

Gerência de configuração: E-Personal

1) Qual é o repositório do seu grupo?

<https://github.com/PedroSilvv/gym-app>

2) Como está estruturado o seu repositório?

O repositório do projeto "gym-app" está estruturado da seguinte forma:

- **TEMPLATES:** Diretório contendo templates HTML ou outros arquivos usados para renderização.
- **academia:** Diretório relacionado a funcionalidades ou modelos específicos da academia.
- **env:** Diretório do ambiente virtual Python que gerencia as dependências do projeto.
- **scripts:** Contém scripts auxiliares para automação, configuração ou outras tarefas do projeto.
- **treino:** Diretório com funcionalidades ou dados específicos relacionados a treinos.
- **user:** Diretório relacionado ao gerenciamento de usuários ou funcionalidades específicas de usuários.
- **Outros Arquivos:**
 - `.gitattributes`: Arquivo de configuração do Git.
 - `db.sqlite3`: Banco de dados.
 - `manage.py`: Script principal para gerenciamento do projeto Django.

1) Quais artefatos do projeto (código-fonte, documentação, scripts de build, etc.) serão os itens de configuração?

Código-Fonte: Arquivos nos diretórios academia, treino, user, entre outros que contenham código Python.

Scripts de Build e Configuração: Arquivos dentro do diretório scripts, o ambiente virtual env, e o arquivo `manage.py`.

Banco de Dados: O arquivo `db.sqlite`.

2)

a) Quando abrir uma Branch (por tarefa ou por desenvolvedor, ou outro critério)?

As branches são abertas por desenvolvedor. Cada membro possui uma branch individual, permitindo que o trabalho seja isolado até estar pronto para ser integrado.

b) A nomenclatura dos branches?

A nomenclatura utilizada é `branchNomeAluno`. Por exemplo, `branchPedro`, `branchGabryel`, `branchBeatriz`, etc.

c) Todos os branches devem ser baseados na (principal)?

Sim, todas as branches são baseadas na principal(main).

3)

a) Quando realiza, e quem realiza (quantos pessoas realizam?)

O merge é realizado quando as alterações em uma branch individual estão concluídas e testadas, prontas para serem integradas à branch principal (main), normalmente ao final de uma tarefa ou funcionalidade específica. É feito por cada um em sua própria branch individual. Normalmente algum outro membro do grupo revisa o código antes do merge.

b) Como tratam os Conflitos de merge?

Quando ocorre um conflito de merge, o responsável pela branch avisa os demais da equipe, a fim de decidir a melhor opção. Isso envolve analisar o código conflitante e decidir qual versão deve ser mantida. Após a resolução, realiza um novo commit e tenta o merge novamente.

4) Inserem alguma mensagem no commit para descrever a modificação e referenciando a tarefa? Qual é o padrão?

Sim, informamos a alteração/adição feita. O padrão que segue é normalmente o verbo (adicionar, alterar, remover) e o que modificado (novo arquivo, função, etc).

5)

a) As versões do software seguirão o padrão de versionamento semântico?

Cada membro cria uma branch em alguma alteração que for necessária, e após faz um pull request para main, onde o Pedro (criador original do repositório) fica encarregado de aceitar.

b) Quando é lançada uma release? Quem aprova a release?

Uma release é lançada quando um conjunto de funcionalidades ou correções está completo e testado. A aprovação é feita por consenso do grupo.

c) Como você identifica os elemento release?

As releases são identificadas pelo número de versão semântica.