

# **RELATÓRIO DO PROJETO INTEGRADOR**

## **Integrantes do grupo**

- Frederico Pessoa Barbosa
- Jorge Luiz Patrocínio dos Santos
- Rafael Victor Redoval de Sousa
- Yago Raphael de Melo Mouro

## **Resumo do projeto**

O projeto consiste no desenvolvimento de uma aplicação de análise e recomendação financeira baseada em inteligência artificial. O objetivo é auxiliar usuários a tomarem melhores decisões financeiras, oferecendo recomendações personalizadas a partir do perfil e histórico de cada um. A aplicação oferece um questionário ao usuário, para que ele descreva seu perfil como investidor, com isso, a plataforma consegue identificar os investimentos mais adequados ao seu grau de risco.

## **Tecnologias utilizadas**

Backend: Python, Flask, Scikit-learn, Pandas, Joblib, Flask-JWT-Extended, Flask-Bcrypt, Flask-CORS, Pymongo, yfinance, python-dotenv  
Front-end: Flutter (Dart)  
Banco de Dados: MongoDB Atlas (nuvem)  
Outras: Jupyter Notebook para análise e treinamento de modelos

## **Como o objetivo foi alcançado**

O objetivo foi atingido por meio da integração de um backend inteligente, responsável por processar dados e gerar recomendações, com um front-end responsivo, que proporciona uma experiência amigável ao usuário. O backend utiliza uma rede neural (MLP) treinada offline para analisar o perfil do usuário e sugerir ações financeiras. Os dados de perfil, histórico e as features dos ativos são armazenados em um banco de dados MongoDB Atlas na nuvem. O front-end, desenvolvido em Flutter, garante ótima usabilidade e eficiência para o usuário.

## **Fluxo principal:**

1. O usuário realiza o cadastro e responde um questionário de perfil (6 perguntas).
2. O backend armazena o perfil e utiliza o modelo de IA para gerar uma lista ranqueada de ações e probabilidades de compatibilidade.
3. O usuário recebe recomendações personalizadas na interface Flutter.

O treinamento do modelo é feito via notebook Jupyter (`supervised_train.ipynb`), que gera os artefatos necessários e popula o banco de dados com dados do Yahoo Finance.

⚠ Este produto é puramente educativo e não constitui recomendação de investimento.

## Instruções para instalação e execução

### Backend

1. Acesse a pasta `backend`.
2. (Opcional) Crie e ative um ambiente virtual:

```
python -m venv .venv
.venv\Scripts\activate # Windows
source .venv/bin/activate # Linux/Mac
```
3. Instale as dependências:

```
pip install -r requirements.txt
```
4. Crie um arquivo `.env` na raiz da pasta `backend` com as variáveis:

```
MONGODB_URI="mongodb+srv://<username>:<password>@<cluster-url>?retryWrites=true&w=majority"
JWT_SECRET_KEY="SUA_CHAVE_SECRETA_LONGA_E_ALEATORIA_AQUI"
PORT=8000
```
5. Certifique-se de que os artefatos de treino (`.joblib`, `.json`) estão na pasta `artifacts` (gerados pelo notebook `supervised_train.ipynb`).
6. Execute a API:

```
python -m api.app
```

A API rodará por padrão em <http://localhost:8000>

### Front-end

1. Acesse a pasta `front-end/finance_ia`.
2. Instale as dependências:

```
flutter pub get
```
3. Execute o app:

```
flutter run
```