

Proposta de Software – Engenharia de Sistemas A

1. Identificação do Projeto

Nome do projeto: Ranking e Análise de Fundos Imobiliários

Nome de aluno: Lucas Sally

Nome de aluno (se houver): _____

Repositório GitHub: <https://github.com/LucasSLL/fiis-mvp.git>

Obs.: O professor deve ser adicionado como colaborador com acesso total.

2. Descrição Geral do Projeto

Resumo: (Breve descrição do software a ser desenvolvido, destacando o problema e o público-alvo.)

Aplicação web para comparar e visualizar FIIs (Fundos Imobiliários). O sistema monta um ranking a partir de métricas de preço (Yahoo Finance) e de fundamentos (Investidor10), e permite pesquisar um FII pelo código do ativo (ex.: KNCR11). Público-alvo: estudantes e investidores iniciantes que desejam uma visão rápida, educacional, para comparar FIIs sem precisar montar planilhas ou importar CSV.

Objetivo: (Descreva o propósito do sistema e os benefícios esperados.)

Entregar uma visão consolidada de cada FII

Oferecer um protótipo navegável que rode localmente com um clique (Streamlit).

Reduzir a fricção de dados (sem depender de CSV)

Escopo: (Liste as principais funcionalidades planejadas.)

- Construir um **ranking interativo de FIIs** com métricas de risco e retorno calculadas automaticamente.
- **Consolidar uma base de dados** dos principais FIIs (fundamentos, preços, dividendos e volume) em arquivo local e/ou MongoDB Atlas.

- Implementar uma **tela de análise detalhada do FII**, acessada por barra de pesquisa, com cartões de métricas e gráficos de preço, drawdown e dividendos.
- Disponibilizar uma **visão simples de carteira**, permitindo ao usuário selecionar seus FIIs e visualizar indicadores agregados (risco médio, retorno médio, quantidade de ativos).
- Documentar bibliotecas, fontes de dados e instruções de execução para que o MVP rode em qualquer máquina com Python.

3. Tecnologias e Ferramentas (Coloque somente categorias que forem relevantes ao seu projeto.)

Categoria	Ferramenta/Tecnologia	Justificativa
Linguagem de programação	Python 3.11+ -	Ecosistema forte para dados e prototipagem rápida.
Framework/Biblioteca	Streamlit -	Cria UI web.
	pandas, numpy -	Transformações e cálculos estatísticos.
	yfinance -	Histórico de preços e dividendos via Yahoo Finance.
	requests, requests-cache -	Coleta HTTP + cache local 24h.
	beautifulsoup4, requests-html, pyppeteer -	Coleta de dados de FIIs em páginas web (fundamentos / P/VP / segmento).
Banco de dados	Arquivo Parquet (ranking_seed.parquet) – Armazena o ranking consolidado para carregamento rápido. MongoDB Atlas – Armazenamento opcional de todos os FIIs com fundamentos e métricas.	
Controle de versão	GitHub -	Colaboração, histórico, issues e versionamento.
Outras ferramentas	VS Code / Anaconda -	Ambiente e gerenciamento de dependências.

4. Processo de Desenvolvimento (Scrum)

O desenvolvimento será conduzido utilizando o framework Scrum, com 5 Sprints, cada Sprint terá duração de 1 semana, encerrando com uma reunião de revisão e demonstração do resultado.

O professor fará o papel de *Product Owner* e de *Scrum Master*.

5. Planejamento dos Sprints (Coloque um planejamento que faça sentido ao seu projeto, é apenas uma estruturação básica. Abaixo segue um exemplo)

Sprint 1 – Definição do Projeto

Objetivo: Definir escopo, requisitos e plano de desenvolvimento.

Entregáveis:

- Proposta completa
- Criação do repositório GitHub
- Backlog inicial definido
- Esboço do sistema com modelos
- Cronograma de Sprints

Sprint 2 – Protótipo e Estrutura Inicial

Objetivo: Implementar a base do projeto e protótipo inicial.

Entregáveis:

- Protótipo navegável
- Setup do ambiente
- Features básicas implementadas

Desenvolvido:

1. Script de coleta e consolidação de dados dos FIIs (fundamentos + preços).
2. Armazenamento dos dados em base local (ranking_seed.parquet) e script de carga para o MongoDB Atlas.
3. Função build_ranking gerando um DataFrame com as principais colunas e métricas.

Tela inicial em Streamlit:

1. Ranking de FIIs em tabela, com ordenação por coluna.
2. Colunas padrão definidas (ativo, segmento, risco, índice de risco, retorno 12m, volatilidade, drawdown, cotação, P/VP).
3. requirements.txt inicial com as bibliotecas principais.

Sprint 3 – Funcionalidades Principais

Objetivo: Desenvolver funcionalidades centrais.

Entregáveis:

- Módulos principais implementados
- Integração parcial
- Testes unitários iniciais

Desenvolvido:

1. Criação da **barra de pesquisa de FII** por ticker, com validação de formato (ex.: MXRF11).
2. Integração com **Yahoo Finance** e com o site de fundamentos para buscar:
 1. histórico de preços,
 2. dividendos,
 3. volume médio 30 dias,
 4. P/VP e segmento.
3. Implementação do módulo de métricas (`metrics_from_prices`) para calcular:
 1. retorno 12 meses (preço + dividendos),
 2. volatilidade anual,
 3. drawdown máximo em 2 anos.
4. Tela “**Análise do FII**” em Streamlit com:
 1. cartões de métricas (índice de risco, retorno 12m, vol. anual, máx. DD 2a, P/VP, DY 12m, liquidez média 30d, segmento),
 2. gráficos de preço (R\$), preço normalizado e drawdown dos últimos 24 meses,
 3. gráfico de dividendos mensais dos últimos 24 meses.

Sprint 4 – Integração e Refinamento

Objetivo: Integrar módulos e aperfeiçoar interface.

Entregáveis:

- Sistema parcialmente funcional
- Ajustes de design
- Documentação técnica atualizada

Desenvolvido:

1. Criação da aba “**Minha carteira**” em Streamlit.
2. Componente de seleção múltipla com **todos os FIIs do ranking** para o usuário informar quais compõem sua carteira.
3. Cálculo e exibição de métricas agregadas da carteira:
 1. quantidade de FIIs,
 2. risco médio (índice de risco),
 3. retorno médio 12m.
4. Tabela com os FIIs da carteira usando as mesmas colunas do ranking.
5. Ajustes de layout geral:
 1. página dividida em duas colunas (ranking à esquerda, análise/carteira à direita),
 2. criação do módulo `layout.py` com estilo global e métricas em formato de cartão,
 3. abas superiores “Análise do FII / Minha carteira” no estilo *pill*.

Sprint 5 – Testes e Entrega Final

Objetivo: Realizar testes finais e preparar a entrega.

Entregáveis:

- Versão final funcional

- Manual do usuário
- Apresentação final
- Registro no GitHub

Desenvolvido:

1. Organização do código em módulos (core/app.py, pacote modulos, pasta data).
2. Refino visual do MVP:
 1. padronização de cores, tipografia e espaçamentos,
 2. estilo das abas em formato *pill* com destaque na aba ativa,
 3. métricas em cartões com sombra leve.
3. Otimizações de desempenho:
 1. cache HTTP com requests-cache (24h),
 2. uso do arquivo ranking_seed.parquet para inicializar o ranking de forma rápida.
4. Criação do arquivo **requirements.txt** com as dependências mínimas do projeto.
5. Criação do script **Executar.bat** e atualização do **README** com instruções para:
 1. criar/usar ambiente virtual,
 2. instalar as dependências,
 3. executar streamlit run Script/core/app.py com um único duplo-clique.

Cronograma Visual dos Sprints

	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5
Definição do Projeto					
Protótipo e Estrutura Inicial					
Funcionalidades Principais					
Integração e Refinamento					
Testes e Entrega Final					

6. Backlog do Produto (Apenas os épicos principais do projeto. Abaixo um exemplo.)

ID	História (épico)	Critério de aceitação (alto nível)	Prioridade
EP01	Como usuário, quero visualizar um ranking de FIIs com as principais métricas de comparação.	Tabela inicial com vários FIIs e colunas de métricas carregando sem erro.	Alta
EP02	Como usuário, quero pesquisar um FII e ver suas métricas detalhadas e gráficos de preço/risco.	Busca por ticker + tela de detalhes com cards e gráficos funcionando.	Alta
EP03	Como usuário, quero ver informações de dividendos dos FIIs, em formato de gráfico.	Gráfico de dividendos carregado para o FII selecionado.	Média
EP04	Como usuário, quero montar uma carteira de FIIs e ver um resumo das métricas dessa carteira.	Aba “Minha carteira” com seleção de ativos e métricas agregadas.	Média
EP05	Como usuário, quero que o sistema seja simples de executar em qualquer máquina (instalação mínima).	Projeto rodando localmente seguindo o README ou via Executar.bat.	Alta
EP06	Como usuário, gostaria de personalizar mais o sistema (favoritos, customização de colunas, gráficos extra).	Funcionalidades de personalização definidas, mesmo que parciais/futuras.	Baixa (futuro)