



# ISTEC - Instituto Superior de Tecnologias Avançadas do Porto

## Projeto Final

Leonardo Pires  
Gonçalo Ferreira

**Curso de Desenvolvimento de Software**  
**Desenvolvimento Ágil de Software**

# Índice

# Introdução

Este relatório documenta a configuração do modelo GitFlow para a gestão de branches em um repositório Git, garantindo um fluxo de trabalho estruturado. É descrito o código utilizado a cada passo. Além disso, com este projeto, conseguimos simular um ambiente de desenvolvimento, contribuindo para o projeto com “hotfixes” ou até mesmo “features” e “releases”.

## Código Git Utilizado para a Criação do Repositório

Neste projeto, a criação do repositório GitHub foi realizada manualmente, e utilizou-se o Git com os seus comandos na linha de comando **Git Bash**. Abaixo encontram-se os comandos utilizados para a criação do repositório e os seguintes passos:

Inicialização do repositório local:

```
git init
```

Depois, para cada um dos integrantes do grupo, utilizou-se o seguinte comando para a ligação com o repositório remoto:

```
git remote add origin https://github.com/leonardojpgires/projeto-final-das
```

Criação do ficheiro .gitignore

```
echo "*.docx" >> .gitignore
```

```
echo "*.doc" >> .gitignore
```

Commit dos ficheiros

```
git add .gitignore
```

```
git commit -m "Adicionado .gitignore para ignorar arquivos .docx e .doc"
```

```
git push origin master
```

Inicialização do Git Flow

```
git flow init
```

Para ter acesso aos ficheiros colocados no repositório:

```
git pull origin develop
```

# Link de Acesso ao Repositório

<https://github.com/leonardojpaires/projeto-final-das>

## Operações efetuadas

Neste ponto, vamos demonstrar as operações que foram realizadas conforme os requisitos do projeto, detalhando cada uma delas:

### Carregamento inicial do ficheiro na branch develop

1. Mudança para a branch develop:  
**git checkout develop**
2. Adição do ficheiro ao repositório local:  
**git add .**
3. Commit da primeira versão do relatório:  
**git commit -m "Primeiro commit do relatório"**
4. Push para o branch develop:  
**git push origin develop**

### Alterações na branch develop com os devidos comentários

1. Edição do conteúdo do relatório  
**git commit -m "Adição da introdução do relatório"**

E assim sucessivamente para as respectivas alterações no relatório

### Release que contribui para o branch master

Após a alterações na branch develop, o código foi "fundido" no branch, criando uma versão final do relatório:

1. Primeiro, o branch master foi atualizado com a versão final:  
**git checkout master**
2. Merging do branch develop no master:  
**git merge develop**

3. Push da versão final para o repositório remoto:  
**git push origin master**

## Hotfix efetuado com a gestão de branches GitFlow

Um hotfix foi feito para corrigir um erro no relatório final após a revisão final:

1. Criação de uma nova branch de hotfix a partir do master:  
**git checkout -b hotfix/ajuste-relatorio master**
2. Correção do erro no relatório (texto justificado):  
**(edição do ficheiro Relatório Projeto Final DAS.pdf)**
3. Commit da correção:  
**git commit -m "Correção da formatação de texto do relatório"**
4. Push do hotfix para o repositório remoto:  
**git push origin hotfix/ajuste-relatorio**
5. Merging do hotfix de volta para o develop e master:  
**git checkout develop**  
**git merge hotfix/ajuste-relatorio**  
**git push origin develop**

E também para o master:

**git checkout master**  
**git merge hotfix/ajuste-relatorio**  
**git push origin master**