

Projecto Final - Git em Equipa

Curso SETIC | Jupiter Desenvolvimento Informático

Contexto

A direcção do SETIC solicitou melhorias urgentes no Portal de Serviços. A vossa equipa de 3 programadores foi designada para implementar as alterações. Cada membro ficará responsável por uma área específica, mas como todos vão trabalhar nos mesmos ficheiros simultaneamente, será necessário coordenar o trabalho usando Git Flow e resolver os conflitos que irão surgir.

Este exercício simula um cenário real de desenvolvimento colaborativo onde os conflitos são inevitáveis e fazem parte do processo.

Objectivos de Aprendizagem

- Aplicar o workflow Git Flow num projecto real
- Trabalhar em equipa com branches paralelos
- Resolver conflitos de merge de forma colaborativa
- Praticar boas práticas de commits e comunicação

Regras do Exercício

- Utilizar Git Flow: branches **main**, **develop** e **feature/***
- Mensagens de commit em português e no modo imperativo
- Fazer commits pequenos e frequentes (mínimo 3 commits por pessoa)
- Comunicar com a equipa antes de fazer merge
- Resolver conflitos em conjunto, não sozinho

Estrutura do Projecto

O repositório contém um site funcional com a seguinte estrutura:

portal-setic/

```
├── index.html    (página principal)
├── css/styles.css (estilos)
├── js/main.js    (JavaScript)
└── README.md    (documentação)
```

Fase 1 – Setup Inicial (20 minutos)

Executar em equipa (um membro lidera, outros observam):

1. Um membro faz fork do repositório para a sua conta GitHub
2. Adicionar os outros 2 membros como colaboradores (Settings > Collaborators)
3. Todos clonam o repositório para as suas máquinas
4. Criar o branch **develop** a partir do **main**
5. Enviar o branch **develop** para o GitHub
6. Cada membro faz pull e cria o seu feature branch a partir de **develop**

Nomenclatura dos branches: feature/navegacao-ux, feature/identidade-visual, feature/conteudo-estrutura

Fase 2 – Desenvolvimento Paralelo (1 hora)

Cada membro trabalha no seu branch de forma independente. **Não fazer merge ainda!**

Membro A – Missão: Navegação e Experiência do Utilizador

Branch: feature/navegacao-ux

Objectivo: Melhorar a navegação e interactividade do site

Tarefas:

1. Redesenhar o menu de navegação no header (pode adicionar ícones, mudar disposição)
2. Melhorar o menu mobile (animações, transições mais suaves)
3. Adicionar efeito de destaque no item de menu activo
4. Implementar navegação suave (smooth scroll) ao clicar nos links
5. Melhorar o efeito do header ao fazer scroll

Ficheiros a alterar: index.html (header/nav), styles.css (estilos de navegação), main.js (interacções)

Membro B – Missão: Identidade Visual

Branch: feature/identidade-visual

Objectivo: Modernizar o visual e criar impacto

Tarefas:

1. Criar nova paleta de cores para o site (alterar as cores principais)
2. Redesenhar a secção Hero com mais impacto visual
3. Melhorar o estilo do header (fundo, sombras, espaçamentos)
4. Redesenhar os botões (cores, hover effects, bordas)
5. Melhorar a tipografia (tamanhos, pesos, espaçamentos)

Ficheiros a alterar: index.html (hero, estrutura), styles.css (cores, tipografia, botões)

Membro C – Missão: Conteúdo e Estrutura

Branch: feature/conteudo-estrutura

Objectivo: Enriquecer o conteúdo e melhorar a estrutura

Tarefas:

1. Adicionar novos itens ao menu de navegação
2. Redesenhar completamente o footer (mais informações, redes sociais, mapa do site)
3. Melhorar o formulário de contacto (novos campos, melhor layout)
4. Adicionar validações mais completas no JavaScript
5. Implementar funcionalidade de scroll to top

Ficheiros a alterar: index.html (nav, footer, form), styles.css (footer, form), main.js (validações, scroll)

Mapa de Conflitos Esperados

A tabela seguinte mostra onde os conflitos vão ocorrer:

Área	Membro A	Membro B	Membro C
Header/Nav	Sim	Sim	Sim
Cores CSS	Sim	Sim	-
Footer	-	-	Sim
main.js	Sim	-	Sim

Nota: Os conflitos são intencionais e fazem parte do exercício!

Fase 3 – Integração e Resolução de Conflitos (1 hora)

Esta fase deve ser feita em conjunto, com todos a observar o ecrã de quem está a resolver o conflito.

1. **Primeiro Merge (Membro A):** Fazer merge de feature/navegacao-ux em develop - Este deve passar sem conflitos
2. Enviar develop actualizado para o GitHub
3. **Segundo Merge (Membro B):** Fazer pull de develop, depois merge de feature/identidade-visual - **CONFLITOS ESPERADOS!**
4. Resolver os conflitos em equipa, decidindo juntos o que manter
5. Fazer commit da resolução e push
6. **Terceiro Merge (Membro C):** Fazer pull de develop, depois merge de feature/conteudo-estrutura - **MAIS CONFLITOS!**
7. Resolver os conflitos em equipa
8. Fazer commit da resolução e push

Dica: Abram o ficheiro em conflito, identifiquem os marcadores <<<<<<, =====, >>>>>>, discutam qual versão (ou combinação) faz mais sentido, editem o ficheiro e removam os marcadores.

Fase 4 – Release Final (20 minutos)

- Criar branch release/v2.0 a partir de develop
- Actualizar o README.md com a versão e lista de alterações
- Testar o site e corrigir pequenos problemas se necessário
- Fazer merge em main com --no-ff
- Criar tag v2.0
- Fazer merge de release/v2.0 também em develop
- Enviar tudo para o GitHub (branches e tags)

Entregáveis

Ao finalizar, a equipa deve ter:

- Repositório GitHub com branches **main** e **develop** actualizados
- Histórico mostrando os merges e resoluções de conflitos
- Tag **v2.0** criada
- Site funcional com todas as melhorias integradas
- README.md actualizado com changelog

Dica Final: Usem **git log --oneline --all --graph** para visualizar o histórico e confirmar que o workflow foi seguido correctamente. O resultado final deve mostrar claramente os feature branches, os merges e a estrutura do Git Flow.