



Alberto Rebello 24/02/2022 12:17 [Publicado na Digital Innovation One](#)

Oi,

Quero compartilhar essa conquista com vocês. Consegui fazer meu primeiro repositório git com uma organização previamente planejada.

Trabalhei de forma ordenada, conseguindo usar os recursos deste VCS para me auxiliar na produtividade, em especial evitando aquele problema de perdermos um código que estava funcionando ao tentar incluir uma nova funcionalidade ou durante refatoração para melhorar sua expressividade.

Já havia assistido a vídeos da DIO, que são ótimos, mas a falta de vivência e de fundamento teórico não me deixavam pôr em prática as vantagens do git. Eu tinha a relação de comandos explicadas nas aulas, entendia o conceito geral mas acredito que alguns "gaps", lacunas no entendimento atrapalhavam.

"Não sei qual é a dificuldade que os devs têm de ler a documentação original antes de buscarem vídeos de tutoriais. Em geral elas são ótimas e bem explicativas."

Essa frase ficou na minha cabeça quando a ouvi há algumas semanas. Concordei com ela e dessa vez baixei o documento do site do git:

<https://git-scm.com/book/en/v2>

Muito bom! me ajudou demais. Respondi a um comentário em um site de educação da JetBrains que me motivou a escrever esse artigo para a comunidade DIO. Nele descrevi o processo que usei para organizar meu trabalho e depois fazer o push no repositório: <https://github.com/AlbertoVictorRebello/academy.jetbrains.kotlinBasics.git>

Espero que seja útil para quem está começando. Se você tem sugestões para eu melhorar este roteiro, por favor me fale. Vou gostar muito de saber.

Contexto: a JetBrains Academy propõe seus problemas em estágios/fases. Neste problema era proposto criarmos uma abstração de uma máquina de café. Ocorre que de um estágio para o outro as funcionalidades anteriormente podem ser modificadas ou abandonadas, não só incluídas novas. Isso é feito porque o problema é pensado para ir ficando mais complexo a cada nova fase. O roteiro de trabalho abaixo propõe uma forma, entre tantas outras possíveis, de usarmos o Git como ferramenta de produtividade. A descrição abaixo faz referência ao projeto Coffee Machine mas pode ser adaptado a qualquer outro:

Obs: antes de cada comando "commit" ser executado, os arquivos que serão rastreados devem ser incluídos usando-se a IDE ou o comando "git add [nome do arquivo]" ou "git add -i" para acionar o modo interativo.

- 1-Crie uma pasta chamada CoffeeMachine
- 2-Inicie a IDE definindo o caminho que direciona a pasta para implantar todos os arquivos do JetBrains nela. Nesta pasta o site da JetBrains vai descarregar todos os arquivos do projeto.
- 3-Crie o repositório git usando o comando IDE ou CLI => git init
- 4-Execute o comando commit para iniciar o trabalho.
- 5-Crie um novo branch chamado, por exemplo, stage01
- 6-Execute o comando checkout para o novo branch stage01.
Faça o trabalho até ser aprovado na verificação do estágio 1 com sucesso.
- 7-Execute o comando commit para o job feito no branch stage01.
- 8-Execute o comando checkout para master branch
- 9-Execute o comando merge para mesclar o branch stage 01 no branch master
- 10- Retorne o item 5 deste roteiro, criando novas branches para as próximas etapas 02, 03, 04, 05 e 06.

Dessa forma, podemos alterar qualquer coisa de um estágio para outro e manter o código de cada estágio disponível para ser executado.

Você também pode criar uma ramificação temporária quando já tiver uma solução parcial funcionando durante um estágio antes de tentar implementá-la. Caso não funcione, você pode voltar ao ponto em que tudo estava funcionando bem. Se estiver ok, apenas faça o merge no branch que você está trabalhando naquele momento.