

1. Gerenciamento de configuração

1.1 Identificação dos itens de configuração

Projeto Django	Arquivo principal de configuração do backend Django	Código-fonte	aep_6s_2b/backend/api_root/settings.py	Django 5.1.2	Equipe de Backend
Dependências (Backend)	Arquivo listando dependências do backend Python	Dependências	aep_6s_2b/backend/requirements.txt	N/A	Equipe de Backend
Modelo de Machine Learning	Modelo treinado em formato serializado para previsões	Modelo Treinado	aep_6s_2b/backend/api_rest/ml/modelos/modelo_treinado.joblib	scikit-learn 1.5.2	Equipe de Backend
Script de Treinamento	Script para treinamento e atualização do modelo de Machine Learning	Código-fonte	aep_6s_2b/treinando_modelo	N/A	Equipe de Backend
Vetorizador do Modelo	Arquivo com vetor de pre-processamento utilizado no modelo	Modelo Treinado	aep_6s_2b/backend/api_rest/ml/vetorizer/vetorizer.joblib	N/A	Equipe de Backend
Projeto Flutter	Arquivo de configuração do projeto Flutter	Código-fonte	aep_6s_2b/frontend/pubspec.yaml	Flutter SDK	Equipe de Frontend
Dependência HTTP	Biblioteca utilizada para fazer requisições	Dependência	aep_6s_2b/frontend/pubspec.yaml	1.2.2	Equipe de Frontend

	HTTP ao backend				
Repositório Git	Repositório contendo o código-fonte do backend e frontend	Controle de Versão	https://github.com/kaikdorvalo/aep_6s_2b	N/A	DevOps
README.md	Documento contendo instruções para instalação e configuração	Documentação	aep_6s_2b/README.MD	N/A	Equipe de Desenvolvimento

1.2 Política de versionamento

Adotamos uma política de versionamento para que fosse possível manter a organização e funcionalidade do sistema. Adotamos um sistema com 3 branches, sendo a Master, Develop e Feature. A master contém o código pronto para produção. A develop é a branch de desenvolvimento, onde serão adicionadas as funcionalidades. A branch feature é onde será criada a funcionalidade.

1.2 Controle de mudanças

Definimos uma política para o controle de mudanças, onde a mudança deverá ser registrada em um quadro do trello, para então ser analisada pela equipe de Gerenciamento de Mudanças. Com base na análise, a equipe integra a mudança ao backlog ou marca como reprovada.

2. Escolha de Paradigma

Por ser um problema de classificação de senhas com comunicação client-server, optamos pelo paradigma orientado à objetos. A linguagem escolhida para as implementações são multi-paradigmas, mas a orientação à objetos aparece em maior parte.

3. Documentação das Manutenções

3.1.1. Redução do Peso do Modelo de Machine Learning

- Objetivo: Reduzir o peso do modelo de Machine Learning.
- Ação Realizada: Otimização e compactação do modelo de Machine Learning utilizado no backend, para reduzir seu tamanho e possibilitar o versionamento dentro do repositório Git.
- Justificativa: A versão anterior do modelo tinha um tamanho elevado.
- Benefícios: A redução do peso permite o versionamento do código no Git.

3.2.1. Aumento da Precisão do Modelo de Machine Learning

- Objetivo: Melhorar a precisão das predições feitas pelo modelo de Machine Learning.
- Ação Realizada: Ajustes foram aplicados ao modelo, incluindo refinamento dos dados.
- Justificativa: A versão anterior apresentava inconsistências e baixa precisão.
- Benefícios: Com a precisão aumentada, o modelo oferece resultados mais confiáveis.

3.2.2. Correção de Validações no Backend

- Objetivo: Corrigir falhas de validação que causavam erros no backend.
- Ação Realizada: Foram implementadas validações adicionais e ajustes de lógica para evitar erros.
- Justificativa: Alguns erros de validação comprometiam a integridade dos dados.
- Benefícios: As validações aprimoradas aumentam a robustez do backend.

3.3.1. Melhoria Estética do Aplicativo Flutter

- Objetivo: Melhorar a experiência visual e a usabilidade do aplicativo.
- Ação Realizada: Ajustes na interface do usuário foram aplicados, incluindo o design dos elementos visuais e o layout das telas.
- Justificativa: A versão anterior do aplicativo tinha uma interface básica.
- Benefícios: Com a estética aprimorada, o aplicativo passa a oferecer uma experiência visual mais agradável.

3.3.2. Adição de Informativos para Usuários

- Objetivo: Facilitar a utilização do aplicativo e esclarecer funcionalidades.
- Ação Realizada: Foram adicionados novos informativos e dicas no aplicativo, guiando o usuário em suas ações.

- Justificativa: A ausência de informativos dificultava a navegação e o entendimento das funcionalidades.
- Benefícios: Os informativos ajudam no aprendizado do usuário

4.1.1. Migração do Framework do Backend para Django

- Objetivo: Adotar um framework que permitisse a integração com o modelo de Machine Learning.
- Ação Realizada: O framework original foi substituído pelo Django, permitindo a integração nativa com o modelo de Machine Learning desenvolvido com Scikit-learn.
- Justificativa: O framework anterior não suportava o modelo de Machine Learning em python
- Benefícios: A migração para Django facilitou a integração com o modelo de Machine Learning.