Instituição de Ensino

ISTEC - Instituto Superior de Tecnologias Avançadas do Porto

Título do Projeto: Trabalho Final

Data de Entrega: 08/03/2025

Disciplina: Desenvolvimento Ágil de Software

Alunos: Hugo Oliveira e Jorge Fernandes

Emails: hugo.oliveira.2024171@my.istec.pt

jorge.fernandes.2024063@my.istec.pt

Índice

Link do Repositório do GitHub

https://github.com/Cruxzit/Trabalho_final_das

Código Utilizado para Criar o Repositório

Primeiro, criamos o repositório no GitHub manualmente e nomeamos como trabalho_final_das. Em seguida, criamos a pasta localmente, também de forma manual.

- git init - Inicializa um novo repositório Git dentro da pasta do projeto.
- git remote add origin https://github.com/Cruxzit/Trabalho_final_das - Adiciona um repositório remoto ao projeto local.
- git flow init - Inicializa o uso do GitFlow, que é um modelo de branching para gerir o desenvolvimento do projeto.
- git flow feature start “nome_da_feature” - Cria uma nova branch para desenvolver uma funcionalidade específica.
- git add “ficheiro” - Adiciona o ficheiro à staging area ou seja, prepara-os para serem “committed”.
- git commit -m “Comentário do desenvoldedor” - Cria um "commit", ou seja, guarda uma versão específica das alterações que fizeste.
- git flow feature finish “nome_da_feature” - Termina a funcionalidade e faz o "merge" da branch "feature/nome_da_feature" na branch develop.
- git push origin develop - Envia da branch develop para o repositório remoto, onde ficam guardadas as tuas alterações.
- git flow release start “v1.0” - Cria uma nova branch de release para preparar a versão do software.
- git add “ficheiro” - Adiciona o ficheiro à staging area ou seja, prepara-os para serem “committed”.

- git commit -m “Comentário do desenvoldedor” - Cria um "commit", ou seja, guarda uma versão específica das alterações que fizeste.
- git flow release finish “v1.0” - Termina a branch de release, realiza o merge com master e develop, e marca a versão com um tag.
- git flow hotfix start “v1.0” - Cria uma nova branch de hotfix a partir de master para corrigir um bug crítico.
- git add “ficheiro” - Adiciona o ficheiro à staging area ou seja, prepara-os para serem “committed”.
- git commit -m “Comentário do desenvoldedor” - Cria um "commit", ou seja, guarda uma versão específica das alterações que fizeste.
- git flow hotfix finish “v1.0” - Termina a branch de hotfix, realiza o merge com master e develop, e marca a versão com um tag.

Conclusão

Este projeto permitiu aplicar na prática o Git e o modelo GitFlow para gestão de branches, garantindo organização e controle de versões. Foram definidos níveis de acesso, revisão obrigatória de código e um .gitignore para manter o repositório limpo. O relatório passou por diversas atualizações no develop, uma release para o master e um hotfix, demonstrando a correta utilização do GitFlow. Com a versão final armazenada no master, o projeto reforçou boas práticas e colaboração em desenvolvimento de software.