**Issues**

Pode criar issue em qualquer repositório, nele você pode sugerir melhorias. Dentro da isssue pode fazer mais coisas que apenas registrar problemas, inclusive palestras. Pode utilizar código markdown.

Se você tem as permissões necessárias dentro do projeto consegue fechar as Issues.

**Fork**

É quando você faz uma cópia de algum repositório em sua conta do Git.

**Pull Request**

Quando um pull request é requesitado, o próprio github avalia se é possível realizar esse commit (não tem divergência entre código), ele vai mostra o que foi adicionado (no código tbm).

O dono do projeto consgue autorizar o pull request e depois precisa realizar um merge com o novo commit.

**Unindo Commit**

Git rebase -i HEAD~3 -> A partir do comando atual (HEAD) quer unir os últimos 3 commits enviados, caso queira pegar a partir de um commmit, copia o anterior que ele vai fazer daquele ponto até o último.

Na página que abriu vai vai aparece a palavra “pick”, a partir do arquivo, substitui os picks das linhas de baixo por “s”.

**Git cherry-pick**

**Git cherry-pick (rash do commit)** -> Pega o commmit selecionado para o branch que esta no momento, com esse comando pega um commit especifico e leva para a branch

Se uma implementação é necessária em sua *branch* e já foi feita em outra *branch*, pode fazer sentido trazer um *commit* específico, utilizando o git cherry-pick.

**Git bisect**

**Git bisect start ->**Com esse comando o git realiza uma busca dos pontos de partida de um commit A até um commit B.

Ex: git bisect HEAD (Ponto que o commit esta ruim)

Git bisect rash (Onde se sabe que estava bom), vai dando **git bisect bad** ate encontrar, quando encontrart da o comando **git bisect good.**

**Git bisect reset ->** Encerra a busca e vai para a branch que estava.

Git show rash\_do\_commit ->Mostra as alterações que foram feitas do commit

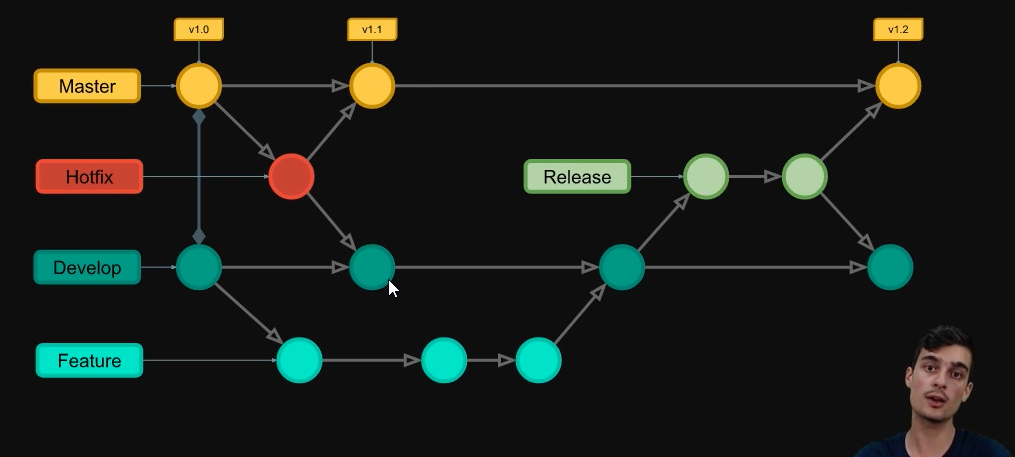
**Encontrando “Culpado”**

**Git blame ->** Ele me mostra quem é o responsável por cada uma das linhas

**Git Flow –** É uma forma bem organizada de seguir com commits

**Trabalhando profissionalmente**

Branch master ou main é a branch que teremos todo nosso código e é a que vai para produção. A codificação não deve ser realizada na master (é uma boa prática).



Mapa mental git flow

