

# INSTITUTO DE COMPUTAÇÃO - UNICAMP

## Especialização em Engenharia de Software INF335 – Ambientes para Concepção de Software

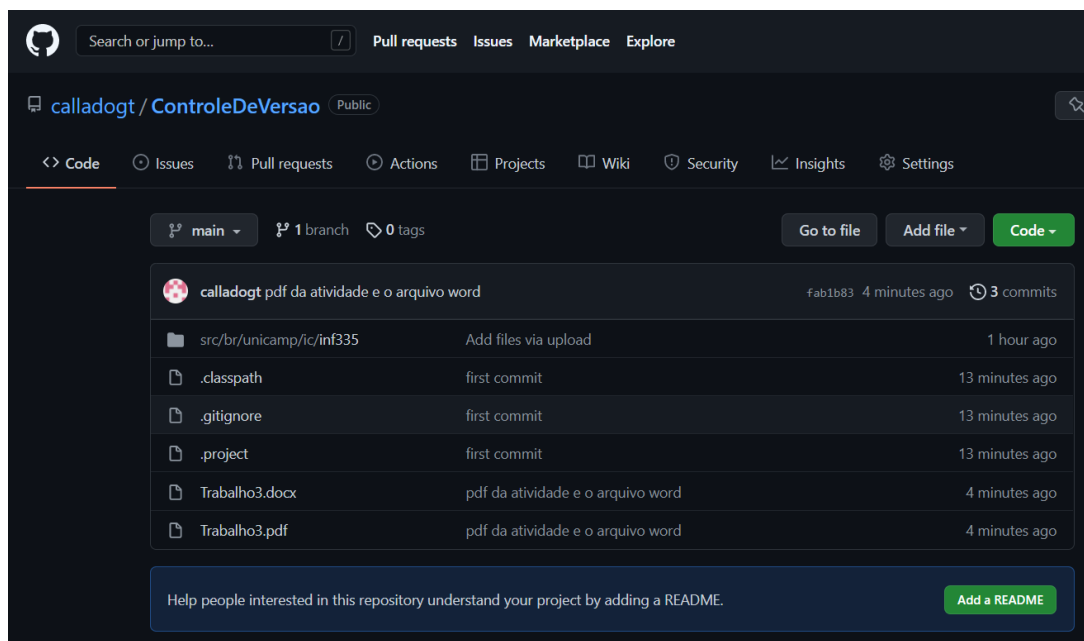
Nome/RG: Gabriel Teixeira Callado – 45.022.276-7

### 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:

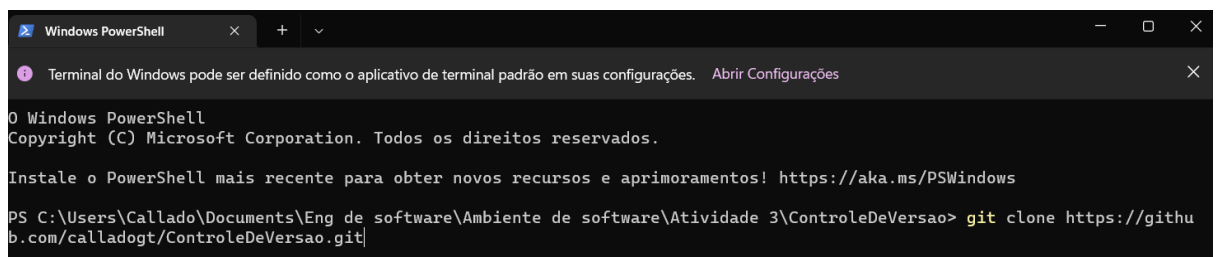
1) *Inserir código no GitHub*

Nesta primeira etapa, criei o repositório no GitHub e inseri o arquivo disponibilizado:



2) *Clonar repositório no próprio computador (local)*

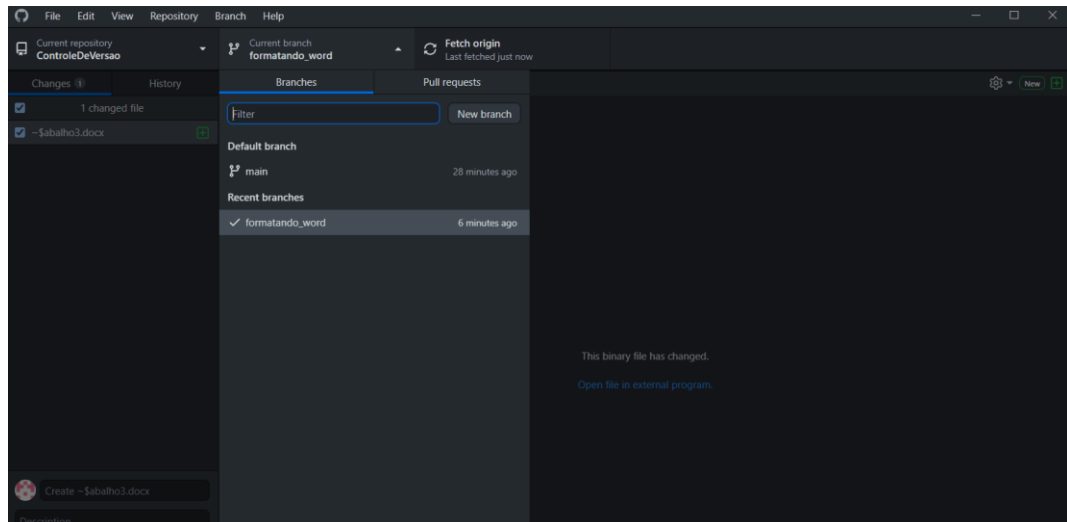
Clonei o repositório abrindo o terminal na pasta do projeto local e dando um “git clone url”:



### 3) Criar nova branch para correções

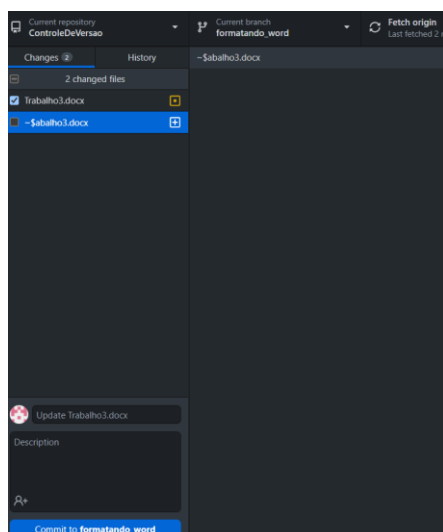
Neste momento, usei o software denominado *GitHub Desktop* para criar as branches, fazer commits e etc. Ele é bem visual e útil, eliminando a necessidade de executar os códigos em linha de comando como feito durante a aula. Até a clonagem do item anterior poderia ser feita diretamente dele.

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).



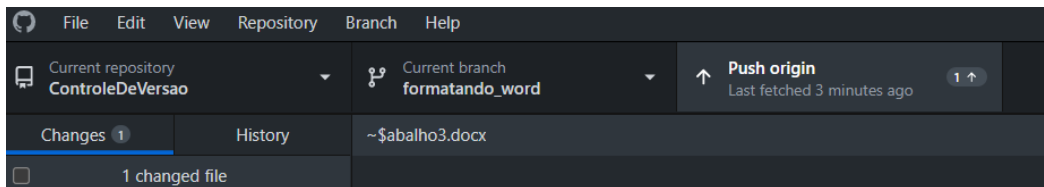
### 4) Alterar localmente, colocar no stage e fazer commit no repositório local

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



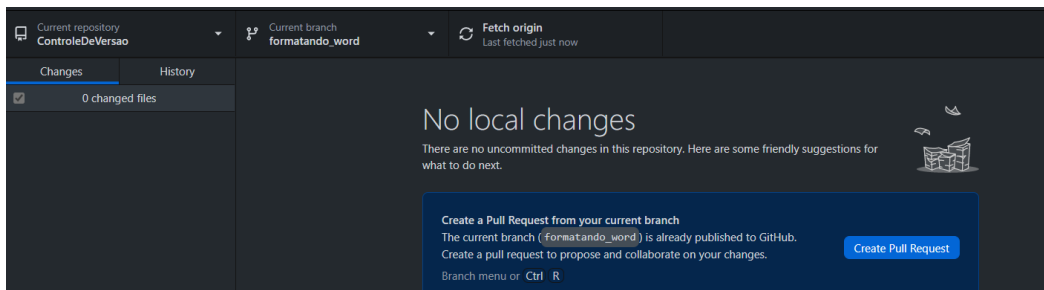
### 5) Atualizar o GitHub (push)

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

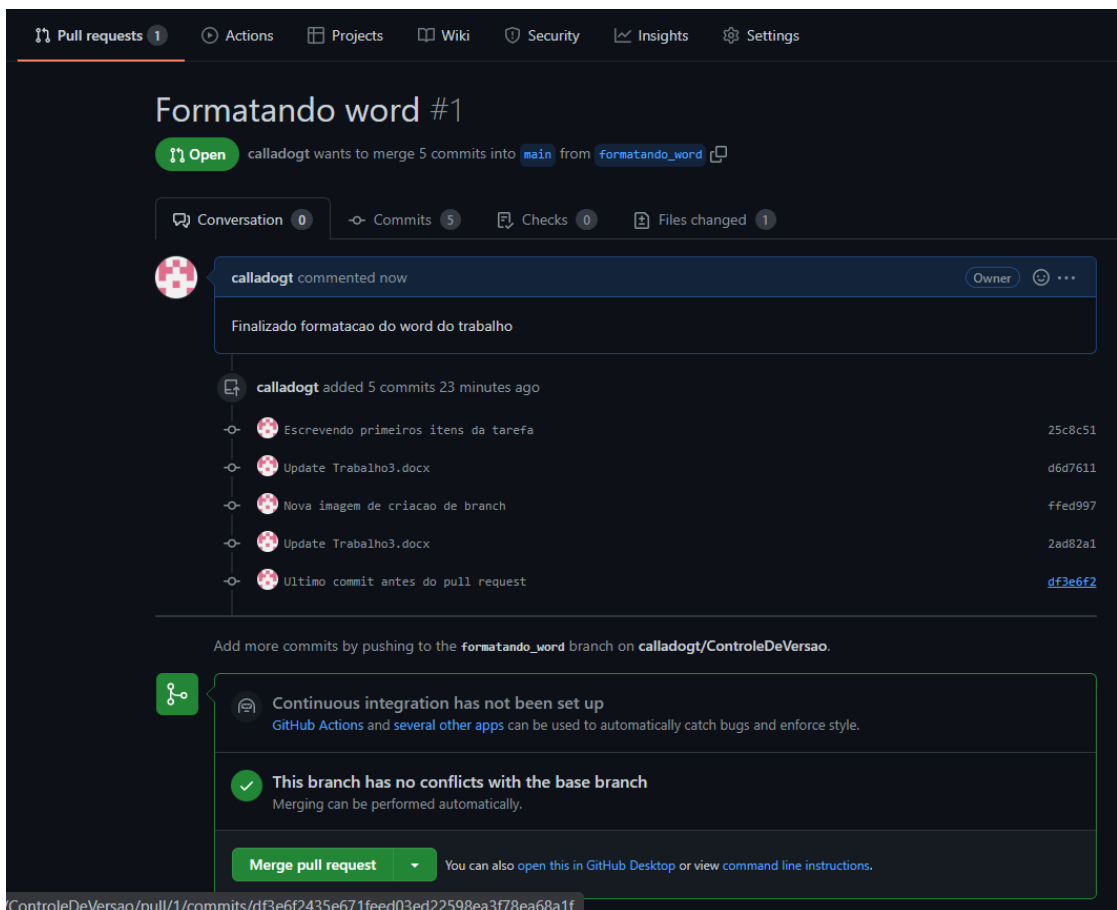


## 6) Fazer merge das branches criadas

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

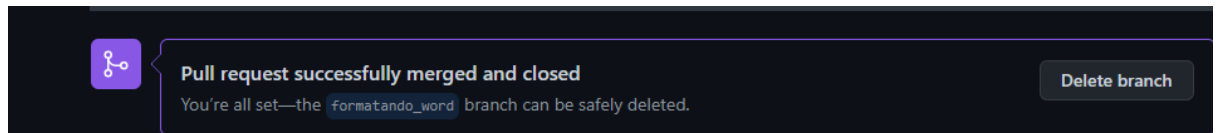


Clicando nela, aparece um campo de texto para descrever a solicitação e, após enviá-la, podemos ver o pull request:

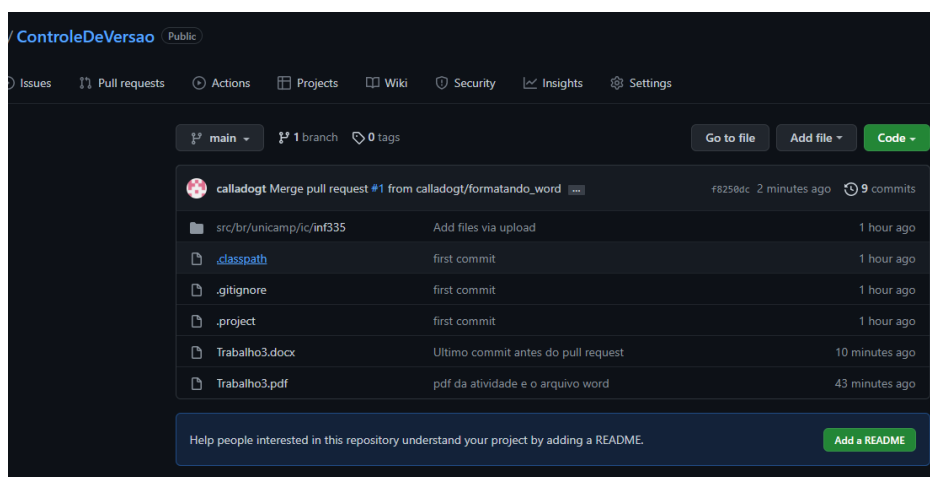


E já podemos aceitar e fazer o merge como mostrado no botão na parte de baixo da imagem acima.

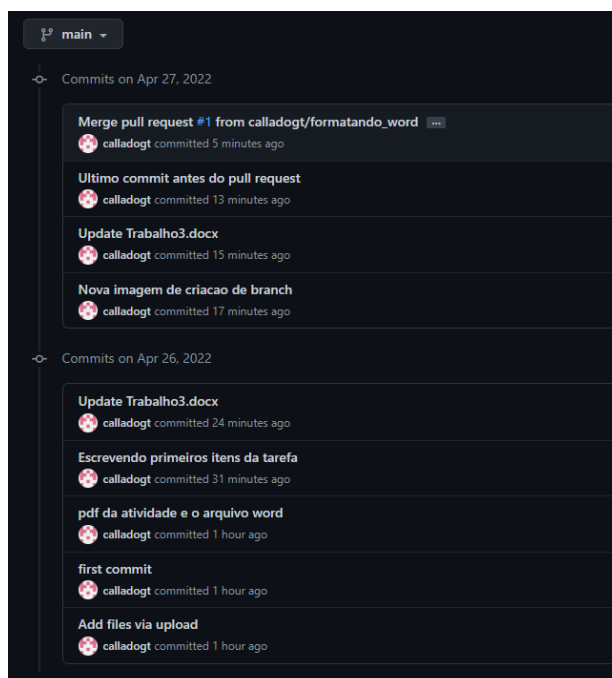
Ao confirmar o merge, temos a branch main atualizada com as mudanças da outra branch. Em seguida, aparece a msg para apagar a branch e podemos deleta-la para ficar apenas com a principal.



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:

```
PS C:\Users\Callado\Documents\Eng de software\Ambiente de software\Atividade 3\ControleDeVersao> git log --oneline --decorate --graph --all
* f8250dc (HEAD -> main, origin/main, origin/HEAD) Merge pull request #1 from calladogt/formatando_word
| \
| * df3e6f2 (formatando_word) Ultimo commit antes do pull request
| * 2ad82a1 Update Trabalho3.docx
| * ffed997 Nova imagem de criacao de branch
| * d6d7611 Update Trabalho3.docx
| * 25c8c51 Escrevendo primeiros itens da tarefa
| /
| * ff0fe15 (refs/stash) On main: !!GitHub_Desktop<main>
| /|
| * 6967c19 index on main: fab1b83 pdf da atividade e o arquivo word
| /
* fab1b83 pdf da atividade e o arquivo word
* dd63e81 first commit
* ac41085 Add files via upload
PS C:\Users\Callado\Documents\Eng de software\Ambiente de software\Atividade 3\ControleDeVersao> |
```

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

```
Instale o PowerShell mais recente para obter novos recursos e aprimoramentos! https://aka.ms/PSWindows
PS C:\Users\Callado\Documents\Eng de software\Ambiente de software\Atividade 3\ControleDeVersao> git add .
PS C:\Users\Callado\Documents\Eng de software\Ambiente de software\Atividade 3\ControleDeVersao> git diff --staged
diff --git a/.add(nPr b/.add(nPr
new file mode 100644
index 0000000..0838563
--- /dev/null
+++ b/.add(nPr
@@ -0,0 +1,50 @@
+diff --git a/src/br/unicamp/ic/inf335/Brecho.java b/src/br/unicamp/ic/inf335/Brecho.java
+index fe7767b..c6c0bed 100644
+--- a/src/br/unicamp/ic/inf335/Brecho.java
++++ b/src/br/unicamp/ic/inf335/Brecho.java
+@@ -11,15 +11,20 @@ public class Brecho {
+
+     public static void main(String[] args) {
+         ProdutoBean nProd = new ProdutoBean("CD00001", "Celular Galaxy S10", "128 Gb, Preto, com Carregador", 125
+         @, "Poucos riscos, estado de novo.");
+     }
+ }
```

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

```
PS C:\Users\Callado\Documents\Eng de software\Ambiente de software\Atividade 3\ControleDeVersao> git commit -m "Corrigindo
do append e loops no Brecho"
[main 3559da6] Corrigindo append e loops no Brecho
3 files changed, 60 insertions(+), 5 deletions(-)
create mode 100644 .add(nPr
create mode 100644 ~$abalho3.docx
PS C:\Users\Callado\Documents\Eng de software\Ambiente de software\Atividade 3\ControleDeVersao> git push
```