**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

**Pós-graduação *Lato Sensu* em Arquitetura de *Software* Distribuído**

Projeto Integrado

Relatório Técnico

Sistema Gerenciador de Investimento

de Renda Variável

Eduardo Sampaio Viana

Belo Horizonte

Março 2023

# Projeto Integrado – Arquitetura de Software Distribuído

**Sumário**

[Projeto Integrado – Arquitetura de Software Distribuído 2](#_Toc116844011)

[1. Introdução 3](#_Toc116844012)

[2. Especificação Arquitetural da solução 5](#_Toc116844013)

[2.1 Restrições Arquiteturais 5](#_Toc116844014)

[2.2 Requisitos Funcionais 5](#_Toc116844015)

[2.3 Requisitos Não-funcionais 6](#_Toc116844016)

[2.4 Mecanismos Arquiteturais 6](#_Toc116844017)

[3. Modelagem Arquitetural 7](#_Toc116844018)

[3.1 Diagrama de Contexto 7](#_Toc116844019)

[3.2 Diagrama de Container 8](#_Toc116844020)

[3.3 Diagrama de Componentes 8](#_Toc116844021)

[4. Avaliação da Arquitetura (ATAM) 9](#_Toc116844022)

[4.1 Análise das abordagens arquiteturais 10](#_Toc116844023)

[4.2 Cenários 10](#_Toc116844024)

[4.3 Evidências da Avaliação 11](#_Toc116844025)

[5. Avaliação Crítica dos Resultados 12](#_Toc116844026)

[6. Conclusão 13](#_Toc116844027)

[Referências 14](#_Toc116844028)

## Introdução

No Brasil os ativos de renda variável são negociados pela B3 que é uma bolsa de valores fundada em São Paulo em 1890. Atualmente o número de investidores na bolsa de valores tem aumentado cada vez mais. As pessoas estão migrando da poupança para investimento de renda variável e renda fixa. Em um cenário atual com uma taxa selic 13,75% ao ano, a renda fixa tornou-se uma excelente oportunidade de ganhos para o curto prazo. A taxa selic é a taxa básica de juros da economia brasileira e influencia as demais taxas de juros do Brasil. Em contrapartida os investimentos de renda variável apresentam também grandes oportunidades devido à taxa selic elevada fazendo com que as ações e fundos imobiliários fiquem bastantes descontados pois se encontram em um ciclo de baixa da bolsa de valores devido a elevação dos juros, porém mesmo descontados apresentam um maior risco e devem ser analisados e gerenciados com detalhes.

O mercado financeiro é muito vasto e a escolha de um ativo de renda variável necessita de muitas informações na escolha de um determinado ativo tornando uma tarefa difícil mesmo para profissionais experientes no mercado financeiro. Um bom investimento depende de boas decisões, errar na escolha de um determinado ativo pode levar o investidor a perder dinheiro. Temos hoje aproximadamente 400 empresas no qual podemos ser sócios através de ações com segmentos variados e 150 fundos imobiliários listados na bolsa de valores tornando a escolha um pouco difícil. O investidor está buscando comprar ações e fundos imobiliários por um bom preço e que paguem bons dividendos. Dividendos são proventos pagos aos acionistas por possuir um determinado ativo. Cada empresa paga seus dividendos em datas diferentes e com frequências diferentes podendo ser de forma anual, semestral, quadrimestral, trimestral ou mensal essas informações são importantes para o investidor ter uma previsibilidade. Um outro problema é a visualização de ativos escolhidos que muitos investidores recorrem a ferramentas como Excel ou outros sites extremamente complexos tornando difícil gerenciamento desses ativos através dessas ferramentas o investidor tenta visualizar a taxa de rentabilidade dos investimentos e sua evolução patrimonial.

O objetivo deste trabalho é permitir que os investidores de renda variável possam ter informações sobre ações e fundos imobiliários que ajudem na sua escolha para montagem e gerenciamento de sua carteira de investimentos e a visualização de pagamentos de proventos, rentabilidade da carteira de investimento e evolução patrimonial. Ações são títulos de renda variável que representam uma fração do capital social de uma empresa quando compramos uma ação nos tornamos sócios desta empresa e também dos seus resultados. Os fundos imobiliários são títulos de renda variável que reúnem seus recursos para que sejam aplicados em conjunto no mercado imobiliário. Uma carteira de investimentos agrupa todos os ativos de um investidor.

A principal motivação deste trabalho é uma tentativa de facilitar e ajudar de maneira social na educação financeira dos brasileiros, pois ao contrário dos Estados Unidos ainda é uma parcela muito pequena da população que conseguem ter conhecimento e dinheiro para realizar investimentos.

O objetivo deste trabalho é apresentar uma descrição do projeto arquitetural sistema gerenciador de investimentos de Renda Variável.

Os objetivos específicos propostos são:

* Visualizar detalhes ações e fundos imobiliários
* Gerenciar carteira de investimentos
* Visualizar a rentabilidade da carteira e de seus ativos
* Visualizar próximos proventos
* Visualizar evolução patrimonial

## Especificação Arquitetural da solução

Esta seção apresenta a especificação básica da arquitetura da solução a ser desenvolvida, incluindo diagramas, restrições e requisitos definidos pelo autor, tal que permitam visualizar a macro arquitetura da solução.

## Restrições Arquiteturais

A abaixo as restrições arquiteturais relativas à aplicação a ser desenvolvida:

R1: Deve utilizar a linguagem utilizada deve ser o Python.

R2: Deve ser utilizado o framework FastApi para fornecer uma API Rest

R3: Deve ser uma aplicação web utilizando framework NextJS

R4: Deve permitir usuário se autenticar através do google ou por um novo cadastro

R5: Deve permitir autorização através token JWT

R6: Deve utilizar api do yahoo finance para obter valores de cotações

R7: Deve ser utilizado banco de dados PostgreSQL para armazenamento de dados

R8: Deve utilizar para gráficos a api do apache echarts

R9: Deve utilizar websocket para valores de cotações

R10: Deve utilizar tailwindcss para parte de estilização de componentes web

## Requisitos Funcionais

Os requisitos funcionais descrevem o como sistema deve ser comportar para atender a expectativas e necessidades do usuário.

Listagem de requisitos funcionais do sistema:

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Descrição Resumida** | **Dificuldade (B/M/A) \*** | **Prioridade**  **(B/M/A) \*** |
| RF01 | Buscar ações pelo nome ou código | B | A |
| RF02 | Buscar fundos imobiliários pelo nome ou código | B | A |
| RF03 | Visualizar detalhes de ações | A | A |
| RF04 | Visualizar detalhes de fundos imobiliários | A | A |
| RF05 | Visualizar próximos pagamentos de proventos | B | M |
| RF06 | Visualizar patrimônio | M | A |
| RF07 | Visualizar rentabilidade da carteira de investimentos | A | A |
| RF08 | Visualizar rentabilidade de ativos individualmente | A | A |
| RF09 | Cadastrar uma nova transação de compra de ativos | B | A |
| RF10 | Cadastrar uma nova transação de venda de ativos | B | A |
| RF11 | Atualizar uma transação de compra ou venda de ativos | B | A |
| RF12 | Listar transação realizadas | B | A |
| RF13 | Cadastrar uma nova data de proventos para um ativo | B | B |
| RF14 | Atualizar data de proventos para um ativo | B | B |
| RF15 | Deletar ativos da carteira de investimento | B | B |
| RF16 | Visualizar índice IBOVESPA | B | M |
| RF17 | Visualizar índice IFIX | M | M |
| RF18 | Visualizar Ações em alta do dia | M | B |
| RF19 | Visualizar Ações em baixa no dia | M | B |
| RF20 | Visualizar Fundos imobiliários em alta do dia | M | B |
| RF21 | Visualizar Fundos imobiliários em baixa no dia | M | B |

\*B=Baixa, M=Média, A=Alta.

## Requisitos Não-funcionais

Os requisitos funcionais são requisitos que estão relacionados ao uso da aplicação que como restrições e aspectos de qualidade.

Listagem de requisitos não funcionais do sistema:

|  |  |  |
| --- | --- | --- |
| **ID** | **Descrição** | **Prioridade**  **B/M/A** |
| RNF01 | Validar transação compra e venda se código do ativo existe | A |
| RNF02 | Validar transação venda se número de ativos vendidos corresponde | A |
| RNF03 | Calcular preço médio de uma ação ao realizar transação compra | M |
| RNF04 | Calcular lucro ou prejuízo ao realizar uma transação de venda | B |
| RNF05 | Realizar cache da listagem de ações para melhorar desempenho | A |
| RNF06 | Realizar cache da listagem de fundos imobiliários para melhorar desempenho | A |
| RNF07 | Calcular preço médio de um fundo imobiliários ao realizar operação compra | M |
| RNF08 | Validar e-mail do usuário existente | A |
| RNF09 | Validar por meio de expressão regulares se o e-mail é válido | A |
| RNF10 | Validar campo de nome tamanho mínimo 4 | A |
| RNF11 | Validar campos de nome tamanho máximo de 30 caracteres | A |
| RNF12 | Criptografar a senha ao cadastrar um novo usuário | A |
| RNF13 | Ao Visualizar patrimônio calcular rentabilidade de cada ativo | M |
| RNF14 | Ao Visualizar patrimônio calcular variação diária de cada ativo | M |
| RNF15 | Ao Visualizar patrimônio calcular variação total de cada ativo | M |
| RNF16 | Ao Visualizar patrimônio calcular percentual da quantidade de cada ativo | M |

## Mecanismos Arquiteturais

Esta seção deve apresentar uma visão geral dos mecanismos que compõem a arquitetura do software, baseando-se em três estados: (1) análise, (2) design e (3) implementação. Em termos de Análise devem ser listados os aspectos gerais que compõem a arquitetura do software, como: persistência, integração com sistemas legados, geração de logs do sistema, ambiente de front end, tratamento de exceções, formato dos testes, formato de distribuição/implantação (deploy), dentre outros. Em Design deve-se identificar o padrão tecnológico a seguir para cada mecanismo identificado na análise. Em Implementação deve-se identificar o produto a ser utilizado na solução, caso ela fosse implementada.

|  |  |  |
| --- | --- | --- |
| **Análise** | ***Design*** | **Implementação** |
| Persistência | ORM | Sqlalchemy |
| Persistência | Banco de dados Relacional | PostgreSQL |
| Cache | Banco de dados chave e valor | Redis |
| Front end | Interface de comunicação com usuário do sistema | Next JS |
| Back end | Linguagem | Python |
| API | Rest | Fastapi |
| Log do sistema | Implementação dos recursos de log | Logoru |
| Autenticação e Autorização | Protocolo de autenticação e autorização | OpenID |
| Segurança Criptografia | Hash | Passlib bcrypt |
| Versionamento | Versionamento de código | Git |
| Deploy | Utilização de container | Docker |

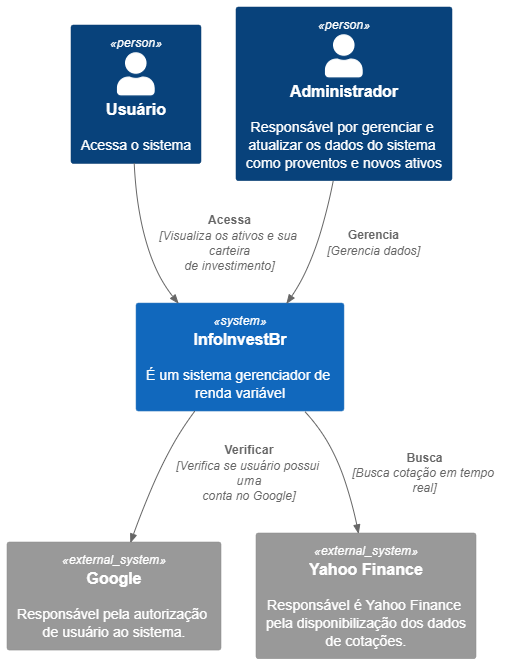
## Modelagem Arquitetural

Esta seção apresenta a proposta da modelagem arquitetural do sistema de gerenciamento de renda variável a solução proposta permite ter uma visão geral do projeto. Para esta modelagem arquitetural optou-se por utilizar o modelo C4 para documentação de arquitetura de software. Dos quatro nível que compõem o modelo C4 três serão apresentados aqui que são diagrama de contexto, diagrama de container e o diagrama de componente.

## 3.1 Diagrama de Contexto

O diagrama de contexto apresenta um visão de geral da macro arquitetura do projeto proposto com nível de abstração bem elevado mostrando o sistema de software e suas responsabilidade, principais usuários e suas dependência.

Apresente um diagrama que mostre a visão geral da solução proposta e, em seguida, explique-o brevemente, de forma textual. Este diagrama não precisa seguir os padrões da UML, e deve ser completo e tão simples quanto possível, apresentando a **macro arquitetura** da solução, como no exemplo:



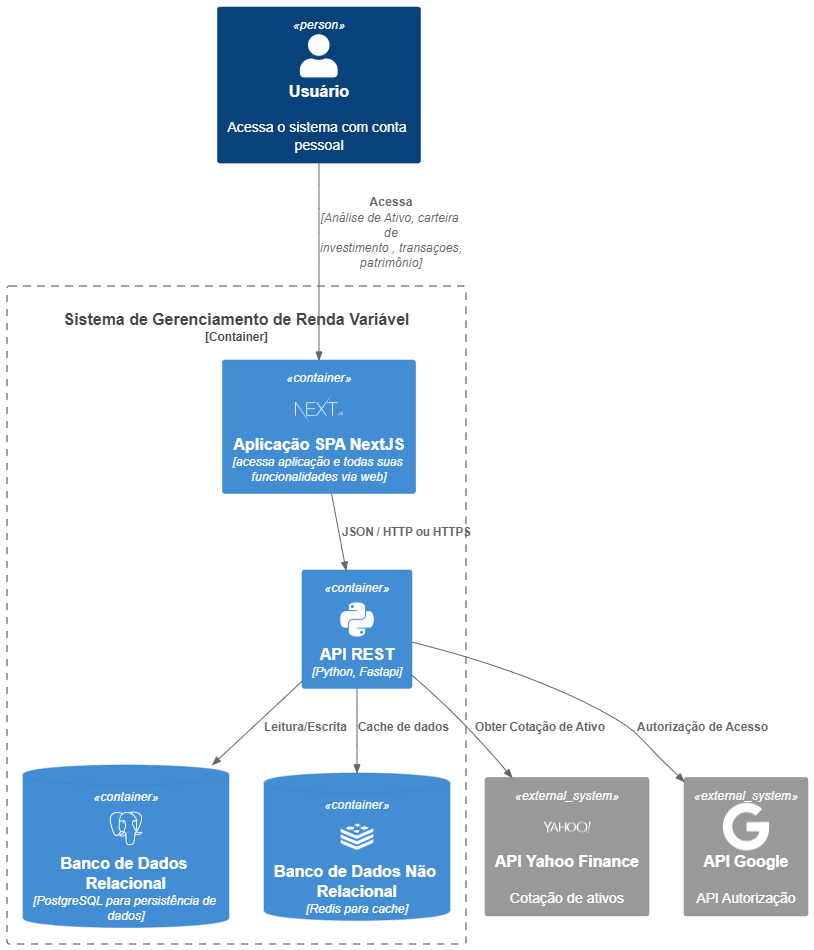
**Figura 1 – Diagrama de Contexto da aplicação**

**Link:** [**Link para imagem**](https://imgur.com/a/vxRz4T9)

A figura 1 mostra a especificação o diagrama geral da solução proposta, com todos seus principais módulos e suas interfaces...

## 3.2 Diagrama de Container

Apresente o Diagrama de Container da aplicação, indicando como os componentes (aplicativos, armazenamentos de dados, microservices, etc.) que compõem esse sistema de software estão distribuídos e organizados. Lembre-se que as decisões de tecnologia que você tomou devem ser contempladas nesse diagrama.

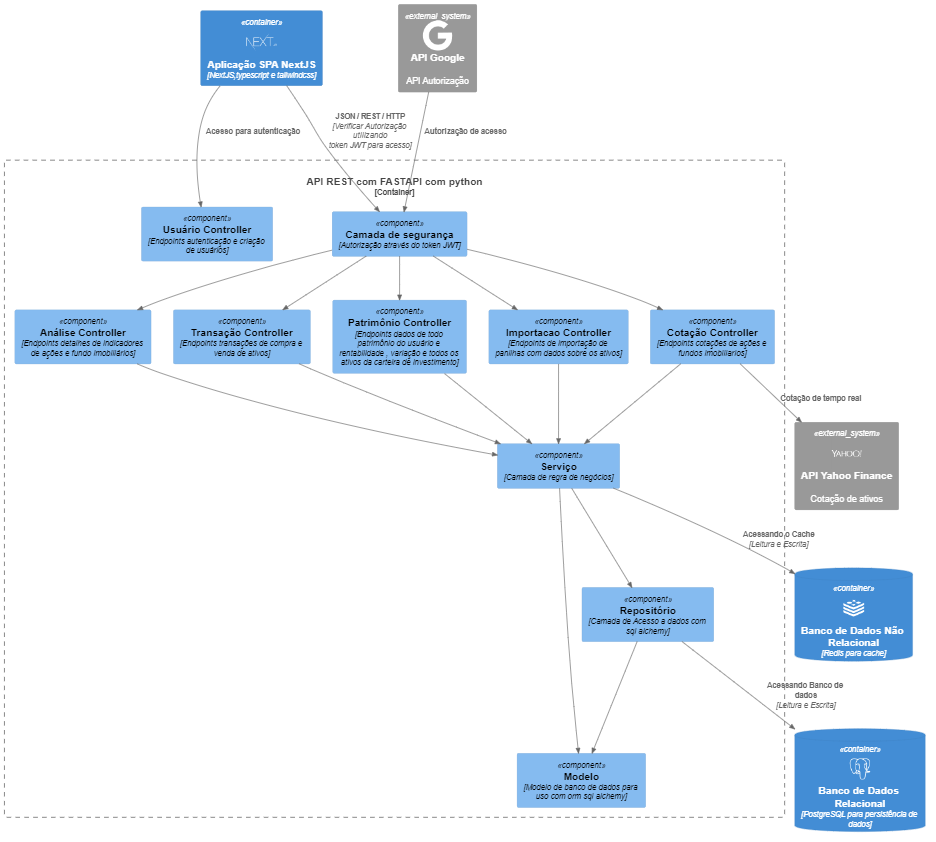


**Figura 2 – Diagrama de container.**

Link: [Link para imagem](https://imgur.com/fg4EFiC)

## 3.3 Diagrama de Componentes

Apresente o Diagrama de Componentes da aplicação (baseado na UML), indicando os elementos da arquitetura e as interfaces entre eles. Liste os estilos/padrões arquiteturais utilizados e faça uma descrição sucinta dos componentes indicando o papel de cada um deles dentro da arquitetura/estilo/padrão arquitetural. Indique também quais componentes serão reutilizados (navegadores, SGBDs, middlewares, etc), quais componentes serão adquiridos por serem proprietários e quais componentes precisam ser desenvolvidos.



**Figura 3 – Diagrama de Componentes.**

[Link para imagem](https://imgur.com/orEWW6t)

## Avaliação da Arquitetura (ATAM)

A avaliação da arquitetura desenvolvida neste trabalho é abordada nesta seção visando avaliar se ela atende ao que foi solicitado pelo cliente, segundo o método ATAM.

## Análise das abordagens arquiteturais

Apresente aqui um breve resumo das principais características da proposta arquitetural. Para isto, utilize o método Architecture Tradeoff Analysis Method (ATAM), no qual são utilizados cenários para fazer essa análise.

Exemplo:

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributos de Quailidade** | **Cenários** | **Importância** | **Complexidade** |
| Interoperabilidade | Cenário 1: O sistema deve se comunicar com sistemas de outras tecnologias. | A | M |
| Usabilidade | Cenário 2: O sistema deve prover boa usabilidade. | M | B |
| Manutenibilidade | Cenário 3: O sistema deve ter a manutenção facilitada. | M | M |

## 

## 4.2 Cenários

Mostre os cenários utilizados na realização dos testes da sua aplicação. Escolha cenários de testes que demonstrem os requisitos não funcionais (atributos de qualidade) sendo satisfeitos. Priorize os cenários para a avaliação segundo critérios quantitativos ou qualitativos.

*Exemplos de cenários:*

*Cenário 1 - Interoperabilidade: Ao acessar a URL do serviço de informações gerenciais via HTTP GET, o mesmo deve retornar as informações no formato JSON.*

*Cenário 2 - Usabilidade: Ao navegar na tela, o sistema deve apresentar boa usabilidade. A navegação deve apresentar facilidade e o acesso as funcionalidades deve ser bem objetivo para a função que precisar ser realizada, o usuário deve ser capaz de efetuar uma compra em no máximo 5 minutos, assim garantindo a agilidade e a usabilidade para ficar de acordo com um dos requisitos não funcionais.*

*Cenário 3 - Manutenibilidade: Havendo a necessidade de alterar o gateway de pagamento somente será necessário fazer alteração no broker da funiconalidade de pagamento, facilitando a manutenção e os testes.*

## 4.3 Evidências da Avaliação

Apresente as medidas registradas na coleta de dados. Para o que não for possível quantificar apresente uma justificativa baseada em evidências qualitativas que suportem o atendimento ao requisito não-funcional.

|  |  |
| --- | --- |
| Atributo de Qualidade: | Interoperabilidade |
| Requisito de Qualidade: | O sistema deve se comunicar com outras tecnologias. |
| Preocupação: | |
| O sistema deve ter como resposta a uma requisição uma saída de fácil leitura por outro componente. | |
| Cenário(s): | |
| Cenário 1 | |
| Ambiente: | |
| Sistema em operação normal | |
| Estímulo: | |
| O sistema de monitoramento envia uma requisição para o serviço REST do módulo de informações gerenciais. | |
| Mecanismo: | |
| Criar um serviço REST para atender às requisições do sistema de monitoramento | |
| Medida de resposta: | |
| Retornar os dados requisitados no formato JSON | |
| Considerações sobre a arquitetura: | |
| Riscos: | Alguma instabilidade na rede pode deixar a conexão lenta ou mesmo a perda de pacotes. |
| Pontos de Sensibilidade: | Não há |
| Tradeoff: | Não há |

Acrescente imagens e descreva os testes realizados, de tal forma que se comprove a realização da avaliação.

Faça isto para todos os cenários apresentados no tópico 6.1.

## Avaliação Crítica dos Resultados

Apresente aqui, de forma resumida, os principais pontos positivos e negativos da arquitetura proposta. Adote uma postura crítica que permita entender as limitações arquiturais, incluindo os prós e contras das tecnologias. Você pode utilizar o formato textual ou produzir um quadro resumo.

Ex. de quadro resumo:

|  |  |
| --- | --- |
| **Ponto avaliado** | **Descrição** |
| xxxxxxxxxxxxxxx | xxxxxxxxxxxxxxxxxxxx |
|  |  |
|  |  |

## Conclusão

Descreva, de forma sucinta, quais foram as lições aprendidas na execução do seu projeto arquitetural. Procure apresentá-las de tal forma que fiquem configurados os *trade-offs* da arquitetura produzida, como por exemplo, Segurança X Desempenho, Granularidade X Manutenibilidade, etc.

Aqui deve ser apresentado também tudo que se aprendeu com esse projeto, de modo a servir como ajuda para outros profissionais.

Também se faz necessário evidenciar as possibilidades de melhoria do projeto, caso se deseje dar continuidade a ele. Nesse sentido, indique possíveis ajustes ou melhorias arquiteturais, que possam vir a ser realizados futuramente.

Lições aprendidas (ex.):

1. xxxxxxxxxxxxxxxxx
2. xxxxxxxxxxxxxxxxx
3. xxxxxxxxxxxxxxxxx

## Referências

Esse trabalho não requer revisão bibliográfica e, por isso, a inclusão das referências não é obrigatória, embora seja recomendada. Caso você deseje incluir referências empregadas em seu trabalho, relacione-as de acordo com as normas ABNT, disponíveis em [www.pucminas.br](http://www.pucminas.br), no *link*: <http://portal.pucminas.br/imagedb/documento/DOC_DSC_NOME_ARQUI20160217102425-n.pdf>.

Exemplo:

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.