

INSTITUTO FEDERAL DO SUDESTE DE MINAS GERAIS
CÂMPUS AVANÇADO BOM SUCESSO
CURSO SUPERIOR EM TECNOLOGIA DE ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

RODRIGO MORAES DOS SANTOS PACHECO

DESENVOLVIMENTO DE UM SOFTWARE PARA GERENCIAMENTO DE
INVESTIMENTOS

BOM SUCESSO, MG

2022

DESENVOLVIMENTO DE UM SOFTWARE PARA GERENCIAMENTO DE INVESTIMENTOS

RODRIGO MORAES DOS SANTOS PACHECO

Trabalho de Conclusão apresentado ao Campus Avançado Bom Sucesso, do Instituto Federal de Educação Ciência e Tecnologia do Sudeste de Minas Gerais, como parte das exigências do Curso de Graduação de Tecnologia em Análise e Desenvolvimento de Sistemas para obtenção do título de Tecnólogo.

Orientador: Prof. Victor Schmidt Comitti

Coorientador: Prof. Pedro Henrique de Oliveira e Silva

BOM SUCESSO, MG
2022

Dados internacionais de catalogação na publicação (CIP)
Bibliotecária responsável Maria de Lourdes Cardoso CRB-6/3242

P116d Pacheco, Rodrigo Moraes dos Santos, 1998 -

Desenvolvimento de um *software* para gerenciamento de investimentos /
Rodrigo Moraes dos Santos Pacheco. -- 2022.

45 f. : il. ; 30 cm.

Orientador: Victor Schimidt Comitti

Coorientador: Pedro Henrique de Oliveira e Silva

Monografia (Graduação) - Instituto Federal do Sudeste de Minas Gerais,
Campus Avançado Bom Sucesso, Coordenadoria de Curso de Tecnologia em
Análise de Desenvolvimento de Sistemas, 2022.

1. Software – Desenvolvimento. 2. Carteiras (Finanças) - Administração. 3.
Investimentos. I. Comitti, Victor Schimidt. II. Silva, Pedro Henrique de Oliveira e. III.
Instituto Federal do Sudeste de Minas Gerais, Campus Avançado Bom Sucesso. IV
Título.

CDD: 005.1



MINISTÉRIO DA EDUCAÇÃO
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO SUDESTE DE MINAS GERAIS

COMPROVANTE DE APROVAÇÃO DO PROJETO POR BANCA EXAMINADORA Nº 3 / 2022 -
BSCCCADS (11.01.10.01.01.02.01)

Nº do Protocolo: NÃO PROTOCOLADO

Juiz de Fora-MG, 04 de Março de 2022

TERMO DE APROVAÇÃO

Rodrigo Moraes dos Santos Pacheco

DESENVOLVIMENTO DE UM SOFTWARE PARA GERENCIAMENTO DE INVESTIMENTOS

Este Trabalho de Conclusão de Curso foi julgado e aprovado como requisito parcial para a obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais - *Campus* Avançado Bom Sucesso.

Bom Sucesso, 04 de março de 2022.

(Assinado digitalmente em 04/03/2022 20:44)

HEBER FERNANDES AMARAL
PROFESSOR ENS BASICO TECN TECNOLOGICO
Matrícula: 1147455

(Assinado digitalmente em 04/03/2022 20:44)

PEDRO HENRIQUE DE OLIVEIRA E SILVA
PROFESSOR ENS BASICO TECN TECNOLOGICO
Matrícula: 1758559

(Assinado digitalmente em 02/05/2022 16:09)

VICTOR SCHMIDT COMITTI
PROFESSOR ENS BASICO TECN TECNOLOGICO
Matrícula: 3082930

Para verificar a autenticidade deste documento entre em <https://sig.ifsudestemg.edu.br/documentos/> informando seu número: **3**, ano: **2022**, tipo: **COMPROVANTE DE APROVAÇÃO DO PROJETO POR BANCA EXAMINADORA**, data de emissão: **04/03/2022** e o código de verificação: **8a34c2d49a**

Dedico este trabalho aos meus pais,
pilares da minha formação como ser
humano.

AGRADECIMENTOS

Agradeço primeiramente à Deus por ter sido meu sustento durante todos os momentos deste trabalho.

Aos meus pais e minha irmã pelo apoio demonstrado, cada um à sua maneira, mas que muito me fortaleceu, por aturarem todos os meus momentos de estresse e serem muito pacientes quando nem mesmo eu era.

Ao meu professor orientador, Victor Schmidt Comitti, por cada palavra, cada conselho, por sempre me motivar principalmente nos inúmeros momentos que pensei em desistir. Nada disso seria possível sem sua atenção e apoio, juntamente com meu professor coorientador Pedro Henrique de Oliveira e Silva, que também não mediu esforços em me apoiar. Duas pessoas que Deus colocou em minha vida, e que serão sempre um espelho para mim.

Aos meus amigos de caminhada, por cada momento compartilhado nesse período de um pouco mais de dois anos e meio, tanto no dentro do Instituto quanto nas reuniões do *Google Meet*.

A todos os demais professores e funcionários do Instituto Federal Câmpus Avançado Bom Sucesso, que de alguma forma participaram da minha caminhada acadêmica.

E por último, mas não menos importante meus amigos de vida, que por muitas vezes só me ouviam falar sobre este trabalho.

A todos, a minha gratidão!

“Não seja refém da motivação, ela é ingrata e traiçoeira. Haverá dias que ela te abandonará. Foque na disciplina, com ela você fará o que precisa ser feito.”

Flávio Augusto

RESUMO

Nos últimos anos as operações no mercado financeiro vem crescendo cada vez mais, alcançando números históricos de investidores ativos, neste cenário comumente pode-se observar que a maioria das pessoas que entram não possuem um conhecimento prévio tanto para iniciar, quanto para gerenciar seu patrimônio. Diante disso, este trabalho apresentará um *software* justamente para este tipo de usuário. O *software* chamado de *Bullseye*, consegue apresentar em gráficos o percentual de representatividade de cada investimento que o usuário inseriu em seu portfólio representa. Esse desenvolvimento foi possível pela aprendizagem adquirida durante todo o período de curso, tanto teórica quanto prática, das metodologias presentes neste trabalho.

Palavras-chave: desenvolvimento de software; investimento; gerenciamento de carteira.

ABSTRACT

In recent years operations in the financial market have been growing more and more, reaching historical numbers of active investors, in this scenario it can commonly be observed that most people who enter do not have prior knowledge both to start and to manage their assets. In view of this, this work will present a software precisely for this type of user. The software called Bullseye, manages to present in graphs the percentage that each investment that the user has entered in his portfolio represents. This development was made possible by the learning acquired throughout the course period, both theoretical and practical, of the methodologies present in this work.

Key-words: software development; investment; portfolio management.

LISTA DE FIGURAS

FIGURA 1. Investidores pessoas físicas na B3 comparado com os índices econômicos.....	18
FIGURA 2. Divisão dos investidores da bolsa por gênero.....	19
FIGURA 3. Evolução da diversificação das mulheres por produto.....	19
FIGURA 4. Evolução da diversificação dos homens por produto.....	20
FIGURA 5. Modelo Espiral	23
FIGURA 6. Diagrama de caso de uso.....	27
FIGURA 7. Diagrama de classe, representando a camada <i>MODEL</i>	28
FIGURA 8. Diagrama de classe, representando a camada <i>VIEW</i>	28
FIGURA 9. Diagrama de classe, representando a camada <i>CONTROLLER</i>	29
FIGURA 10. Diagrama Lógico de Banco de Dados do <i>software</i>	30
FIGURA 11. Fragmento do Código implementado	31
FIGURA 12. Inter-relação entre as camadas da arquitetura MVC.....	32
FIGURA 13. Arquitetura MVC aplicada no <i>software</i>	32
FIGURA 14. Tela de Login.....	33
FIGURA 15. Tela de Cadastro de Usuário.....	34
FIGURA 16. Tela Principal.....	35
FIGURA 17. Tela Minha Carteira.....	35
FIGURA 18. Adicionar Título Renda Fixa.....	36
FIGURA 19. Criar Título Renda Fixa.....	36
FIGURA 20. Adicionar Ativo Renda Variável.....	37
FIGURA 21. Criar Ativo Renda Variável.....	37

LISTA DE SIGLAS

CDB - Certificado de Depósito Bancário

CEI - Centro Eletrônico do Investidor

ETF - *Exchange Traded Funds*

FII - Fundos de Investimentos Imobiliário

JCP - Juros Sobre Capital Próprio

LCA - Letras de Crédito Agronegócio

LCI - Letras de Crédito Imobiliário

MVC - *Model-View-Controller*

SGBD - Sistema de Gerenciamento de Banco de Dados

UML - *Unified Modeling Language*

SUMÁRIO

1 INTRODUÇÃO	14
1.1 OBJETIVOS	15
1.1.1 Objetivo Geral	15
1.1.2 Objetivos Específicos	15
1.2 ORGANIZAÇÃO DO TRABALHO	16
2. REVISÃO DE LITERATURA	17
2.1 PESSOA FÍSICA NA BOLSA	17
2.2 APLICATIVOS DE GERENCIAMENTO DE CARTEIRAS DE INVESTIMENTO	20
2.2.1 Kinvo	21
2.2.2 TradeMap	21
2.2.3 Gorila Invest	21
2.2.4. IM+	22
3. DESENVOLVIMENTO	22
3.1. SOBRE O SISTEMA	22
3.2. PROCESSO DE SOFTWARE	22
3.3. FASES REALIZADAS DURANTE O DESENVOLVIMENTO DO SISTEMA	25
3.3.1. Comunicação	25
3.3.2. Planejamento	26
3.3.3. Modelagem	26
3.3.3.1. Diagrama de Caso de Uso	27
3.3.3.2. Diagrama de Classes	28
3.4. CONSTRUÇÃO	30
3.4.1 Banco de Dados	30
3.4.2. Modelo de entidade e relacionamento	30
3.4.3. Codificação	31
3.4.3.1. Arquitetura do Sistema	32
4 RESULTADOS	34
5 CONCLUSÃO E TRABALHOS FUTUROS	39

REFERÊNCIAS	40
APÊNDICE A - Trecho do código implementado na camada <i>MODEL</i>	43
APÊNDICE B - Trecho do código implementado na camada <i>VIEW</i> .	44
APÊNDICE C - Trecho do código implementado na camada <i>CONTROLLER</i> .	45

1 INTRODUÇÃO

De maneira simplificada podemos definir investimentos como uma forma de comprometimento de recursos no presente com a expectativa da obtenção de maiores recursos em um determinado horizonte de tempo -- seja ele de curto, médio ou longo prazo (BODIE; KANE; MARCUS, 2014).

No mercado existem diversos tipos de investimentos que podem ser classificados como de renda fixa ou renda variável. Os investimentos de renda fixa são aqueles nos quais os rendimentos são conhecidos previamente; seja através de juros prefixados ou de indexadores como a taxa de câmbio, a taxa DI, algum índice de preços ou a taxa SELIC -- taxa básica de juros da economia brasileira. Exemplos de produtos de renda fixa incluem o Tesouro Direto, a Caderneta de Poupança, o Certificado de Depósito Bancário (CDB) e as Letras de Crédito Imobiliário ou do Agronegócio (LCI e LCA). Já os investimentos de renda variável são uma categoria de investimentos mais dinâmica e que envolve um risco maior uma vez que os ativos financeiros pertencentes a esta categoria não oferecem retornos previsíveis para o investidor. Os principais produtos de renda variável são ações negociadas em bolsas de valores, Fundos de Investimento Imobiliários (FIIs), *Exchange Traded Funds* (ETFs), derivativos, criptomoedas entre outros ativos.

A bolsa de valores brasileira atingiu a marca de 3,7 milhões de investidores pessoas físicas, o que representa uma alta de 54% entre abril de 2020 e abril de 2021 comparado com o período equivalente um ano antes. Este aumento de investidores se deve muito à era da tecnologia que estamos vivendo pois, com o vasto leque de acesso às informações que a internet nos proporciona, aliado ao momento que o mundo vem passando, as pessoas ou estão se dando conta de quanto uma educação financeira é importante e até necessária, ou pelo fato de se ter a possibilidade real de estar aumentando seu capital/patrimônio através dos investimentos (CAMPOS, 2021).

O Brasil é um país com histórico de juros altos. Porém, nos últimos anos, a SELIC saiu de 14,25% ao ano em Agosto de 2016 para 2% ao ano em Agosto de 2020. Esse movimento levou muitas pessoas a abandonar a renda fixa e entrar no mercado da renda variável na busca por retornos maiores para seus investimentos. A renda variável, entretanto, apresenta desafios bem diferentes em relação a outras

modalidades mais conservadoras de investimento. Neste mercado o investidor se depara com dois riscos principais: o risco de mercado e o risco idiossincrático de cada ativo. O primeiro é sistemático e, portanto, não pode ser mitigado; já o segundo pode ser eliminado através da diversificação dos ativos na carteira. Existe ainda o risco de liquidez: aquele que está relacionado à dificuldade que o detentor de um ativo irá encontrar para transformá-lo em dinheiro. Em face dessas dificuldades, é essencial que o investidor seja capaz de fazer a gestão de risco de sua carteira para se proteger da volatilidade do mercado (SALLES; LIU; RODRIGUES, 2009).

Contudo, ter uma carteira de investimentos com um grande número de ativos pode se tornar bastante desafiador, principalmente para o pequeno investidor que dispõe de pouca experiência e, na maioria das vezes, não possui o auxílio de um profissional da área. Na ânsia de conseguir ganhos rápidos, o investidor iniciante aloca em sua carteira ativos de altíssimo risco, o que pode levar a perdas significativas de seu capital.

Tendo como base esses fatores, surge a ideia do desenvolvimento de um *software* que auxilie justamente esse público, no gerenciamento de seus investimentos.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Desenvolver um software para auxiliar novos investidores a gerenciar suas aplicações.

1.1.2 Objetivos Específicos

- Analisar o perfil de novos investidores;
- Realizar um levantamento de requisitos;
- Realizar um levantamento dos tipos de aplicações;
- Análise e comparativo de softwares voltados para o gerenciamento de aplicações;
- Definição das funcionalidades do aplicativo;
- Modelagem das funcionalidades do software e

- Implementação das funcionalidades do software;

1.2 ORGANIZAÇÃO DO TRABALHO

O presente trabalho foi dividido em 5 capítulos. O capítulo 2 trata-se da revisão de literatura, no qual são abordados dados relevantes sobre a presença de pessoas físicas na bolsa de valores e é realizado o levantamento de aplicativos existentes atualmente dentro do mercado que o *software* irá atuar. O capítulo 3 fornece informações sobre todas as metodologias e ferramentas utilizadas para o desenvolvimento do *software* iniciando pela fase da escolha do processo de *software*, depois pela comunicação em que houve o levantamento de requisitos, pelo planejamento onde foi definido qual arquitetura utilizar, qual modelagem para o sistema, qual estrutura de banco de dados e por fim qual linguagem de programação, chegando aos resultados que são apresentados no capítulo 4. E enfim no capítulo 5 é apresentada a conclusão.

2. REVISÃO DE LITERATURA

2.1 PESSOA FÍSICA NA BOLSA

Nos últimos anos, observou-se um grande aumento do número de investidores Pessoa Física (PF) na Bolsa de Valores brasileira. O Quadro 1 mostra que no período 2010-2020 esse tipo de investidor praticamente quintuplicou. Segundo dados da B3 - a controladora da Bolsa de Valores brasileira em São Paulo --, o início da pandemia do COVID-19, em março de 2020, marcou um grande influxo de contas PF na bolsa que, em agosto de 2021, chegam a 3,2 milhões de CPFs (BERTÃO, 2020).

Quadro 1 - Crescimento de pessoas físicas na bolsa de valores

Ano	Quantidades de CPFs
2002	85.249
2003	85.478
2004	116.914
2005	155.183
2006	219.634
2007	456.557
2008	536.483
2009	552.364
2010	610.915
2011	583.202
2012	587.165
2013	589.276
2014	564.116
2015	557.109
2016	564.024
2017	619.625
2018	813.291
	Continua...

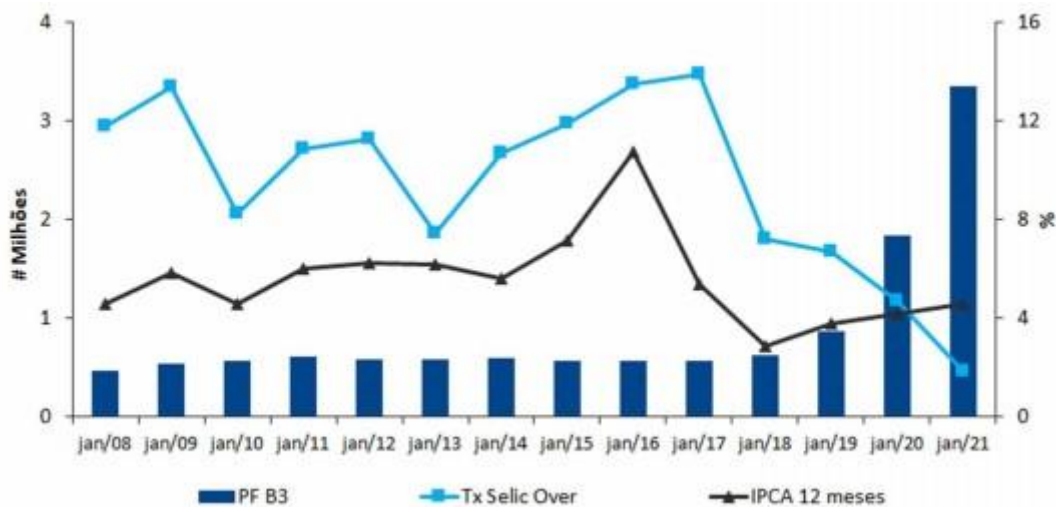
2019	1.681.033
2020	2.958.442
2021 até agosto	3.200.000 aproximadamente

Fonte: Autoria própria adaptado de B3

Esse crescimento pode ser explicado pelo fato de que as pessoas estão sempre buscando maneiras de aumentar o próprio patrimônio. Zanotelli (2021) ressalta que, devido ao ciclo de juros baixos pelo qual o Brasil passou entre 2018 e 2021, o ganho real em modalidades conservadoras de investimento foi muito baixo ou até mesmo negativo. Por esse motivo, houve um grande movimento de PFs que migraram da renda fixa para a renda variável em busca por rentabilidades maiores. Podendo-se observar no Quadro 1 dois períodos onde há uma grande expansão do número de pessoas físicas ingressando na bolsa, que são os períodos entre 2006 e 2007 qual apresentou um crescimento bastante significativo e o outro entre os anos de 2019 e 2020 atingindo número históricos, até o início de 2021 período que houve uma desaceleração nesse crescimento (RYNGELBLUM, 2021).

A Figura 1 mostra o crescimento da presença de investidores PF na bolsa de valores brasileira (contas de custódia abertas na B3) e a trajetória de dois indicadores econômicos: Taxa Nominal de Juros e o Índice de preços ao consumidor. Nota-se um estreitamento da taxa de juros real, que segundo a ExpertXP (2021) é o indicador que representa o verdadeiro lucro em um investimento, pelo fato de subtrair do rendimento os impactos da inflação, que a partir de 2018 coincide com a elevação da presença do investidor pessoa física no mercado de ações.

Figura 1. Investidores pessoas físicas na B3 comparado com os índices econômicos



Fonte: Rodrigo Queiroz, 2021

Porém, o pequeno e novo investidor está entrando na bolsa de valores sem o mínimo de conhecimento prévio o que pode acarretar problemas futuros uma vez que esse tipo de investidor frequentemente coloca suas reservas em risco apostando em ativos complexos e de alta volatilidade. Uma alternativa para esse tipo de investidor seria buscar um acompanhamento com um profissional da área para assim ter alguma noção e um direcionamento.

A Figura 2 mostra a divisão por gêneros dos investidores na bolsa de valores, pode-se observar que apesar das mulheres ainda serem minoria, vem aumentando gradativamente a cada ano que passa.

Figura 2. Divisão dos investidores da bolsa por gênero



Fonte: Autoria própria adaptado de B3

A Figura 3 e Figura 4 mostram o perfil por gênero dos investidores da B3. É possível observar que não só o crescimento da presença feminina na bolsa como também o fato de que elas vêm diversificando seus investimentos cada vez mais. Esse cenário mostra um nível de amadurecimento das investidoras que se reflete também na inserção das mulheres no mercado financeiro em posições de destaque como gestoras de grandes fundos e economistas-chefe de bancos (JACONELI, 2021).

Figura 3. Evolução da diversificação das mulheres por produto

	2016	2017	2018	2019	2020	2021 (2tri)	2021 (3tri)
Ações	77%	73%	67%	50%	54%	51%	50%
FII	10%	12%	15%	19%	12%	12%	12%
ETF	1%	1%	2%	1%	1%	1%	2%
Outros	2%	2%	1%	0%	0%	1%	1%
Ações + FII	7%	10%	13%	23%	25%	24%	24%
Ações + ETF	2%	2%	2%	2%	2%	3%	4%
FII + ETF	0%	0%	0%	1%	1%	1%	1%
Ações + FII + ETF	1%	1%	1%	4%	5%	7%	7%
	0,1%	0,1%	0,2%	0,3%	0,7%	0,9%	1,0%

Fonte: Autoria própria adaptado de relatórios da B3

Figura 4. Evolução da diversificação dos homens por produto

	2016	2017	2018	2019	2020	2021 (2tri)	2021 (3tri)
Ações	79%	75%	68%	52%	53%	50%	49%
FII	6%	7%	10%	12%	8%	8%	8%
ETF	1%	1%	1%	1%	1%	1%	1%
Outros	2%	1%	0%	0%	0%	1%	1%
Ações + FII	9%	12%	17%	29%	30%	27%	27%
Ações + ETF	2%	2%	2%	2%	2%	2%	4%
FII + ETF	0%	0%	0%	1%	1%	4%	4%
Ações + FII + ETF	1%	1%	2%	5%	6%	9%	9%
	0,4%	0,4%	0,6%	1,1%	1,9%	2,2%	2,3%

Fonte: Autoria própria adaptado de relatórios da B3

2.2 APLICATIVOS DE GERENCIAMENTO DE CARTEIRAS DE INVESTIMENTO

Atualmente existem no mercado de software alguns aplicativos que possuem algumas semelhanças com o proposto neste trabalho. Nesta seção, iremos apresentar os principais aplicativos desta classe e mostrar suas principais funcionalidades e limitações.

2.2.1 Kinvo

Um deles é o Kinvo que é um aplicativo para o gerenciamento de investimentos, que foi desenvolvido pela fintech Kinvo. Ele é dividido em duas versões, uma gratuita e outra premium. Na sua versão gratuita tem como funcionalidades acompanhar os ativos do usuário, que podem ser inseridos tanto manualmente como pela exportação através do Centro Eletrônico do Investidor (CEI); disponibiliza uma análise de ativos, para o melhor entendimento do investidor, conta também com um sistema de cálculo automatizado para visualização dos ativos em carteira, além de uma ferramenta para se visualizar os proventos recebidos por ativos e ainda na versão premium oferece serviços adicionais de consultoria e projeções para o usuário (NUNES, 2021).

2.2.2 TradeMap

O TradeMap, assim como o Kinvo, permite ao usuário tanto adicionar ativos manualmente como importar sua carteira de ativos direto do CEI, para seu melhor acompanhamento. Também é possível buscar por algum ativo que não esteja em sua carteira e acompanhá-lo em tempo real. Disponibiliza ainda diversas informações a respeito dos diversos produtos financeiros como seus indicadores fundamentalista. Também possibilita a visualização de todos os títulos de renda fixa que existem no mercado, além de uma aba que apresenta notícias relacionadas ao mercado de financeiro e que foram divulgadas nos mais diversos portais de notícias (NUNES, 2021).

2.2.3 Gorila Invest

No Gorila, assim como em outros aplicativos de gerenciamento de carteiras, é possível adicionar manualmente seus diversos investimentos ou também importá-los para dentro da plataforma. Tem como diferencial a funcionalidade eventos de carteira, na qual o investidor consegue observar o montante do retorno obtido em cada aplicação em um determinado período, seja juros sobre capital próprio (JCP), dividendos, rendimentos dos FII's entre outros.

2.2.4. IM+

O IM+ é um aplicativo de gerenciamento de carteiras, que surgiu no mercado com a junção do InfoMoney, um portal financeiro do Brasil, com o Fliper que já era um aplicativo de gerenciamento de ativos bem consolidado no mercado. Têm como funcionalidades, criar um vínculo com diversas contas, bancos e corretoras que o investidor venha a possuir, a partir daí o aplicativo apresenta resumos das aplicações existentes, sejam elas como um todo ou individuais, separadas por contas, por tipos de investimentos entre outras possibilidades. Apresenta quais ativos obtiveram maiores resultados, ou menores, em um determinado período de tempo, apresenta também a rentabilidade e a liquidez da carteira.

3. DESENVOLVIMENTO

Neste capítulo serão abordados os métodos que foram utilizados no trabalho quanto às suas características e funcionalidades, bem como as etapas realizadas durante o desenvolvimento do software de gerenciamento de aplicações.

3.1. SOBRE O SISTEMA

O sistema consiste em um *software* capaz de receber informações sobre investimentos de um determinado usuário, e apresentar por meio de gráficos a alocação relativa de cada ativo em sua carteira.

3.2. PROCESSO DE SOFTWARE

Um dos fatores principais para o sucesso no desenvolvimento de um software é escolher o processo de *software* que mais se adequa ao sistema que será desenvolvido. Segundo Pfleeger (2001) o processo de software é um conjunto de atividades realizadas para construir *software*, levando em consideração os produtos sendo construídos, as pessoas envolvidas, e as ferramentas com as quais trabalham. Neste sentido, o modelo de processo de *software* escolhido para o desenvolvimento do sistema proposto, foi o modelo de processo evolucionário espiral.

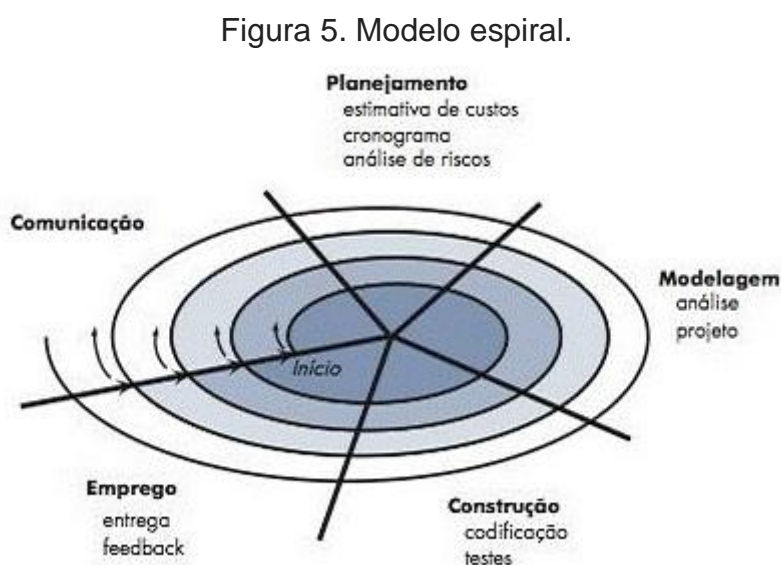
Os modelos de processos evolucionários são caracterizados por serem iterativos e apresentarem características que possibilitam o desenvolvimento de

versões cada vez mais completas do *software*. A escolha por este modelo se dá pela característica do *software* evoluir ao longo do tempo e conforme o seu desenvolvimento avança também há mudanças nas necessidades de negócio e de produtos.

Segundo Devmedia (2013) o modelo espiral é um modelo de processo de software evolucionário iterativo com aspectos sistemáticos e controlados do modelo cascata. O modelo espiral fornece um grande potencial para que possamos ter rápido desenvolvimento de versões cada vez mais completas.

Segundo Devmedia (2013) o modelo espiral se inicia no centro e segue no sentido horário. Os riscos são considerados à medida que cada evolução é realizada. A primeira atividade se dá com o desenvolvimento de uma especificação de produto, as próximas passagens podem ser usadas para desenvolver um protótipo e, assim sucessivamente, dando origem a versões cada vez mais sofisticadas do *software*. Cada passagem pela parte de planejamento, por exemplo, resulta em ajustes no planejamento do projeto. O custo e o cronograma são sempre ajustados de acordo com o *feedback* obtido do cliente após uma entrega. Também há um ajuste no número de iterações planejadas para completar o *software*.

O modelo espiral possui diversas atividades definidas pela engenharia de *software* sendo que cada uma dessas atividades representa um segmento do caminho espiral. A Figura 5 apresenta o modelo espiral.



A fase de comunicação ou fase de levantamento de requisitos, é a fase na qual é realizada uma interação com o usuário, para se apurar qual é o interesse principal do *software*, e quais serão os requisitos necessários para o seu desenvolvimento. De acordo com Pressman (2006), esse processo de levantamento de requisitos tem como objetivo garantir que, através dos requisitos levantados, os mesmos possam atender satisfatoriamente às expectativas e principalmente às necessidades do usuário.

De acordo com Sommerville (2007), requisitos são condições que definem as funcionalidades e os recursos do sistema. Através deles os usuários irão ter ideia de como o software se comporta em cada situação.

Os requisitos podem ser classificados como funcionais ou não funcionais. Os funcionais abrangem, como o nome diz, todas as funcionalidades esperadas no processo de execução do software; já os não funcionais são aqueles associados ao desempenho, à confiabilidade, à segurança do sistemas e às tecnologias utilizadas em sua criação.

A fase que vem logo após ao levantamento de requisitos é a do planejamento na qual, segundo Pressman (2006), deve-se determinar quais serão as atividades a serem realizadas de acordo com o modelo de processo, definir quais ferramentas serão utilizadas com base nas atividades e, se houver equipe, definí-la. Tudo isso para que, caso haja algum imprevisto, não venha a ser algo que impeça de maneira decisiva o andamento do projeto.

Logo após, passando para a fase da modelagem do *software*, que para Rumbaugh, Booch, e Jacobson (2000) é a fase central de todas as atividades que conduzem à implantação de um bom *software*. Uma boa modelagem de *software* permite com que os desenvolvedores consigam realizar um trabalho bastante eficaz, pois através dos diagramas, se forem bem modelados, vão conseguir obter uma fácil compreensão de todas as informações, pelo fato da modelagem conseguir representar um sistema real através de um formato abstrato e simples, tornando-se assim uma espécie de guia de revisão.

Na fase da construção engloba duas etapas, a da codificação e a de testes. A parte da codificação transpõe a representação do projeto com uma linguagem de programação, utilizando geralmente uma linguagem de alto nível, que segundo

Almeida (2014) nesta fase deve-se desenvolver os componentes e classes, em obediência aos requisitos levados anteriormente.

A etapa de testes de um sistema de acordo com Myers (2004), é uma parte onde se observa, e chega-se a conclusão sobre a qualidade que o *software* está entregando em seu funcionamento, e ainda Farias et al (2012) ressalta o quão necessário são os testes, pois verificam se os requisitos estão sendo atendidos, mas que independentemente da importância desta fase, em muitos projetos não recebe o destaque esperado. Para a realização desta etapa foram introduzidos dados fictícios no *software* para testá-lo, justamente para se observar e corrigir problemas de funcionamento da versão antes que passe para a etapa de entrega.

Na fase do emprego ou entrega no modelo espiral se dá em versões. Assim, de acordo com a Devmedia (2013), cada versão pode ser adaptada em cada entrega, até que o cliente fique satisfeito com o resultado, podendo-se assim declarar que o *software* foi totalmente concluído.

3.3. FASES REALIZADAS DURANTE O DESENVOLVIMENTO DO SISTEMA

3.3.1. Comunicação

Fase que deu início ao desenvolvimento do software na qual foram levantadas diversas questões sobre a sua real necessidade, para quem seria direcionado e como ele seria desenvolvido.

Após estes questionamentos, e partindo da ideia de que o sistema servirá para que um investidor que esteja ingressando no mercado de investimento, possa ter uma visualização simples e intuitiva de como está sendo dividido seu portfólio. Então, depois disso, começou-se a pensar sobre os requisitos necessários para o desenvolvimento do sistema e, só após pesquisa, foram determinados os requisitos, que estão apresentados no Quadro 2.

Quadro 2. Requisitos Funcionais e Não Funcionais

REQUISITOS FUNCIONAIS	
Cadastro de Usuários	<p>O sistema possibilitará cadastrar usuários (investidores).</p> <p>O sistema permitirá ao usuário realizar alterações em seu cadastro.</p> <p>O sistema permitirá ao usuário realizar a exclusão de seu próprio cadastro.</p> <p>O Cadastro dos usuários deverá constar (nome, CPF, data nascimento, renda telefone, email principal e alternativo, endereço e senha para acesso do sistema).</p>
Login	O sistema permitirá o acesso através da validação em seu banco de dados do CPF (login) e a senha criada pelo usuário.
Composição da Carteira	<p>O sistema possibilitará o usuário cadastrar seus ativos tanto de renda fixa, quanto renda variável.</p> <p>O usuário poderá adicionar ou remover ativos que desejar de sua carteira de investimento.</p> <p>O sistema apresentará a porcentagem que cada ativo ocupa na carteira através de gráficos.</p>
REQUISITOS NÃO FUNCIONAIS	
O sistema deve apresentar um interface bastante intuitiva.	
O sistema deve consumir o mínimo de recursos do computador.	
O sistema deve possuir um tempo de resposta baixo em cada operação.	

Fonte: Autoria própria.

3.3.2. Planejamento

A fase de planejamento teve como objetivo definir quais ferramentas e qual arquitetura de software iriam ser utilizadas para o desenvolvimento do projeto, de modo torná-lo o mais funcional possível. Para a arquitetura de *software* optou-se pelo modelo arquitetural *Model View Controller* (MVC), que será detalhado no tópico 3.4.3.1.

3.3.3. Modelagem

Para a modelagem do sistema, que é a parte de criação de diagramas que ilustram o processo de funcionamento do software em desenvolvimento, foi utilizada a *Unified Modeling Language (UML)* - Linguagem de Modelagem Unificada -, que é uma linguagem bastante empregada para se modelar sistemas orientados a objetos e que para Guedes (2018) é fundamental para fase que engloba o levantamento de requisitos do projeto.

Segundo Guedes (2018) a UML é composta por diagramas divididos em duas categorias:

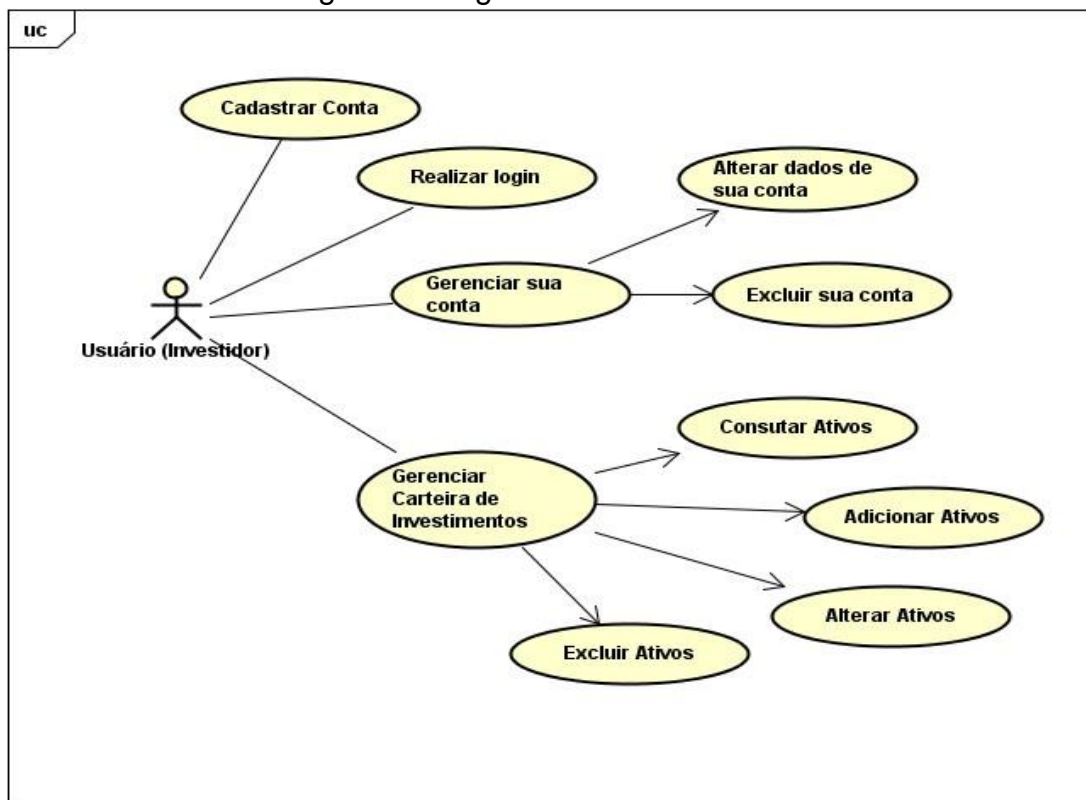
- Os Diagramas Estruturais que servem para mostrar como é a parte estática do software - um exemplo clássico dessa categoria é o diagramas de classe -;
- Diagramas Comportamentais que são utilizados para mostrar a parte dinâmica do software, como por exemplo o diagrama de Caso de Uso .

Para a criação dos diagramas utilizados no *software* em questão foi utilizado o Astah Professional 8.3.0, por ser uma ferramenta bastante versátil pelo fato de dispor de todos os diagramas pertencentes à UML.

3.3.3.1. Diagrama de Caso de Uso

Levando em conta os requisitos levantados anteriormente, foi possível realizar a modelagem do sistema através de um diagrama de caso de uso (Figura 6) que define todas as funcionalidade que ele possuirá bem como o seu funcionamento básico.

Figura 6. Diagrama de Caso de Uso

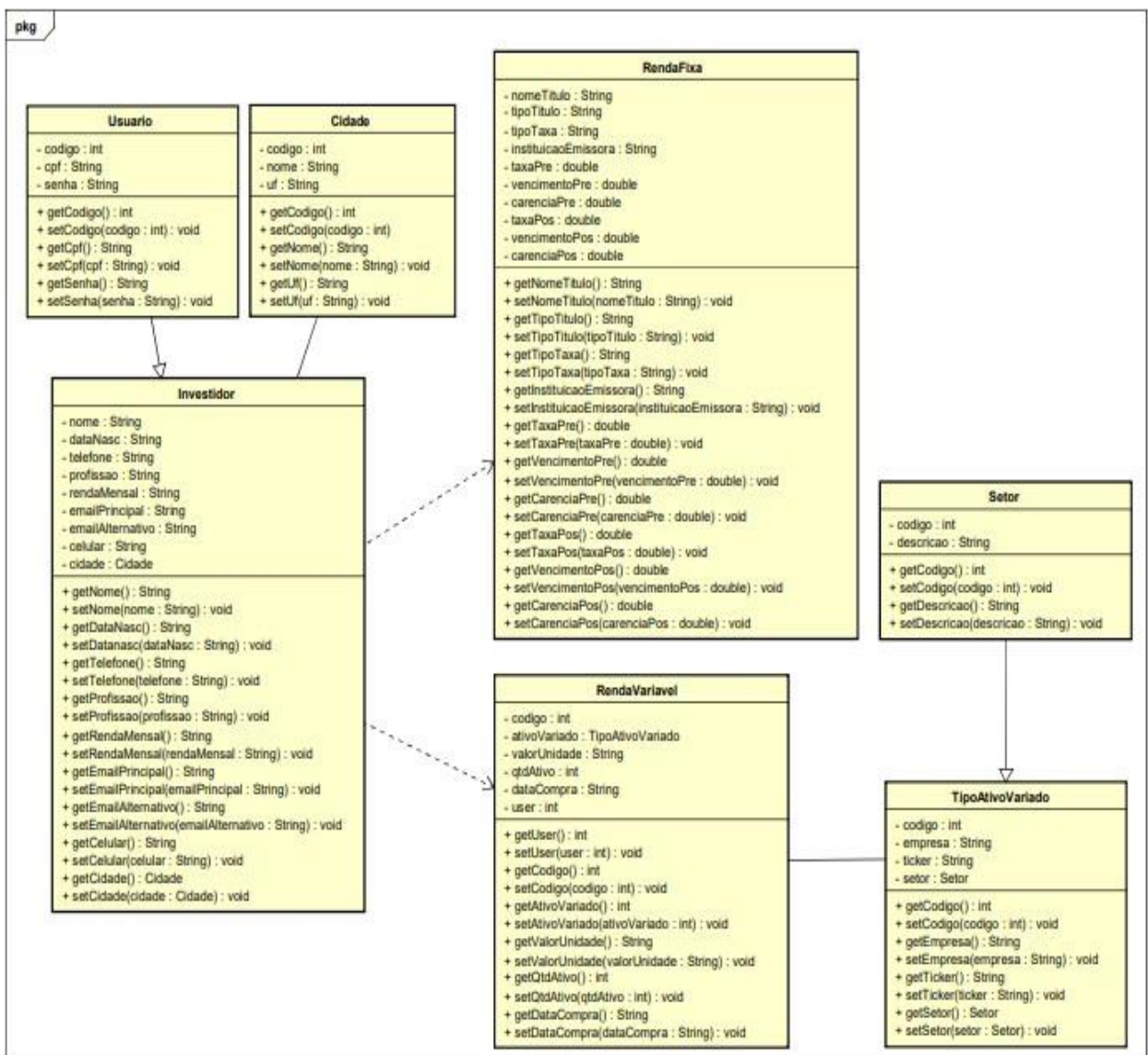


Fonte: Autoria própria

3.3.3.2. Diagrama de Classes

Como já citado anteriormente, existem diagramas utilizados para demonstrar a parte estática do software nos quais são apresentadas todas as classes existentes no sistema, seus atributos, operações e seus relacionamentos entre objetos. Para este trabalho foi escolhido para a representação da parte estática o diagrama de classes, que é um diagrama de fácil entendimento, e se parece com o modelo de banco de dados, o que o torna muito prático. A Figura 7, 8 e 9 apresentam os diagramas de classe das camadas *model*, *view*, *controller* respectivamente do sistema proposto.

Figura 7. Diagrama de Classe, representando a camada *MODEL* do *software*



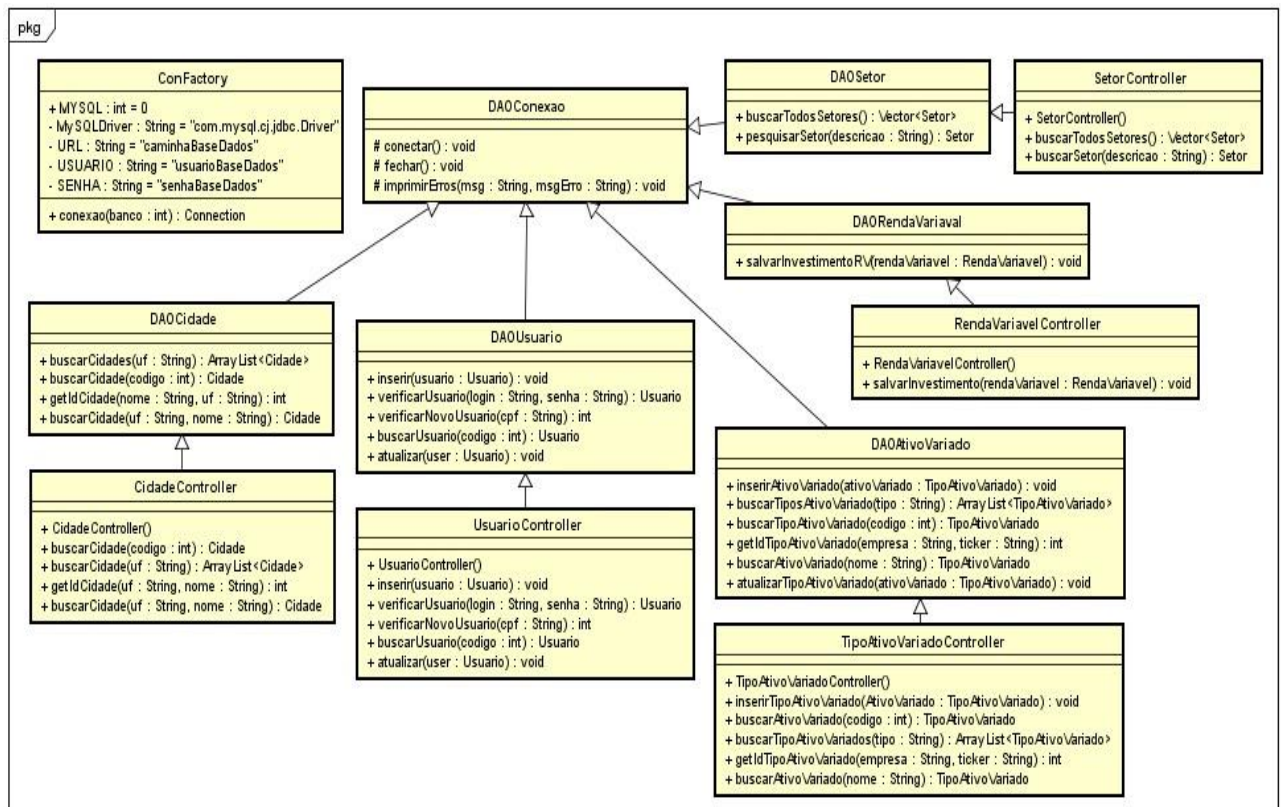
Fonte: Autoria própria

Figura 8. Diagrama de Classe, representando a camada VIEW do software



Fonte: Autoria própria

Figura 9. Diagrama de Classe, representando as camadas CONTROLLER e DAO do Software



Fonte: Autoria própria

3.4. CONSTRUÇÃO

3.4.1 Banco de Dados

De maneira simples, banco de dados nada mais é que um sistema de armazenamento de dados que tem por finalidade facilitar a organização de informações que possuem alguma relação entre si. Essa relação pode ser, por exemplo, direta, por hierarquia ou por categoria (ROB, 2011). Para a organização e armazenamento dos dados do sistema optou-se pela utilização do Banco de Dados MYSQL, que possui uma boa e eficiente arquitetura de banco de dados e um sistema de gerenciamento de banco de dados (SGBD) de fácil gerenciamento e bastante consistente.

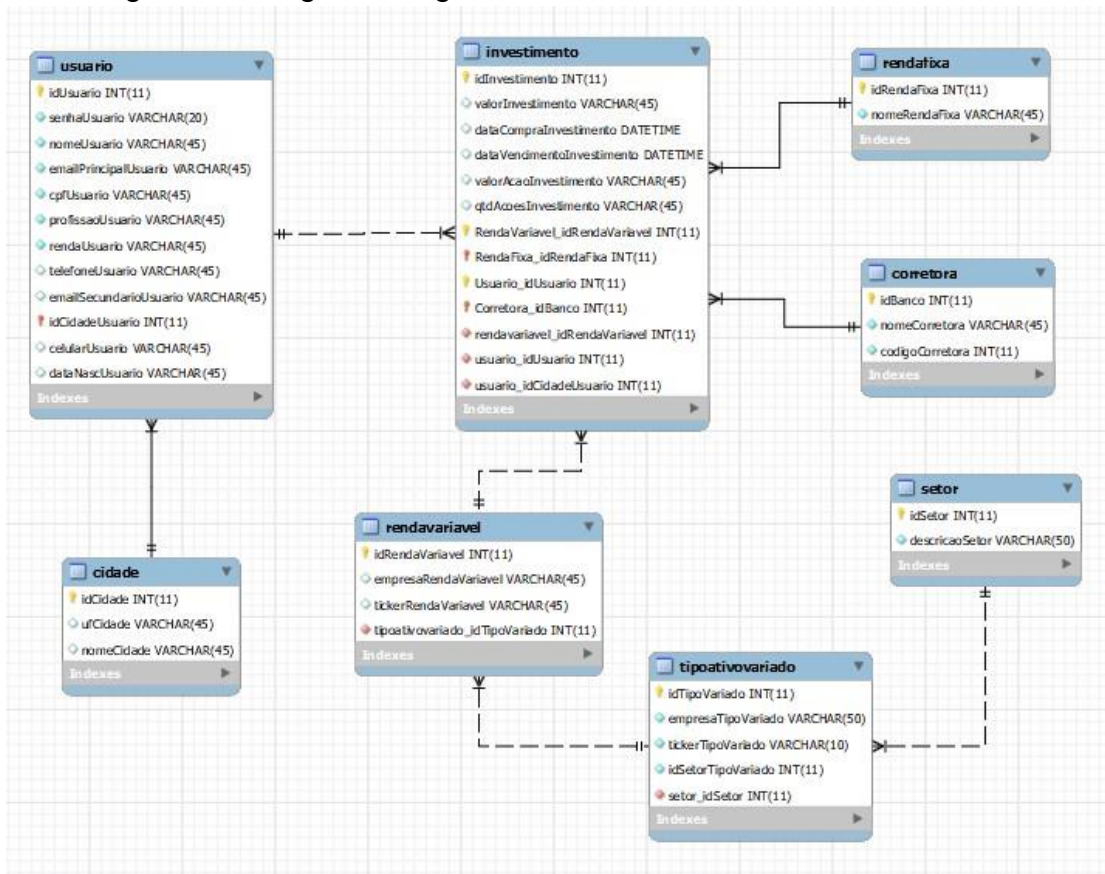
Existem diversos modelos para a criação desse sistema de armazenamento de dados. Alguns mais comuns são, o modelo relacional, o modelo hierárquico, o modelo de rede, o modelo entidade-relacionamento, o modelo de banco de dados orientado para objetos entre outros. Para a criação e representação do banco de dados utilizou-se o modelo de entidade e relacionamento.

3.4.2. Modelo de entidade e relacionamento

O modelo de entidade e relacionamento de acordo com Maia (2013), foi proposto pelo chinês Peter Pin-Shan Chen em 1976. Esse modelo surgiu pela necessidade de estruturas mais elaboradas de banco de dados, e com isso a necessidade de ferramentas de *design* para o mesmo.

Segundo Maia (2013), esse modelo tem como objetivo criar uma estrutura do banco de dados por meio de uma interpretação gráfica. Baseando-se na percepção de um mundo real através de suas entidades e seus relacionamentos. E que foi escolhida para o *software* proposto por ser muito útil para mapear graficamente as interações de empresas reais em um modelo conceitual e ainda, de acordo com Laender e Flynn (1993), o modelo entidade e relacionamento tem como base princípios simples e de entendimento fácil, possibilitando uma abordagem muito eficaz para a modelagem de dados, o que o torna um modelo amplamente aceito para a modelagem dos dados. A Figura 10 representa o diagrama do banco de dados do *software*.

Figura 10. Diagrama Lógico de Banco de Dados do Software.



Fonte: Autoria própria

3.4.3. Codificação

Para a codificação do sistema foi escolhida a linguagem de programação JAVA que foi trabalhada na plataforma de desenvolvimento Netbeans IDE 12.5. Essa linguagem foi escolhida devido à sua versatilidade, ao aspecto multiplataforma e à facilidade em se reaproveitar códigos-fonte sem que haja necessidade de reescrevê-lo, por ser uma linguagem orientada a objetos. Além disso, o Java possui uma das maiores plataformas *open source*. Já a escolha pelo Netbeans foi motivada por suas inúmeras ferramentas que facilitam o desenvolvimento de diferentes tipos de software facilitando o desempenho dos programadores que o utilizam. A Figura 11 apresenta um fragmento de código.

Figura 11. Fragmento do código implementado

```
public class RendaVariavelController {  
  
    DAORendaVariavel dao;  
  
    public RendaVariavelController() {  
        dao = new DAORendaVariavel();  
    }  
  
    public void salvarInvestimento(RendaVariavel rendaVariavel) {  
        dao.salvarInvestimentoRV(rendaVariavel);  
    }  
}
```

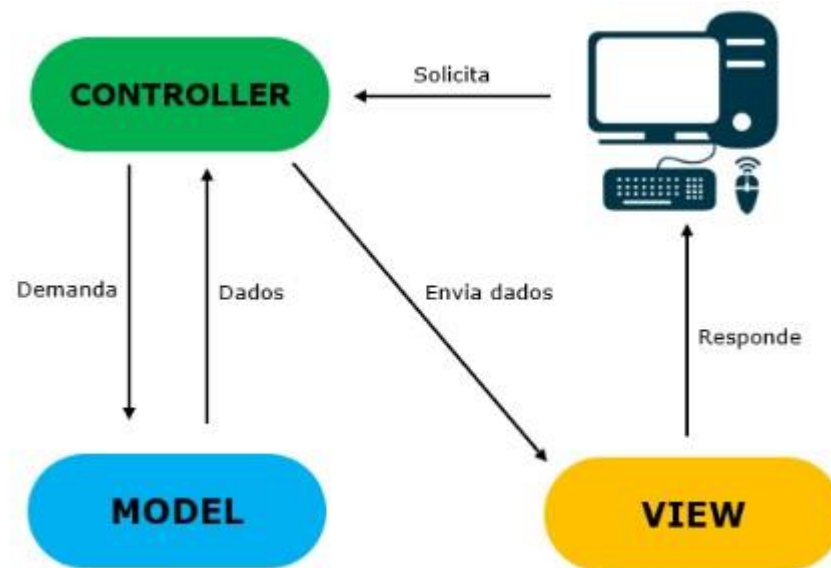
Fonte: Autoria própria

3.4.3.1. Arquitetura do Sistema

Para a parte arquitetural do sistema foi escolhido o padrão *Model View Controller* (MVC). Essa é uma arquitetura que surgiu no início dos anos 80, sendo idealizada por Trygve Reenskaung, funcionário da empresa Xerox Parc, vindo a se popularizar na criação de aplicações WEB (LUCIANO e ALVES, 2017).

De acordo com Luciano e Alves (2017), o MVC é um padrão que possui uma execução simples, com todos os recursos de suas aplicações direcionados para a camada *Controller*, a qual acessa a camada *Model* onde por sua vez ocorre a manipulação de dados, chegando no final do processo em que o resultado é apresentado ao usuário final na camada *View*. Esse processo está ilustrado na Figura 12. Pode-se dizer que o padrão MVC divide as camadas em de apresentação, lógica, de negócios e de gerenciamento do fluxo da aplicação. Através dessa divisão é possível aumentar a capacidade de manutenção e de reutilização do projeto.

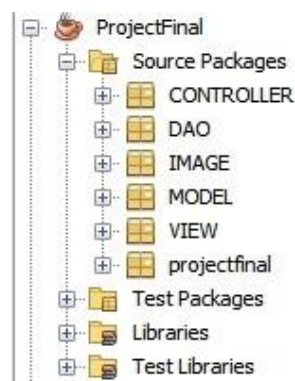
Figura 12 - Inter-relação entre as camadas da arquitetura MVC



Fonte: Luciano e Alves (2017)

A arquitetura MVC, foi proposta para o *software*, principalmente pelo fator de possibilitar que se divida o projeto em camadas muito bem estabelecidas, e que tanto o *Model*, o *Controller* e o *View*, executem somente o que será definido em ambas e nada mais que isso. E ainda como ressalta Devmedia (2013) o padrão MVC pode ser utilizado em diversos tipos de projetos, sejam eles, *desktop*, *web* ou *mobile*.

Figura 13 - Arquitetura MVC aplicada no *software*



Fonte: Elaboração própria

4 RESULTADOS

Neste capítulo serão apresentadas as principais telas do sistema de gerenciamento de investimentos *Bullseye*, assim como suas aplicações. O desenvolvimento desse *software* respeitou a metodologia apresentada no capítulo 3, buscando cumprir todos os requisitos propostos e apresentados no quadro 2.

A Figura 14, mostra a primeira tela apresentada pelo sistema – a tela de login –, na qual o usuário tem a opção de acessar o sistema caso já possua um cadastro, apenas inserindo seu Cadastro de Pessoa Física (CPF) juntamente com sua senha de acesso; ou, se ainda não possuir um cadastro, ele poderá fazê-lo através do botão NOVO USUÁRIO que irá direcioná-lo para a tela de cadastro de usuário (Figura 15).

Figura 14. Tela Login



Gerenciamento de Carteira

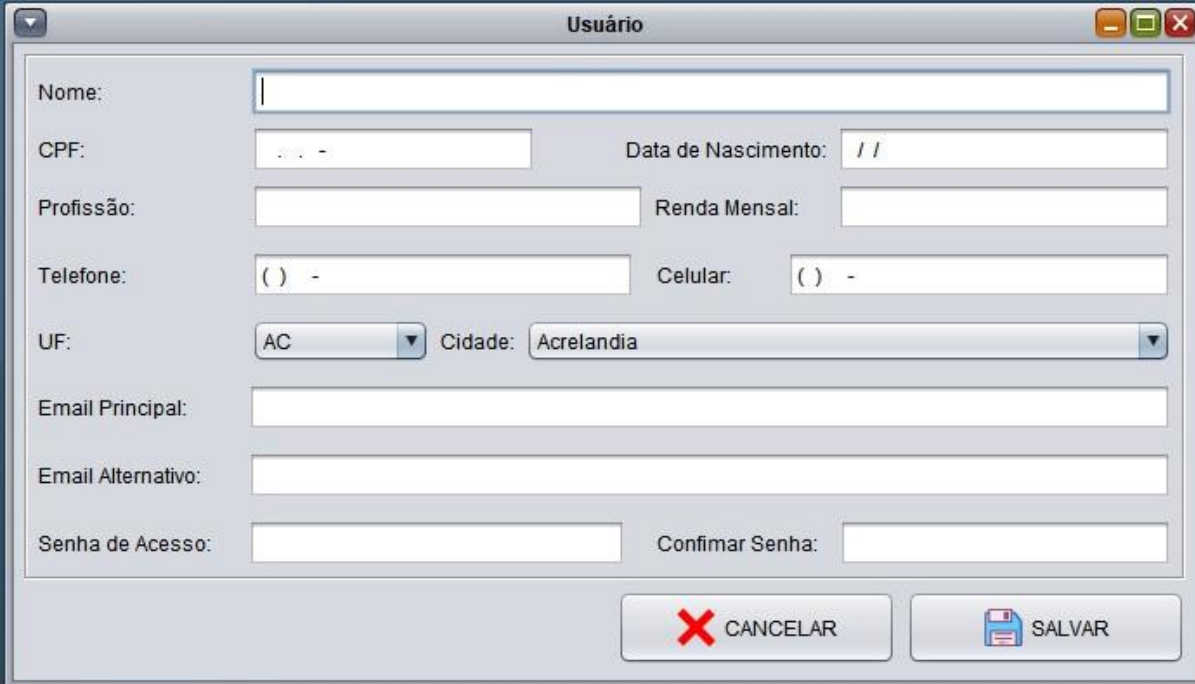
Bullseye

Digite o seu cpf:

Digite a sua senha:

Fonte: Autoria própria

Figura 15. Tela de Cadastro de Usuário



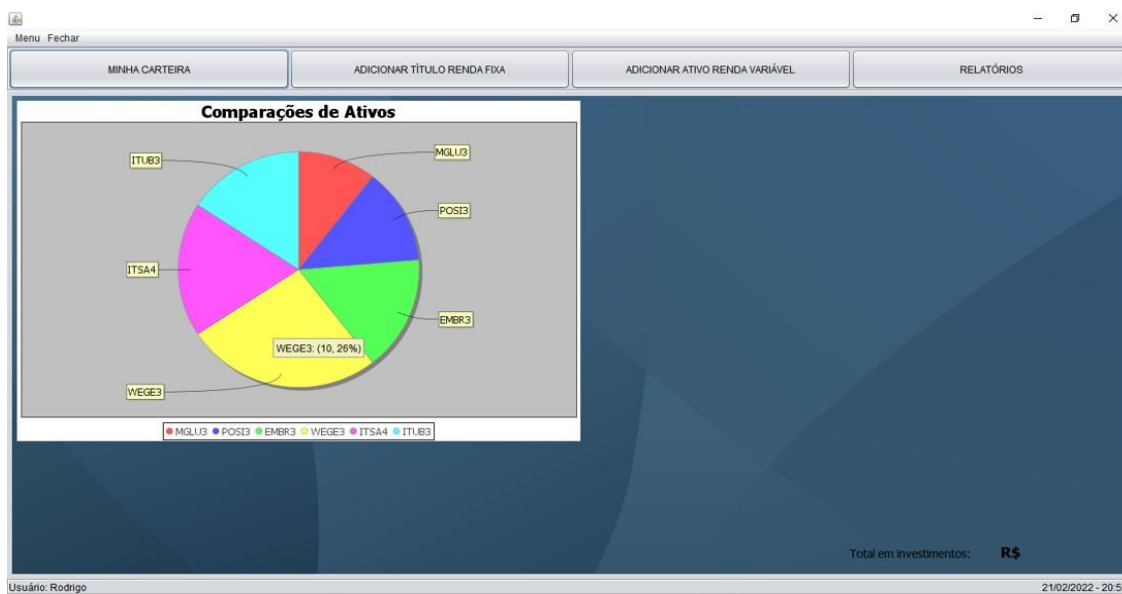
A screenshot of a user registration window titled "Usuário". The window contains several input fields for user information: "Nome:" (Name), "CPF:" (CPF), "Data de Nascimento:" (Date of Birth), "Profissão:" (Profession), "Renda Mensal:" (Monthly Income), "Telefone:" (Phone), "Celular:" (Cellular), "UF:" (State) with a dropdown menu showing "AC", "Cidade:" (City) with a dropdown menu showing "Acrelandia", "Email Principal:" (Main Email), "Email Alternativo:" (Alternative Email), "Senha de Acesso:" (Access Password), and "Confirmar Senha:" (Confirm Password). At the bottom, there are two buttons: "CANCELAR" (Cancel) with a red 'X' icon and "SALVAR" (Save) with a floppy disk icon.

Fonte: Autoria própria

A tela para realizar o cadastro dos usuários será a mesma para alteração de dados do usuário logado no *software* caso houver necessidade.

A tela principal, apresentada na Figura 16, possui um gráfico no qual será mostrado a quantidade de ativos existente no portfólio do usuário e também um menu com todas as funcionalidades disponíveis, sendo que para as principais existem botões de atalho na própria tela principal.

Figura 16. Tela Principal



Fonte: Autoria própria

A Figura 17 apresenta a tela MINHA CARTEIRA, onde o usuário conseguirá visualizar todos os ativos alocados em seu portfólio e terá a opção de adicionar outros ou remover algum existente.

Figura 17. Tela Minha Carteira

The screenshot shows the "Minha Carteira" window with two tabs: RENDA VARIÁVEL (selected) and RENDA FIXA. Below the tabs is a table listing assets:

Ticker	Empresa	Nº de Ações	Preço Médio
POSI3	Positivo	5	8,62
EMBR3	Embraer	6	18,35
WEGE3	WEG	10	29,52
ITSA4	Itausa	4	10,21
ITUB3	Itau	6	22,09

At the bottom of the window, summary statistics are displayed: Valor total renda fixa: R\$ 00,00; Valor total variável fixa: R\$ 621,78; Valor total da carteira: R\$ 621,78. Below these statistics are three buttons: CANCELAR (with a red X icon), REMOVER ATIVO (with a trash icon), and ADICIONAR ATIVO (with a plus icon).

Fonte: Autoria própria

Para adicionar investimentos de renda fixa em sua carteira, o usuário precisará ir até a tela ADICIONAR TÍTULO RENDA FIXA, e preencher o que se pede (Figura

18), caso não encontre o título que procura, terá que cadastrar manualmente esse título, através do botão CRIAR TÍTULO, que irá lhe redirecionar para a tela de criação do título apresentada pela Figura 19.

Figura 18. Adicionar Título Renda Fixa

ADICIONAR TÍTULO RENDA FIXA

Pesquisar: Procure o ativo de renda fixa que deseja.

Nome:

Instituição Detentora: Tipo:

Taxa de Juros: Valor da Unidade:

Data de Compra: Quantidade:

Vencimento: Carência:

Fonte: Autoria própria.

Figura 19. Criar Título Renda Fixa

RENDA FIXA

Nome do Ativo:

Tipo: Pré-Fixado Pós-Fixado

Instituição Emissora:

Pré-Fixado

Taxa:

Vencimento

Carência

Pós-Fixado

IPCA +

Vencimento

Carência

Fonte: Autoria própria

Para os ativos de renda variável o procedimento é semelhante: o usuário vai até a tela ADICIONAR ATIVO RENDA VARIÁVEL (Figura 20), para localizar o ativo. Caso não o encontre, terá que cadastrá-lo manualmente através do botão CRIAR ATIVO, o qual irá lhe redirecionar para a tela de criação do ativo desejado (Figura 21).

Figura 20. Adicionar Ativo Renda Variável

A interface 'ADICIONAR ATIVO RENDA VARIÁVEL' apresenta um formulário com os seguintes campos e botões:

- Barra de pesquisa: 'Pesquisar:' seguida de um campo de texto e um botão 'Pesquisar'.
- Botão 'Criar ativo' localizado à direita da barra de pesquisa.
- Campos de entrada: 'Empresa:', 'Ticker:', 'Valor da Unid...'
- Campos de entrada: 'Data de Compra:' (contendo '//') e 'Quantidade:'.
- Botões de ação: 'CANCELAR' (com ícone de X vermelho) e 'SALVAR NA CARTEIRA' (com ícone de disquete).

Fonte: Autoria própria

Figura 21. Criar Ativo Renda Variável

A interface 'Cadastrar Ativo' apresenta um formulário com os seguintes campos e botões:

- Barra de pesquisa: 'Pesquisar:' seguida de um campo de texto e um botão 'Pesquisar'.
- Campos de entrada: 'Empresa:', 'Ticker:', 'Setor:' (com o valor 'Bens industriais' selecionado).
- Botões de ação: 'CANCELAR' (com ícone de X vermelho), 'EDITAR' (com ícone de lápis) e 'SALVAR' (com ícone de disquete).

Fonte: Autoria própria

5 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho teve por objetivo a criação de um *software* para o gerenciamento de carteiras de investimento tendo como público alvo os investidores que estejam iniciando no mercado financeiro, auxiliando-os a terem uma visualização simples de cada ativo dentro do seu portfólio, ou seja, apresentando a parcela que cada investimento representa dentro de sua carteira. Com base neste contexto foi desenvolvido o *software* de gerenciamento de investimentos *Bullseye*. O resultado final obtido pode ser considerado bastante satisfatório, principalmente pela interface simples do aplicativo que contribui para torná-lo bem intuitivo.

Todas as metodologias e ferramentas utilizadas, se mostraram muito eficazes quando utilizadas de maneira conjunta, o que foi de suma importância para o desenvolvimento do trabalho, que transcorreu de forma harmoniosa, possibilitando o cumprimento de todos os requisitos estabelecidos.

Para trabalhos futuros pretende-se implementar, a emissão de relatórios e principalmente, uma opção para que o investidor possa importar sua carteira de investimento diretamente de sua corretora, para que não haja a necessidade da criação de ativos. Acredita-se que com essa implementação o *software* ganhará maior visibilidade e se tornará mais ágil.

REFERÊNCIAS

ALMEIDA, Carla Cássia de Jesus *et al.* **Qualitas**: um modelo de processo de desenvolvimento de software orientado a modelos. 2014.

BERTÃO, Naiara. VALOR INVESTE. **Número de pessoas físicas na bolsa cresce ainda mais em agosto e encosta em 3 milhões**. Disponível em: <https://valorinveste.globo.com/mercados/renda-variavel/noticia/2020/09/02/numero-de-pessoas-fisicas-na-bolsa-cresce-ainda-mais-em-agosto-e-encosta-em-3-milhoes.ghtml>. Acesso em: 6 jul. 2021.

BODIE, Zvi; KANE, Alex; MARCUS, Alan J. **Fundamentos de investimentos**. 9. ed. [S. l.]: AMGH Editora Ltda, 2014.

CAMARGO, Robson. Gerenciamento de Projetos. **Robson Camargo**. [S, l.], p. 1-15, 29 nov. 2019. Disponível em: <https://robsoncamargo.com.br/metodocascata>. Acesso em: 17 jan. 2022.

CAMPOS, Álvaro. Bolsa atinge 3,7 milhões de investidores, alta de 54% no ano, **VALOR INVESTE**. Disponível em: <https://valorinveste.globo.com/mercados/renda-variavel/bolsas-e-indices/noticia/2021/05/13/bolsa-atinge-37-milhoes-de-investidores-alta-de-54percent-no-ano.ghtml>. Acesso em: 15 jul. 2021.

DEVMEDIA, **introdução aos processos de software e o modelo incremental e evolucionário**. Disponível em: <https://www.devmedia.com.br/introducao-aos-processos-de-software-e-o-modelo-incremental-e-evolucionario/29839> Acesso em: 12 jan. 2022.

DEVMEDIA, **Introdução ao padrão MVC**. Disponível em: <https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308#:~:text=Entre%20as%20diversas%20vantagens%20do,do%20c%C3%B3digo%20sempre%20limpo%20etc.> Acesso em: 13 jan. 2022.

EXPERTXP, **Taxa de juros real**: o que é e como calcular. Disponível em: <https://conteudos.xpi.com.br/aprenda-a-investir/relatorios/juros-real/>. Acesso: 12 jan. 2022.

FARIAS, Virgínia; MOREIRA, Carla; COUTINHO, Emanuel; SANTOS, Ismayle. **iTest learning**: um jogo para o ensino do planejamento de testes de software. *In* Fórum de Educação em Engenharia de Software. Simpósio Brasileiro de Engenharia de Software. 2012.

GUEDES, G. T. A. (2018) **UML 2**: uma Abordagem Prática. 3. ed. Rio de Janeiro: Novatec.

HAZAN, C. *et al.* Roteiro SERPRO de contagem de pontos de função e

estimativas. Serviço Federal de Processamento de Dados.

JACONELI, Matheus. NOVA FUTURA INVESTIMENTOS. **A evolução das mulheres no mercado financeiro brasileiro**. Disponível em: <https://www.novafutura.com.br/a-evolucao-das-mulheres-no-mercado-financeiro-brasileiro/>. Acesso em: 20 nov. 2021.

LAENDER, A. H. F.; FLYNN, D. J., **A semantic comparison of modeling capabilities of the ER and NIAM models**. In: ELMASRI, R.; KOURAMAJIAN, V.; THALHEIM, B. (eds.). Entity-Relationship Approach - ER'93, Springer-Verlag, p. 242-256. 1993.

LEMOS, Maxmilian Ferreira de; OLIVEIRA, Patrícia Carvalho de; RUELA, Leandro César; SANTOS, Matheus da Silva; SILVEIRA, Thallis Carvalho; REIS, José Cláudio de Sousa. Aplicabilidade da arquitetura mvc em uma aplicação web (webapps). **Revista Eletrônica Científica de Ciência da Computação**, 2013. Disponível em: <http://revistas.unifenas.br/index.php/RE3C/article/view/54/13> Acesso em: 25 fev. 2021.

LUCIANO, J., ALVES, W. J. B. **Padrão de ArquiteturaMVC: Model-View-Controller**.

MAIA, Regina Maria da Costa Smith. **Inter Complementação do sistema categorial de Ranganathan e do modelo entidade-relacionamento de Chen para a modelagem conceitual**: uma aplicação no domínio do biomonitoramento do Projeto Manuelzão. UFMG nas águas da Bacia do Rio das Velhas. 2013.

MYERS, G. **The Art of Software Testing**. 2. ed. John Wiley & Sons,

NUNES, Melissa. IDINHEIRO. **KINVO vale a pena? Conheça as funcionalidades do app!** Disponível em: <https://www.idinheiro.com.br/kinvo-vale-a-pena/>. Acesso em: 12 set. 2021.

NUNES, Melissa. IDINHEIRO. **TRADEMAP gratuito vale a pena? É seguro? Confira a análise completa!** Disponível em: <https://www.idinheiro.com.br/trademap/>. Acesso em: 12 set. 2021.

PFLEEGER, S. L. **Software Engineering: Theory and Practice**. 2. ed. Upper Saddle River, New Jersey: Prentice-Hall, 2001.

PRESSMAN, Roger S. **Engenharia de software: Uma abordagem profissional**. 7. ed. São Paulo: McGraw-Hill, 2011.

PRESSMAN, Roger S. PENTEADO, Rosângela Delloso (trad.). **Engenharia de software**. 6. ed. Rio de Janeiro: McGrawHill, 2006.

QUEIROZ, Rodrigo de Campos. **Dilema do pequeno investidor**. 2021. Dissertação (Mestrado em Economia) - Fundação Getúlio Vargas Escola de Economia de São Paulo, São Paulo, 2021.

ROB, P.; CORONEL, C. **Sistemas de banco de dados**. São Paulo: Cengage Learning, 2011.

RUMBAUGH, J.; BOOCH, G.; JACOBSON, I. **UML: guia do usuário**. Rio de Janeiro: Editora Campus Ltda, 2000.

RYNGELBLUM, Ivan. Montante aportado por pessoas físicas na bolsa volta a subir em março. **Seu Dinheiro**. Disponível em: <https://www.seudinheiro.com/2021/bolsa-dolar/montante-aportado-por-pessoas-fisicas-na-bolsa-volta-a-subir-em-marco/>. Acesso em: 19 ago. 2021.

SALLES, André Assis de; LIU, Adriana; RODRIGUES, Caroline Xavier de Abreu. **O risco idiossincrático e o risco sistemático de mercados emergentes: evidências acerca do bloco BRIC**. In: XXIX ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO. [S. l.], p. 1-12.

SOMMERVILLE, Ian. **Engenharia de Software**. 8 ed. São Paulo. Pearson Addison-Wesley, 2007.

TAKAI, Osvaldo Kotaro, ITALIANO Isabel Cristina e FERREIRA João Eduardo. **Introdução a banco de dados**. São Paulo: Eletrônica, 2005.

ZANOTELLI, Márcia Virginia Almeida. **Uma análise do perfil conservador do investidor brasileiro pessoa física**. 2021. 63f. Dissertação (Mestrado em Economia de Empresas) - Faculdade de Economia, Administração, Atuária e Contabilidade - FEAAC, Programa de Economia Profissional - PEP, Universidade Federal do Ceará - UFC, Fortaleza (CE), 2021.

APÊNDICE A - Trecho do código implementado na camada *MODEL*

```
public class TipoAtivoVariado {
    private int codigo;
    private String empresa;
    private String ticker;
    private Setor setor;

    public int getCodigo() {
        return codigo;
    }

    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }

    public String getEmpresa() {
        return empresa;
    }

    public void setEmpresa(String empresa) {
        this.empresa = empresa;
    }
}
```

APÊNDICE B - Trecho do código implementado na camada VIEW.

```
private void jBSalvarActionPerformed(java.awt.event.ActionEvent evt) {  
    TipoAtivoVariado ativoVariado = new TipoAtivoVariado();  
  
    ativoVariado.setEmpresa(jTFEmpresa.getText());  
    ativoVariado.setTicker(jTFTicker.getText());  
  
    SetorController controller = new SetorController();  
  
    Setor setr = controller.buscarSetor(jCBSetor.getSelectedItem().toString());  
  
    ativoVariado.setSetor(setr);  
  
    if(jTFEmpresa.getText().equals("")){  
        JOptionPane.showMessageDialog(null, "O campo EMPRESA não pode ficar vazio.");  
        jTFEmpresa.setFocusable(true);  
    }else if(jTFTicker.getText().equals("")){  
        JOptionPane.showMessageDialog(null, "O campo TICKER não pode ficar vazio.");  
        jTFTicker.setFocusable(true);  
    }else{  
        TipoAtivoVariadoController controllerAtivoVariado = new TipoAtivoVariadoController();  
  
        controllerAtivoVariado.inserirTipoAtivoVariado(ativoVariado);  
  
        TelaAddRendaVariavel telaRV = new TelaAddRendaVariavel(idUser);  
        TelaPrincipal.jDPPrincipal.add(telaRV);  
        telaRV.setVisible(true);  
  
        this.dispose();  
    }  
}
```

APÊNDICE C - Trecho do código implementado na camada CONTROLLER.

```
public class TipoAtivoVariadoController {  
  
    DAOAtivoVariado dao;  
  
    public TipoAtivoVariadoController() {  
        dao = new DAOAtivoVariado();  
    }  
  
    public void inserirTipoAtivoVariado (TipoAtivoVariado AtivoVariado) {  
        dao.inserirTipoAtivoVariado (AtivoVariado);  
    }  
  
    public TipoAtivoVariado buscarAtivoVariado(int codigo) {  
        return dao.buscarTipoAtivoVariado(codigo);  
    }  
  
    public ArrayList<TipoAtivoVariado> buscarTipoAtivoVariados(String tipo) {  
        return dao.buscarTiposAtivoVariado(tipo);  
    }  
  
    public int getIdTipoAtivoVariado(String empresa, String ticker) {  
        return dao.getIdTipoAtivoVariado(empresa, ticker);  
    }  
  
    //pesquisar por nome da empresa ou ticker  
    public TipoAtivoVariado buscarAtivoVariado(String nome) {  
        return dao.buscarAtivoVariado(nome);  
    }  
}
```