

Trabalhando com Branches!

Até agora, aprendemos a inicializar um repositório, adicionar arquivos, "commitar" e enviar nosso código para o GitHub. Isso é ótimo para projetos individuais, mas o verdadeiro poder do Git brilha quando trabalhamos em equipe ou em funcionalidades complexas.

Neste material, vamos aprender a usar Branches que são a base para o versionamento e para colaboração em projetos maiores.

O Poder das Branches

1. O que é uma Branch?

Imagine que a branch principal do seu projeto, chamada main (ou master, lembra de quando criamos o repositório no github?), é a linha do tempo oficial e estável do seu código. É a versão que está em produção ou que funciona perfeitamente (versão principal).

Uma branch é uma cópia dessa linha do tempo, uma ramificação que você cria para trabalhar em algo novo sem afetar a linha principal.

Pense nela como um universo paralelo:

Você pode fazer qualquer alteração, experimento ou nova funcionalidade na sua branch.

Pode errar, testar, quebrar o código... tudo isso sem medo de estragar a versão principal (main).

Quando sua funcionalidade estiver pronta e testada, você pode "juntar" (fazer o merge) da sua branch de volta na main.

2. Por que usar Branches?

Isolamento: Permite desenvolver uma nova feature (ex: "login com Google") sem interferir em outra feature que um colega está desenvolvendo (ex: "sistema de pagamento").

Segurança: Mantém a branch main sempre limpa e funcional. Só entra nela código que já foi revisado e aprovado.

Organização: Cada branch pode representar uma tarefa, uma correção de bug ou uma nova funcionalidade. Fica fácil saber no que cada pessoa está trabalhando.

3. Comandos Essenciais para Branches

Vamos supor que estamos num projeto e precisamos criar uma nova tela de perfil de usuário.

- Listar todas as branches locais:

git branch

- Criar uma nova branch. Vamos chamar nossa branch de feature/tela-de-perfil. O prefixo feature/ é uma convenção para indicar que é uma branch de nova funcionalidade:

git branch feature/tela-de-perfil

- Mudar para a nova branch. Criar a branch não te move para ela. Para isso, usamos o switch. Ele será o comando que nos ajudará a navegar entre as branches. (Você deve notar no terminal uma mudança apontando que agora você se encontra na branch recém criada)

git switch feature/tela-de-perfil

Dica: você pode criar e já mudar para a nova branch com um único comando usando a flag -b com checkout:

git checkout -b feature/tela-de-perfil

Agora, qualquer git add e git commit que você fizer será registrado **apenas na branch feature/tela-de-perfil, mantendo a main intacta.**

Ciclo de vida de uma feature

Vamos simular o fluxo completo de trabalho com uma branch:

Após feitos os passos acima temos uma nova branch “feature/tela-de-perfil”.

Trabalhe nela fazendo tudo que for necessário, criando arquivos, novos códigos, editando códigos que já existam, etc.

Lembre-se de ir fazendo commits a cada pequenas alterações e edições do código, **estes commits ficarão ligados a branch atual que você está.** Basta fazer como sempre fez, “git add .” , “git commit -m”.

Com sua funcionalidade pronta, agora é hora de integrá-la a linha do tempo principal.

- 1 Volte para a branch principal:

git switch main

- 2 Garanta que a main está atualizada:

git pull origin main

- 3 Traga as alterações da sua branch para a main:

git merge feature/tela-de-perfil

Se não houverem conflitos agora sua nova feature faz parte do código principal (main)!

- 4 Se tudo tiver ocorrido sem conflitos, já podemos deletar a branch local, ela não é mais necessária já que o código agora faz parte da main.

git branch -d feature/tela-de-perfil

- 5 Agora é só fazer o push da sua main para o github:

git push origin main

Como já salientado em aula, para aprender a fazer é necessário praticar. Pratiquem os comandos acima para entenderem o fluxo do processo e tudo ficará muito mais fácil (eu prometo!)

Com o domínio deste fluxo de processos, podemos dar um passo adiante e contribuir em repositórios com vários desenvolvedores, sejam privados ou públicos.