Criação de Repositório GitHub

DAS



Guilherme Ribeiro-2024094

Duarte Palha-2024049

[Intruducão 2](#_Toc1155596492)

[1-Criar o Repositório no GitHub 3](#_Toc293674298)

[2- Criar e Enviar as Branches Necessárias 4](#_Toc478602408)

[4- Configurar Níveis de Acesso no GitHub 5](#_Toc1759341426)

[5-Criar um “.gitignore” 5](#_Toc1635677789)

[6- Controlar Versões 5](#_Toc1497081352)

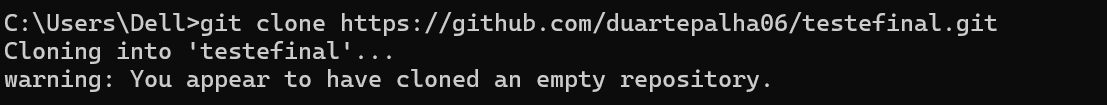
[7-Combinar alteracões 6](#_Toc850323580)

# 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

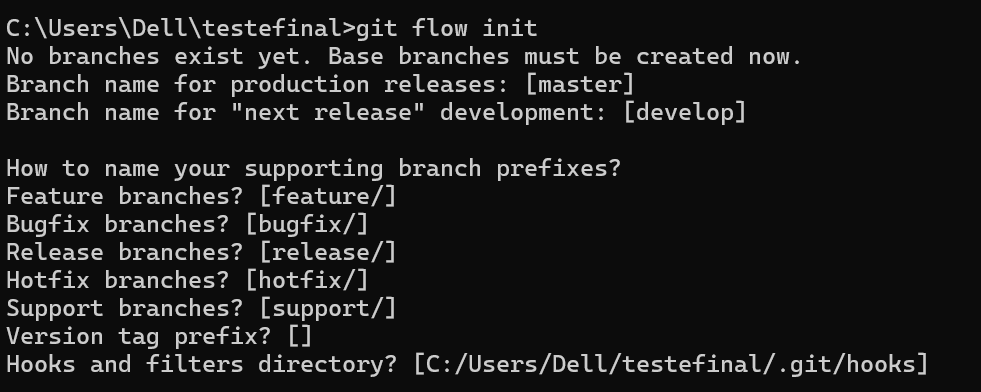
Para comecar fazemos git clone para criar pasta localmente. De seguida fazemos cd para conseguirmos aceder à pasta.

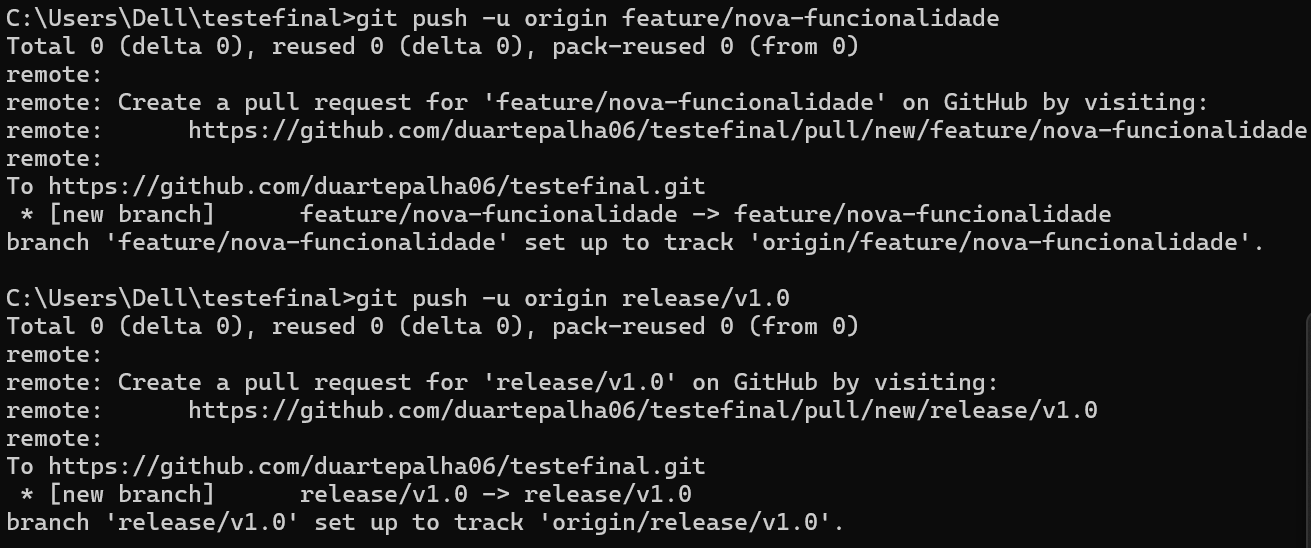




# 2- Criar e Enviar as Branches Necessárias

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.

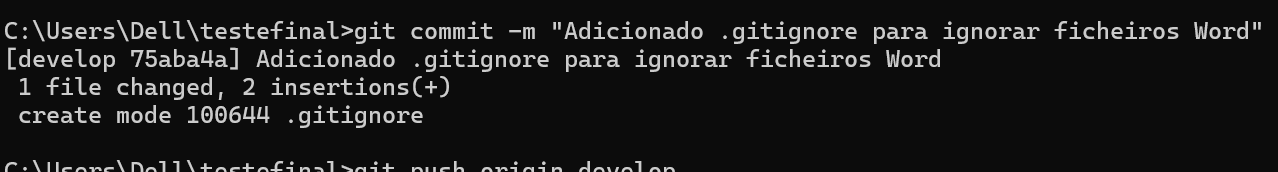
Adiciona os developers com permissão de Write (podem submeter código, mas não alterar configurações).

Mantém apenas os proprietários com permissão de Admin.

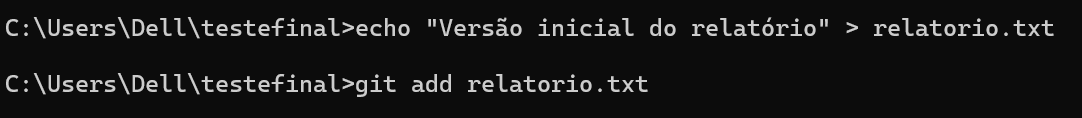
# 5-Criar um “.gitignore”

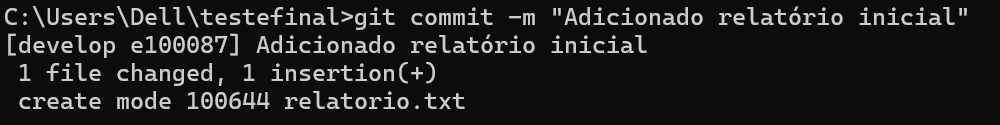
O **.gitignore** é um ficheiro utilizado no Git para indicar quais ficheiros e pastas devem ser ignorados pelo controlo de versões.

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.

