um e-book produzido pela:

Code Prestige

Deu ruim no git

e agora?!

Situações corriqueiras no git e como resolvê-las

Agradecimentos

Seja muito bem-vindo!

O git é uma ferramenta sensacional e fundamental para qualquer desenvolvedor(a) hoje em dia. No entanto, também é uma ferramenta complexa e que pode nos deixar em algumas situações complicadas (até mesmo para os mais experientes). Este e-book retrata algumas dessas situações e como podemos resolvê-las sem perder a sanidade mental.

Ficou curioso? Então continue a leitura e bons códigos!

Diego Martins de Pinho
Fundador da Code Prestige
@DiegoPinho

Sumário de situações

As soluções apresentadas são apenas sugestões. De forma alguma são soluções definitivas e não devem ser consideradas como tal. Use-as por sua conta e risco!

Acabei de fazer um commit... mas esqueci uma coisa!

É muito comum ao finalizar uma tarefa nos darmos conta de que alguma coisinha ficou para trás. Geralmente isso acontece poucos segundos após darmos o enter no comando de commit.

Mas não há nada a temer, conseguimos resolver isso facilmente com as opções **--amend** e **--no-edit** do comando commit.

```
# faça as mudanças necessárias
git add . # adicione as mudanças
git commit --amend --no-edit
# agora seu último commit contém as mudanças!
```

** você também pode usar o --amend para trocar a mensagem do commit!*

Commitei... mas era pra ter sido em outra branch!

Trabalhar com branches no git é algo poderoso, mas como já dizia o tio Ben, com grandes poderes vem grandes responsabilidades.

Para resolver este problema, vamos combinar os comandos **stash** e **reset**.

```
# refaça o último commit, mas deixe as mudanças disponíveis
git reset HEAD~ --soft
git stash
# vá para a branch correta
git checkout nome-da-sua-branch
git stash pop
git add . # adicione as mudanças
git commit -m "mensagem do seu commit";
# agora as mudanças estão na branch correta!
```

Eu preciso reverter um commit antigo!

No mundo de desenvolvimento de software absolutamente nada é garantido. Aquilo que parecia certo um dia, pode ser descoberto como um bug crítico algumas sprints depois.

Se esse for o caso, podemos usar os comandos **log** e **revert** para nos ajudar a desfazer as besteiras.

```
# encontre no histórico o commit que você quer deletar
# use as setas do teclado para navegar entre eles
# ao encontrar o commit, salve a sua HASH
git log
git revert HASH
# o git criará um novo commit que desfaz o escolhido
# basta editar a mensagem e commitar
```

Acho que eu c**uei todo o projeto, quero resetar!

Se você é como eu, com certeza já deve ter estragado todo o projeto fazendo experimentos para tentar resolver um problema cabeludo.

Todo mundo já fez isso, rs.

Para voltar ao estado funcional do seu projeto, usamos uma combinação poderosa de comandos.

Só cuidado porque isso não é reversível, hein?!

```
# volta ao estado original da sua master local
git reset --hard origin/master
# pega quaisquer novas mudanças
git fetch origin
# deleta quaisquer arquivos e diretórios
# que não foram adicionados ao git (trackeados)
git clean -d --force
```

Em algum ponto eu me perdi! Tem como voltar no passado?!

Como já é de se imaginar, o git registra absolutamente tudo o que fazemos nos projetos e com isso ganhamos algumas possibilidades, como a de voltar no tempo, por exemplo.

Para tal usamos os comandos **reflog** e **reset**.

```
# com isso você verá uma lista com TUDO
# que você já fez no git
# cada mudança é identificada com um INDEX
git reflog
# identifique a mudança anterior a qual você quer voltar
git reset HEAD@{INDEX}
# está feito
```

Referências

- https://ohshitgit.com/
- https://git-scm.com/docs/
- https://www.atlassian.com/br/git/tutorials/rewriting-history
- https://git-scm.com/book/pt-br/v2/Git-Tools-Stashing-and-Cleaning
- https://git-scm.com/docs/git-log
- https://brorlandi.github.io/git-desfazendo-commits

Gostou do Material?

Se você gostou deste material, pedimos encarecidamente para que compartilhe com os seus colegas de trabalho e ajude a disseminar o conhecimento!

Dúvidas? Sugestões? Críticas?

Fale conosco através do e-mail:
codeprestige.oficial@gmail.com

Nos acompanhe nas redes sociais para mais conteúdo!

Confira outros e-books, vídeos e cursos nas nossas redes sociais!