Trabalho Prático Individual Final - Git

Desenvolvimento Ágil de Software

Desenvolvimento de Software

Pedro Marques

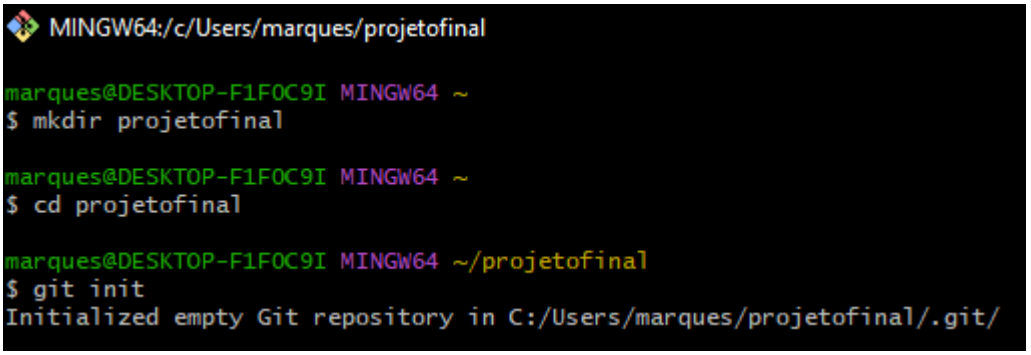
ISTEC Instituto Superior de Tecnologias Avançadas do Porto

Índice

Trabalho Prático Individual Final - Git

Introdução

Devemos começar por criar um repositório local com os seguintes comandos:



Depois de criado o repositório local devemos acessar o GitHub e criar o nosso repositório em: **Your repositories >>** 

Assim que concluído, para ligar o nosso repositório local ao nosso repositório do GitHub devemos digitar o seguinte comando:

1. Criação das Branches para ser possível a utilização do modelo GitFlow.

Para criar todas as branches necessárias e configurar o modelo gitflow devemos começar por digitar o seguinte comando:

```
marques@DESKTOP-F1F0C9I MINGW64 ~/projetofinal (master)
$ git flow init
No branches exist yet. Base branches must be created now.
Branch name for production releases: [master]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/marques/projetofinal/.git/hooks]
```

Quando digitado o comando é recomendado pressionar **ENTER** até ao fim da configuração. Após o procedimento, estaremos por diante de 2 branches: Master e Develop.

2. Níveis de acesso no GitHub

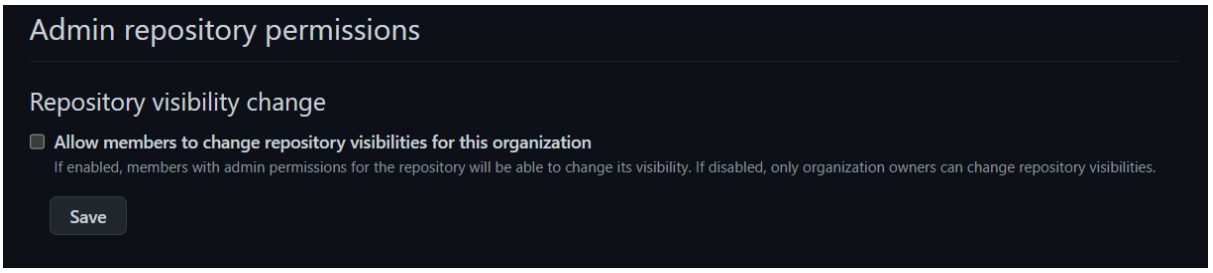
Para criar níveis de acesso no GitHub, teremos de criar uma organização e associar o nosso repositório à organização.

Para criar uma organização devemos acessar
Your organizations >> New Organization

Quando criada devemos transferir o nosso repositório para a nova organização
Your Repositories >> (O repositório) >> Settings >> Transfer ownership

Pronto! Agora é só acessar a nossa nova organização e ir a:
Settings >> Member Privileges

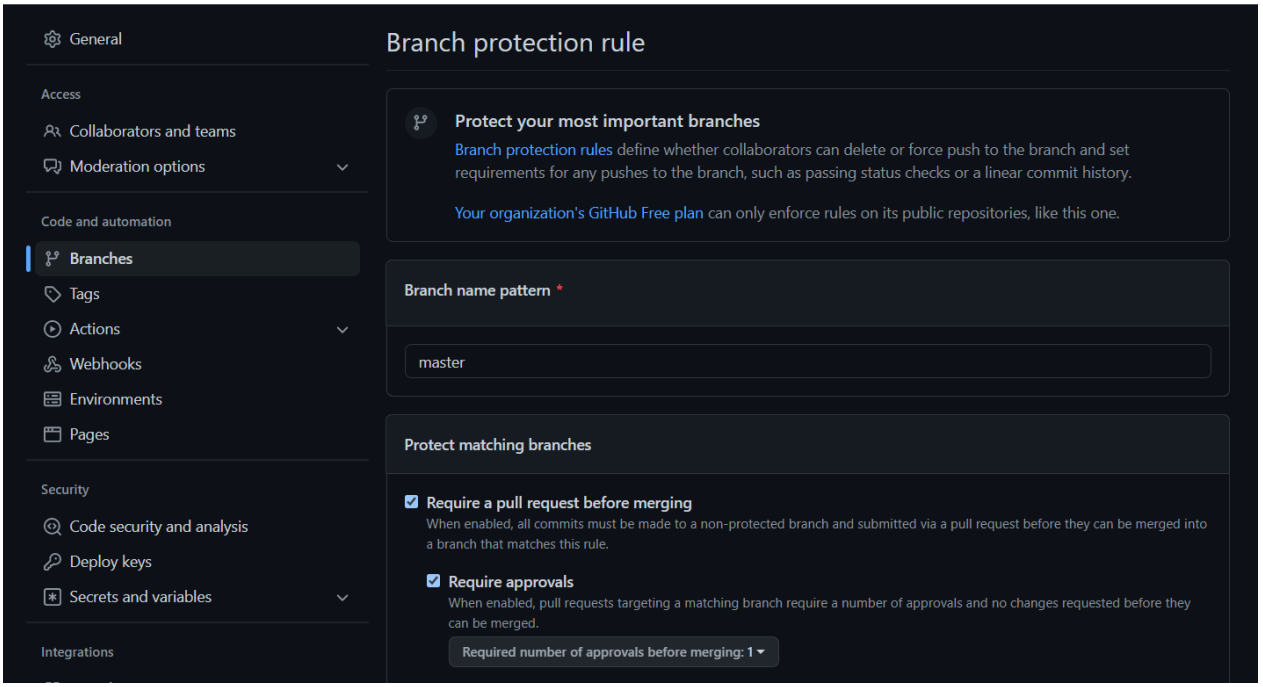
E dar untick da seguinte opção:



3. Revisão de código antes de um pull request

Para tornarmos obrigatória a revisão de código antes de um pull request é necessário acessarmos:
Your Organization >> (A organização que contém o repositório) >> Repositories >> (O repositório) >> Settings >> Branches >> Add rule

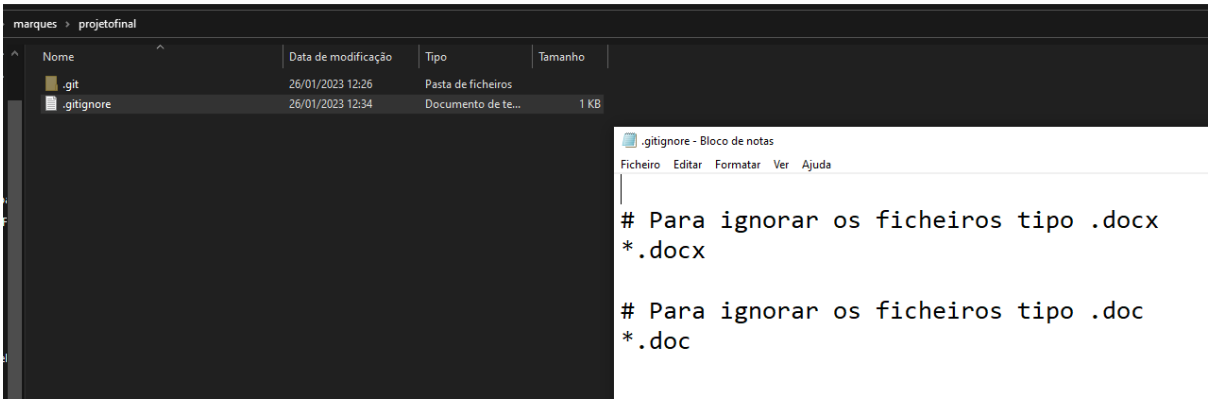
Para aplicar a regra a todas as branches é só aplicá-la na branch master visto que todas as branches provêm desta mesma. Devemos dar tick à regra Require a pull request before merging e Require approvals (escolhendo quantos aprovals queremos na revisão de código para que o pull request seja aprovado)



4. Adicionar GitIgnore

Para adicionar um ficheiro .gitignore começamos por criá-lo no repositório local e abrir como documento de texto para poder escrever quais ficheiros deseja ignorar.

Para ignorar ficheiros .docx e .doc devemos escrever o seguinte no ficheiro texto:



Assim feito, vamos agora proceder ao commit e dar push para a branch master

Para isso escrevemos estas linhas de comando:

```
marques@DESKTOP-F1FOC9I MINGW64 ~/projetofinal (master)
$ git add .gitignore

marques@DESKTOP-F1FOC9I MINGW64 ~/projetofinal (master)
$ git commit -m ".gitignore adicionado para ignorar ficheiros .docx e .doc"
[master 9850145] .gitignore adicionado para ignorar ficheiros .docx e .doc
 1 file changed, 6 insertions(+)
 create mode 100644 .gitignore
```

```
marques@DESKTOP-F1FOC9I MINGW64 ~/projetofinal (master)
$ git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 463 bytes | 463.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/pedrogpmarques/projetofinal
 * [new branch]      master -> master
```