

# Introduction à GIT



Everytime I do a git pull  
& there is a conflict





Swift

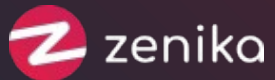
Flutter



Président 2022-2023



Master 2023



Nicolas Marinho  
Ingénieur Logiciel



QUARKUS



Vue



Kotlin

Capgemini



Sheikoop



Président actuel

**MIAGE** RÉSEAU  
DES MIAGE  
DE FRANCE  
Grenoble

Responsable P&C M1 MIAGE



**ATRALTECH**



Nicolas Marinho

Ingénieur Logiciel

Président 2022-2023



**MIAGE** RÉSEAU  
DES MIAGE  
DE FRANCE  
Grenoble

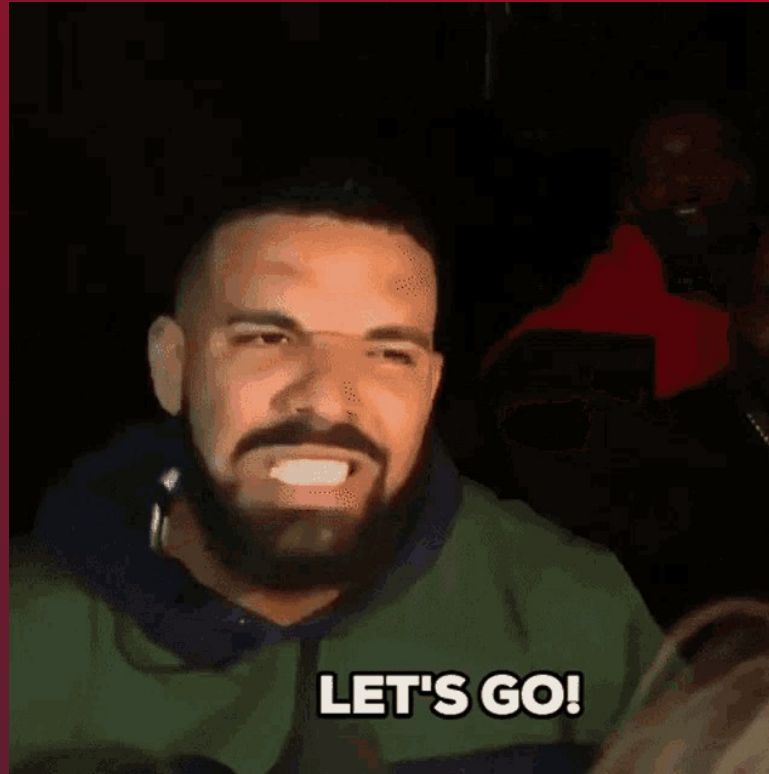
Master 2023

**zenika**

**MIAGE** RÉSEAU  
DES MIAGE  
DE FRANCE  
Grenoble

**UGA**  
IM<sup>2</sup>AG  
UFR  
Université  
Grenoble Alpes

# On s'y met ?





# git





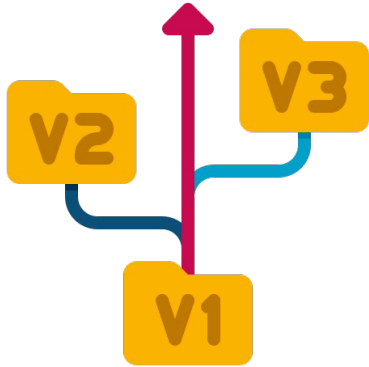
Linus Torvalds

En avait marre de CVS

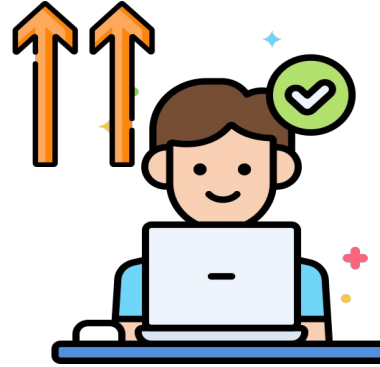


Créé de manière libre

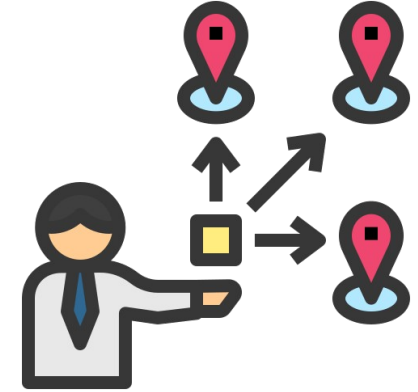




**Versionning**



**Travail collaboratif  
en parallèle**

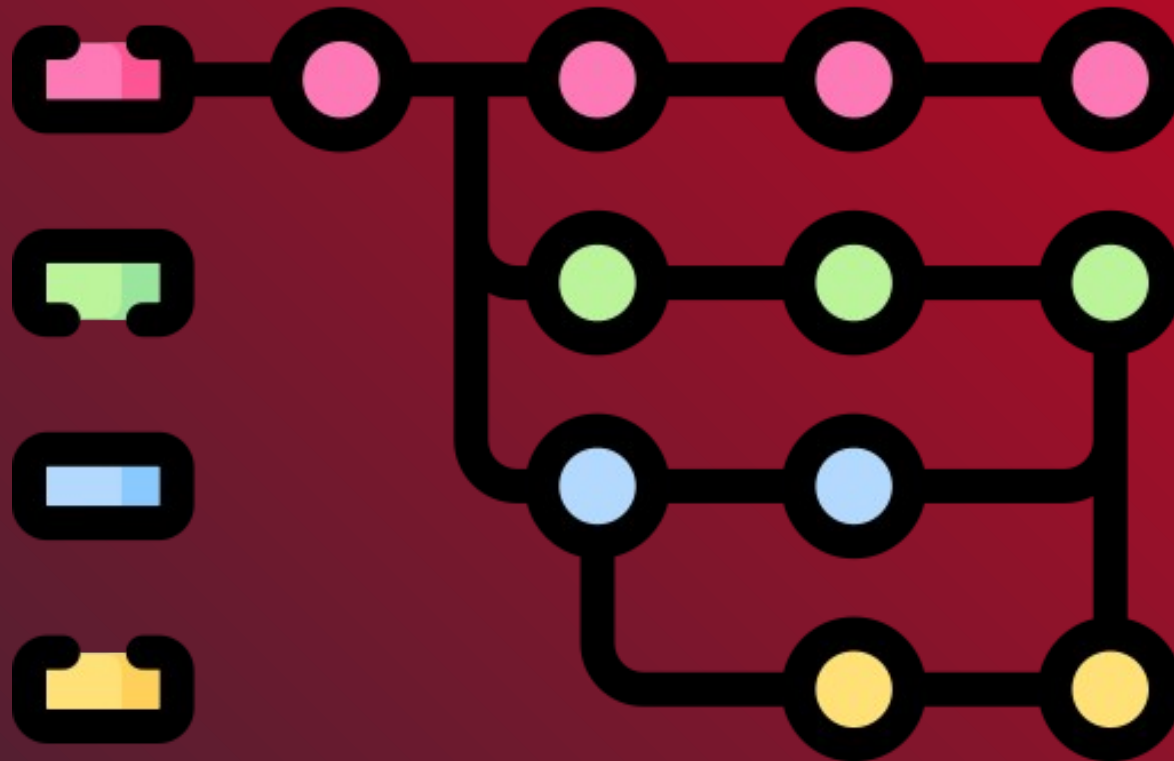


**Distribuer son  
code**



**Une grande histoire**

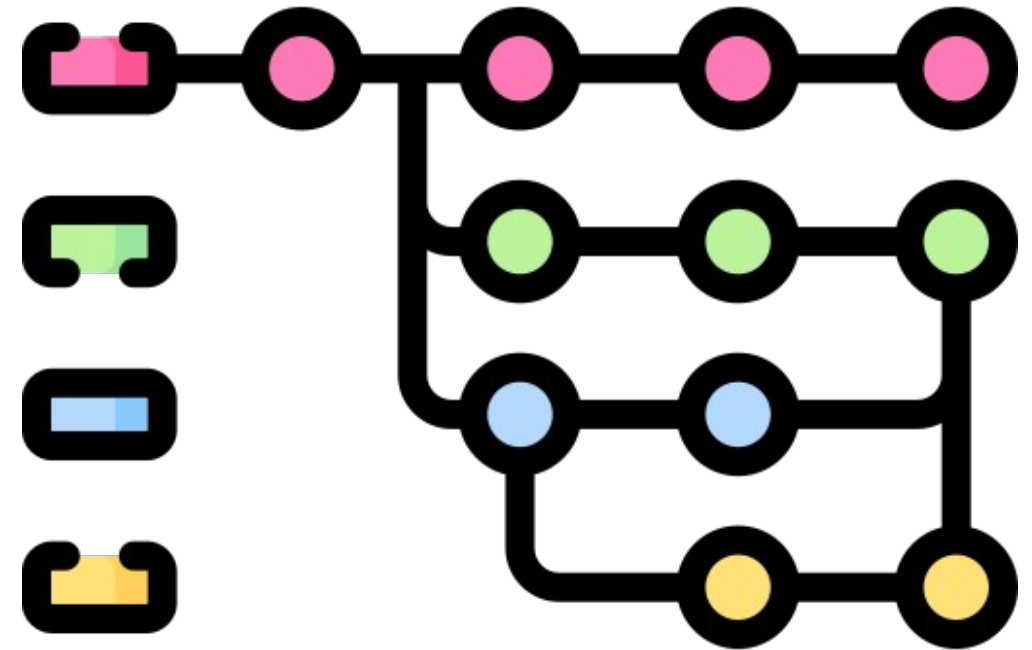
# Le branching





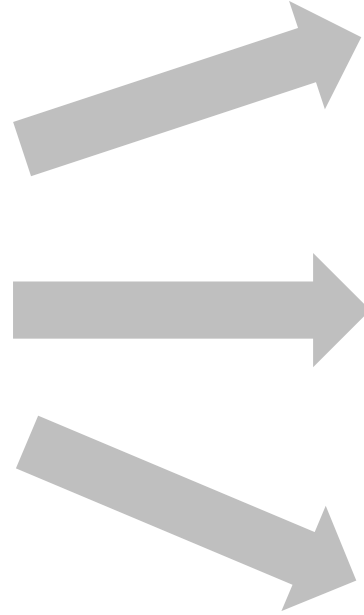


## Une représentation graphique du versionning





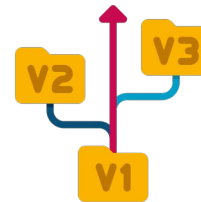
Un ensemble de  
fichiers



Suivis



Distribués

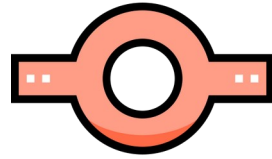


Versionnés

git

clone <https://git/mon-repo.git>

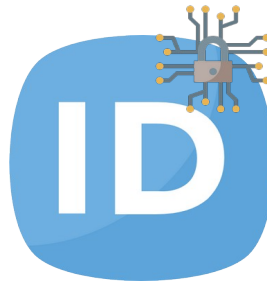
# Un commit



Un commit



Ensemble de lignes  
modifications



Réunies sous un  
identifiant unique



Permet donc de versionner des changements

```
git commit -m "mon message"
```

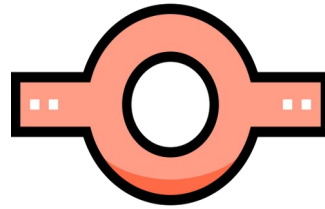


Git graph vision





Permet de voir  
l'état



Repository local



Ensemble de  
modifications

**git**  
**status**



```
→ airport-si git:(main) ✕ git status
```

```
On branch main
```

```
Your branch is up to date with 'origin/main'.
```

```
Changes not staged for commit:
```

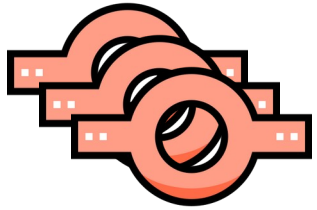
```
(use "git add <file>..." to update what will be committed)
```

```
(use "git restore <file>..." to discard changes in working directory)
```

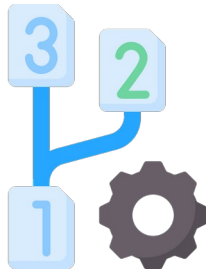
```
modified: pom.xml
```

```
modified: src/main/kotlin/fr/uga/miage/m1/data/entities/UserEntity.kt
```

# Les Branches



Ensemble de commits



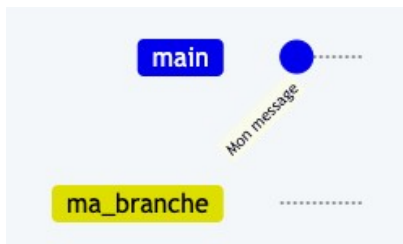
Dans un ordre précis



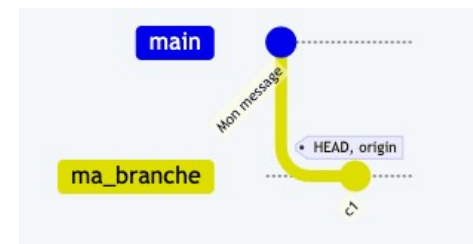
Travail à part du code fonctionnel

`git branch ma_branche`

`git checkout ma_branche`

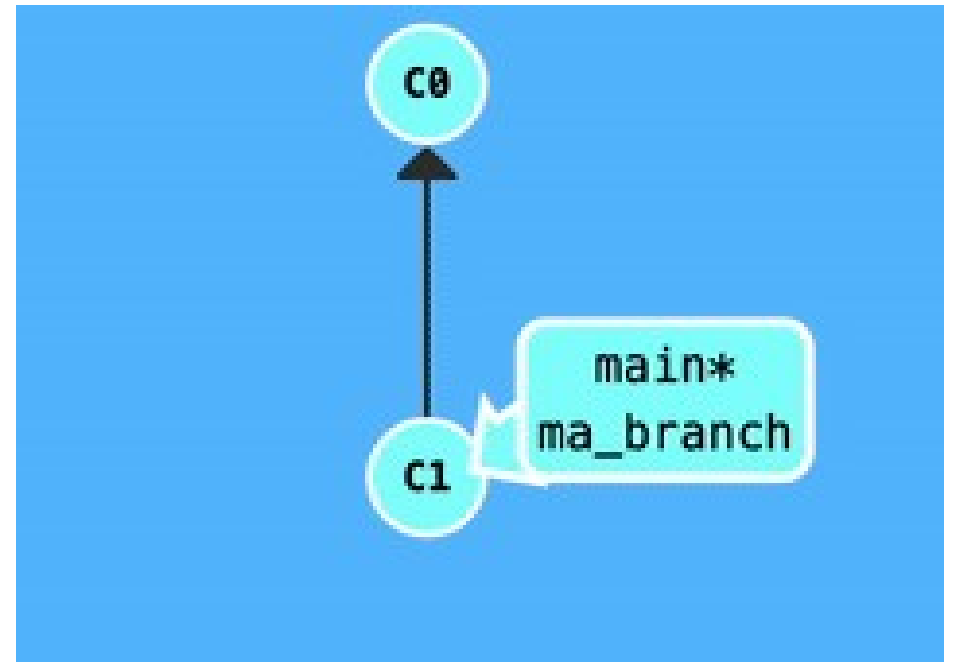
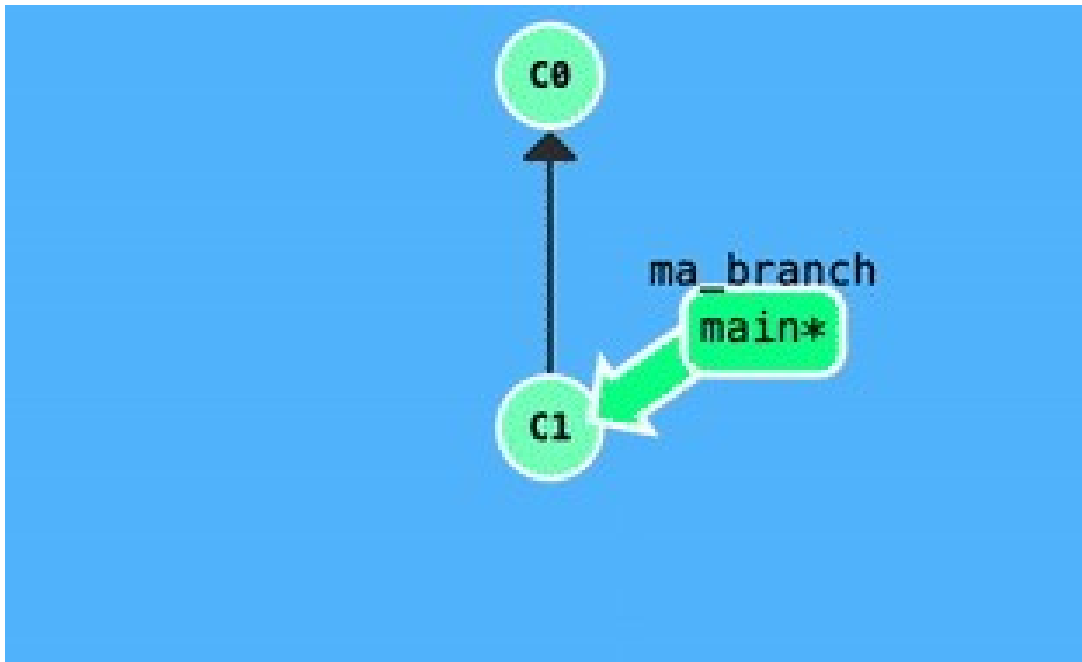


Git graph vision



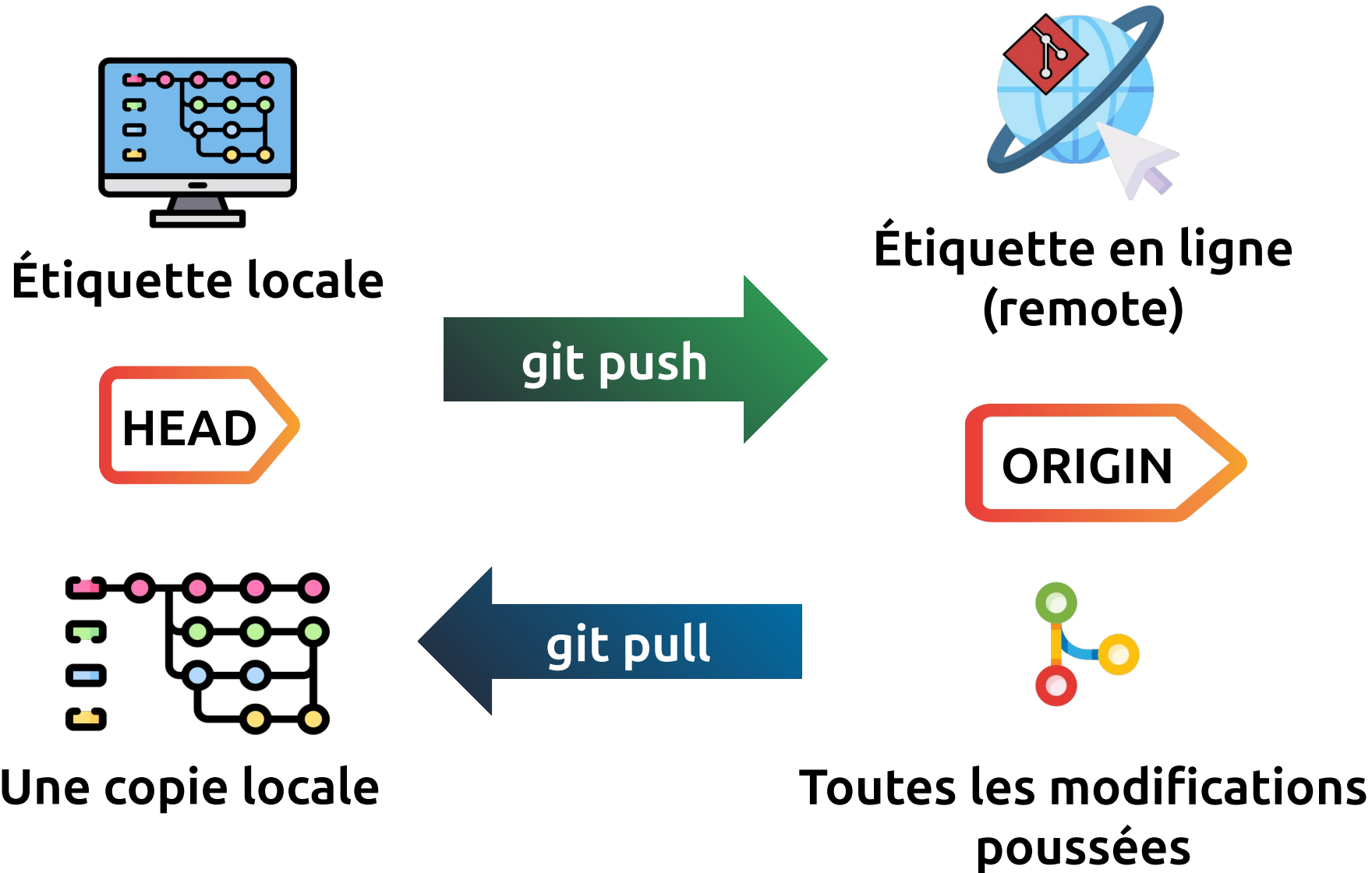
```
git branch ma_branch
```

```
git checkout ma_branch
```

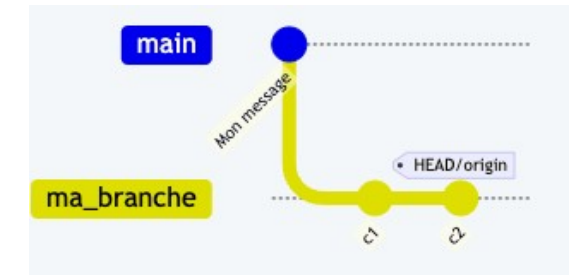
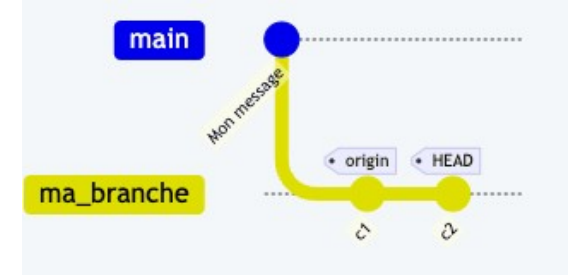




# Les étiquettes



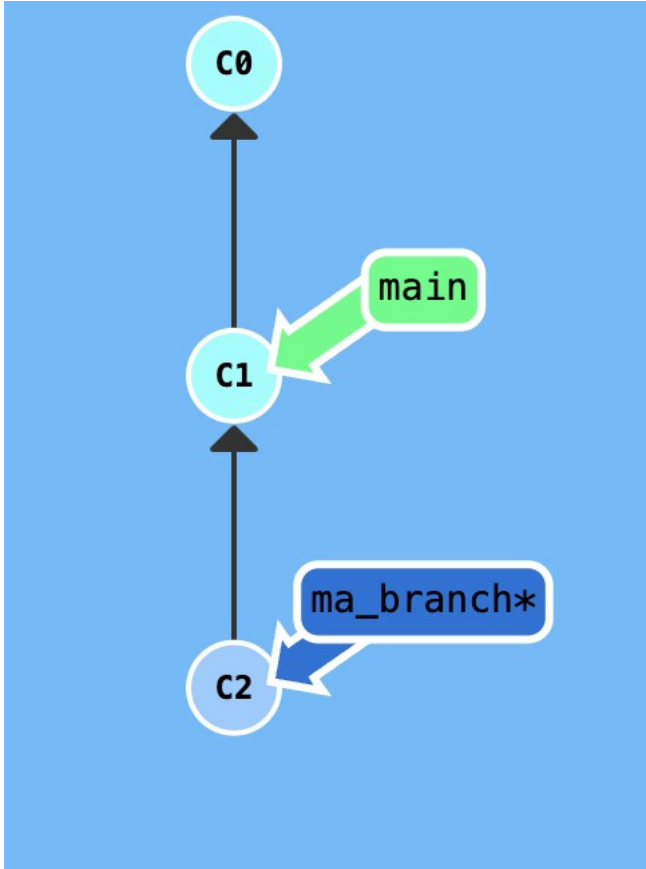
Git graph vision



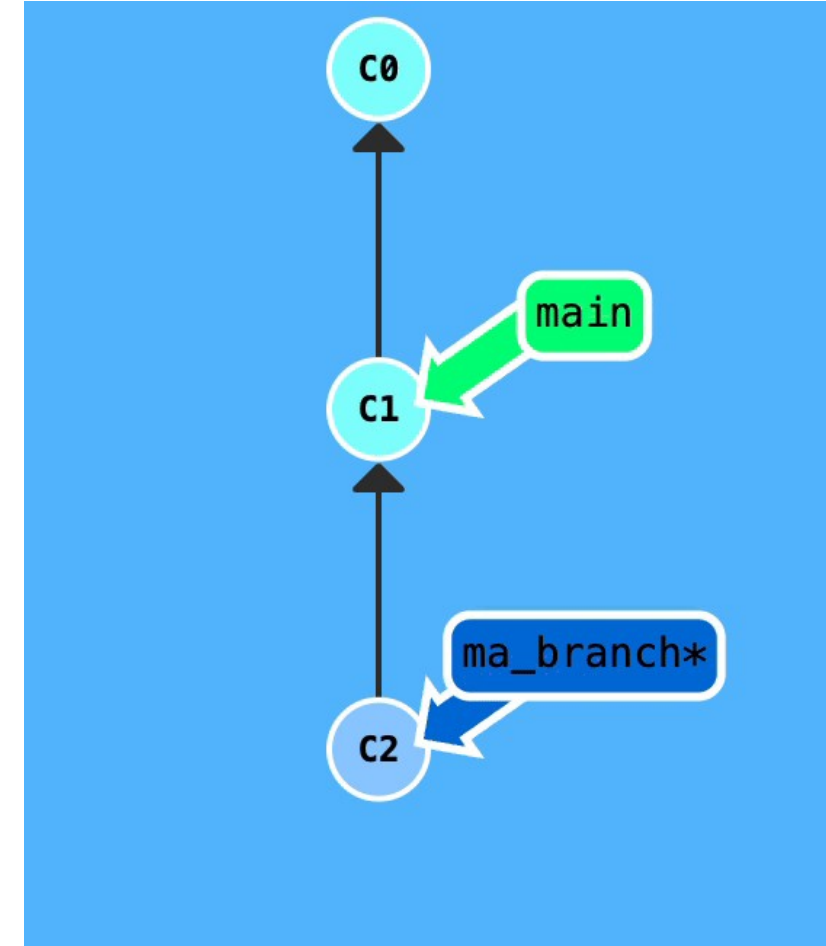
# Les étiquettes



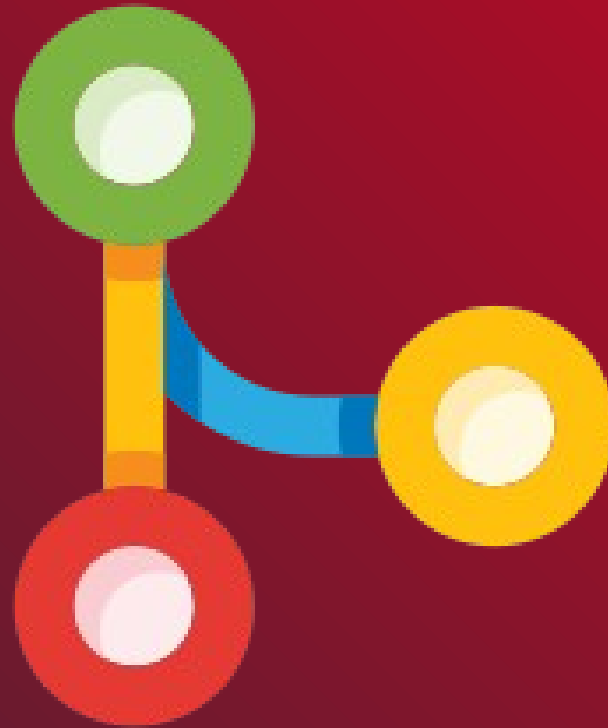
Git graph vision



`git chekout C0`



# Les merges





Une branche  
source

Injecter les modifications dans une autre

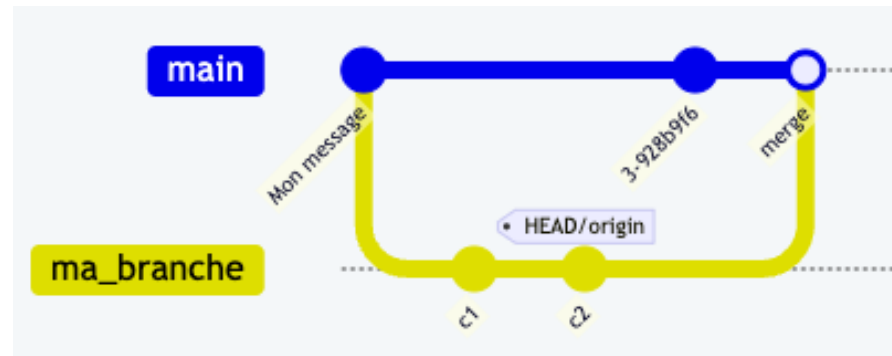


Une branche  
destination

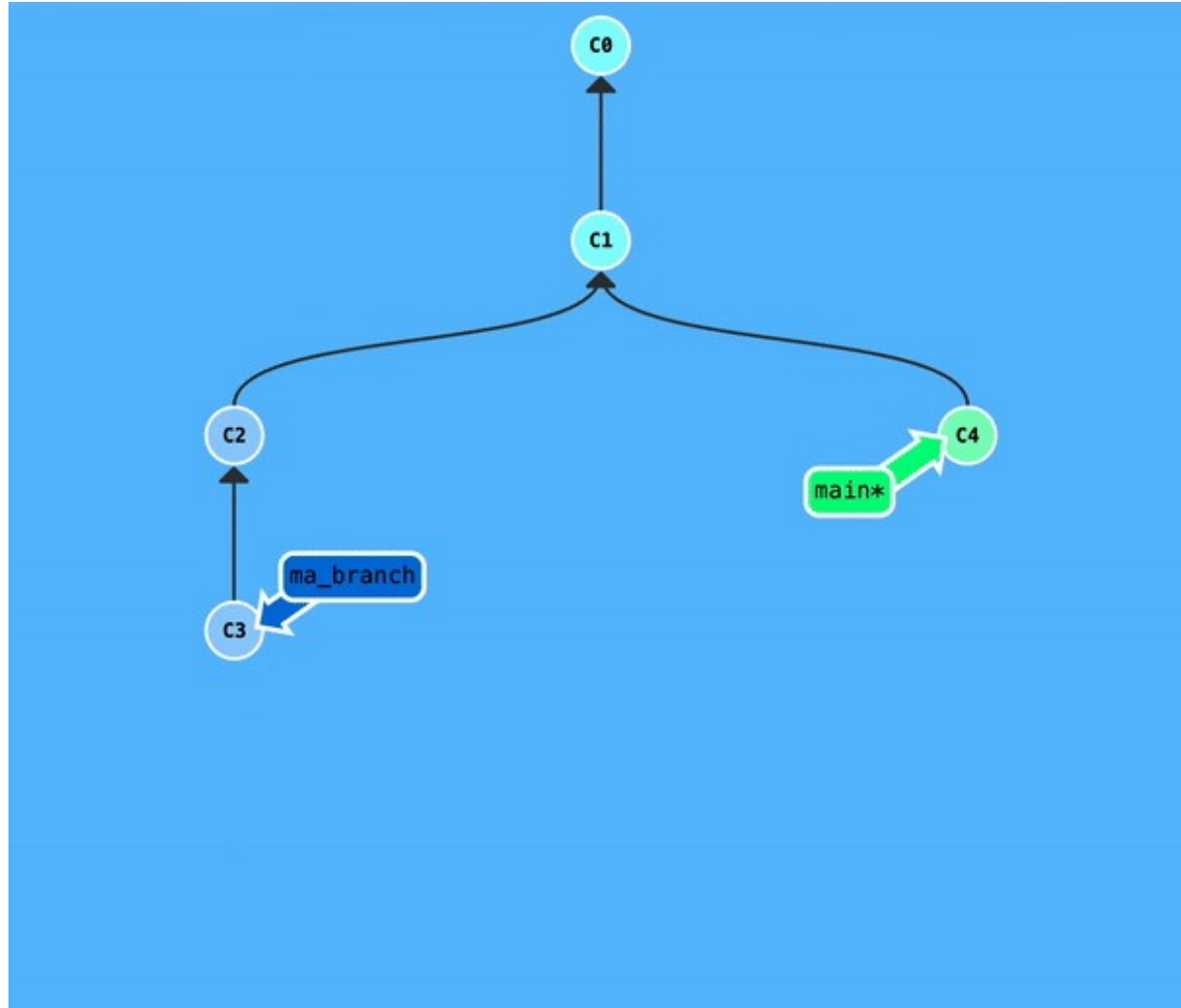
**git merge**



Git graph vision



# Les merges



# Les rebases

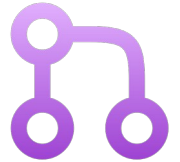






Une branche à  
rebase

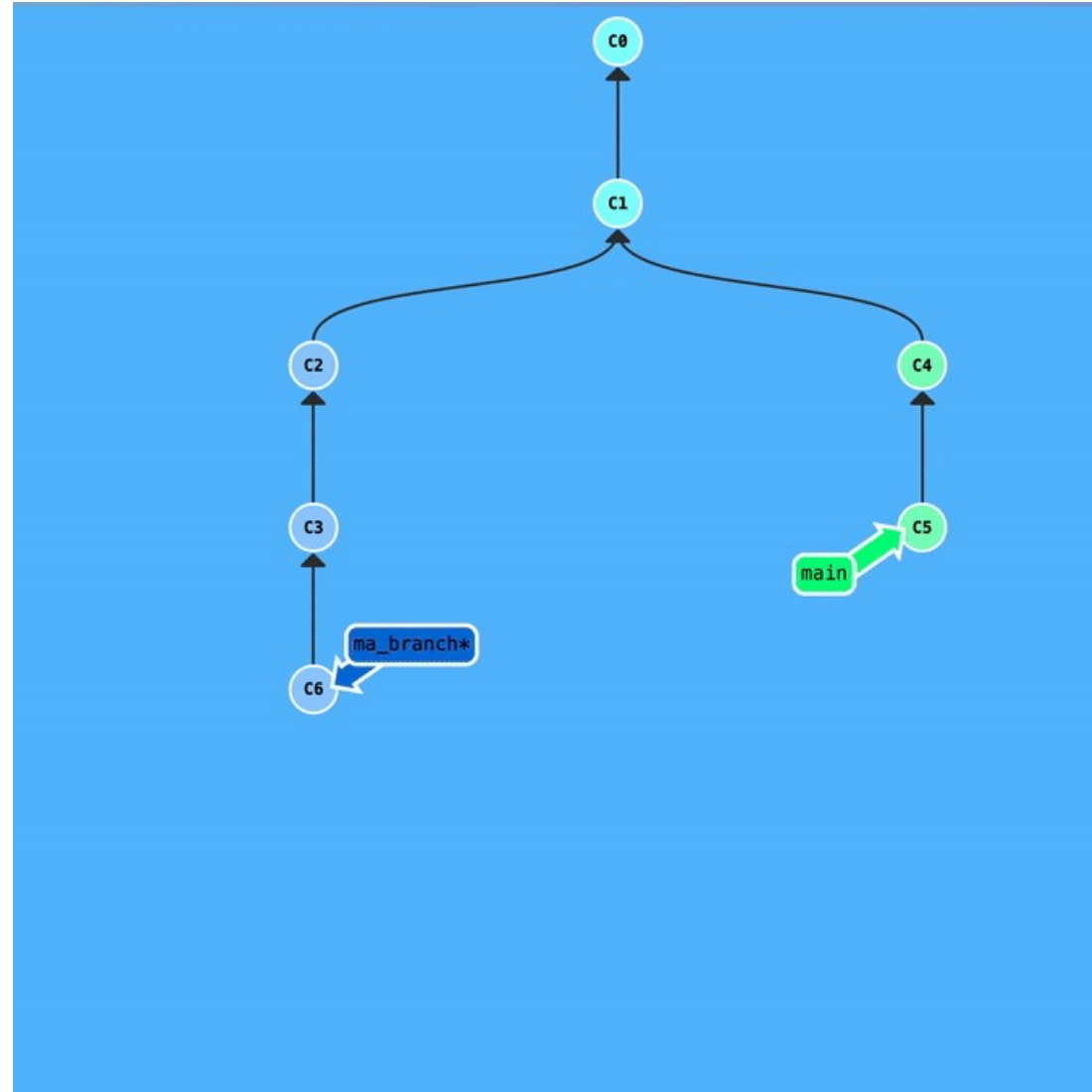
Ramener tous les commits sur la tête



Une branche  
De base

```
git rebase  
main
```

# Les rebases



# Les conflits



# Le plus chiant



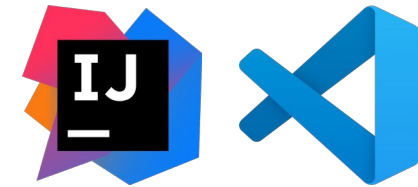
# Les conflits

2 méthodes

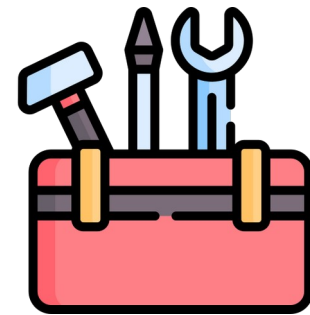


Par les fichiers

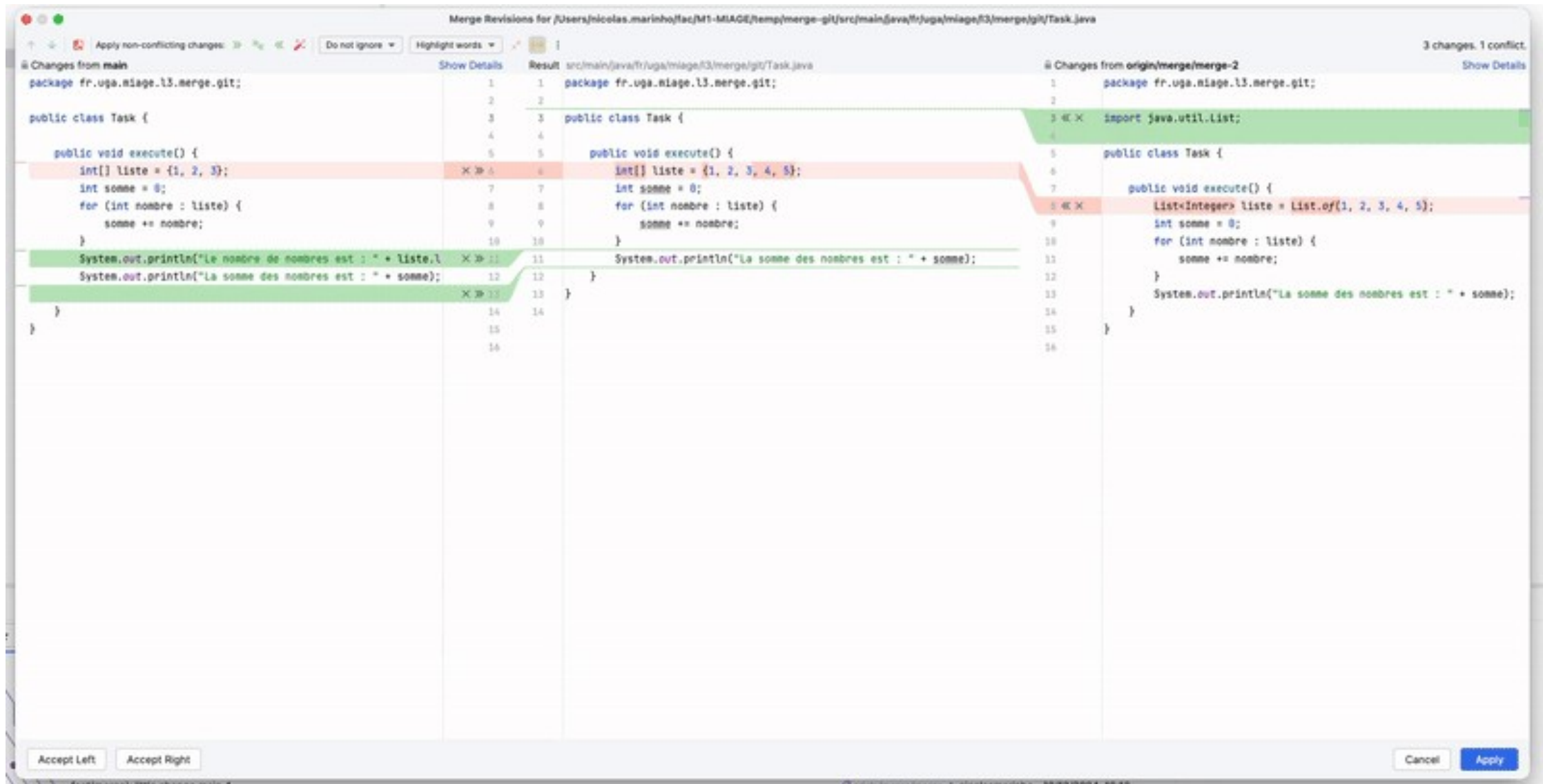
```
<<<<<< HEAD
  int[] liste = {1, 2, 3, 4, 7, 9, 0, 9};
=====
  int[] liste = {1, 2, 3};
>>>>>> origin/merge/merge-1
```



Par les IDE

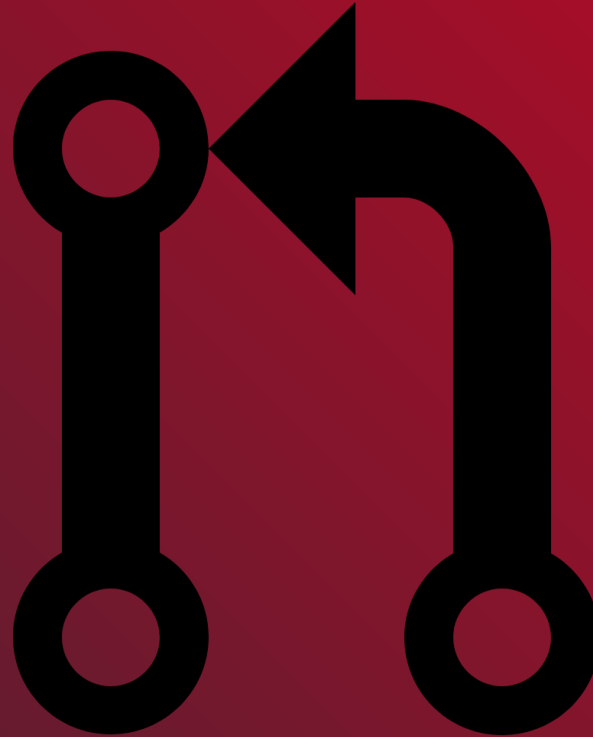


Des outils pour

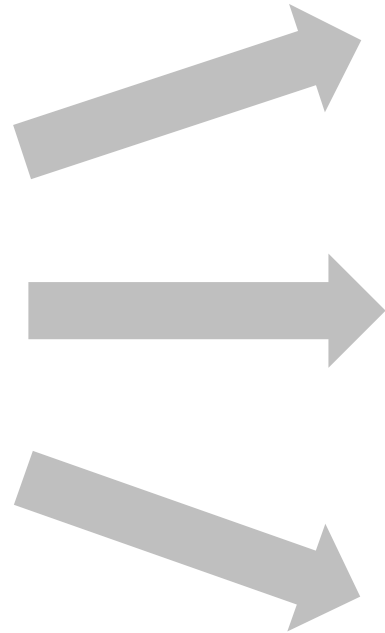
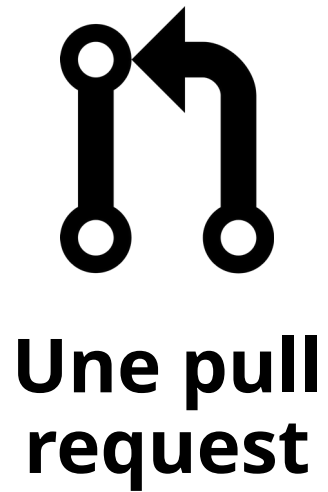




# Une pull request



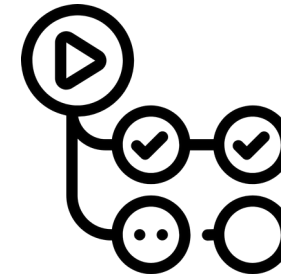
# Une pull request



A une branche source et  
destination



La validation par les pères



La validation par les pères

The screenshot shows a GitHub pull request interface. At the top, there's a navigation bar with links to Code, Issues, Pull requests (selected), Actions, Projects, Security, Insights, and Settings. A notification banner at the top right states: "GitHub users are now required to enable two-factor authentication as an additional security measure. Your activity on GitHub includes you in this requirement. You will need to enable two-factor authentication on your account before January 01, 2026, or be restricted from account actions." with an "Enable 2FA" button.

The main heading is "Feedback #1". Below it, a status bar indicates "Open" and "github-classroom wants to merge 64 commits into feedback from main". There are tabs for Conversation (0), Commits (64), Checks (3), and Files changed (151). A summary shows "+18,537 -165" changes.

On the left, a file tree shows the repository structure: .DS\_Store, .github/workflows (containing pr-dev-back.yml, pr-dev-front.yml, push-main.yml), backend (containing .gitattributes, .gitignore, .java-version), and .mvn/wrapper (containing maven-wrapper.properties, docker-compose.yml, mvnw, mvnw.cmd, pom.xml).

The main area displays the diff for the selected file, ".github/workflows/pr-dev-back.yml". The diff shows a new file being added, with a summary of changes: "44 additions, 0 deletions, 0 moves". The content of the file is shown in a code editor, starting with a comment about workflow execution time and a reference to GitHub documentation. The workflow is named "PR dans dev Backend" and is triggered on a pull request to the dev branch.

## Ajout d'un fichier au commit

*git add mon\_fichier*

## Pousser les commits sur la remote

*git push*

## Merge une branche

*git merge branche-à-merge*

## Créer un commit avec un message

*git commit -m « mon message »*

## Récupérer les commit distant

*git pull*

## Changer de branche

*git checkout branche\_destination*

## Voir le status

*git status*

## Créer une branche

*git branch nom\_branche*

## Rebase une branche

*git rebase branche-de-base*

# Conclusion

