

## COLINHA GIT E COMMITS

### 1. Tipos de Commits (Conventional Commits)

- feat: adiciona nova funcionalidade.
- fix: corrige bug.
- refactor: altera código sem mudar funcionalidade.
- style: mudanças de formatação.
- docs: documentação.
- test: testes.
- perf: performance.
- chore: tarefas técnicas.
- build: ajustes de build.
- hotfix: correção urgente.

### 2. Estrutura ideal de commit

tipo: descrição

Ex: feat: cria tela de login

### 3. Comandos principais do Git

- git status: mostra status.
- git add . : adiciona tudo ao commit.
- git add arquivo: adiciona arquivo específico.
- git commit -m "msg": cria commit.
- git push origin branch: envia commit.
- git pull origin branch: baixa mudanças.
- git merge branch: mescla branch.
- git log: histórico.
- git restore arquivo: restaura mudanças não commitadas.

- git reset --soft HEAD~1: desfaz commit mantendo alterações.
- git reset --hard HEAD~1: desfaz commit removendo alterações.

#### 4. Branches

- git checkout -b feature/nome: cria branch.
- git checkout nome: troca de branch.
- git push -u origin branch: sobe branch.

#### 5. Fluxo de trabalho recomendado

1. git checkout main

2. git pull

3. git checkout -b feature/minha-feature

4. desenvolver e commitar

5. git push

6. abrir Pull Request