# **INSTITUTO DE COMPUTAÇÃO - UNICAMP**

**Especialização em Engenharia de Software**

**INF335 – Ambientes para Concepção de Software**

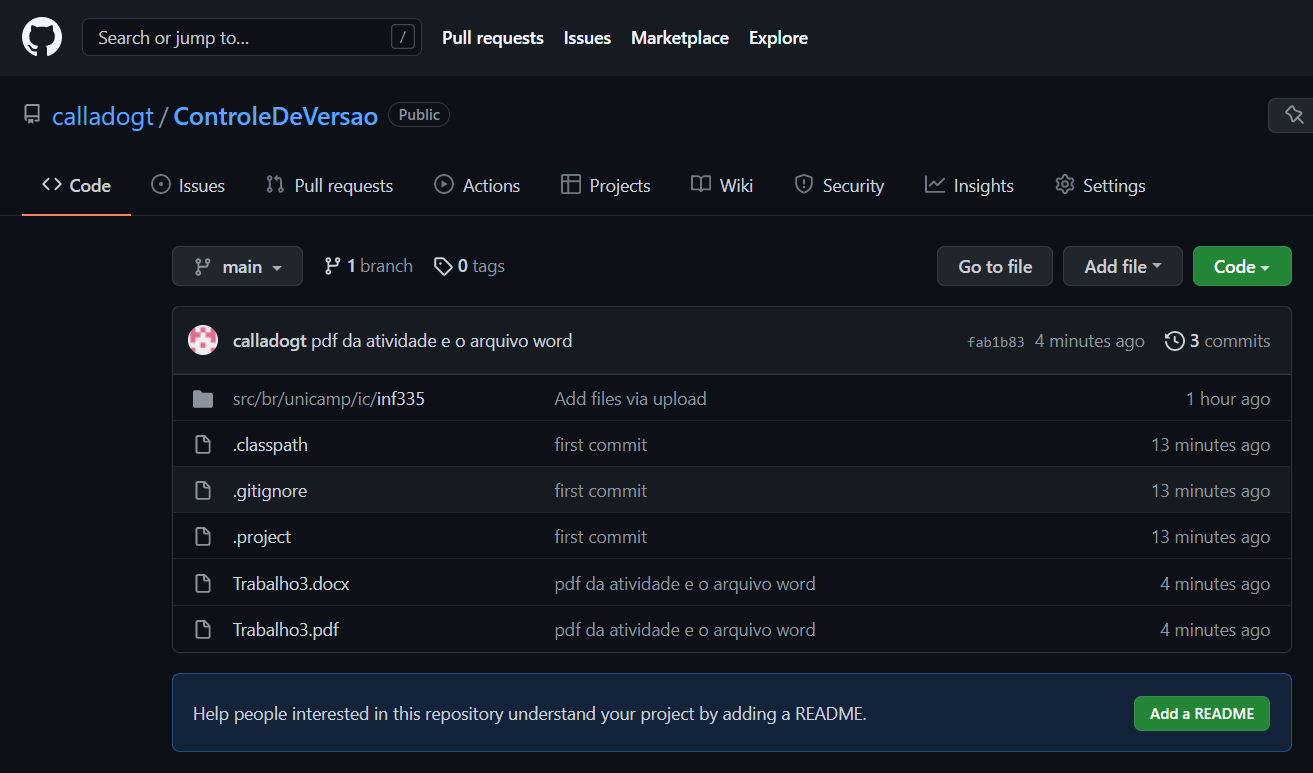
**Nome/RG**: Gabriel Teixeira Callado – 45.022.276-7

# **Resposta Atividade 3**

A atividade 3 tem como objetivo desenvolver habilidades de uso de sistema de controle de versão “Git”. Para realizar o mesmo, as seguintes atividades foram levantadas como necessárias:

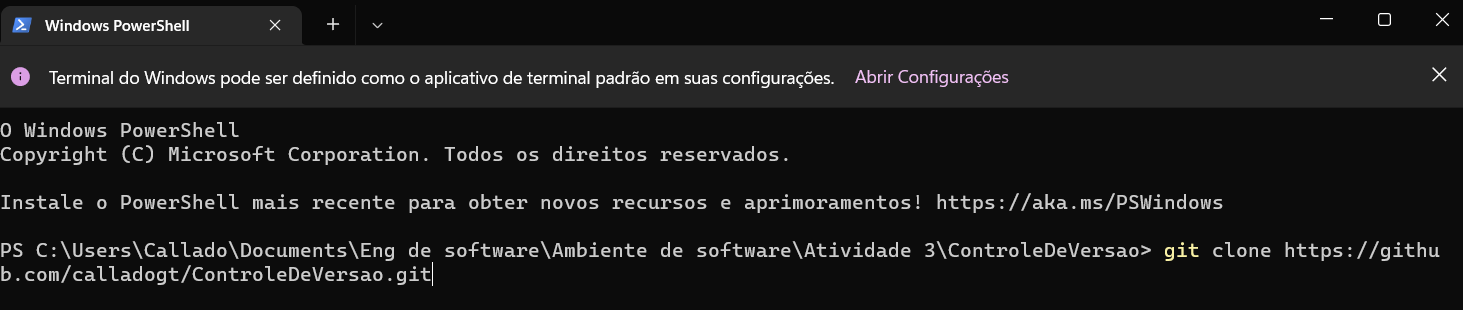
1. *Inserir código no GitHub*

Nesta primeira etapa, criei o repositório no GitHub ***(https://github.com/calladogt/ControleDeVersao***) e inseri o arquivo disponibilizado:



1. *Clonar repositório no próprio computador (local)*

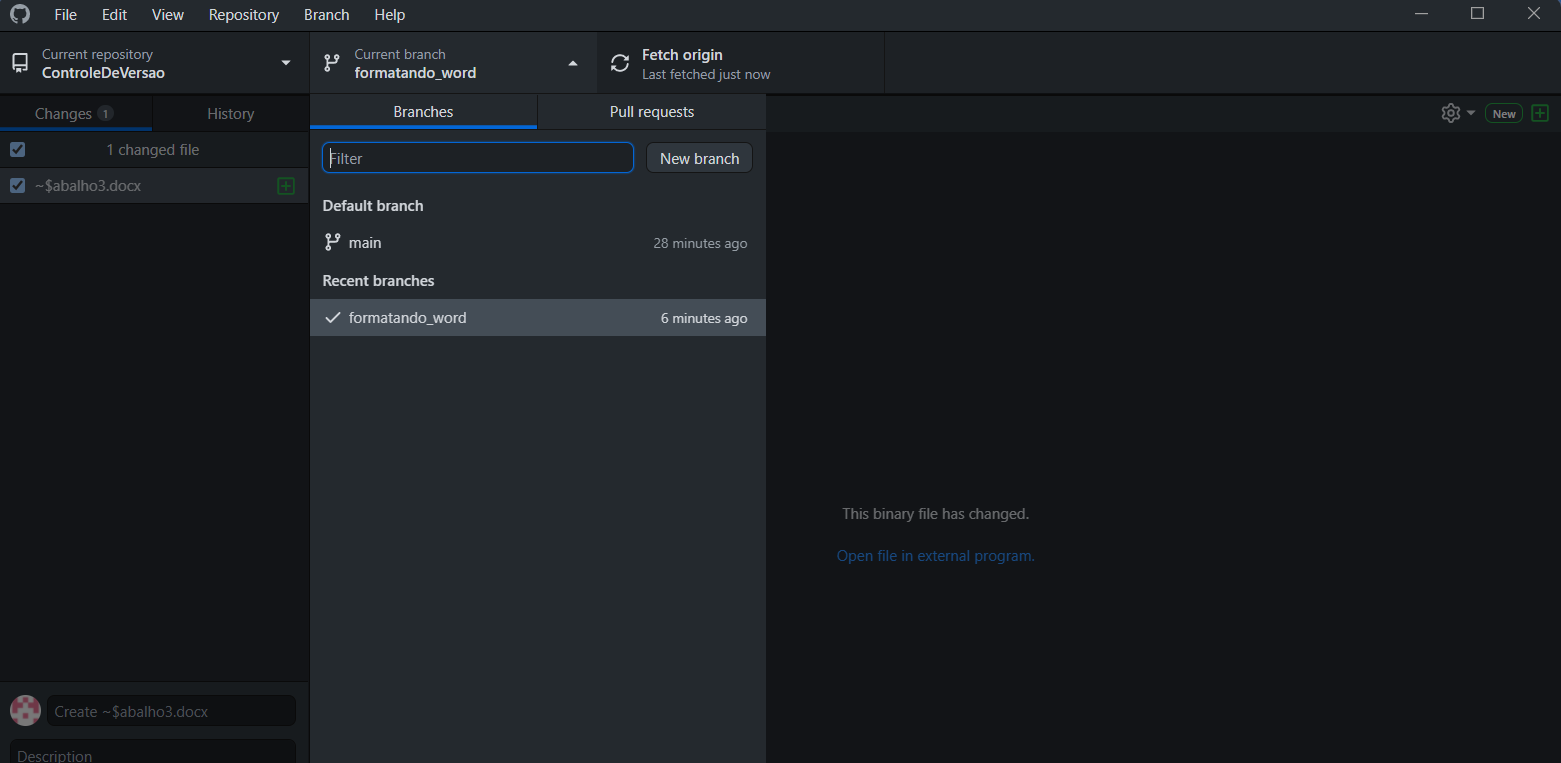
Clonei o repositório abrindo o terminal na pasta do projeto local e dando um “git clone url”:



1. *Criar nova branch para correções*

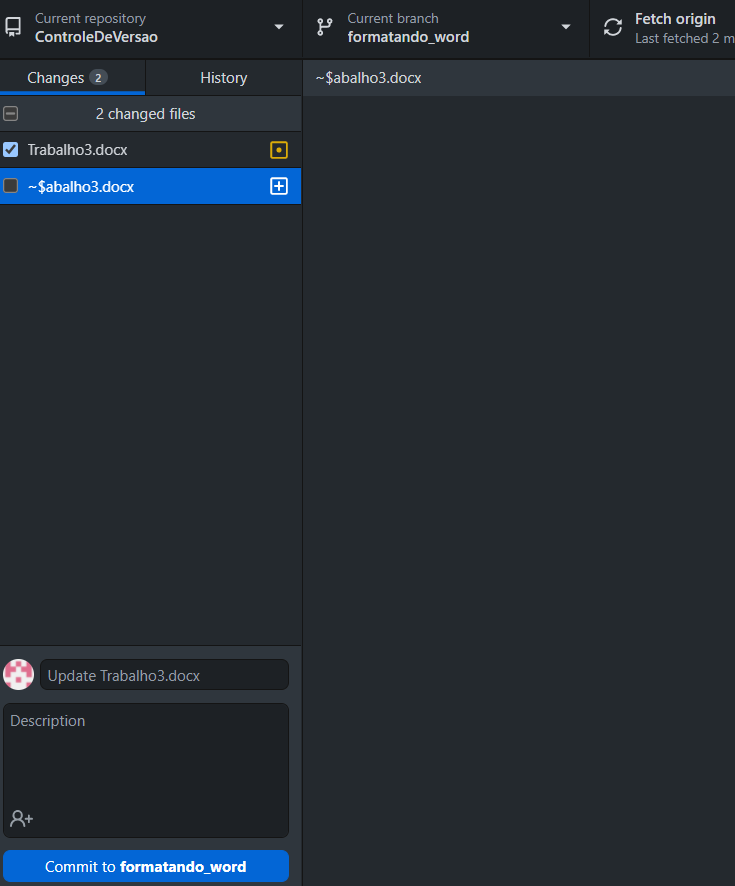
Neste momento, usei o software denominado *GitHub Desktop* para criar as branches, fazer commits e etc. Ele é bem visual e útil, eliminando a necessidade de executar os códigos em linha de comando como feito durante a aula. Até a clonagem do item anterior poderia ser feita diretamente dele.

Na imagem abaixo, temos o menu do *GitHub Desktop* onde vemos a possibilidade de criar uma branch (eu já tinha criado uma chamada “formatando\_word” para colocar as mudanças da formatação deste relatório).



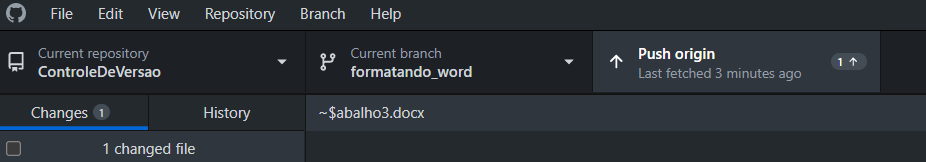
1. *Alterar localmente, colocar no stage e fazer commit no repositório local*

À cada alteração feita, aparecia no meu *GitHub Desktop* a opção de dar o commit com uma mensagem.



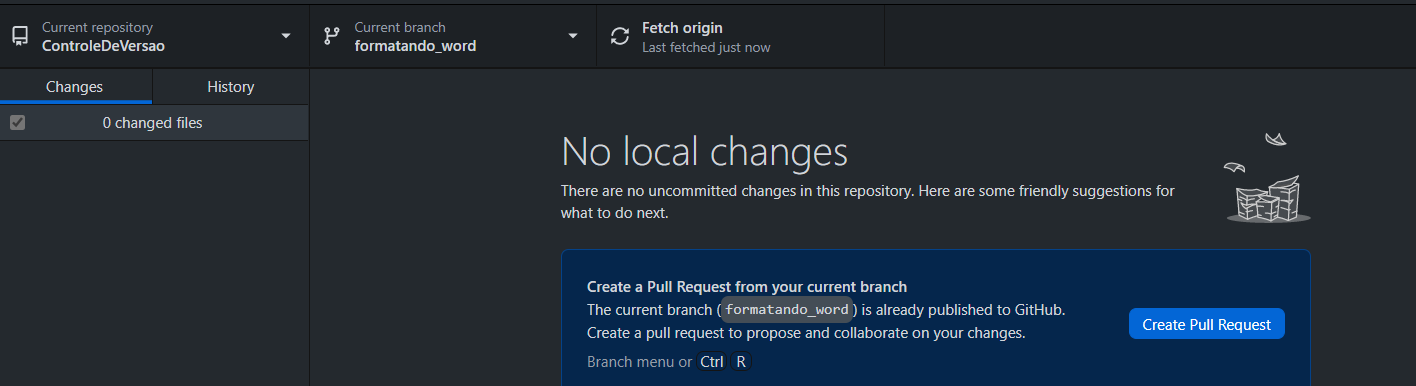
1. *Atualizar o GitHub (push)*

Logo em seguida, fiz o push pro repositório para o caso de alguém estar trabalhando na mesma branch.

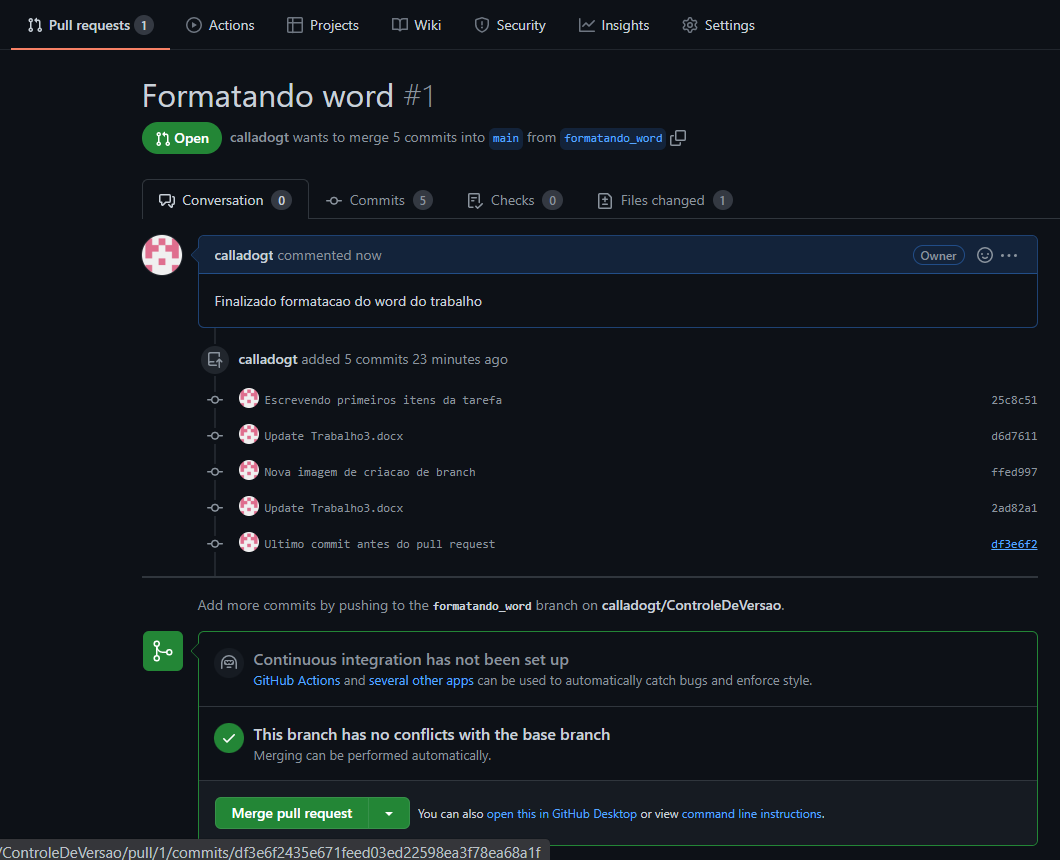


1. *Fazer merge das branches criadas*

Logo em seguida, após commitar tais mudanças da branch, nos aparece a opção pull request:

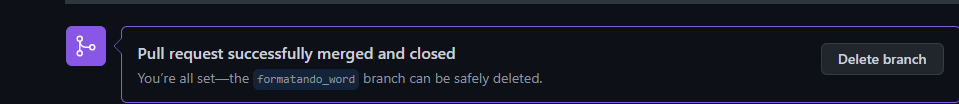


Clicando nela, aparece um campo de texto para descrever a solicitação e, após enviá-la, podemos ver o pull request:

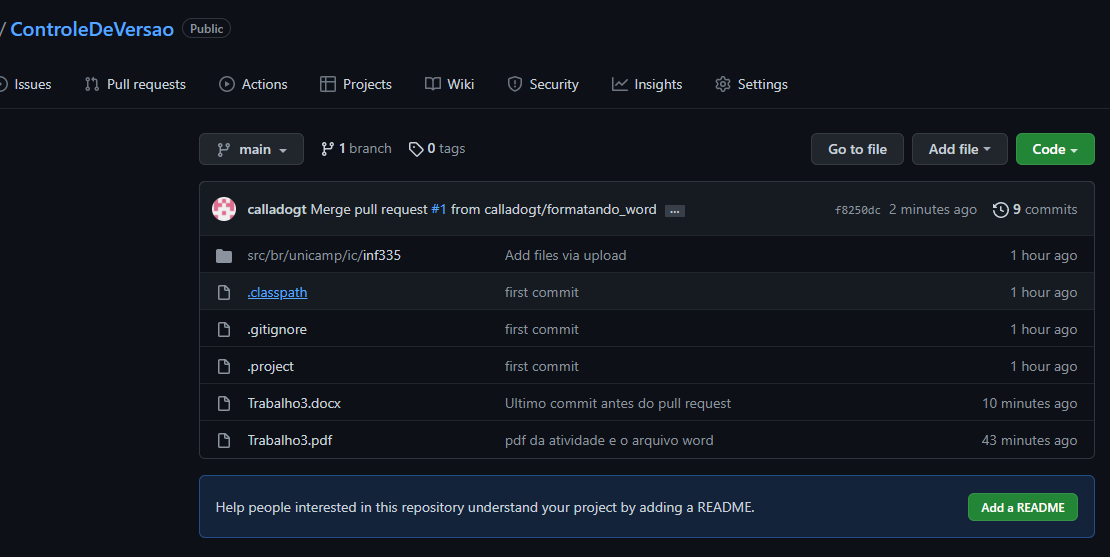


E já podemos aceitar e fazer o merge como mostrado no botão na parte de baixo da imagem acima.

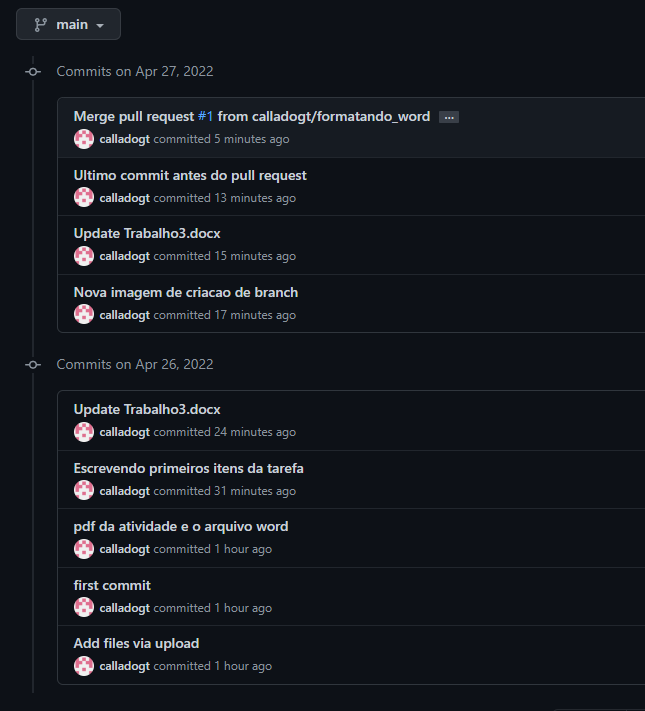
Ao confirmar o merge, temos a branch main atualizada com as mudanças da outra branch. Em seguida, aparece a msg para apagar a branch e podemos deleta-la para ficar apenas com a principal.



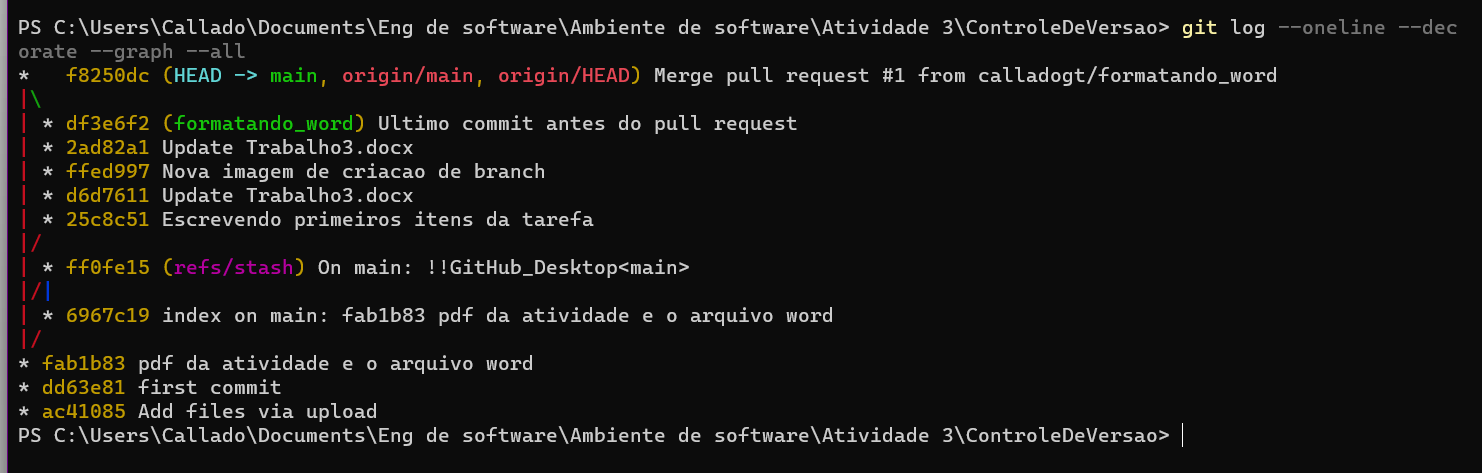
Por fim, vemos o projeto no *GitHub* com o merge feito:



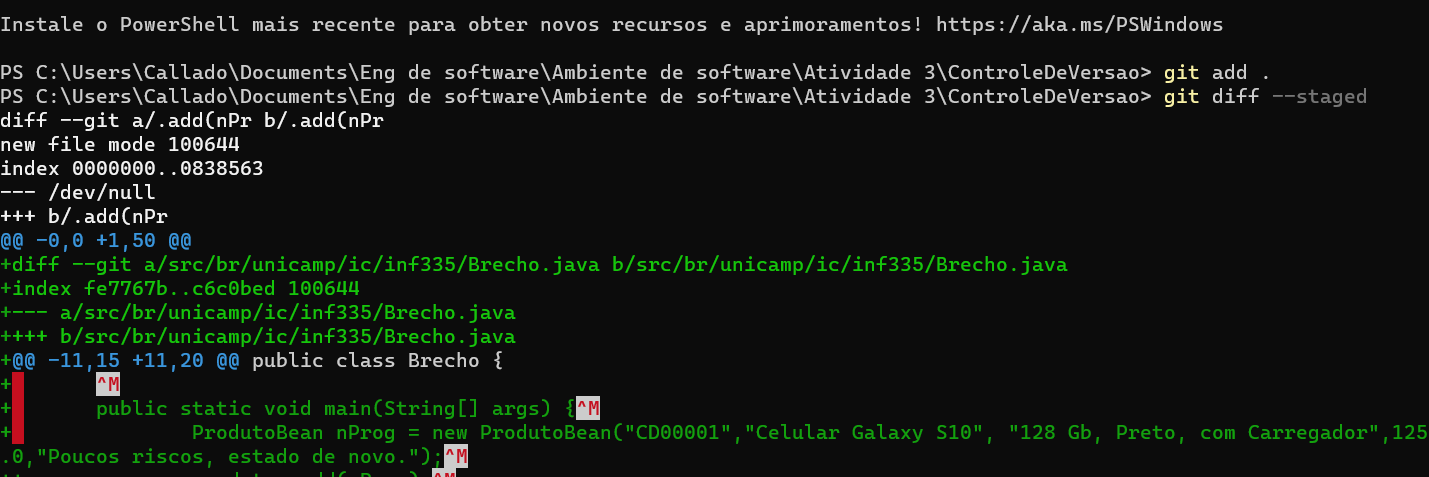
e o histórico mostrando que foi executado tudo com sucesso:



O print com o log das alterações conforme solicitado:



Em linha de comando, vou fazer as alterações do arquivo do Brecho java e aí colocar no stage:



Logo em seguida já dou o commit e o push pro repositório:

