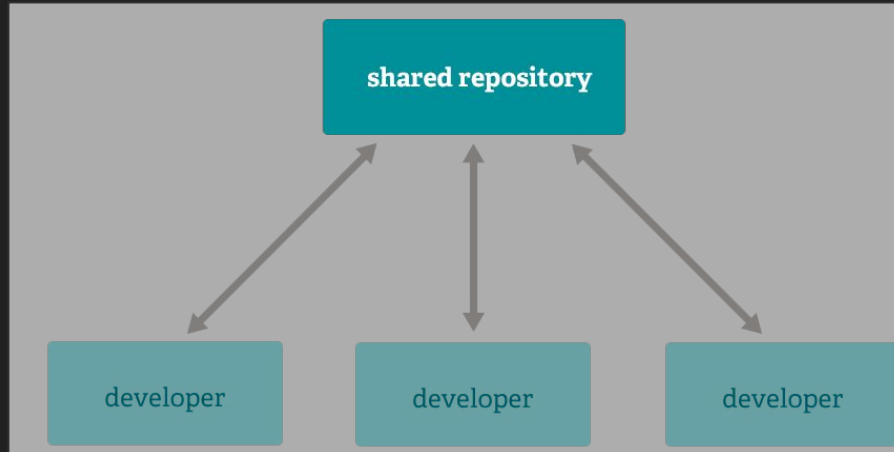


Git

Crash Course

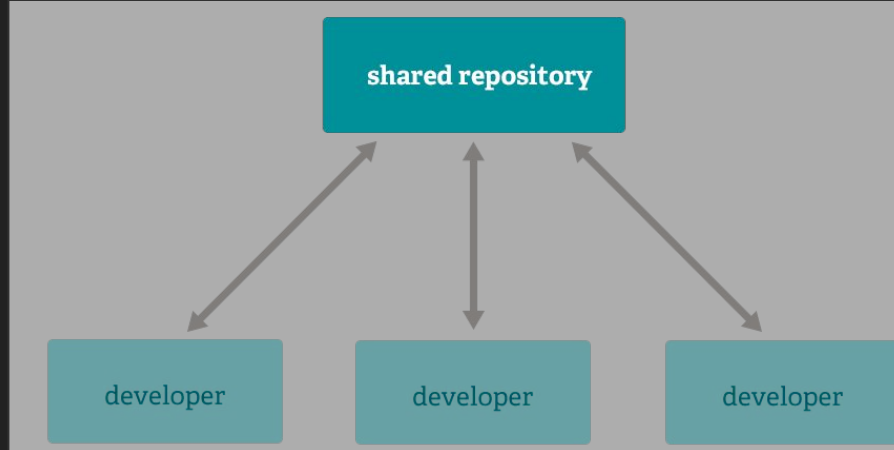
¿Qué? ¿Por qué? ¿Cómo?

- Software de **control de versiones** (Linus Torvalds, 2005)
- Permite desarrollo **no-lineal** y una **gestión distribuida** del proyecto (cada usuario tiene una copia local del proyecto).



¿Gestión distribuida?

- El trabajo local no afecta a los demás (permite experimentar diferentes soluciones)
- Tracking de cada línea modificada de cada archivo

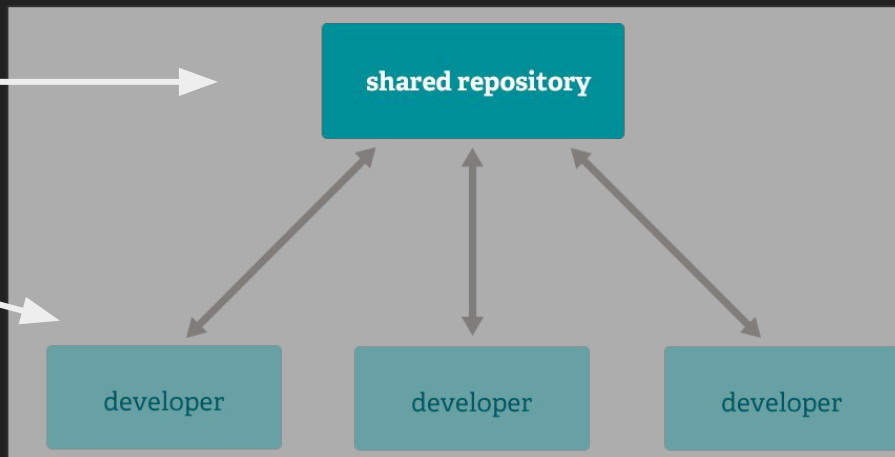


Conceptos

Repositorio: Contenedor/folder de un conjunto de archivos correspondientes a diferentes versiones de un proyecto.

Repositorio remoto:

Repositorio local:



Conceptos

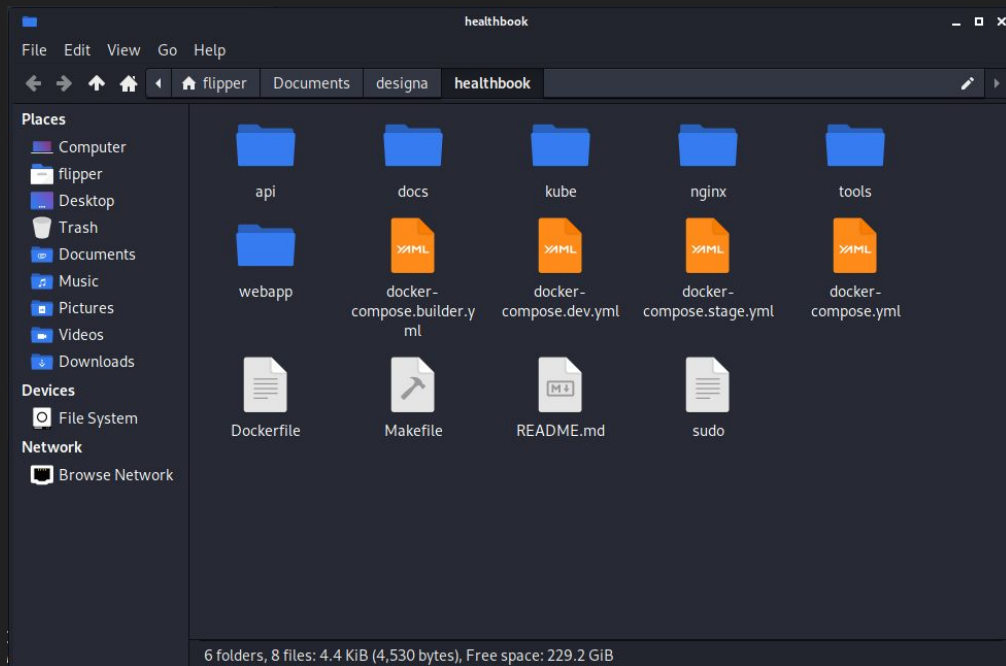
Repositorio remoto:

The screenshot shows the GitLab web interface for a repository named 'designamx'. The left sidebar contains a menu with options: Project information, Repository, Issues (0), Merge requests (0), CI/CD, Security & Compliance, Deployments, Monitor, Infrastructure, Analytics, Wiki, and Settings. The main content area displays the repository details, including the project ID (829488), a 'Leave project' link, and statistics (1 Commit, 1 Branch, 0 Tags, 205 KB Files, 256 KB Storage). A notification for 'Auto DevOps' is visible, stating it will automatically build, test, and deploy applications based on predefined CI/CD configurations. Below this, there are buttons for 'History', 'Find file', 'Web IDE', and 'Clone'. A section titled 'add repo' shows the repository was authored 5 years ago. At the bottom, there are buttons for 'Upload File', 'Add README', 'Add LICENSE', 'Add CHANGELOG', 'Add CONTRIBUTING', 'Add Kubernetes cluster', 'Set up CI/CD', and 'Configure Integrations'. A table lists the repository's files and their last commit details.

Name	Last commit	Last update
app	add repo	5 years ago
assets	add repo	5 years ago
.gitignore	add repo	5 years ago
app.js	add repo	5 years ago
bowerjson	add repo	5 years ago

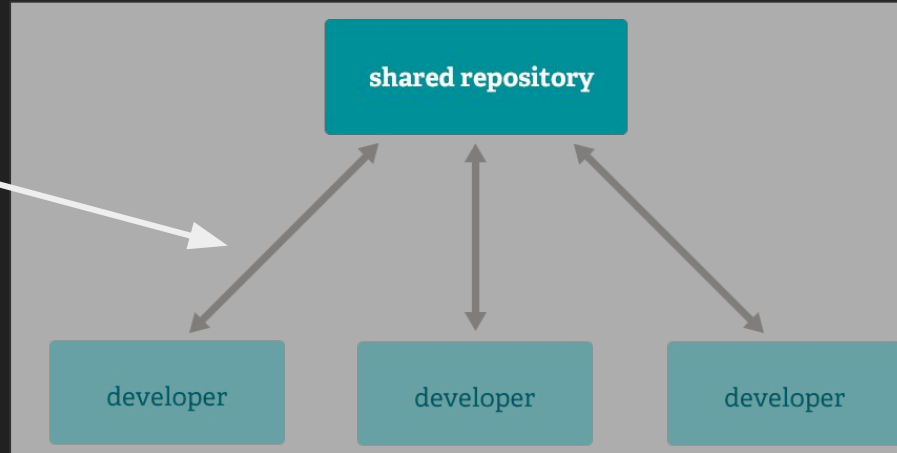
Conceptos

Repositorio local:



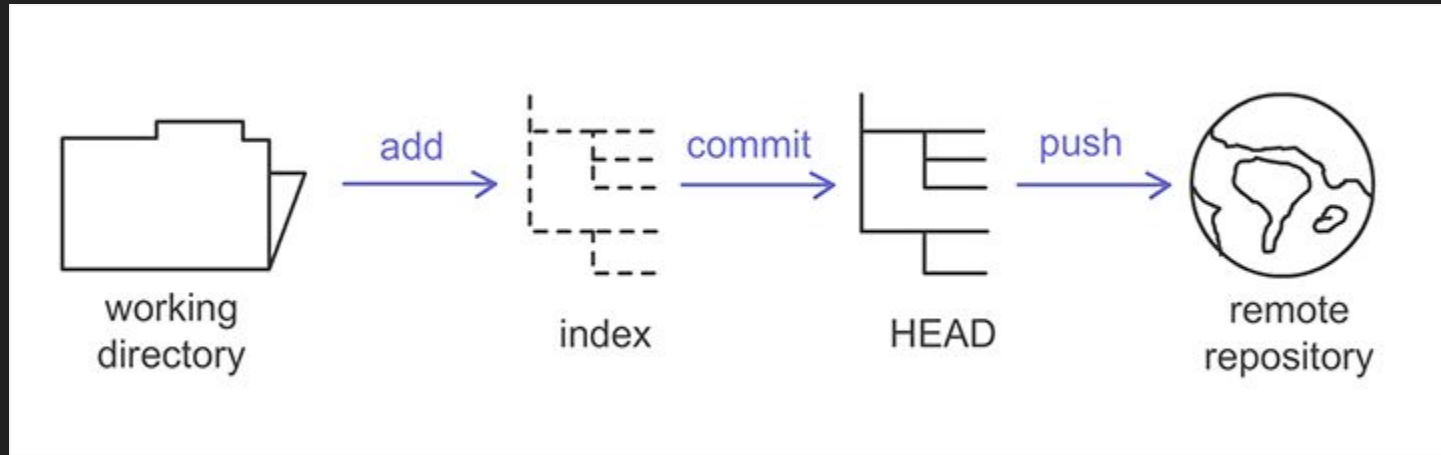
Conceptos

Commit:

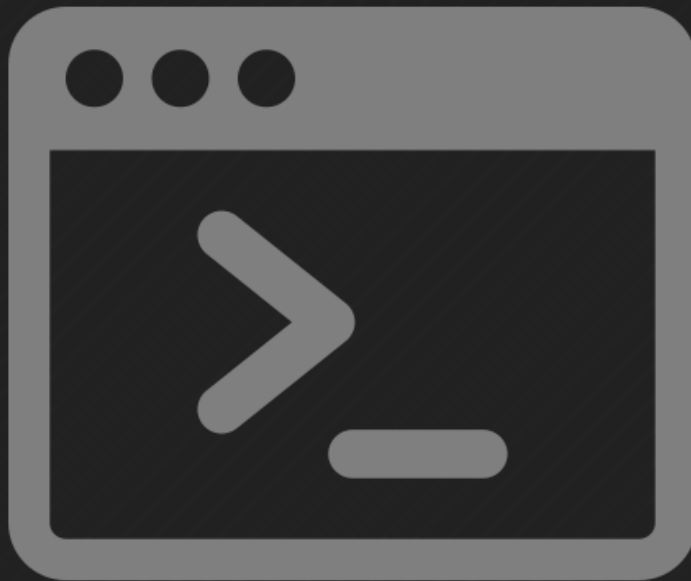


Flujo de trabajo

4 estados: Workspace, index/stage, HEAD, remote

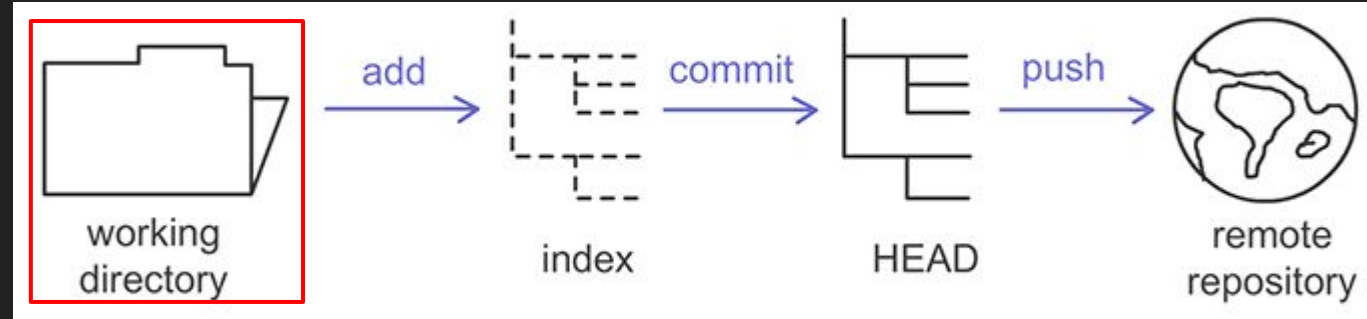


I ain't afraid of no terminal... right?



Flujo de trabajo

Working directory



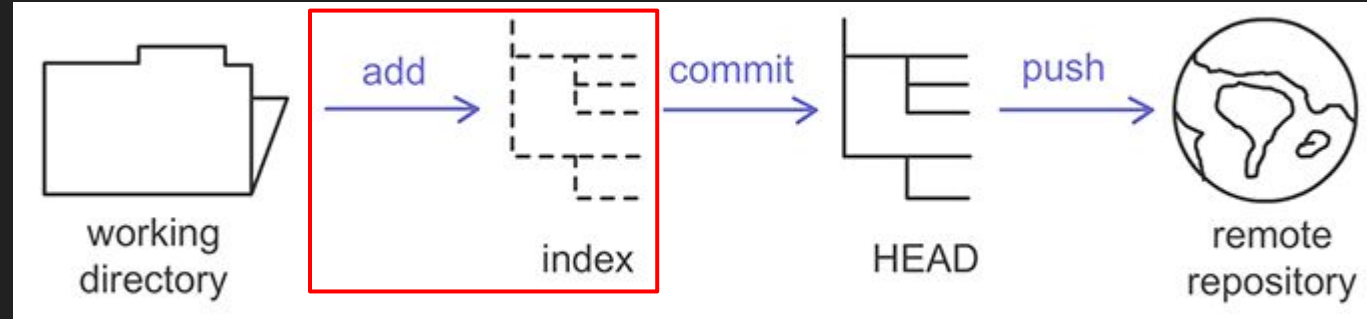
```
flipper@sud@ku-solver (master) $ git status
On branch master
Your branch is up to date with 'origin/master'.

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:   src/game_logic.cpp
        modified:   src/grid.cpp

no changes added to commit (use "git add" and/or "git commit -a")
flipper@sud@ku-solver (master) $
```

Flujo de trabajo

Stage/index



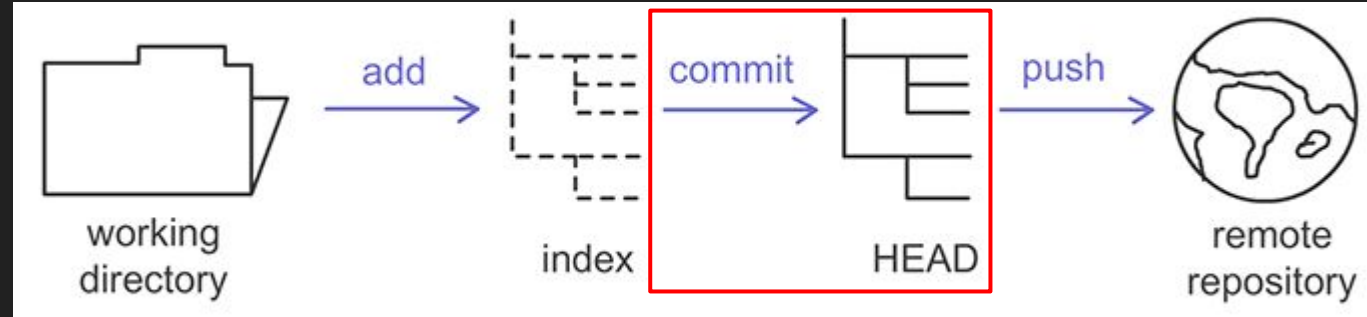
```
flipper@sud0ku-solver (master) $ git add src/game logic.cpp
flipper@sud0ku-solver (master) $ git add src/grid.cpp
flipper@sud0ku-solver (master) $ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   src/game_logic.cpp
    modified:   src/grid.cpp

flipper@sud0ku-solver (master) $
```

Flujo de trabajo

HEAD

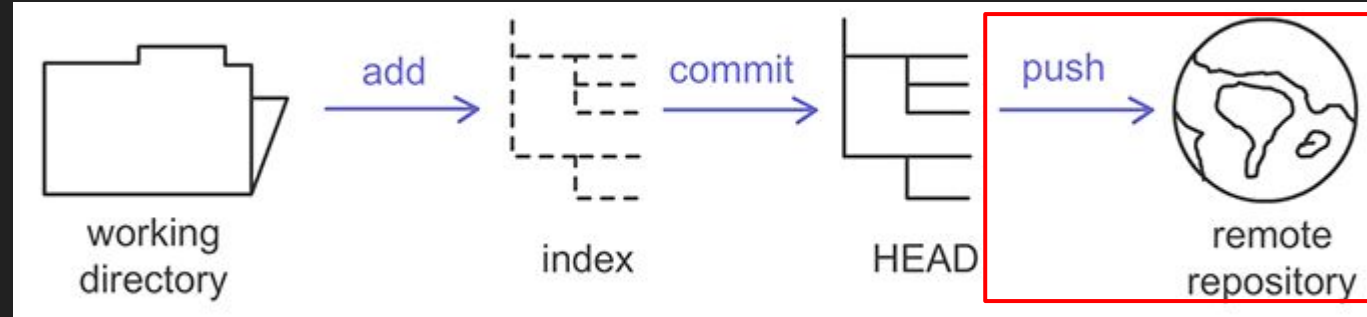


```
flipper@sud0ku-solver (master) $ git commit -m'add visualization of solve progress'
[master ca591c8] add visualization of solve progress
 2 files changed, 7 insertions(+), 1 deletion(-)
flipper@sud0ku-solver (master) $ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
flipper@sud0ku-solver (master) $
```

Flujo de trabajo

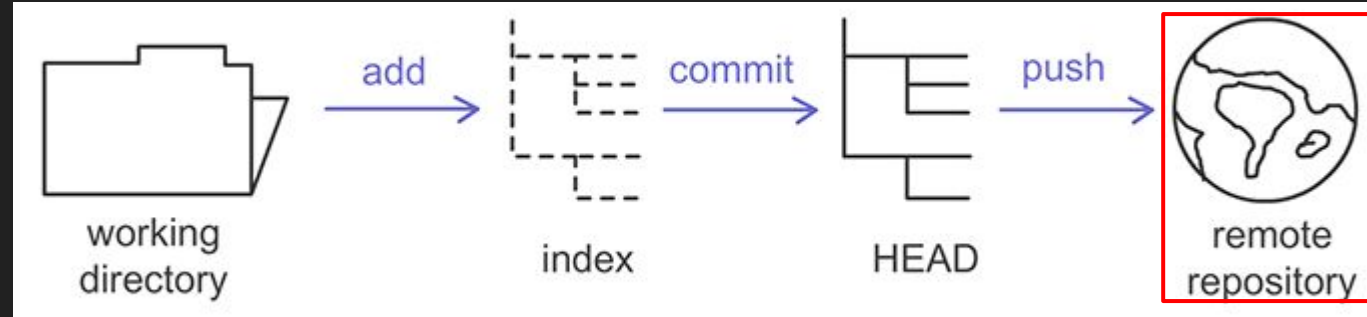
Remote



```
flipper@sud0ku-solver (master) $ git push origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 567 bytes | 567.00 KiB/s, done.
Total 5 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:mitsiu-carreno/sud0ku-solver.git
   6ef8a74..ca591c8 master -> master
flipper@sud0ku-solver (master) $
```

Flujo de trabajo

Remote



master ▾



Commits on Aug 26, 2021

add visualization of solve progress



mitsiu-carreno committed 4 minutes ago



ca591c8



Flujo de trabajo

Remote [Fuente](#)

add visualization of solve progress

Browse files

master

mitsiu-carreno committed 12 minutes ago

1 parent 6ef8a74 commit ca591c8f73c9cd6a5f45dcbe627fba14e3fdc3a

Showing 2 changed files with 7 additions and 1 deletion.

Unified

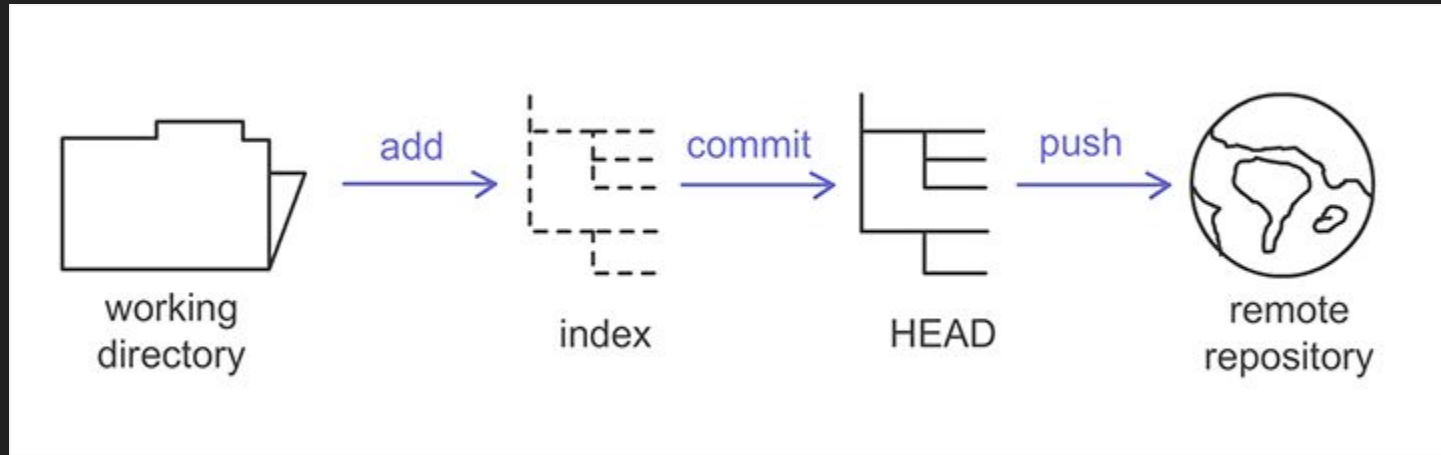
Split

6 src/game_logic.cpp

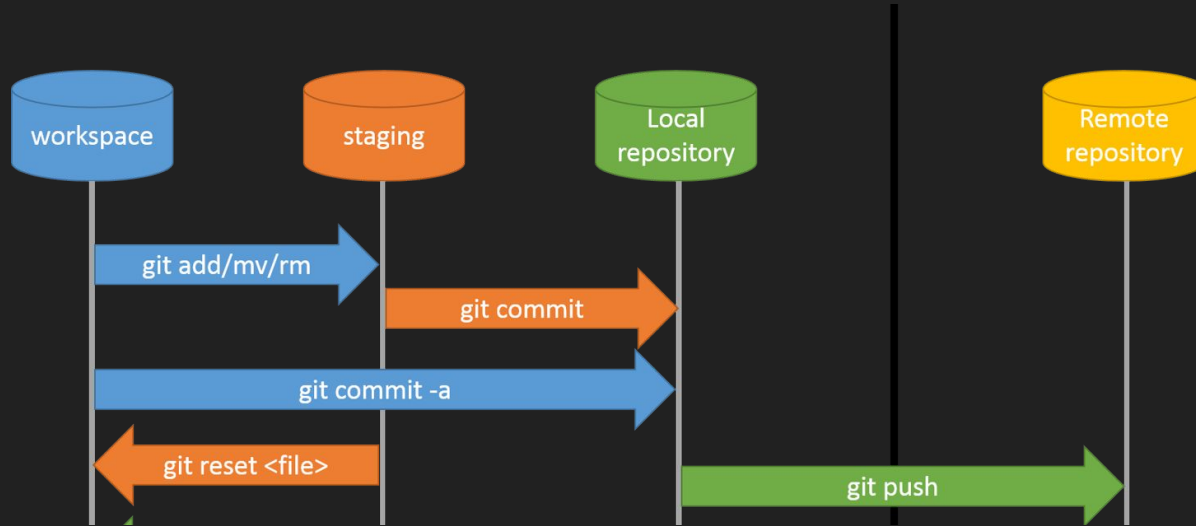
```
@@ -397,6 +397,12 @@ namespace game_logic{
397 397         if(ValidNumber(meta.grid, node_row, node_col, reinterpret_cast<cell::Cell*>(meta.solution_path[solution_path_index]->value)->backlog_values[next_val
398 398
399 399         reinterpret_cast<cell::Cell*>(meta.solution_path[solution_path_index]->value)->current_value = reinterpret_cast<cell::Cell*>(meta.solution_path[sc
400 +
401 +         clear();
402 +         grid::PrintGrid(meta.grid, false);
403 +         refresh();
404 +         napms(1);
405 +
406 406         ++solution_path_index;
```

Flujo de trabajo

4 estados: Workspace, stage/index, HEAD, remote



Flujo de trabajo



Instalación

<https://git-scm.com/downloads>

<https://gitlab.com>

Review

Crear un repositorio local
git init

Clonar un repositorio remoto
git clone <repository-URL>

Obtener el estatus actual del repositorio
git status

Agregar un archivo al repositorio para ser
trackeado
git add <file>

Agregar todos los archivos al repositorio
git add --all

Crear un nuevo commit y agregar un mensaje
git commit -m '<Explain changes>'

Obtener los cambios del repositorio remoto
git pull <repository> <branch>

Compartir cambios al repositorio remoto
git push <repository> <branch>

Práctica

1. Ayudarme a crear un repositorio remoto
2. Clonar dicho repositorio
3. Generar de manera individual un archivo cuyo título sea su nombre, dentro del archivo escribir un Hola mundo en el lenguaje de su preferencia
4. Subir sus cambios al repositorio remoto (commits con nombres descriptivos)
5. Actualizar su repositorio local con los Hola mundo de **todos** sus compañeros

Conflictos!

Casos:

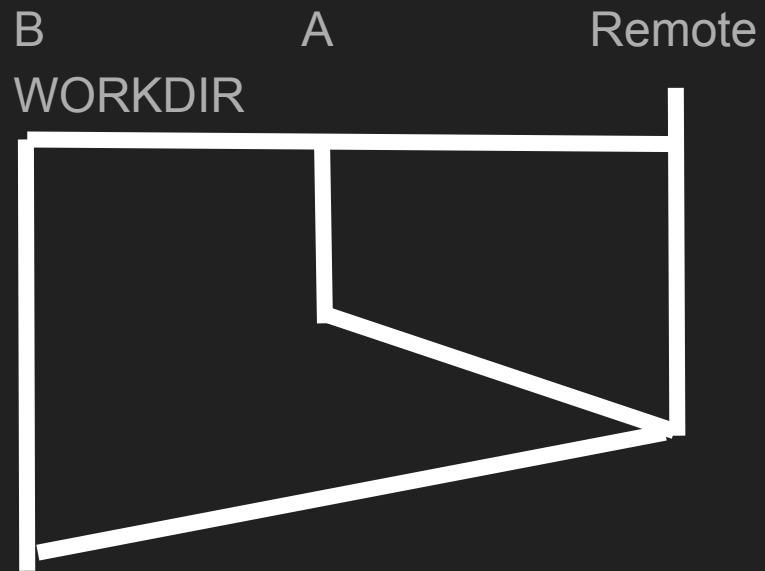
1. Tienen cambios en workspace y cuando hacen pull otro commit modificó los mismos archivos.
- 2.- Hacen un nuevo commit y cuando hace pull otro commit modificó los mismos archivos.

Conflictos!

Caso 1) Tienen cambios en workspace y cuando hacen pull otro commit modificó los mismos archivos.

Prog A y Prog B tienen archivo X en versión 1

1. [Prog A] Modifica archivo X generando versión 2 y actualiza en repositorio remoto
2. [Prog B] Trabaja en workspace el archivo X pero sigue con la versión 1
3. [Prog B] Hace pull



Conflictos!

Caso 1) Tienen cambios en workspace y cuando hacen pull otro commit modificó los mismos archivos.

[Prog B]

```
flipper@sud0ku-solver ((6ef8a74...)) $ git pull origin master
From github.com:mitsiu-carreno/sud0ku-solver
 * branch          master      -> FETCH_HEAD
Updating 6ef8a74..ca591c8
error: Your local changes to the following files would be overwritten by merge:
    src/game_logic.cpp
Please commit your changes or stash them before you merge.
Aborting
flipper@sud0ku-solver ((6ef8a74...)) $ █
```

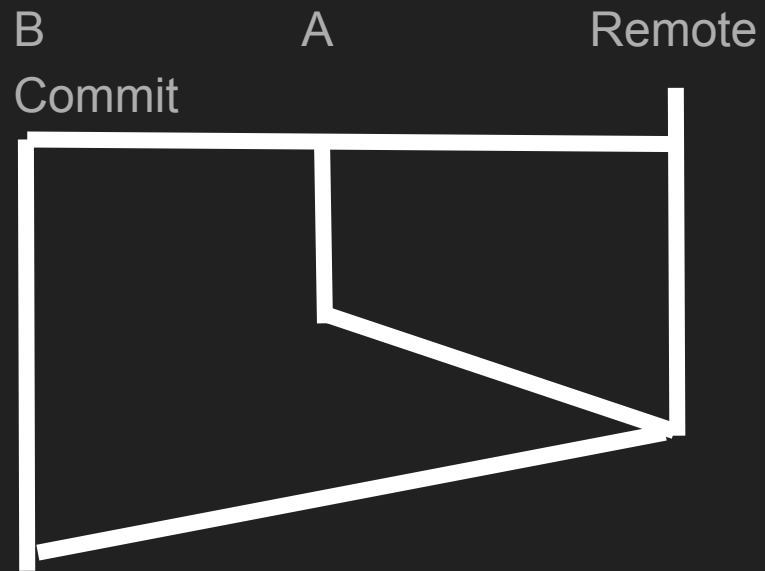
[Prog B] Puede a) hacer commit b) hacer stash

Conflictos!

Caso 2) Hacen un nuevo commit y cuando hace pull otro commit modificó los mismos archivos.

Prog A y Prog B tienen archivo X en versión 1

1. [Prog A] Modifica archivo X generando versión 2 y actualiza en repositorio remoto
2. [Prog B] También modifica el archivo X generando su propia versión 2 (diferente de la del prog A)
3. [Prog B] Hace commit, pull



Conflictos!

Caso 2) Hacen un nuevo commit y cuando hace pull otro commit modificó los mismos archivos.

[Prog B]

```
flipper@sud0ku-solver ((a7ffd5f...)) $ git pull origin master
From github.com:mitsiu-carreno/sud0ku-solver
 * branch          master      -> FETCH_HEAD
Auto-merging src/grid.cpp
CONFLICT (content): Merge conflict in src/grid.cpp
Automatic merge failed: fix conflicts and then commit the result.
flipper@sud0ku-solver ((a7ffd5f...)|MERGING) $ git status
```

Conflictos!

Caso 2) Hacen un nuevo commit y cuando hace pull otro commit modificó los mismos archivos.

[Prog B]

```
flipper@sud0ku-solver ((a7ffd5f...)|MERGING) $ git status
HEAD detached from 6ef8a74
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Changes to be committed:
  modified:   src/game_logic.cpp

Unmerged paths:
  (use "git add <file>..." to mark resolution)
  both modified:   src/grid.cpp

flipper@sud0ku-solver ((a7ffd5f...)|MERGING) $
```

Conflictos!

Caso 2) Hacen un nuevo commit y cuando hace pull otro commit modificó los mismos archivos.

[Prog B]

```
106         if(PrintDivisionHere(g_row)){
107 <<<<<<< HEAD
108         char divider[constants::kGridSize + constants::kGridSection +2];
109 =====
110         char divider[constants::kGridSize + constants::kGridSection +1];
111 >>>>>>> ca591c8f73c9cd6a5f45dcbef627fba14e3fdc3a
112         std::fill(divider, divider + constants::kGridSize + constants::kGridSection
113         if(show_guides){
```

Conflictos!

Caso 2) Hacen un nuevo commit y cuando hace pull otro commit modificó los mismos archivos.

[Prog B]

```
106     if(PrintDivisionHere(g_row)){
107 <<<<<< HEAD
108         char divider[constants::kGridSize + constants::kGridSection +2];
109 =====
110         char divider[constants::kGridSize + constants::kGridSection +1];
111 >>>>>> ca591c8f73c9cd6a5f45dcbef627fba14e3fdc3a
112         std::fill(divider, divider + constants::kGridSize + constants::kGridSection, ' ');
113         if(show_guides){
```

[Prog B] Resolución de conflicto

```
105     printf("\n");
106     if(PrintDivisionHere(g_row)){
107         char divider[constants::kGridSize + constants::kGridSection +1];
108         std::fill(divider, divider + constants::kGridSize + constants::kGridSection, ' ');
109         if(show_guides){
```

Conflictos!

Caso 2) Hacen un nuevo commit y cuando hace pull otro commit modificó los mismos archivos.

```
flipper@sud0ku-solver ((a7ffd5f...)|MERGING) $ git status
HEAD detached from 6ef8a74
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)
```

Antes

```
Changes to be committed:
  modified:   src/game_logic.cpp

Unmerged paths:
  (use "git add <file>..." to mark resolution)
  both modified:   src/grid.cpp
```

Después

```
flipper@sud0ku-solver ((a7ffd5f...)|MERGING) $ git status
HEAD detached from 6ef8a74
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
  modified:   src/game_logic.cpp
  modified:   src/grid.cpp
```

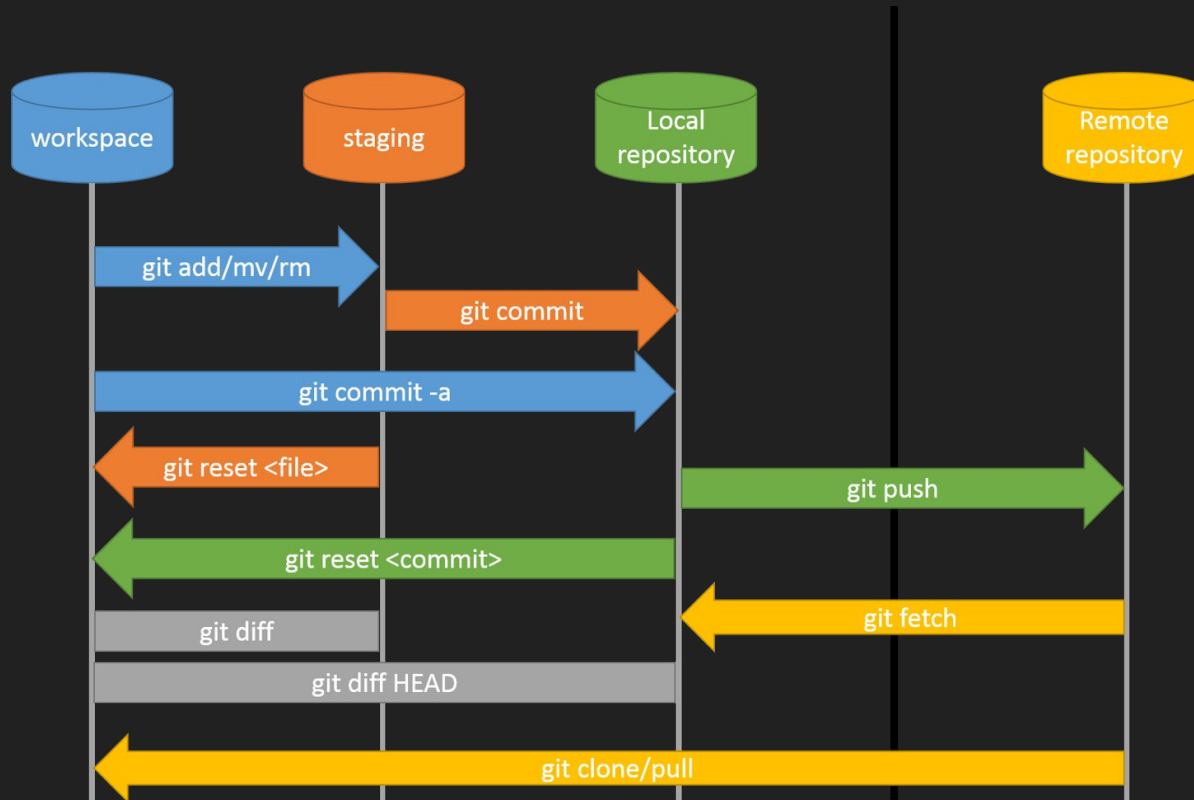
Conflictos!

Caso 2) Hacen un nuevo commit y cuando hace pull otro commit modificó los mismos archivos.

[Prog B]

```
flipper@sud0ku-solver ((a7ffd5f...)|MERGING) $ git commit -m 'merge'
```


Flujo de trabajo

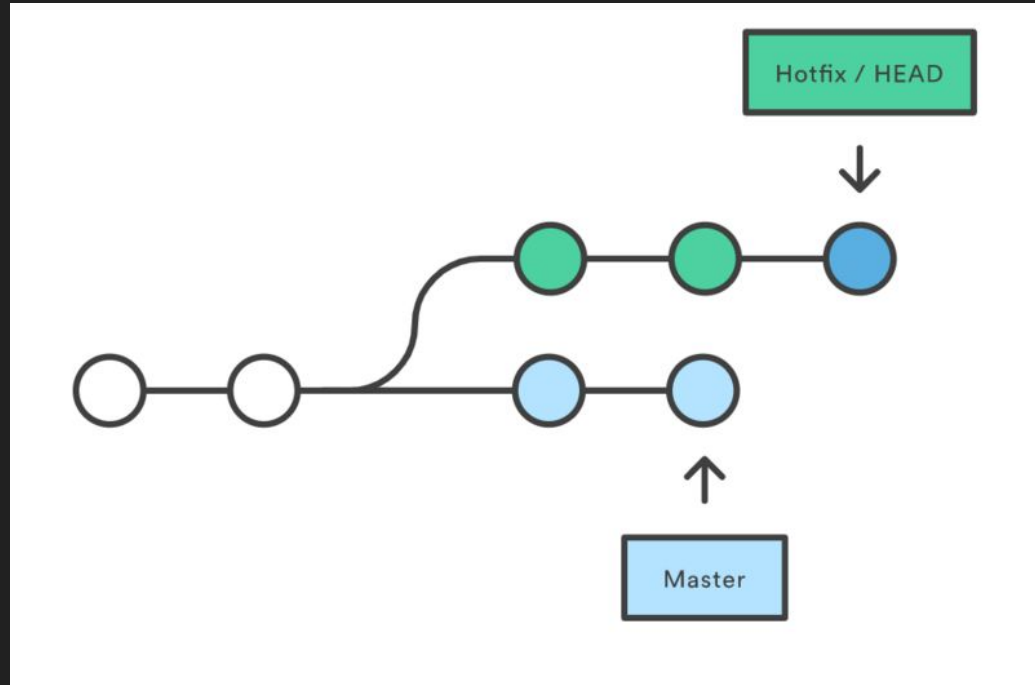


Práctica

6. Esperen a que modifique sus Hola mundo
7. Sin hacer pull, modifiquen sus propios Hola mundo para que imprima en pantalla "Hello world by Designa"
8. Guarden sus cambios, hagan commit y hagan pull
9. Resuelvan los conflictos

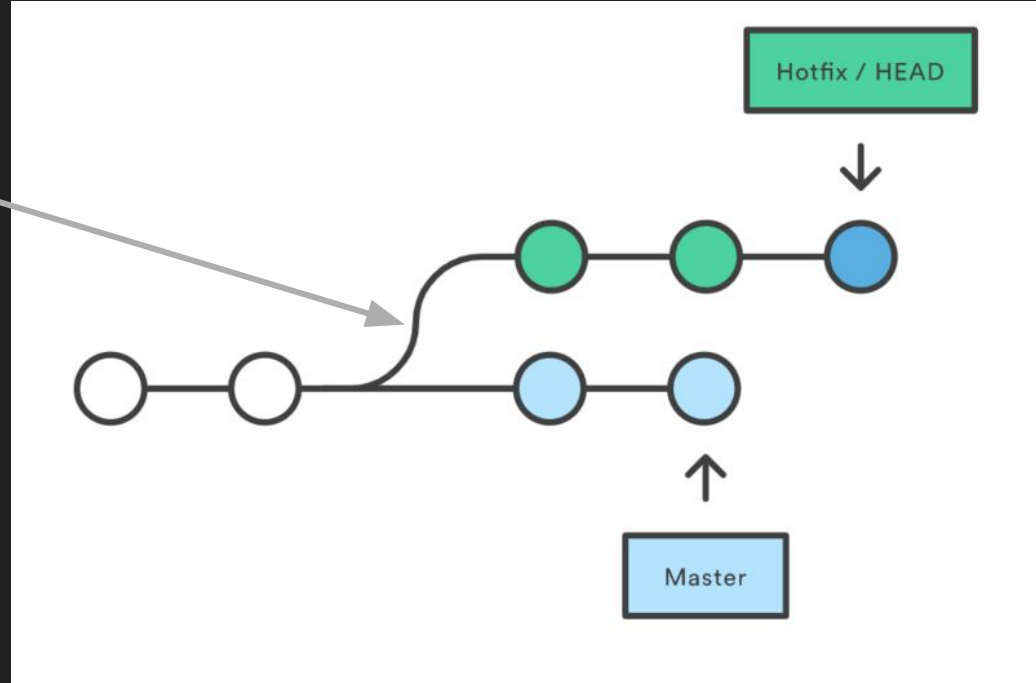
Conceptos

Branch:



Commands

`git checkout -b <new-branch>`



Commands

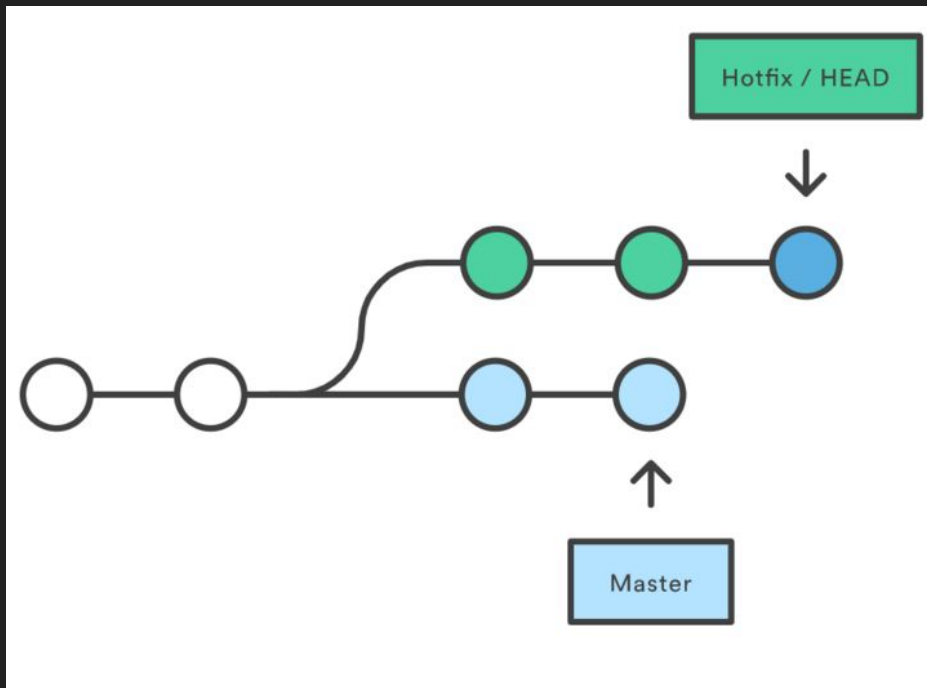
`git branch --all`

* Hotfix

Master

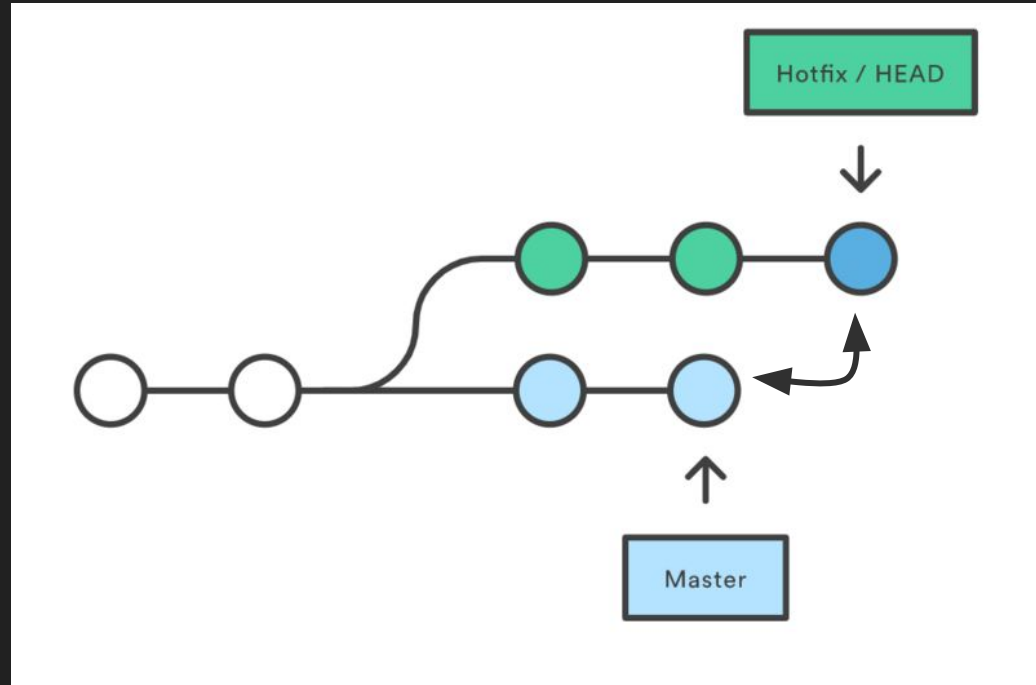
`remotes/origin/HEAD -> origin/Hotfix`

`remotes/origin/master`



Commands

`git checkout <branch>`



Commands

Current branch: Feature

git checkout main

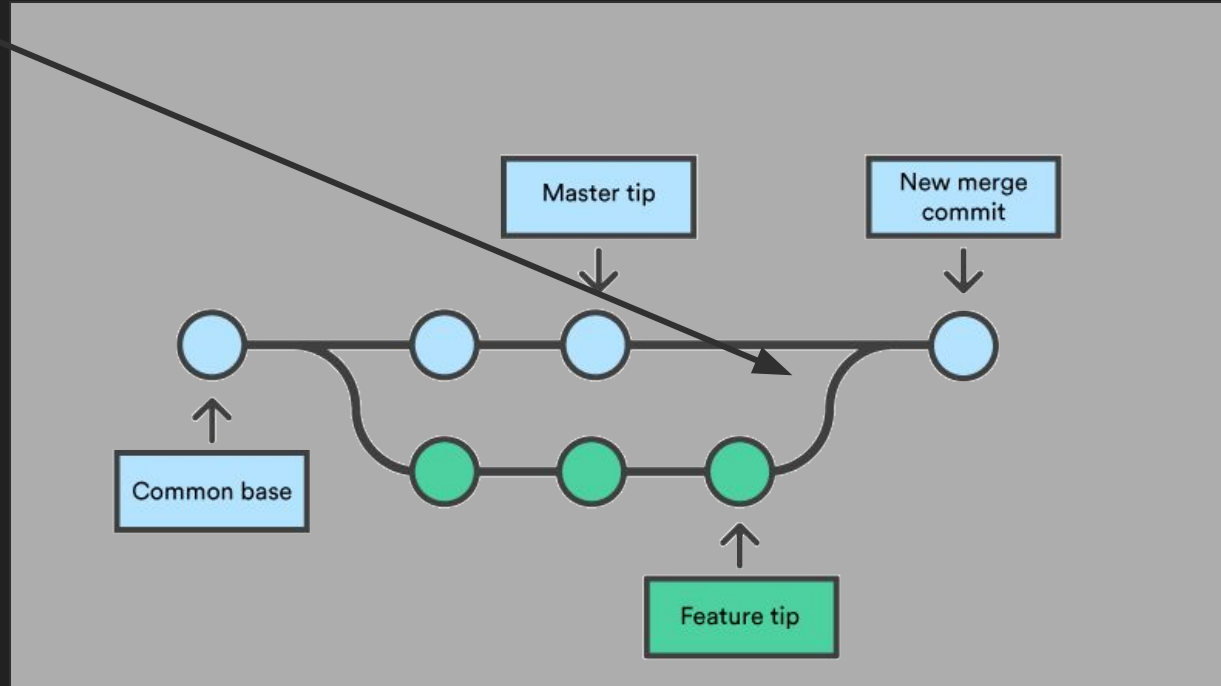
git pull origin main

git merge <branch>

git status (branch main con
cambios de feature)

git push origin main

git branch -d <branch>



Conceptos

Branches



Review

Crear un repositorio local
git init

Clonar un repositorio remoto
git clone <repository-URL>

Obtener el estatus actual del repositorio
git status

Agregar un archivo al repositorio para ser
trackeado
git add <file>

Agregar todos los archivos al repositorio
git add --all

Crear un nuevo commit y agregar un mensaje
git commit -m '<Explain changes>'

Obtener los cambios del repositorio remoto
git pull <repository> <branch>

Compartir cambios al repositorio remoto
git push <repository> <branch>

Review

Crear una nueva rama, y cambiar el workspace
git checkout -b <new-branch>

Mostrar todas las ramas locales y remotas
git branch --all

Cambiar el workspace de rama
git checkout <branch>

Unir dos ramas (genera nuevo commit)
git merge <branch>

Mostrar los cambios entre workspace y HEAD
git diff <file>

Mostrar el registro de commits
git log

Revertir los cambios del workspace a HEAD
git checkout <file>

La documentación es su amiga :)

<https://www.git-scm.com/docs>

Práctica

10. Crear una rama con su nombre

11. Subir sus documentos de estadías en su rama personal

Gracias

In case of fire



<https://c.google.com/c/stefan-schubert/>



1. `git commit`



2. `git push`



3. leave building