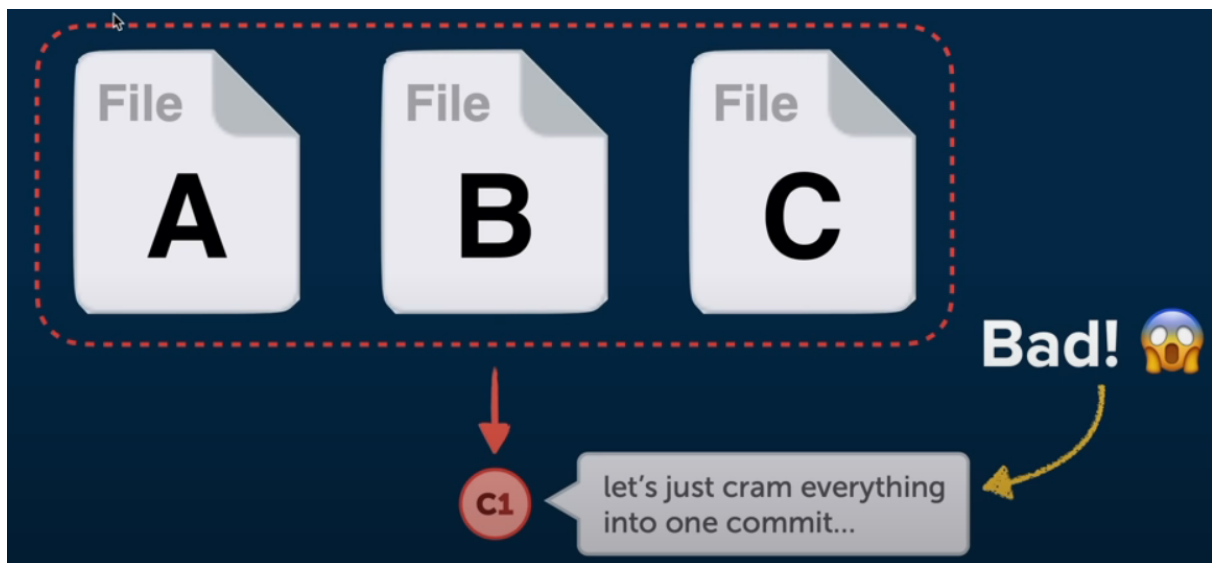




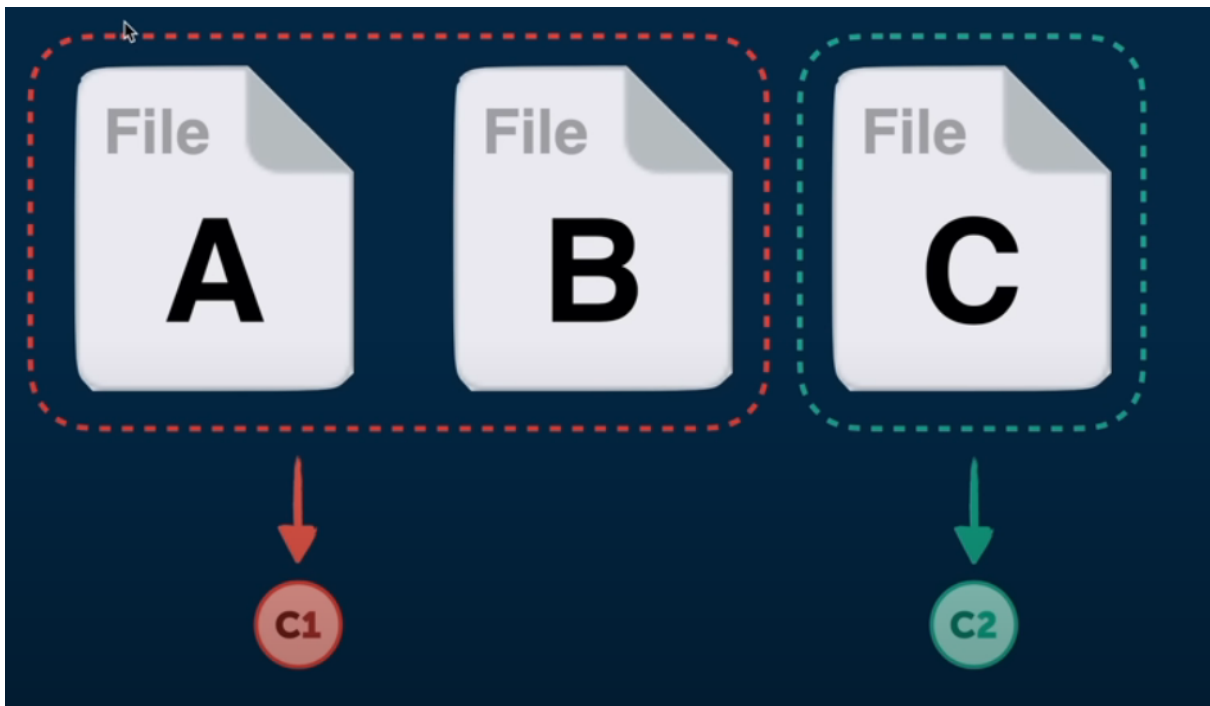
Como criar o commit perfeito

- Adicionar um commit por arquivo.
- Adicionar uma boa mensagem no commit.

Evitar:



Utilizar:



Necessário respeitar esta regra para dar o commit conciso:

- 1 **Subject = *concise summary of what happened***
- 2 **Body = *more detailed explanation***
 - What is now different than before?
 - What's the reason for the change?
 - Is there anything to watch out for / anything particularly remarkable?

Convenção GIT

A Written Convention

Agree on a Branching Workflow in Your Team

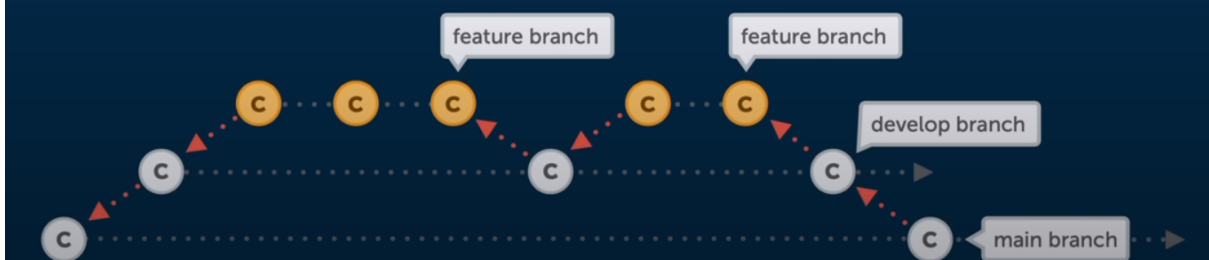
- 1 Git allows you to *create branches* - but it doesn't tell you *how to use* them!
- 2 You need a written best practice of how work is ideally structured in your team - to avoid mistakes & collisions.
- 3 It highly depends on your team / team size, on your project, and how you handle releases.
- 4 It helps to onboard new team members ("this is how we work here").

criar mais branch

State, Release, and Feature Branches

Branches Enhance Structures & Workflows

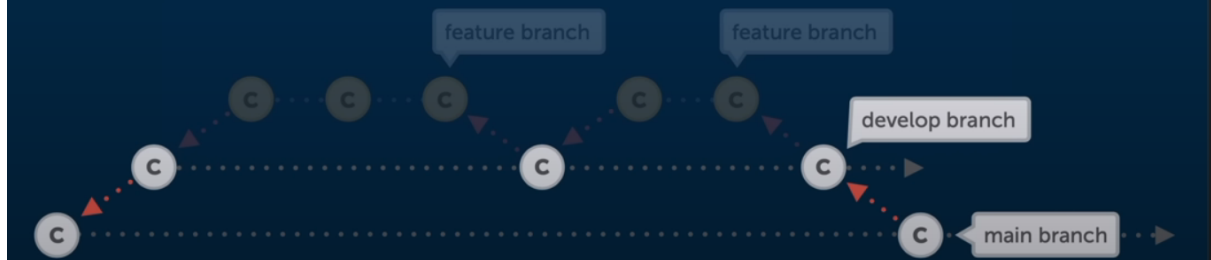
- different types of branches...
- ...fulfill different types of jobs



Não adicionar códigos sem testar no galho principal (main branch)

Long-Running & Short-Lived Branches

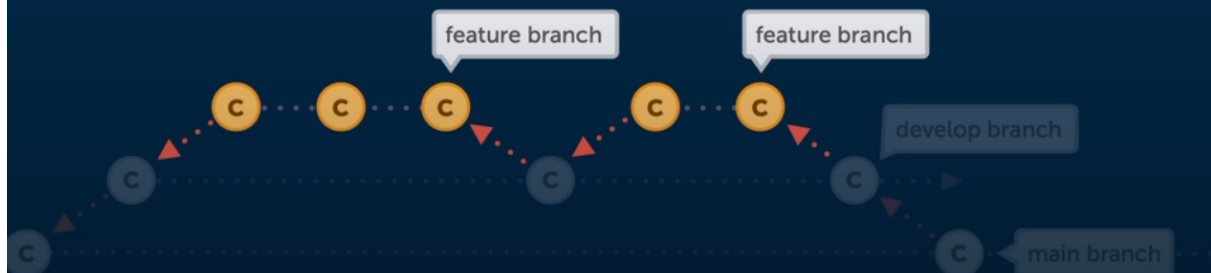
- exist through the complete lifetime of the project
- often, they mirror “stages” in your dev life cycle
- common convention: no direct commits!



os Branchs de feature serão deletados após a integração.

Long-Running & Short-Lived Branches

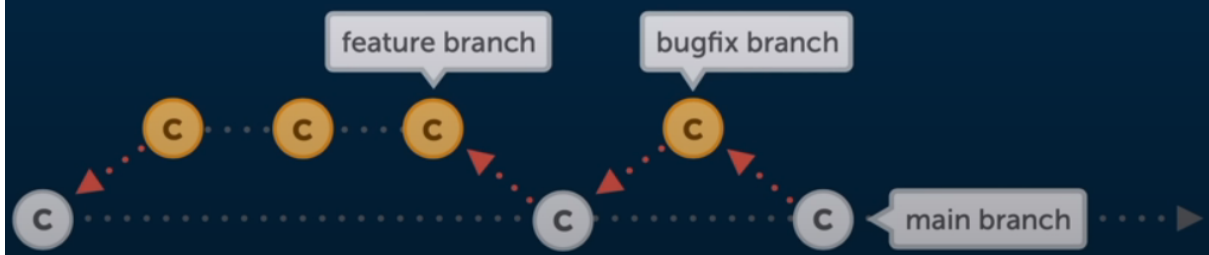
- for new features, bug fixes, refactorings, experiments...
- will be deleted after integration (merge/rebase)



Normalmente terá apenas um longo branch

GitHub Flow

- very simple, very lean: only one long-running branch (“main”) + feature branches



Modelo Gitflow mais estruturado com mais regras

GitFlow

- more structure, more rules
- long-running: “main” + “develop”
- short-lived: features, releases, hotfixes

