 Istec-porto.pt

Curso Técnico Superior Profissional (CTeSP) de Desenvolvimento de Software

**Criação de Repositório GitHub**

Desenvolvimento Ágil de Software

Trabalho realizado por: Rafael Filipe Correia da Silva – 2022075 – D.S.

**Processo de criação do Repositório**

Localmente, comecei por escolher onde queria que o meu repositório local ficasse localizado e iniciei o Git Bash nessa pasta. Uma vez o Git Bash aberto, inicializei o repositório local com o comando:

$ git init

Depois, no browser acessei o GitHub e criei um novo repositório remoto. Para adicionar o repositório remoto ao repositório local, fiz o seguinte comando:

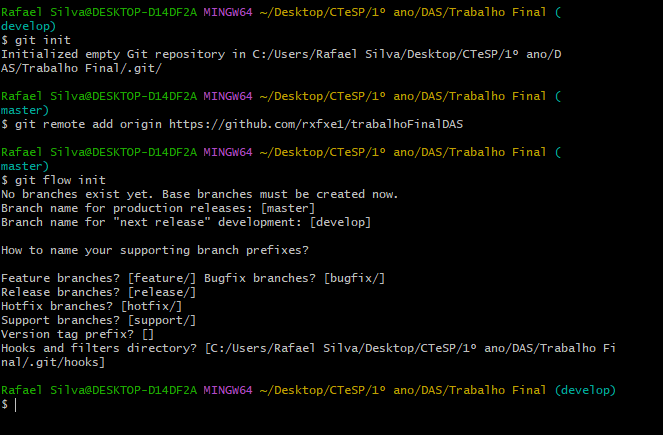
$ git remote add origin <url do meu repositório GitHub>

De seguida, inicializei o modelo de fluxo de trabalho GitFlow com o comando:

$ git flow init

Este comando pergunta sobre algumas configurações, como a criação das 2 branches principais master/main e develop, e também sobre o prefixo dos branches de feature, release, hotfix e sufixo de suporte e versionamento. Essas configurações podem ser deixadas como default ou personalizadas

Deixo em baixo, um print do GitBash com todos os comandos:

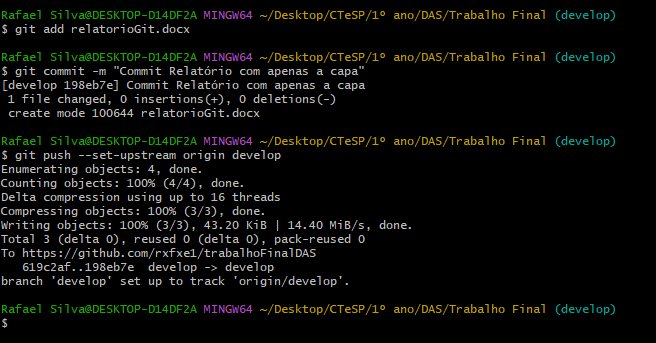


**Link de acesso ao repositório GitHub:**

<https://github.com/rxfxe1/trabalhoFinalDAS>

**Controlo de versões do Relatório usando o Repositório:**

Carregamento inicial do Relatório no branch develop:

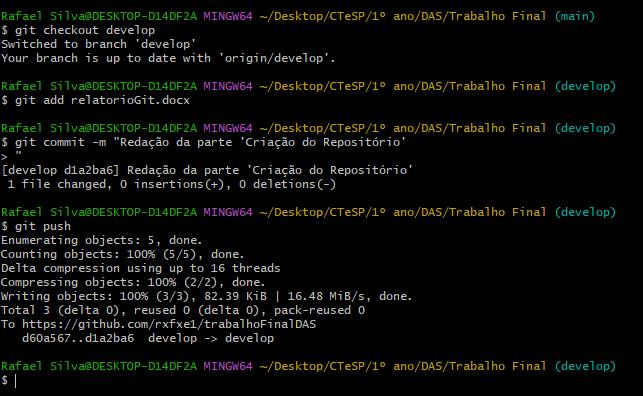


5 alterações no próprio branch develop com os devidos comentários:

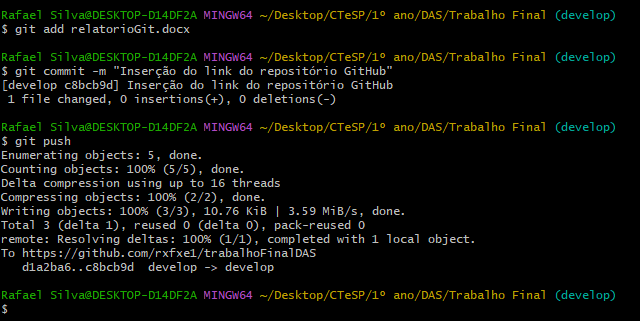
Para todas as alterações que vamos ver nesta lista, todas foram feitas da mesma maneira, seguindo os mesmos passos, estes sendo:

1. Verificar que estamos no branch main
2. Fazer as alterações no Relatório
3. No Git Bash fazer o comando: $ git add relatorioGit.docx
4. Em seguida, o commit com: $ git commit -m “*comentário das alterações*”
5. E finalizar com: $ git push ou $ git –set-upstream origin develop

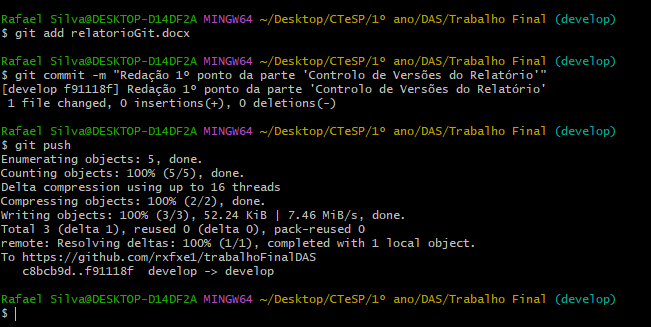
A 1ª alteração é a redação da secção ‘Processo de Criação do Repositório’, podemos ver na imagem em baixo todo o processo no Git Bash.



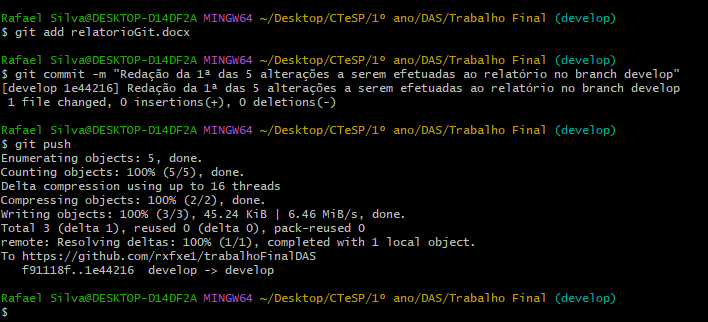
A 2ª alteração é também a redação de um das partes do relatório, neste caso é a redação da secção “Link de acesso ao Repositório”.



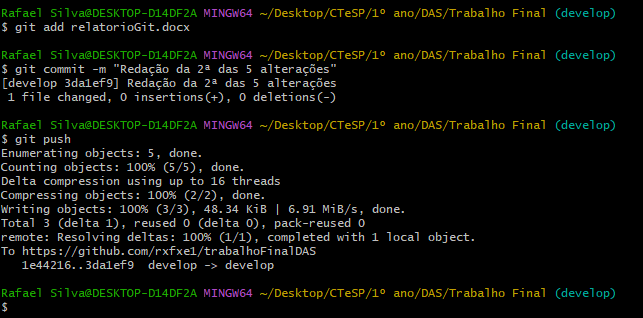
A 3ª alteração do Relatório no branch develop trata-se da adição da secção Controlo de Versões do Relatório e a redação do seu 1º ponto.



Na 4ª alteração temos a redação e inserção de uma imagem demonstrativa do 1º dos 5 pontos que estamos a apresentar neste momento.



Para finalizar, a 5ª alteração é a adição 2ª alteração e respetivo texto e imagem.



Release que contribui para o branch master:

Para fazer um release que contribua para o branch main devemos estar no branch develop: $ git checkout develop

Iniciamos uma nova release com o comando "$ git flow release start [versão]"; eu escolhi utilizar a versão 0.5 então utilizei o seguinte comando: $ git flow release start 0.5



Fazemos as alterações no Relatório e fazemos commit para a release.

O 2º passo é fazer $ git add e $ git commit para adicionar e fazer commit das alterações.

Quando as alterações estiverem prontas, finalizamos a release com o comando

$ git flow release finish 0.5

Isso automaticamente fará o merge das alterações do branch release/0.5 para o branch master e o branch develop, além disso, irá criar uma tag com o nome da release

Isso fará com que as alterações da release sejam mescladas para o branch master, permitindo que a versão mais recente do seu código esteja disponível nesse branch. Lembre-se de fazer o merge do branch develop após a finalização da release.

**Hotfix efetuado utilizando a gestão de braches GitFlow**

Podemos seguir os seguintes passos para fazer um hotfix utilizando a gestão de branches do GitFlow no Git Bash:

Certificar de que estamos no branch master: $ git checkout master

Iniciar um novo hotfix com o comando: $ git flow hotfix start 1



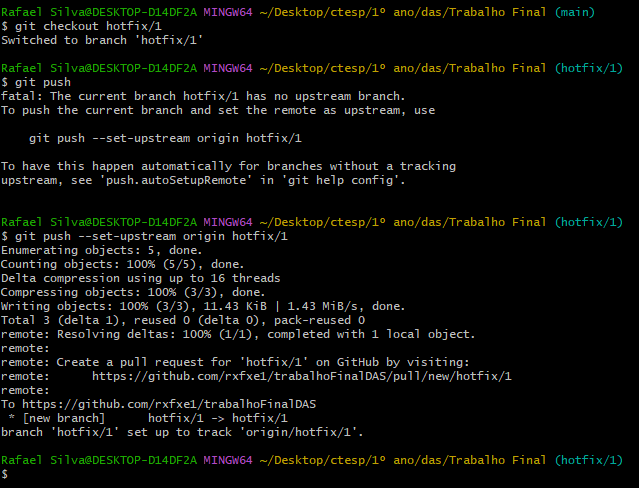
Depois fazemos as alterações e commits necessários e utilizamos os comandos $ git add e $ git commit.

Quando as alterações estiverem prontas, finalizamos o hotfix com o comando

$ git flow hotfix finish 1

Isso automaticamente fará o merge das alterações do branch hotfix/1 para o branch master e o branch develop, além disso, irá criar uma tag com o nome do hotfix.

A imagem descritiva destes passos encontra-se na página seguinte.



**FIM**