

Situação Atual do Projeto e Arquitetura Tecnológica

1. Introdução

Este documento tem como finalidade apresentar a situação atual do projeto e registrar oficialmente as decisões relacionadas à arquitetura tecnológica e às ferramentas utilizadas, bem como as justificativas técnicas e estratégicas para cada escolha.

2. Situação Atual

O projeto encontra-se em fase de estruturação técnica e organização dos primeiros módulos funcionais. As definições iniciadas já foram consolidadas em relação às tecnologias que comporão o back-end, front-end, automação de coleta de dados (web scraping), interface visual (protótipos) e armazenamento de código-fonte.

3. Tecnologias Definidas

3.1. Back-end: Django (Python)

O back-end será desenvolvido utilizando o framework Django, construído sobre a linguagem Python.

Justificativas:

- Django é um framework robusto, maduro e com excelente documentação.
- Proporciona desenvolvimento rápido e seguro com recursos nativos como autenticação, ORM, administração e roteamento.
- A integração com bibliotecas Python voltadas a Machine Learning e Web Scraping será facilitada.

3.2. Web Scraper: Python

O módulo de coleta automatizada de dados será inteiramente desenvolvido em Python, por meio de bibliotecas como requests, BeautifulSoup, Selenium ou Scrapy.

Justificativas:

- Python é amplamente utilizado para automação e raspagem de dados na web.
- Há grande disponibilidade de bibliotecas e comunidade ativa.
- Integração direta com o backend em Django e com módulos futuros de análise.

3.3. Front-end: Kotlin

A camada de apresentação do aplicativo será desenvolvida em Kotlin, conforme orientação acadêmica da supervisão do projeto.

Justificativas:

- Kotlin é uma linguagem moderna, segura e oficialmente suportada para desenvolvimento Android.
- Atende aos requisitos estabelecidos pela orientação docente.
- Possibilita uma arquitetura multiplataforma futuramente, caso desejado.

3.4. Comunicação entre Back-end e Front-end

A comunicação entre o back-end desenvolvido em Django e o front-end em Kotlin será feita por meio de APIs RESTful, implementadas com Django REST Framework (DRF).

Justificativas:

- APIs RESTful oferecem interoperabilidade, independência de plataforma e leveza na troca de dados.
- O Django REST Framework proporciona uma forma estruturada, segura e eficiente de expor endpoints que serão consumidos pelo aplicativo em Kotlin.
- O app em Kotlin utilizará bibliotecas como Retrofit ou Ktor para realizar as requisições HTTP e consumir os dados JSON retornados pelas APIs.

Formato da comunicação:

- Formato dos dados: JSON

- Método de requisições: HTTP (GET, POST, PUT, DELETE)
- Autenticação prevista: Token JWT ou OAuth2 (a definir)
- Exemplo de fluxo:
- O usuário se autentica via app → o app envia um POST para /api/login/
- O back-end retorna um token de sessão
- O app usa esse token para consumir os dados com segurança via API

3.5. Interface e Design: Figma

O desenvolvimento dos protótipos visuais será realizado em Figma.

Justificativas:

- Ferramenta amplamente utilizada no mercado com foco em prototipagem rápida e colaborativa.
- Permite compartilhamento ágil entre equipe de design e desenvolvimento.
- Oferece bibliotecas de componentes reutilizáveis e suporte para design responsivo.

3.6. Repositório e Versionamento: GitHub

O controle de versão e armazenamento dos arquivos será centralizado no GitHub.

Justificativas:

- Permite colaboração entre membros da equipe com rastreamento de mudanças.
- Suporte nativo a integração contínua, pull requests e gerenciamento de issues.
- Amplamente aceito em projetos acadêmicos e profissionais.

4. Tecnologias Pendentes de Definição

4.1. Machine Learning / Inteligência Artificial

A escolha do modelo de machine learning ainda está em avaliação. O projeto prevê o uso de algoritmos de clusterização e processamento de linguagem natural (NLP), porém o modelo e a estratégia final de uso (local ou via API) serão definidos após testes exploratórios.

5. Conclusão

A definição das ferramentas adotadas busca garantir uma base sólida, escalável e com alto grau de interoperabilidade entre os módulos. A integração entre as tecnologias será cuidadosamente planejada, de forma a otimizar o desenvolvimento e garantir a qualidade da solução final.

Este documento poderá ser atualizado conforme novas decisões técnicas forem tomadas.