

E T E P O R T O D I G I T A L

MINICURSO GITHUB

Cloves e os Quatro Capangas



CARLOS ALVAREZ
Estudante do 3A



IGOR OLIVEIRA
Estudante do 3C



JORGE FREITAS
Estudante do 3A

Cloves e os Quatro Capangas



THIAGO MARQUES

Estudante do 3C



CLOVES ROCHA

Professor e orientador

Objetivos

O que deve ser esperado desta talk, quais conhecimentos serão adquiridos e como será sua estrutura.

Entender o básico de GitHub

- 01 Entendimento dos seus objetivos, comandos e a base de seu vocabulário técnico, tal como instalar, criar uma conta, um repositório e um projeto.
-

Compreender a importância do GitHub

- 02 Compreensão de sua importância para o mercado de trabalho, projetos pessoais e aprimoramento de soft skills como trabalho de equipe e capacidade de liderança.
-

Importância da colaboração e gestão

- 03 A requisição do mercado por profissionais colaborativos e com um bom gerenciamento, portfólio pessoal com utilização do GitHub e conhecimentos essenciais.

Contexto histórico

Versão preliminar

Uma versão de demonstração, que visava apresentar a ideia do código.

Versão de testes

A versão na qual são efetuados testes acerca do código, também buscando uma melhor performance e qualidade.

Versão oficial

A versão pronta para vendas, com a conclusão da aplicação desenvolvida.



Contexto histórico

Versão preliminar

Uma versão de demonstração, que visava apresentar a ideia do código.

Versão de testes

A versão na qual são efetuados testes acerca do código, também buscando uma melhor performance e qualidade.

Versão oficial

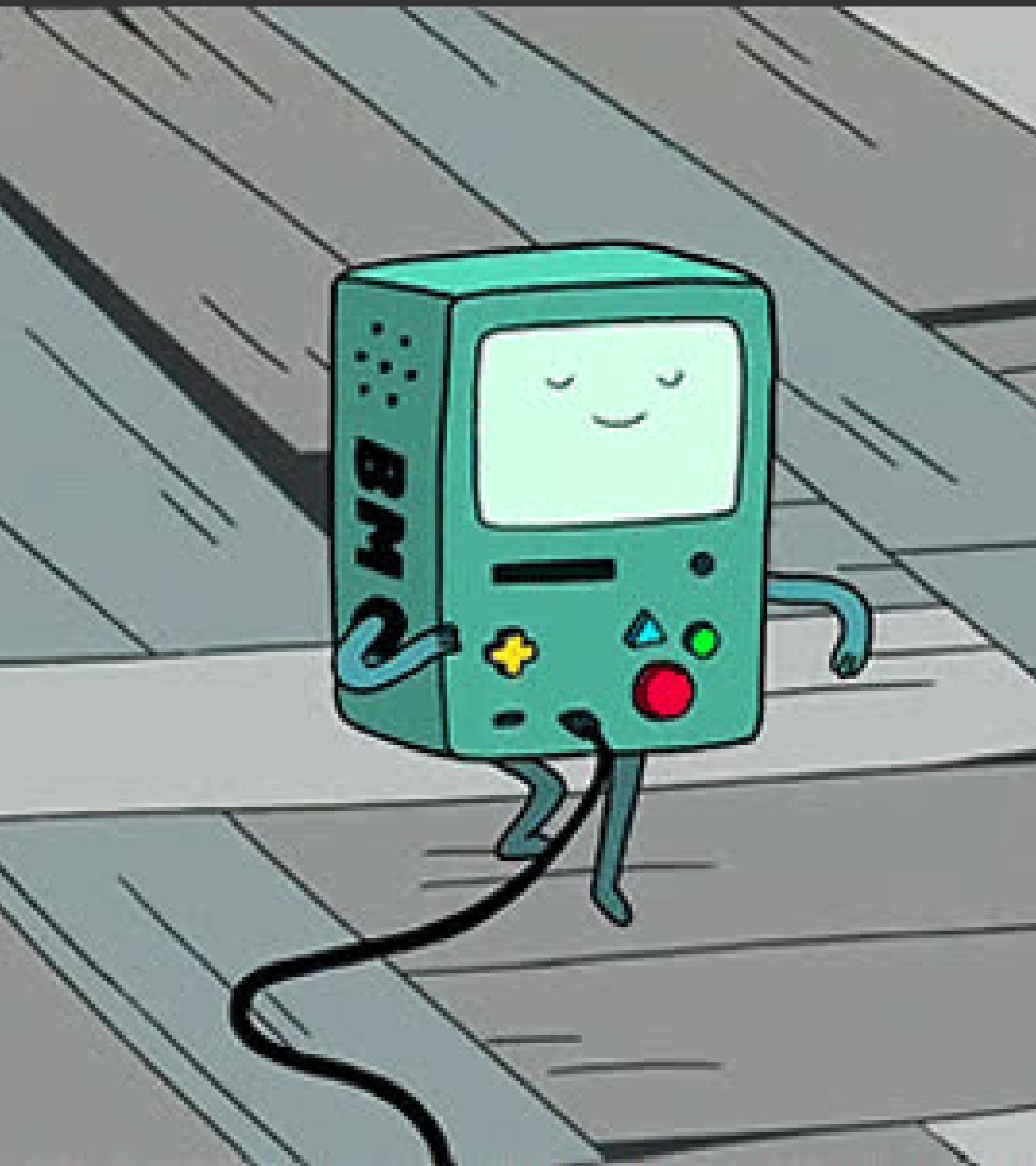
A versão pronta para vendas, com a conclusão da aplicação desenvolvida.

Posteriormente, eram desenvolvidas novas versões...

Tava demorando, mas
chegou o problema...



Nascem os
versionadores de código



História dos versionadores

O desenvolvimento dos versionadores de código, seu funcionamento, sua lógica, disponibilidade para comunidade geral e até mesmo seus pequenos conflitos...



SISTEMA DE VERSÕES
CONCORRENTES (CVS)
1985



SUBVERSION (SVN)
2000



BITKEEPER
2000



GIT/GITHUB
2005/2008

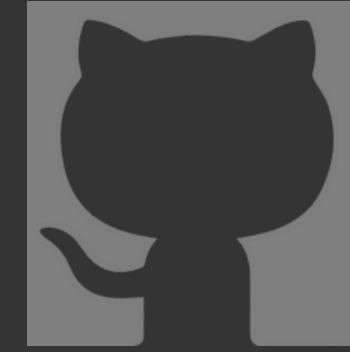


Com o uso do Git,
isso pode parar

arquivo.py
arquivo_2.py
arquivo_v2_legendado.py
main_legendado_dublado_dual_-
rip.py
arquivo_v3_aaaaaaaaa.py
arquivo_final_def_real.py
arquivo_socorro.py
;-;.py
aaaaaaaaaa.py
tristeza.py



Tá, mas qual a diferença entre **Git** e **GitHub**?



Git

- Ferramenta de VCS (Software de Controle de Versão)
- Responsável pelo controle de arquivos de um projeto
- Repositório Local na sua máquina

GitHub

- Rede social para programadores
- Maior comunidade de Open Source do Mundo
- Melhor forma de mostrar seu trabalho como programador

Mas...
Como o Git
funciona?



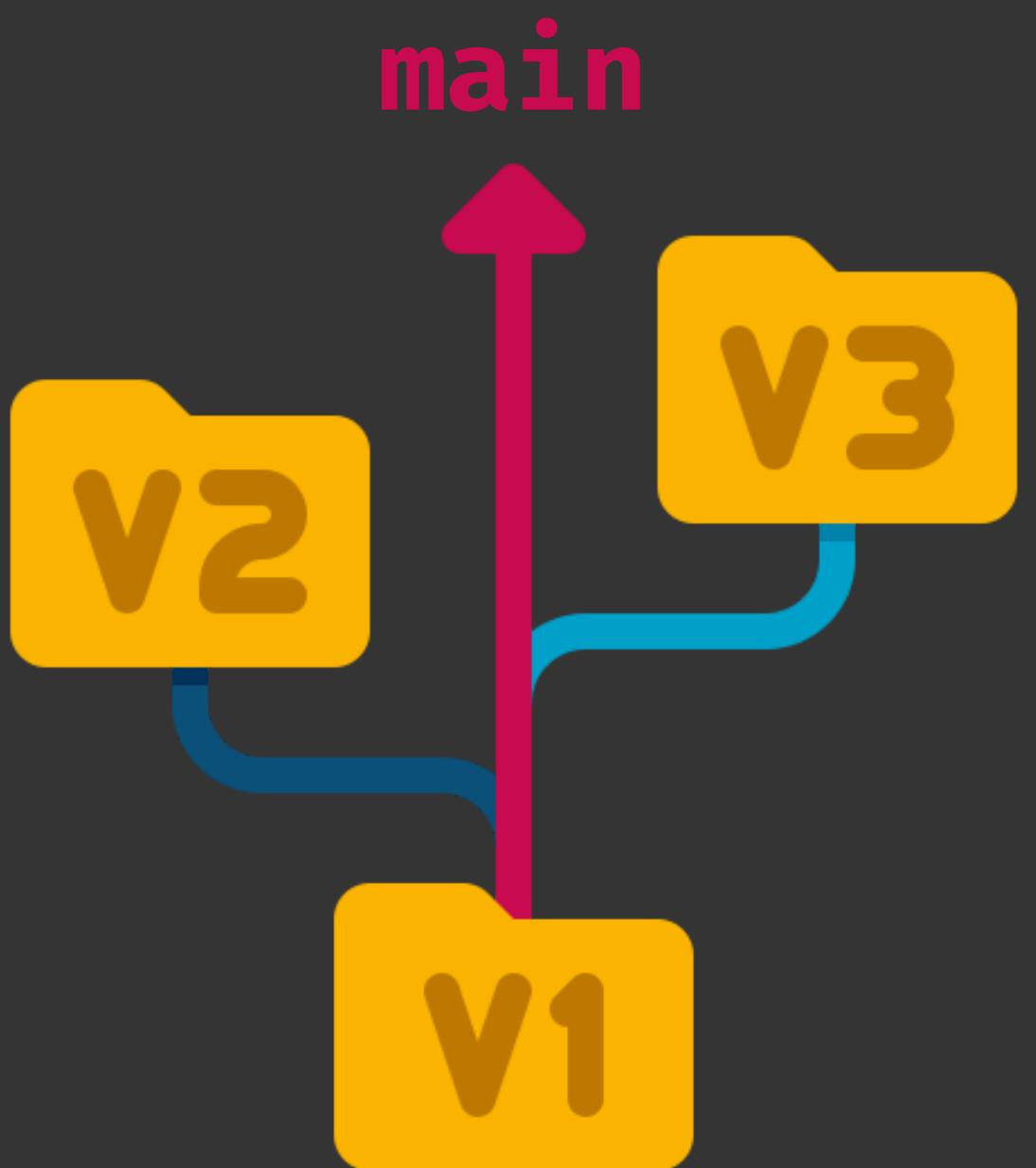
Repositórios

São os galpões do projeto, é onde ele será armazenado.



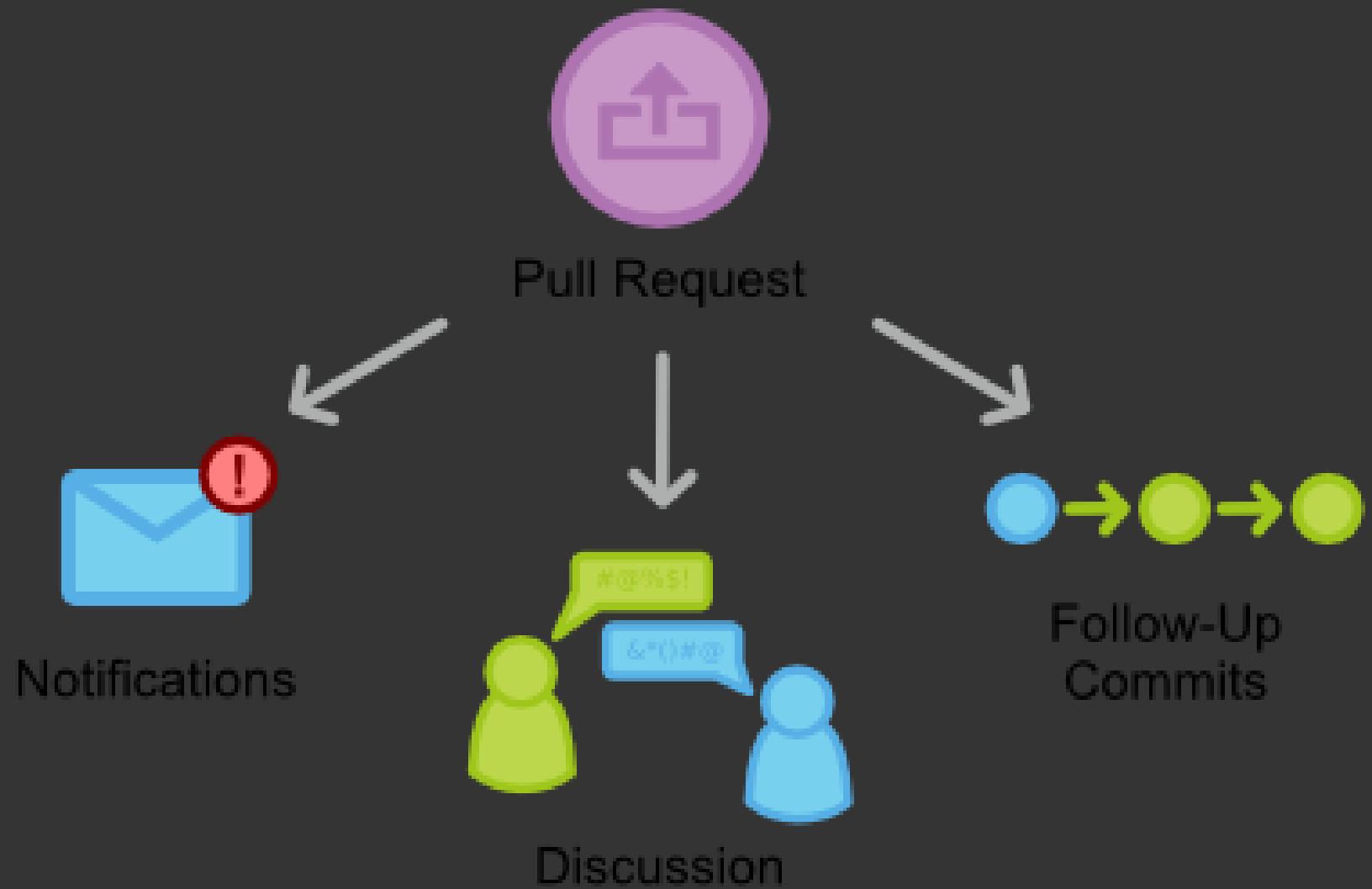
Branches

São as linhas de produção, é onde os times de dev são divididos.

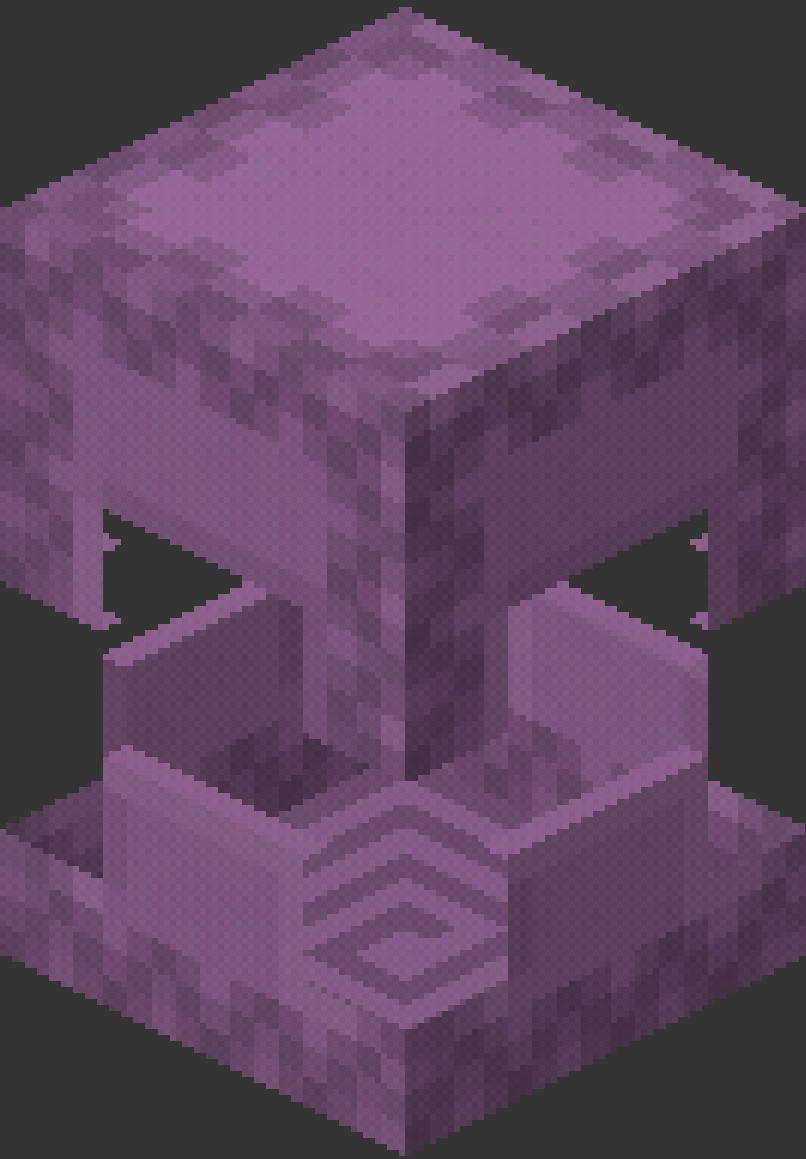


Pull Request (PR)

Quando há o pedido de inserção **de um código de uma branch qualquer para a main**



Como
podemos
transferir as
caixas?



Crie seu galpão (repositório) local

```
git init galpao_local
```



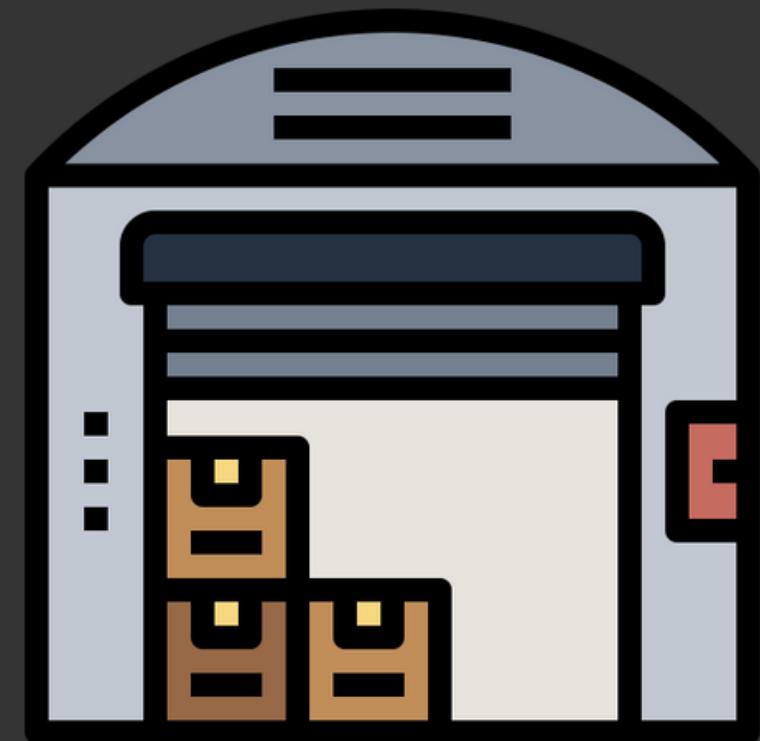
Estabeleça a conexão dos galpões

```
git remote add galpao_remoto  
https://link.galpao2
```



Pegue as informações do galpão central

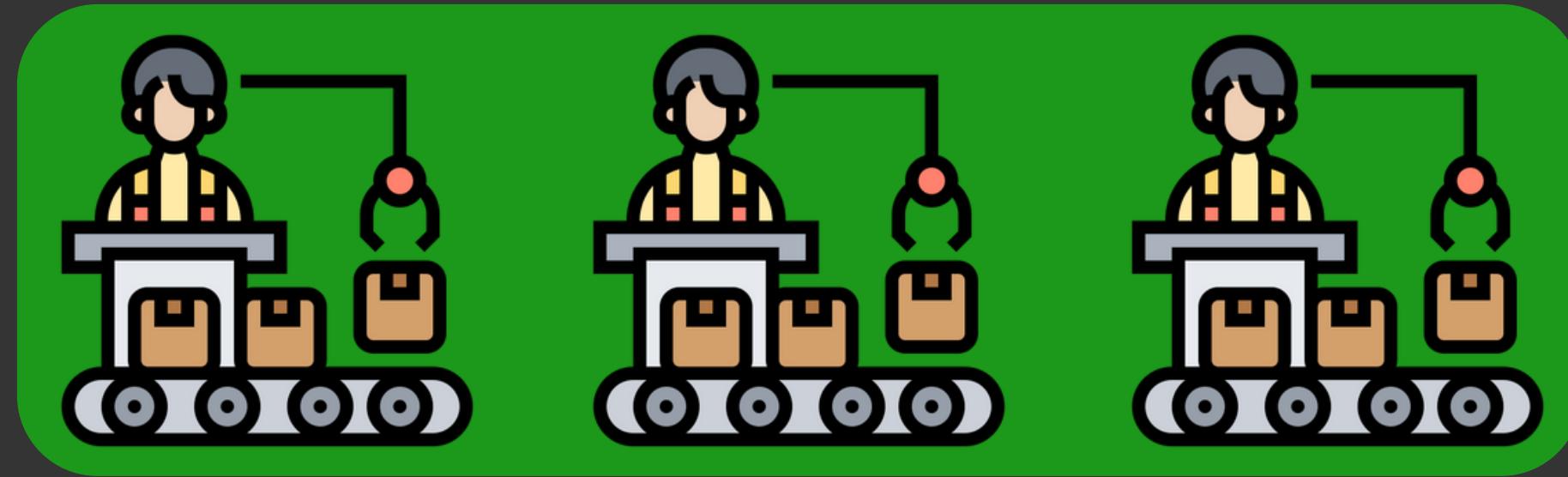
```
git pull galpao_remoto main
```



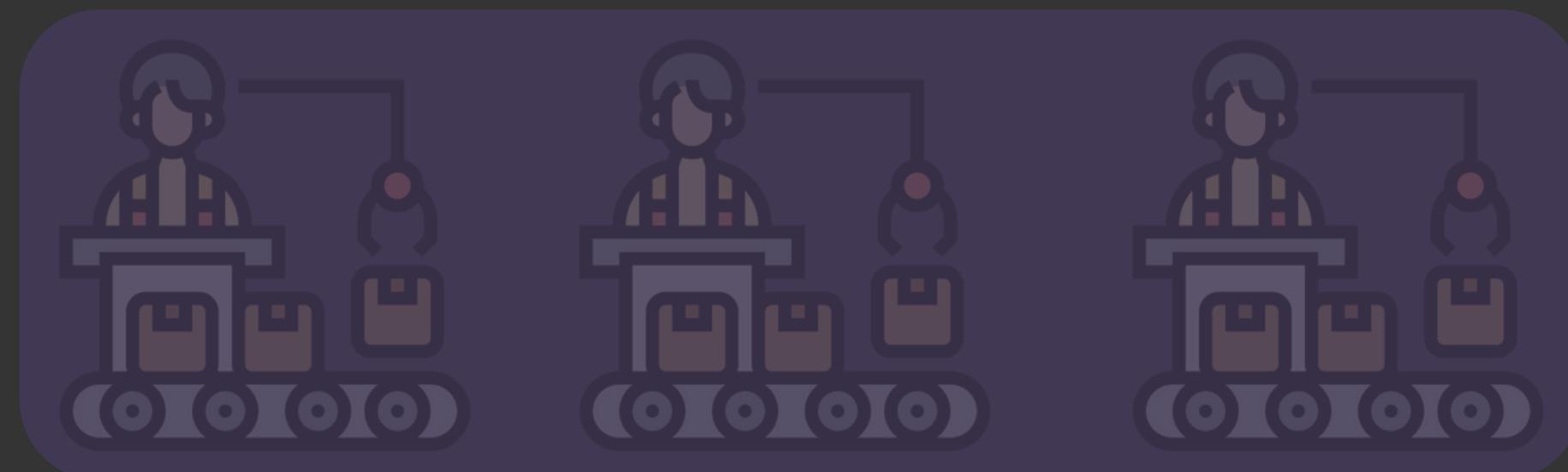
O galpão
local tem
várias linhas
de produção.



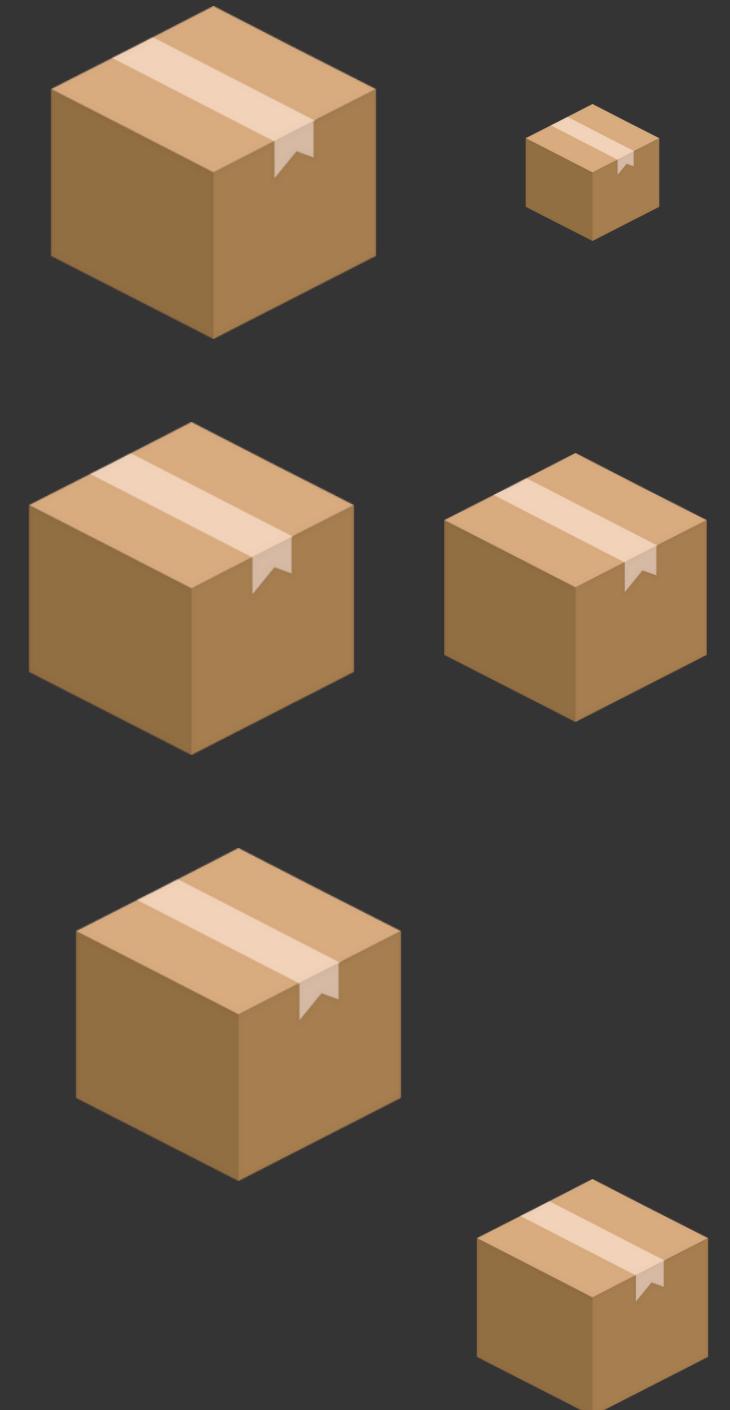
branches



Por padrão,
usamos a
linha "main".
↓



Separe quais caixas serão distribuídas



Separe quais caixas serão distribuídas



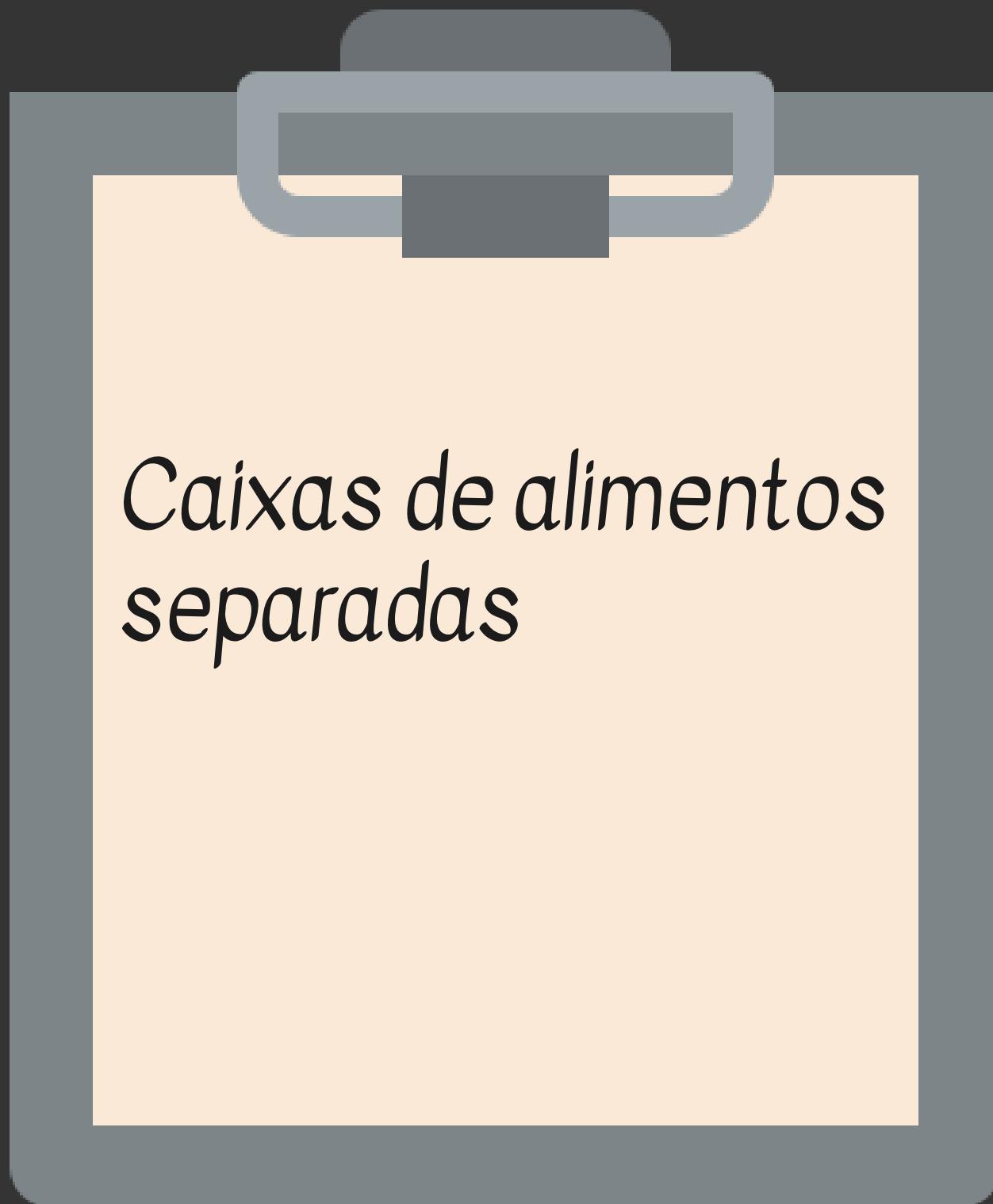
git
add
caixa1
caixa4
caixa5



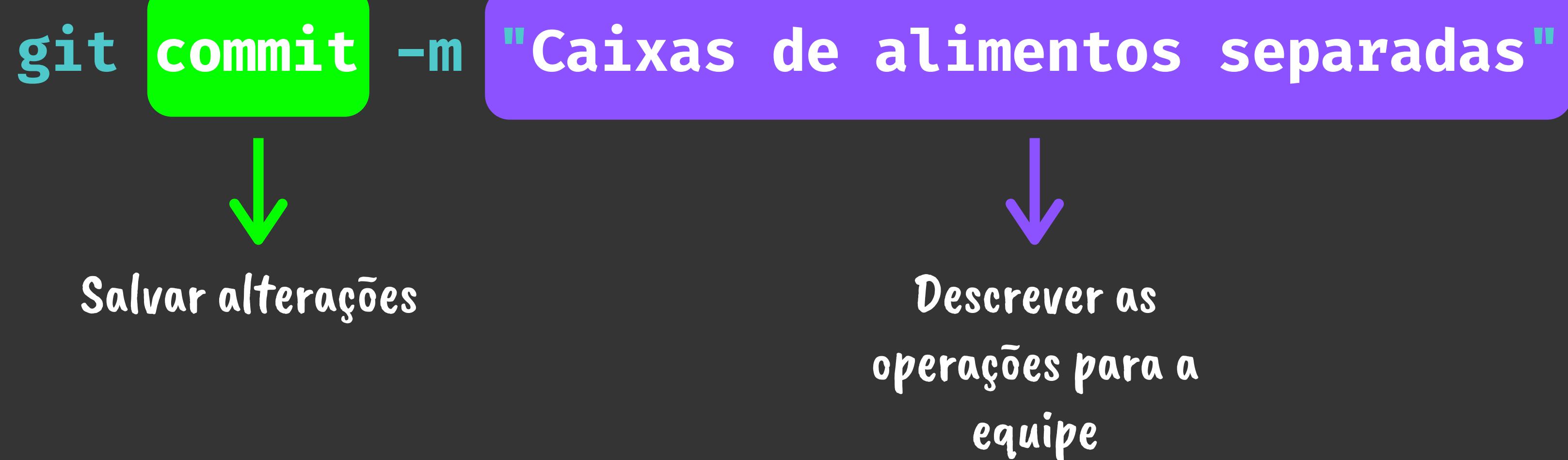
Descreva a operação a ser realizada



Descreva a operação a ser realizada

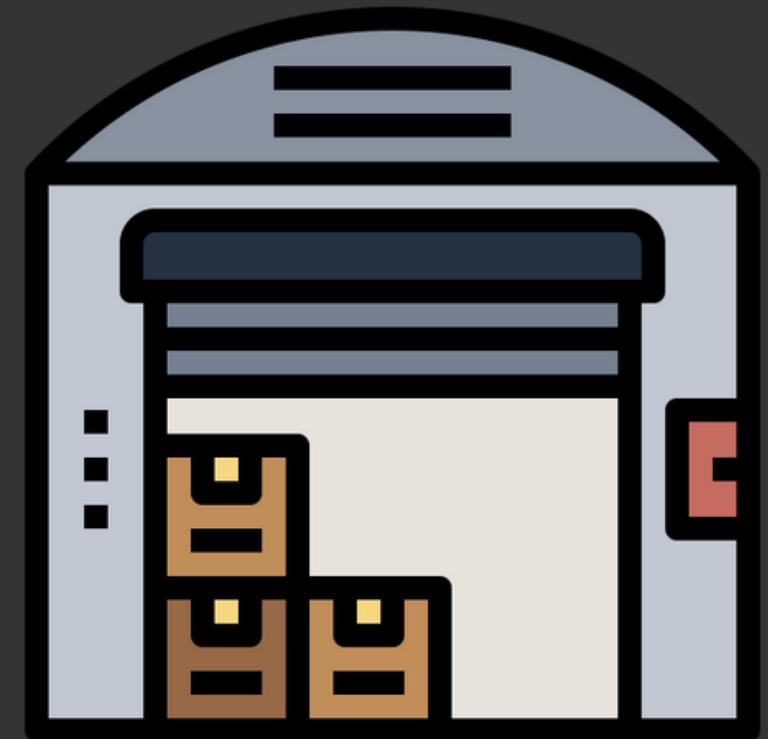


Descreva a operação a ser realizada



Entregue as caixas

```
git push galpao_remoto caixas_repo_local
```



E o processo
se repete



Entendeu bença?

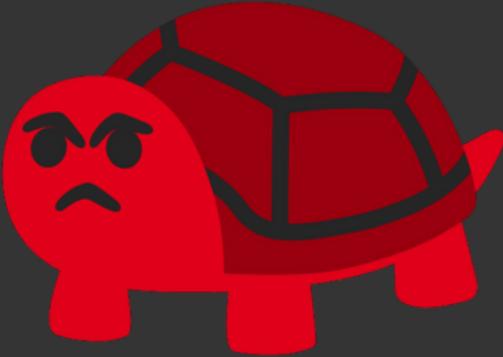


Como criar
uma conta?

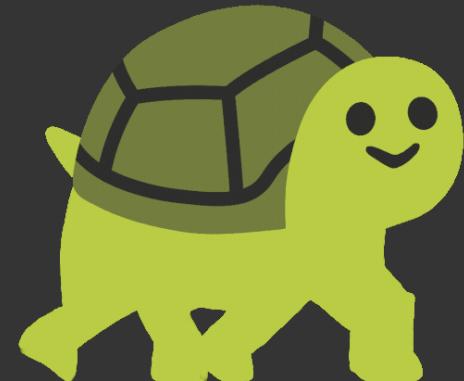
Como criar um
repositório?

Calma lá que lá vem a fera compartilhando, grande Jorge :)





Colaboração em projetos abertos
Ajudar em um projeto que nem é meu?



Aprimoramento de Soft skills

Proatividade

Contribuir com projetos abertos, sem qualquer pressão ou obrigação, implica muito em um profissional proativo, que toma decisões e que sempre está além do comum.

Trabalho em equipe

Trabalhar em projetos abertos, com pessoas de toda personalidade e de qualquer lugar no mundo, mostra bastante que você sabe trabalhar em equipe, e lidar com cada um.

Comunicação

Trabalhar em projetos open source é uma ótima forma de melhorar sua comunicação, e sempre tentar interagir e integrar cada vez melhor uma equipe qualquer.

Outras habilidades e ganhos

Conhecimento

Trabalhar em diversas áreas do conhecimento com distintas pessoas atrai uma carga imensa de conhecimentos que podem ser utilizadas em outros projetos ou simplesmente na vida cotidiana.

Visibilidade

Empresas podem utilizar bots para identificar perfis colaboradores e com habilidades para preencher suas vagas. Os ganhos de notoriedade na comunidade são imensos.

**Lembre-se:
Tenha modos.**

Clean Code

Ao realizar alterações no código, tente sempre deixá-lo o mais limpo e comprehensível possível, com o intuito de somar.

Comentários no código

Caso necessário, comente seu código, deixando-o legível e comprehensível por todos aqueles que irão ler.

Especifique suas ações.

Ao realizar um pull request para inserir seu código na branch, lembre-se sempre de especificar suas ações, dê uma boa descrição do que você mudou, e como aquilo irá afetar.



Issues?

Ferramenta de **colaboração**, como
utilizá-las e qual sua importância?



Pera, pera pera...

Lá vem mais nomezinho chato!?
O que raios é fork meu amigo!?

Git e GitHub são
importantes para o
mercado?

Sim! Demais! Muito! Demasiadamente! Pra cara-



Maior da Classe!

O GitHub desde 2013 se tornou a ferramenta mais utilizada para compartilhar projetos

Empresas Cobram!

Muitas empresas hoje cobram conhecimento prévio de Git e GitHub dos seus funcionários.

Você se torna mais visível

Mais fácil de ver o portfólio, habilidades e trabalho em equipe, pois o GitHub inclusive pode te contactar de acordo com sua atividade no perfil e oferecer uma vaga de emprego.

Hora da Analogia!

Utilização do GitHub para gerenciamento de seu projeto

Boas práticas para gerenciamento e contribuição

Utilização do README.md

README.md é um arquivo em markdown que visa explicar, exemplificar e documentar seu projeto, também servindo como um cartão de visitas ou um atrativo para o mesmo.

Utilização do CONTRIBUTING.md

Um arquivo presente na maioria dos projetos open source, normalmente contendo o "código de conduta" para um contribuidor daquele projeto.

Criação de um diário de bordo

Em várias aplicações ou até mesmo em sites, pode ser utilizada a linguagem markdown para criação de outros arquivos, em um destes, você pode desenvolver um diário de bordo, contendo os passos já dados no projeto, para que a comunidade possa entender o que foi feito e o que ainda falta.