

Roadmap de Desenvolvimento do InvestControl

Data: 16 de Setembro de 2025

Autor: Manus AI

Este documento apresenta o roadmap de desenvolvimento completo e detalhado para o projeto InvestControl, servindo como um guia e roteiro para todas as etapas de implementação. Ele é baseado no Documento de Especificação de Requisitos de Software (SRS) e na Proposta de Stack Tecnológica, garantindo alinhamento com os objetivos do produto e a escolha de ferramentas robustas e eficientes.

O roadmap está organizado em fases sequenciais, priorizando o desenvolvimento do backend robusto e das funcionalidades essenciais de Aportes Sob Demanda, conforme a visão estratégica do projeto.

Visão Geral do Projeto

O InvestControl é uma plataforma de gestão de carteiras e recomendações de investimento focada em dividendos, com a funcionalidade central de **Aportes Sob Demanda**. O objetivo é capacitar investidores a tomar decisões informadas, oferecendo análises de dividend-yield, preço-teto (método Barsi) e calendário de proventos, tudo isso através de uma interface intuitiva e um backend de alta performance.

Stack Tecnológica Resumida

Categoria	Tecnologia Principal	Justificativa Chave
Backend	Python (FastAPI)	Performance, ecossistema de dados, documentação automática.
Banco de Dados	PostgreSQL (SQLAlchemy, Alembic)	Confiabilidade, integridade de dados financeiros, migrações controladas.
Frontend (Web)	Vue.js	Simplicidade, reatividade, experiência de usuário amigável.
Mobile (Opcional)	Flutter	Desenvolvimento multiplataforma, performance nativa.
Infraestrutura	Docker, Kubernetes (produção)	Portabilidade, escalabilidade, alta disponibilidade.

Observabilidade	Prometheus, Grafana, Loki	Monitoramento proativo, dashboards, logs estruturados.
CI/CD	GitHub Actions / GitLab CI/CD	Automação de build, teste e deploy.
Controle de Versão	Git (GitHub/GitLab)	Colaboração e rastreabilidade.

Fases de Desenvolvimento

Fase 0: Preparação e Configuração do Ambiente (1-2 semanas)

Objetivo: Estabelecer o ambiente de desenvolvimento, a estrutura do projeto e as ferramentas básicas para garantir um início de projeto sólido e reproduzível.

Atividades Chave:

- Configuração do ambiente de desenvolvimento local com Docker e Docker Compose.
- Inicialização do repositório Git (GitHub/GitLab) e configuração de branches (main, develop, feature branches).
- Estruturação inicial do projeto FastAPI (backend) e Vue.js (frontend).
- Configuração do PostgreSQL via Docker Compose.
- Configuração inicial do Alembic para gerenciamento de migrações de banco de dados.
- Configuração básica de observabilidade com Prometheus e Grafana via Docker Compose.
- Implementação de um pipeline de CI básico (testes unitários e linting) com GitHub Actions/GitLab CI/CD.

Tecnologias Envolvidas: Docker, Docker Compose, Git, FastAPI, PostgreSQL, Alembic, Prometheus, Grafana, GitHub Actions/GitLab CI/CD.

Entregáveis:

- Ambiente de desenvolvimento funcional e documentado.
- Estrutura de projeto base para backend e frontend.
- Banco de dados PostgreSQL configurado e acessível.
- Ferramentas de observabilidade operacionais.
- Pipeline de CI básico funcionando.

Fase 1: Backend - Lógica Central: Ativos e Transações (2-3 semanas)

Objetivo: Implementar a base do registro de ativos e transações, que são fundamentais para a gestão de carteiras.

Atividades Chave:

- Modelagem das tabelas de **Ativos** e **Transações** no PostgreSQL.
- Implementação dos schemas Pydantic para validação de dados de entrada e saída na API.
- Desenvolvimento das operações CRUD (Create, Read, Update, Delete) para **Ativos** e **Transações** usando SQLAlchemy.
- Criação dos endpoints da API RESTful no FastAPI para gerenciamento básico de ativos e transações.
- Escrita de testes unitários e de integração abrangentes para as funcionalidades implementadas.

Tecnologias Envolvidas: Python, FastAPI, PostgreSQL, SQLAlchemy, Alembic, Pydantic, Pytest.

Requisitos do SRS Atendidos: FR-02 (Gestão de Carteiras e Ativos), NFR-06 (Manutenibilidade).

Entregáveis:

- Modelos de dados para Ativos e Transações implementados no banco de dados.
- Endpoints CRUD para Ativos e Transações na API.
- Testes automatizados para as funcionalidades de ativos e transações.

Fase 2: Backend - Lógica Central: Carteiras e Cálculos Básicos (2-3 semanas)

Objetivo: Implementar o conceito de **Carteira** e os cálculos financeiros básicos associados, integrando-os com ativos e transações.

Atividades Chave:

- Modelagem da tabela de **Carteiras** e estabelecimento das relações com **Ativos** e **Transações**.
- Implementação da lógica para calcular o valor total da carteira, preço médio dos ativos e rentabilidade.

- Criação de endpoints da API no FastAPI para gerenciamento de carteiras e consulta dos cálculos básicos.
- Desenvolvimento de testes para validar a precisão dos cálculos e a integridade das relações entre carteiras, ativos e transações.

Tecnologias Envolvidas: Python, FastAPI, PostgreSQL, SQLAlchemy, Pydantic, Pytest.

Requisitos do SRS Atendidos: FR-02 (Gestão de Carteiras e Ativos), NFR-01 (Desempenho - base para cálculos rápidos).

Entregáveis:

- Modelo de dados para Carteiras implementado.
 - Endpoints para gerenciamento de carteiras e cálculos básicos.
 - Testes automatizados para cálculos financeiros.
-

Fase 3: Backend - Integração de Dados de Mercado (2-3 semanas)

Objetivo: Automatizar a coleta, armazenamento e atualização de dados de mercado (cotações, dividendos, proventos) de fontes externas.

Atividades Chave:

- Pesquisa e seleção de APIs de mercado ou fontes para web scraping de cotações, dividendos e proventos.
- Implementação de módulos de integração para coletar cotações em tempo real e histórico de dividendos/proventos.
- Desenvolvimento de um sistema de cache e atualização periódica para garantir a atualidade dos dados e otimizar o acesso (NFR-01).
- Criação de endpoints na API para consulta de dados de mercado, com mecanismos de rate-limit e cache.
- Implementação de validação e coerência temporal dos dados (NFR-08).

Tecnologias Envolvidas: Python (requests, bibliotecas de scraping/APIs), FastAPI, PostgreSQL, Celery (para tarefas assíncronas de atualização), Redis (para cache).

Requisitos do SRS Atendidos: FR-03 (Integração de Dados de Mercado), NFR-01 (Desempenho), NFR-08 (Qualidade de Dados).

Entregáveis:

- Módulos de integração com APIs de mercado funcionais.
- Sistema de cache e atualização de dados de mercado operante.
- Endpoints da API para consulta de dados de mercado.

Fase 4: Backend - Sistema de Aportes Sob Demanda (PRIORIDADE) (3-4 semanas)

Objetivo: Implementar a funcionalidade principal de recomendação de aportes sob demanda, com algoritmos e justificativas claras.

Atividades Chave:

- Desenvolvimento do algoritmo de recomendação que considera: valor disponível, preço atual, distância ao preço-teto (Barsi), DY atual e histórico, proximidade de datas com/ex, composição da carteira e quedas recentes (FR-01).
- Implementação do sistema de ranqueamento de oportunidades de compra.
- Criação de endpoints na API para receber o valor de aporte e retornar as recomendações ranqueadas com justificativas.
- Desenvolvimento de testes abrangentes para o sistema de recomendação, validando a lógica e a performance (NFR-01).
- Otimização do algoritmo para garantir o tempo de resposta de $\leq 2,0$ segundos (P95).

Tecnologias Envolvidas: Python, FastAPI, PostgreSQL, SQLAlchemy, bibliotecas de análise de dados (Pandas, NumPy).

Requisitos do SRS Atendidos: FR-01 (Aportes Sob Demanda), NFR-01 (Desempenho), NFR-08 (Qualidade de Dados - precisão dos cálculos).

Entregáveis:

- Algoritmo de Aportes Sob Demanda funcional e testado.
 - Endpoints da API para recomendações de aporte.
 - Relatórios de performance do algoritmo.
-

Fase 5: Backend - Recomendações Complementares e Calendário de Proventos (2-3 semanas)

Objetivo: Implementar algoritmos complementares de recomendação e a funcionalidade de calendário de proventos.

Atividades Chave:

- Desenvolvimento de algoritmos para cálculo de dividend yield atual e histórico, e comparações (FR-04).
- Implementação do algoritmo de cálculo de preço-teto (método Barsi) (FR-04).

- Criação de lógica para identificar oportunidades de compra (abaixo do teto, quedas, alta de DY, consistência histórica) (FR-04).
- Desenvolvimento da funcionalidade de calendário de proventos, coletando datas de pagamento e identificando datas com/ex (FR-05).
- Implementação de sugestões de janelas e projeções de renda com base nos proventos (FR-05).
- Criação de endpoints na API para acessar essas funcionalidades.

Tecnologias Envolvidas: Python, FastAPI, PostgreSQL, SQLAlchemy, Pandas.

Requisitos do SRS Atendidos: FR-04 (Motor de Regras Complementares), FR-05 (Calendário de Proventos).

Entregáveis:

- Algoritmos de DY, Preço-Teto e Oportunidades implementados.
 - Funcionalidade de Calendário de Proventos completa.
 - Endpoints da API para as novas funcionalidades.
-

Fase 6: Backend - Gestão de Usuários e Segurança (2-3 semanas)

Objetivo: Implementar o sistema de autenticação, autorização e gestão de perfis de usuário, garantindo a segurança da plataforma.

Atividades Chave:

- Implementação de autenticação de usuários utilizando JWT (JSON Web Tokens) (FR-09, NFR-05).
- Desenvolvimento de autorização e controle de acesso baseado em papéis (RBAC) para proteger endpoints (FR-09, NFR-05).
- Criação de funcionalidades para gerenciamento de perfis e preferências de usuários (FR-09).
- Implementação de criptografia em repouso e em trânsito para dados sensíveis (NFR-05).
- Configuração de logs de segurança e auditoria para monitorar acessos e tentativas inválidas (NFR-04, NFR-05).

Tecnologias Envolvidas: Python, FastAPI, JWT, PostgreSQL, SQLAlchemy, Pydantic.

Requisitos do SRS Atendidos: FR-09 (Gestão de Usuários e Segurança), NFR-05 (Segurança e Privacidade).

Entregáveis:

- Sistema de autenticação e autorização funcional.

- Gerenciamento de perfis de usuário.
 - Endpoints protegidos e seguros.
-

Fase 7: Frontend Web - Interface de Aportes Sob Demanda e Gestão Básica (3-4 semanas)

Objetivo: Desenvolver a interface web principal, focando na funcionalidade de Aportes Sob Demanda e na gestão básica de carteiras.

Atividades Chave:

- Criação da interface de usuário para entrada de valor de aporte e visualização das recomendações ranqueadas com justificativas (FR-01, FR-10).
- Desenvolvimento de componentes para exibir detalhes de ativos e proventos (FR-01, FR-05).
- Implementação da interface para CRUD básico de carteiras, ativos e transações (FR-02).
- Integração com a API do backend para todas as funcionalidades implementadas até a Fase 6.
- Garantia de uma experiência de usuário fluida e responsiva (NFR-01 - Usabilidade).

Tecnologias Envolvidas: Vue.js, HTML, CSS, JavaScript, Axios (para requisições HTTP).

Requisitos do SRS Atendidos: FR-01, FR-02, FR-05, FR-10, NFR-01 (Usabilidade).

Entregáveis:

- Aplicação web funcional para Aportes Sob Demanda e gestão básica de carteiras.
 - Interface de usuário intuitiva e responsiva.
-

Fase 8: Frontend Web - Relatórios, Alertas e Personalização (2-3 semanas)

Objetivo: Expandir a interface web com funcionalidades de acompanhamento, relatórios, alertas inteligentes e personalização.

Atividades Chave:

- Desenvolvimento de dashboards e relatórios para acompanhamento de performance vs. benchmarks e histórico (FR-07).
- Implementação da interface para configuração e gerenciamento de alertas inteligentes (FR-08).

- Criação de funcionalidades de personalização de recomendações e preferências de usuário (FR-06, FR-09).
- Refinamento da experiência do usuário e otimização de desempenho da interface.

Tecnologias Envolvidas: Vue.js, bibliotecas de gráficos (ex: Chart.js, ApexCharts), HTML, CSS, JavaScript.

Requisitos do SRS Atendidos: FR-06, FR-07, FR-08, FR-09, FR-10.

Entregáveis:

- Funcionalidades de relatórios e acompanhamento implementadas na web.
 - Sistema de gerenciamento de alertas na interface web.
 - Opções de personalização de usuário na web.
-

Fase 9: Mobile (Opcional) - MVP com Notificações (3-4 semanas)

Objetivo: Desenvolver um MVP (Produto Mínimo Viável) da aplicação mobile, com foco em notificações e acesso rápido às informações essenciais.

Atividades Chave:

- Desenvolvimento da aplicação mobile com Flutter, focando nas funcionalidades mais críticas (ex: visualização de recomendações de aporte, alertas) (FR-10).
- Implementação de notificações push para alertas inteligentes (FR-08, FR-10).
- Integração com a API do backend.
- Garantia de uma experiência de usuário nativa e de alta performance.

Tecnologias Envolvidas: Flutter, Dart.

Requisitos do SRS Atendidos: FR-08, FR-10.

Entregáveis:

- Aplicação mobile MVP para iOS e Android.
 - Funcionalidade de notificações push operante.
-

Fase 10: Implantação, Monitoramento e Otimização Contínua (Contínuo)

Objetivo: Garantir a implantação eficiente, monitoramento contínuo e otimização da plataforma em produção.

Atividades Chave:

- Configuração de ambiente de produção (servidores, Kubernetes, etc.).
- Implantação da aplicação backend e frontend em ambiente de produção.
- Configuração avançada de Prometheus e Grafana para monitoramento em tempo real.
- Implementação de solução de logging (Loki/ELK Stack) para logs estruturados e análise.
- Estabelecimento de alertas e dashboards para monitorar KPIs (Latência de recomendação, Uptime, Atualidade de dados, etc.) (NFR-01, NFR-03, NFR-04).
- Otimização contínua de performance e escalabilidade (NFR-01, NFR-02).
- Revisões de segurança e auditorias periódicas (NFR-05).

Tecnologias Envolvidas: Docker, Kubernetes, Prometheus, Grafana, Loki/ELK Stack, Ferramentas de CI/CD.

Requisitos do SRS Atendidos: NFR-01, NFR-02, NFR-03, NFR-04, NFR-05, NFR-07.

Entregáveis:

- Ambiente de produção configurado e operacional.
- Sistema de monitoramento e alertas ativo.
- Logs centralizados e acessíveis.
- Plano de recuperação de desastres documentado.

Considerações Finais

Este roadmap é um documento vivo e pode ser ajustado conforme novas informações, desafios ou prioridades surgirem. A comunicação contínua entre as equipes de Produto, Engenharia e Dados será fundamental para o sucesso do projeto. A priorização do backend e das funcionalidades de Aportes Sob Demanda nas fases iniciais visa entregar valor rapidamente e construir uma base sólida para futuras expansões.

Referências

- [1] Documento SRS Inicial - InvestControl (documento_srs_inicial.txt)
- [2] Roadmap Atualizado do InvestControl: Backend Robusto com Aportes Sob Demanda. Disponível em: https://chatgpt.com/s/dr_68c9b9e66a24819195e0d5d958995e81
- [3] Proposta de Stack Tecnológica para o InvestControl (Proposta_Tecnologica_InvestControl.md)