Criação de Repositório GitHub

DAS



Guilherme Ribeiro-2024094

Duarte Palha-2024049

[Intruducão 2](#_Toc538385707)

[1-Criar o Repositório no GitHub 3](#_Toc956517136)

[2- Criar e Enviar as Branches Necessárias 4](#_Toc1248250650)

[4- Configurar Níveis de Acesso no GitHub 5](#_Toc2071887474)

[5-Criar um “.gitignore” 5](#_Toc478852257)

[6- Controlar Versões 5](#_Toc2055086415)

[7-Combinar alteracões 6](#_Toc224580508)

[8-Atualizar 6](#_Toc246980687)

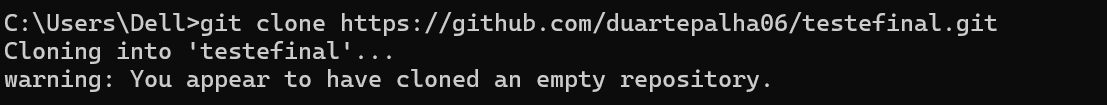
[Conclusão 7](#_Toc634339026)

# Intruducão

Neste projeto pretende-se criar um repositório no GitHub, usando em primeira instância o command line do Git.

# 1-Criar o Repositório no GitHub

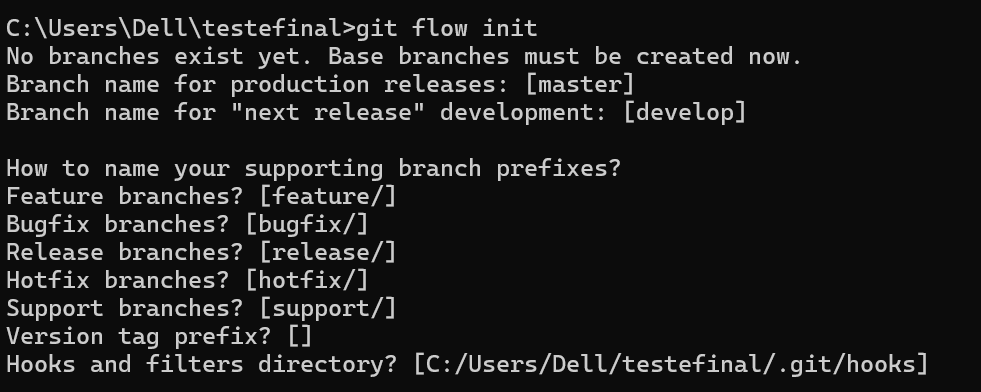
Para comecar fazemos git clone para criar pasta localmente. De seguida fazemos cd para conseguirmos aceder à pasta.

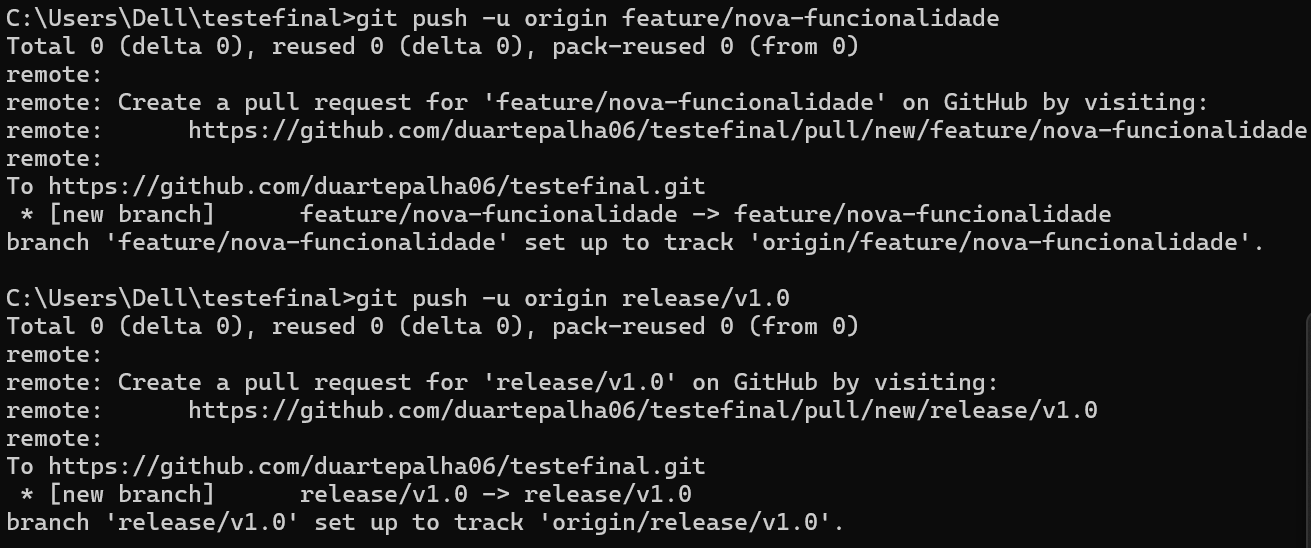




# 2- Criar e Enviar as Branches Necessárias

Os comandos criam a branch develop a partir da main e enviam-na para o GitHub. De seguida, criam branches para novas funcionalidades, lançamentos e correções urgentes, que são também enviadas para o repositório remoto.





# 4- Configurar Níveis de Acesso no GitHub

No GitHub, vamos a configurações> gerir acessos.

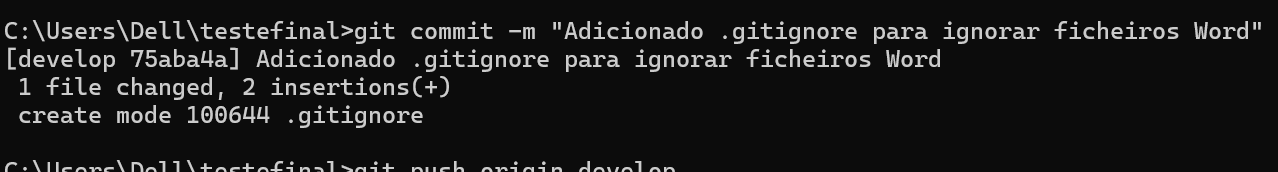
Adiciona os developers com permissão de Write (podem submeter código, mas não alterar configurações).

Mantém apenas os proprietários com permissão de Admin.

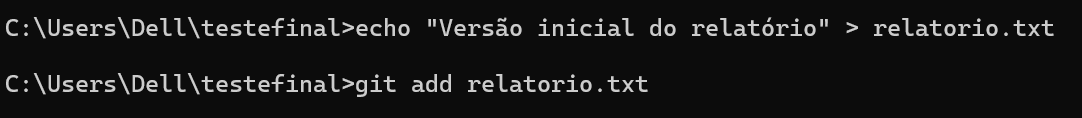
# 5-Criar um “.gitignore”

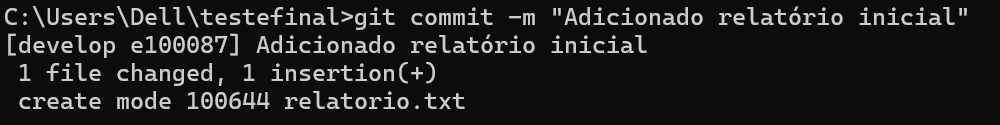
O **.gitignore** é um ficheiro utilizado no Git para indicar quais ficheiros e pastas devem ser ignorados pelo controlo de versões.

Neste vamos ignorar ficheiros Word.



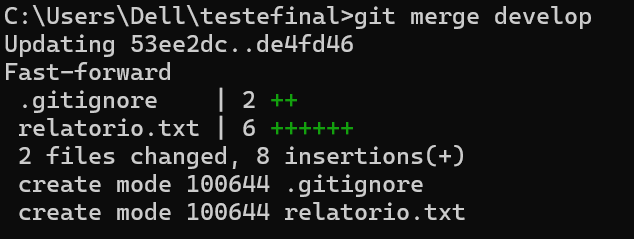
# 6- Controlar Versões





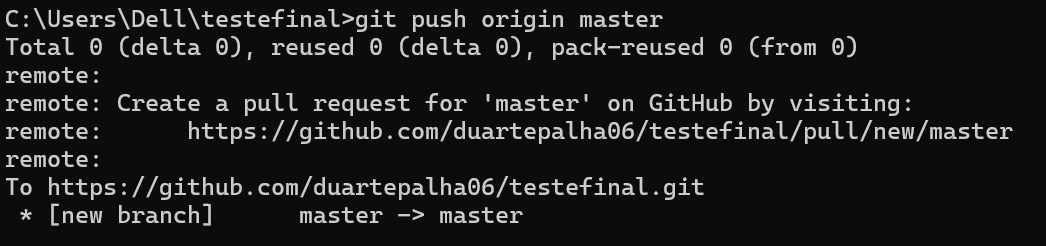
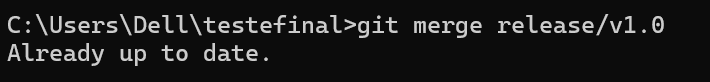
# 7-Combinar alteracões

O comando git merge é usado para **combinar alterações** de diferentes ramos (branches) num único ramo.



# 8-Atualizar

Por fim atualizar tudo



# Conclusão

Para concluir este trabalho permitiu-nos aprofundar o nosso conhecimento sobre o Git, compreendendo melhor o seu funcionamento e a importância da gestão de versões no desenvolvimento de software. Explorámos conceitos fundamentais, como git merge, .gitignore e a resolução de conflitos, o que nos proporcionou uma visão mais clara sobre boas práticas na utilização desta ferramenta essencial para colaboração e controlo de código.