el historial de commits. La fusión automática en Git es muy útil para integrar cambios de diferentes ramas cuando no hay conflictos, ya que permite combinar fácilmente el trabajo sin intervención manual.