

Boas Práticas e Fluxo de Trabalho no GitHub

Projeto -> Gestor de Ginásio (nineTeam)

Este documento serve como orientação de boas práticas, descrevendo o fluxo de trabalho adotado pela equipa **nineTeam** para o desenvolvimento do projeto **Gestor de Ginásio**. Baseia-se na arquitetura MVC (Krasner & Pope) e utiliza as bibliotecas **JSON.NET** e **PDFsharp**.

➤ Estrutura Organizacional

- **Organização GitHub:** nineTeam
- **Repositório Principal:** GestorDeGinasio

Equipas

- **Coordenação:** Pedro (Líder)
- **Desenvolvimento:** César, Daniel, Cláudia
- **Verificação e QA:** Eugénio

➤ Estrutura do Repositório

GestorDeGinasio/

— Model/	# Classes de dados, lógica de negócio, acesso JSON
— View/	# Interfaces, layouts e visualização de dados
— Controller/	# Controladores que ligam View ao Model
— Services/	# Serviços como PDF, autenticação, utilitários
— Data/	# Ficheiros .json com dados persistentes
— Tests/	# Testes de unidade e scripts de QA
— Docs/	# Documentação do projeto (inclui este ficheiro)
— .gitignore	# Exclusão de ficheiros não rastreados (bin, obj, etc.)
— README.md	# Descrição geral do projeto
— GestorDeGinasio.sln	# Solução do Visual Studio

➤ Fluxo de Branches

- *main*: Versão estável e final
- *dev*: Integração de todas as funcionalidades testadas
- *feature/nome*: Nova funcionalidade (ex: feature/login-view)
- *bugfix/nome*: Correção de erros (ex: bugfix/login-error)

➤ Passos para Desenvolvedores

1. **Criar nova branch** a partir de dev:

```
git checkout dev
git pull
git checkout -b feature/nome-da-funcionalidade
```

2. **Desenvolver funcionalidade** na pasta apropriada (ex: Model/, View/)
3. **Testar localmente** no Visual Studio
4. **Fazer commit e push:**

```
git add .
git commit -m "Implementa funcionalidade X"
git push origin feature/nome-da-funcionalidade
```

5. **Criar Pull Request para dev** no GitHub

➤ Tarefas do Verificador

- Rever Pull Requests
- Executar testes em Tests/
- Aprovar ou solicitar correções
- Garantir que dev é estável antes do merge com main

➤ Tarefas do Líder

- Acompanhar o progresso no Discord e GitHub Projects
- Gerir conflitos e organizar reuniões semanais
- Validar merge final de dev para main:

```
git checkout main  
git pull  
git merge dev  
git push origin main
```

➤ Ferramentas Utilizadas

- **Visual Studio:** Desenvolvimento C#
- **Git + GitHub:** Versionamento
- **GitHub Projects:** Gestão de tarefas
- **Discord:** Comunicação
- **PlantUML:** Diagramas
- **JSON.NET / PDFsharp:** Manipulação de dados e relatórios

➤ Regras e Boas Práticas

- Não trabalhar diretamente em main ou dev
- Cada funcionalidade deve ter a sua branch
- Commits claros e frequentes
- Atualizar os READMEs das pastas
- Seguir a arquitetura MVC definida

Esta orientação deve ser seguida por todos os membros da equipa, de forma a garantir coesão e organização durante o desenvolvimento. Em caso de dúvida, contacte o líder ou abra uma discussão no Discord.