

Trabalho Prático Individual Final

DAS

Gonçalo Pereira

Nº2022051 1ºDS

DAS

André Carvalhas

Índice

<i>Repositório</i>	3
<i>Criação Repositório -</i>	3
<i>Criação Branches Gitflow -</i>	4
<i>Criação Regras</i>	4
<i>Criação Ficheiro GitIgnore</i>	5
<i>Link para Repositório</i>	6

Repositório

Criação Repositório -

Para a criação do meu repositório utilizei diretamente o GitHub para a formação do mesmo. Para o conseguir utilizar realizei um clone do repositório diretamente para o meu computador com o prompt de comando.

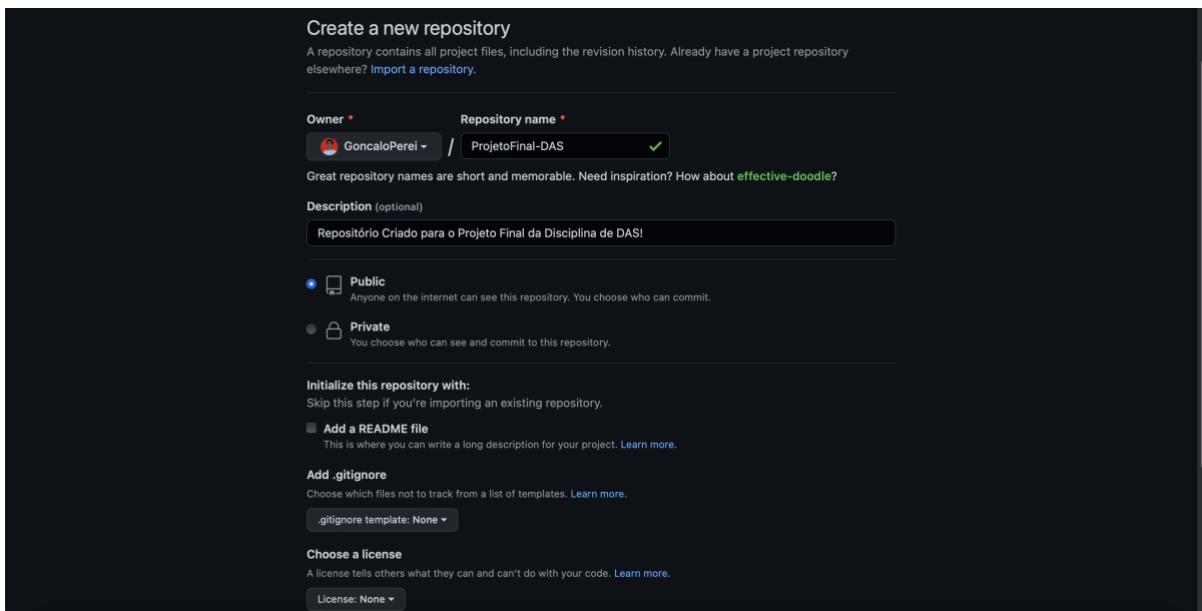


Figura 1 - Criação Repositório no GitHub

A screenshot of a terminal window titled 'GitHub — zsh — 80x24'. The window shows a command-line session. The user has logged in and navigated to their home directory. They then run the command 'git clone https://github.com/GoncaloPerei/ProjetoFinal-DAS.git'. The terminal displays the output of the cloning process, including object enumeration, counting, compressing, and receiving objects, until it finally completes with a success message.

Figura 2 - Clone do Repositório



Figura 3 - Repositório Local

Criação Branches Gitflow -

Para a criação do modelo Gitflow utilizei o comando “Git Flow Init”. Defini todos os valores das branches como “default”.

```
Projetofinal-DAS -- zsh -- 80x24
Last login: Tue Jan 24 15:33:03 on ttys000
goncalopereira@Goncalos-MacBook-Air ~ % cd /Users/goncalopereira/Documents/GitHub
goncalopereira@Goncalos-MacBook-Air GitHub % cd ProjetoFinal-DAS
goncalopereira@Goncalos-MacBook-Air ProjetoFinal-DAS % git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
goncalopereira@Goncalos-MacBook-Air ProjetoFinal-DAS %
```

Figura 4 - Criação Modelo Git Flow

```
[goncalopereira@Goncalos-MacBook-Air ProjetoFinal-DAS % git branch -a
* develop
  main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
```

Criação Regras

Para permitir a developers a submissão de código, mas nunca a alteração de visibilidade do repositório criei o meu repositório dentro de uma organização e mudei os privilégios de membros.

The screenshot shows the 'Integration access requests' section with the 'Allow integration requests from outside collaborators' option checked. It also shows the 'Admin repository permissions' section with the 'Repository visibility change' option checked. Both sections have a 'Save' button at the bottom.

Para tornar obrigatória a revisão de código antes de um pull request ser aprovado mudei as regras das branches dentro do meu repositório.

The screenshot shows the 'Merge rules' section with the 'Require conversation resolution before merging' option checked. A note below states: 'When enabled, all conversations on code must be resolved before a pull request can be merged into a branch that matches this rule.' There is a 'Learn more' link.

Criação Ficheiro GitIgnore

Para a criação do ficheiro gitignore utilizei o comando “touch ~/.gitignore” e de seguida editei o ficheiro colocando para serem ignorados os ficheiros “.docx” e “.doc”.

```
goncalopereira@Goncalos-MacBook-Air ProjetoFinal-DAS % touch ~/.gitignore
```

```
.DS_Store
#Ignorar Ficheiros Docx
*.docx

#Ignorar Ficheiros Doc
*.doc
~
```

Link para Repositório

<https://github.com/GoncaloPerei/ProjetoFinal-DAS>