UNIVERSIDADE DO VALE DO TAQUARI
CURSO DE ENGENHARIA DE SOFTWARE

DISPONIBILIZAÇÃO DE DADOS DE FUNDOS DE INVESTIMENTOS ATRAVÉS DE UMA API

Andrei Cioqueta Mileto

Lajeado, junho de 2021

‘

Andrei Cioqueta Mileto

DISPONIBILIZAÇÃO DE DADOS DE FUNDOS DE INVESTIMENTOS ATRAVÉS DE UMA API

Projeto de Monografia apresentado na disciplina de Trabalho de Conclusão de Curso I, do curso de Engenharia de Software, da Universidade do Vale do Taquari - Univates, como parte da exigência para a obtenção do título de bacharel em Engenharia de Software.

Orientador: Evandro Franzen.

Lajeado, junho de 2021

‘

RESUMO

Investir através de plataformas online se tornou uma facilidade, gerando um maior interesse de consultas e análises de dados na busca de lucratividade e maior retorno. Devido a essa demanda no mercado há uma diversidade de plataformas que auxiliam na análise de rendimentos de aplicações, mas quando se trata de um grande número de dados, a disponibilização torna-se difícil por ter uma rentabilidade histórica. A história, apesar de não dar nenhuma segurança futura, é uma base para compreender a integridade e confiabilidade dos fundos de investimentos, tanto quanto os dados atuais. A manipulação de uma grande quantia de registros, devido às atualizações diárias da fonte de origem: o site do governo, torna-se um desafio, que para sua viabilização demandou o desenvolvimento de duas aplicações, em que a primeira é responsável pela busca de informações, validações, tratamentos e inserções dos dados, e a segunda pela disponibilização de registros através de uma API que possibilita à *softwares* terceiros a busca de informações em menos de vinte ms com uma base de dados que contém 4491242 registros só no ano de 2020.

Palavras-chave: Fundo de investimento, API, Banco de dados, Volume de dados.

ABSTRACT

Investing through online platforms has become an ease, generating greater interest in queries and data analysis in search of profitability and greater returns. This demand in the market requires a variety of platforms that help in the analysis of application applications, but when it comes to a large amount of data, an availability becomes difficult because of its historical profitability. History, while giving no future, is a basis for understanding the trust and confidence of investment funds as much as current data. The handling of a large amount of records, due to daily updates from the source of origin: the government website, becomes a challenge, which required the development of two applications for its feasibility, in which the first is responsible for searching for information , data validations, treatments and insertions, and the second by the availability of records through an API that allows third-party software to search for information in less than twenty ms with a database that contains 4491242 records in the year 2020 alone.

Keywords: Investment fund, API, Database, Data volume.

‘

LISTA DE FIGURAS

Figura 1 - Ilustração do processo de ETL 28
Figura 2 - Gerações de ETL 29
Figura 3 - Ciclo da Curadoria Digital 31
Figura 4 - Índice B-Tree 35
Figura 5 - Índice Bitmap 36
Figura 6 - REST API Architecture 38
Figura 7- Relação entre recursos URIs 39
Figura 8 - Relação URI única 40
Figura 9 - Fluxograma de requisições 41
Figura 10 - Exemplo de JSON 42
Figura 11 - Exemplo de JSON 42
Figura 12 - Gerenciamento de SUAP 44
Figura 13 - Extração dos dados através de uma API do SUAP 45
Figura 14 - União das vantagens dos cenários 46
Figura 15 - Análise de dados Big Data 47
Figura 16 - Top Down e Bottom Up 49
Figura 17 - BDS API Function. 51
Figura 18 - Sistema de comunicação usando BDS-API 52
Figura 19 - Fluxograma sentenças de linguagem 60
Figura 20 - Diagrama de processo 61
Figura 21 - Dados diários CVM 62
Figura 22 - Resultado dados fundos de investimento 63

Figura 23 - Modelo banco de dados 65
Figura 24 - Cadastros duplicados cancelados CNPJ 01.603.578/0001-03 68
Figura 25 - Cadastros duplicados gestores CNPJ 01.392.020/0001-18 68
Figura 26 - Quotas de fundos de Janeiro/2020 69
Figura 27 - Quotas de fundos de Abril/2020 69
Figura 28 - Inserção cadastro de fundos 71
Figura 29 - Cadastro Quotas Fundos 73
Figura 30 - Consulta SQL Aplicação API-REST 77
Figura 31 - Chamada API via Postman 77
Figura 32 - Consulta através do banco de dados 78
Figura 33 - Consulta através da API após criação de índice. 79
Figura 34 - Consulta através do banco de dados após criação de índice 79

LISTA DE TABELAS

Tabela 1 - Classificação de Fundos ANBIMA 21
Tabela 2 - Comparativos Trabalhos Relacionados 54

‘

LISTA DE GRÁFICOS

Gráfico 1 - Histórico de registros do ano de 2020 74
Gráfico 2 - Histórico de tempo de inserção do ano de 2020 75
Gráfico 3 - Comparativo de quantidade de registros x tempo de inserção - Dados de 2020 76

‘

LISTA DE ABREVIATURAS

API	Application Programming Interface
BACEN	Banco Central do Brasil
BD	Banco de dados
BI	Business Intelligence
CDB	Certificado de Depósito Bancário
CKAN	Comprehensive Knowledge Archive Network
CPF	Cadastro de pessoa física
DM	Data Marts
DW	Data warehouse
ETL	Extract, Transform and Load
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IGPM	Índice Geral de Preços do Mercado
IPCA	Índice de Preços ao Consumidor Amplo
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
PDF	Portable Document Format
REST	Representational State Transfer
SDW	Sistema de data warehouse
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	Structured Query Language
SELIC	Sistema Especial de Liquidação e de Custódia

TR	Taxa Referencial Mensal
UML	Unified Modeling Language
URL	Uniform Resource Locator
URI	Uniform Resource Identifier
SOAP	Simple Object Access Protocol
SPI	Serviço Interface do Provedor
SUAP	Sistema Unificado de Administração Pública
XML	Extensible Markup Language
WSDL	Web Services Description Language

SUMÁRIO

1 INTRODUÇÃO 13

1.1 Problema de pesquisa e/ou hipótese 15

1.2 Objetivos 16

1.3 Objetivos Específicos 16

1.4 Justificativa da Pesquisa 16

1.5 Estrutura do trabalho 17

2 FUNDAMENTAÇÃO TEÓRICA 19

2.1 Investimento 19

2.1.1 Fundos de Investimento 20

2.1.2 Caderneta de Poupança 23

2.1.3 Certificado de depósito bancário (CDB) 23

2.1.4 Títulos públicos 24

2.1.5 Análise de investimento 24

2.1.6 Identificando o perfil de investidor 24

2.1.7 Variáveis do investimento 25

2.2 Extração de Dados (ETL, BI) 25

2.3 *Big Data* 29

2.4 Consultas banco de dados relacionais 33
2.4.1 Índice 34
2.4.2 Índice B-tree 35
2.4.3 Índice Bitmap 35
2.4.4 Utilização de *functions*, *stored procedures* e *package* 36
2.4.7 *View* 36
2.4.7 View Materializada 37
2.5 APIs e Web Services 37
2.5.1 JSON 41
3 TRABALHOS RELACIONADOS 43
3.1 Open Data Processor: Uma ferramenta para a automatização do processo de extração e publicação de Dados Abertos 43
3.2 Inovação dos negócios por meio da análise de Big Data. 46
3.3 Fundamentos e aplicação do Big Data: como tratar informações em uma sociedade de yottabytes. 48
3.4 Aumentando a acessibilidade aos sistemas de Big Data por meio de um API de serviços comuns. 50
3.5 Modelo UML de uma API padrão para aplicativo de computação em desenvolvimento em nuvem 52
4 MATERIAIS E MÉTODOS 55
4.1 Pesquisa quanto aos métodos científicos 55
4.1.1 Pesquisa quanto à Natureza 55
4.1.2 Pesquisa quanto a Abordagem 56
4.1.3 Pesquisa quanto aos Objetivos 56
4.1.4 Pesquisa quanto aos Procedimentos Técnicos 57
4.2 Tecnologias 58
4.2.1 Banco de dados PostgreSQL 58
4.2.2 Linguagem de programação Java 59

4.2.3 Postman 60
4.3 Desenvolvimento 61
4.3.1 Consultas através da API 62
4.4 Testes e análise dos resultados 63
5. RESULTADOS PRELIMINARES OU ESPERADOS 67
5.1 Análise dos resultados 67
5.1.1 Dados de inserção cadastro de Fundos de Investimentos 67
5.1.2 Dados de Inserção Quotas Fundos 68
5.1.3 Gerador FI - Cadastro de Fundos 70
5.1.4 Gerador FI - Cadastro Quotas Fundos 72
5.1.5 Chamada API 76
6 CONCLUSÕES 80
REFERÊNCIAS 81

1 INTRODUÇÃO

Investimento é qualquer despesa ou aplicação de recursos que irá lhe trazer um retorno futuro, um desembolso em que se tem expectativa de certo resultado, no qual é necessário investir tempo, energia, estudos e atenção no seu acompanhamento. Na área de finanças, define-se como uma aplicação de dinheiro na espera de produzir certos rendimentos, que se torna possível por conta do efeito dos juros compostos, que fazem com que o dinheiro se multiplique, em outras palavras, é o que acontece em sua poupança do banco.

No cotidiano não podemos evitar os gastos imprevistos, como despesas de última hora com saúde, manutenção de carro ou desemprego. Para que possa estar preparado para uma situação dessas, deve-se criar um fundo de emergência, seja com o propósito para uma garantia de orçamento familiar como também alcançar novos objetivos. Segundo Research XP 2020, para calcular um fundo de emergência, deve-se estimar os valores fixos gastos em um mês e multiplicar por seis, que seriam os meses de gastos que haveria garantia para evitar futuras dívidas.

O conceito de investimento envolve tanto dinheiro, como capital intelectual, social ou natural. Para investir não é obrigatório ser um especialista em finanças, mas é importante ter conhecimentos básicos do que é investimento, conforme Wisniewski 2011, pois com todo marketing aplicado na internet e redes sociais, influenciando as pessoas a se tornarem investidores, as opções tornaram-se cada vez mais fáceis, visíveis e práticas, além de contatos de corretores que auxiliam em melhores aplicações para que as pessoas leigas façam a melhor escolha de acordo a segurança que necessita.

Uma corretora de valores serve para intermediar compras e vendas de títulos financeiros, possibilitando fazer aportes em Fundos de Investimentos, ações e outros. Tem-se a disponibilidade para investimento em diversas corretoras como também de exclusividade em uma única. Porém, muitas vezes torna-se necessário ter aplicações em mais de uma corretora, devido a disponibilidade dos investimentos de cada empresa.

Conforme Nascimento 2015, os fundos de investimento são um conjunto de investidores, que reúnem os recursos de várias pessoas para que possam ser aplicados em conjunto no mercado financeiro e de capitais. Possuem alta quantidade de aplicações financeiras disponíveis, nos quais podem-se escolher o mais adequada com o perfil dos investidores, como, conservador (prioriza as em suas aplicações a segurança, contudo traz um rendimento de menor retorno), moderador (preserva a segurança, porém, possui tolerância de riscos à longo prazo), arrojado ou agressivo (entende que as perdas a curto prazo são momentâneas e necessárias para aproveitar lucros mais altos a longo prazo).

Os rendimentos de baixo retorno, são uma categoria utilizada por investidores mais conservadores devido à segurança. Conforme o Jornal Contábil, obteve no ano de 2020 sua maior queda histórica para 2% ao ano, ao mesmo tempo que a poupança de acordo com a revista Infomoney, obteve um rendimento anual que é de 70% da taxa Selic + TR (Taxa Referencial Mensal). Enquanto a Selic encontra-se abaixo de 8,5% ao ano, observa-se que desde 2018 a TR encontrou-se zerada segundo BACEN (Banco Central do Brasil), o que totalizou um rendimento na poupança de 1,4% ao ano.

Devido a essas condições econômicas muitas pessoas optam por aplicações de maior retorno, como os fundos de investimentos, entre eles os Fundos Alaska Black Institucional FIA que conforme Alaska Asset Management 2021, teve um rendimento anual de 31,12% em 2019 e a Icatu Vanguarda Pré Fixado FIRF LP que teve seu rendimento anual de 16,91% no mesmo ano, segundo relatório divulgados por Icatu Vanguarda 2021.

Para um maior controle estatístico desses tipos de investimentos que possuem diariamente grande variação, utiliza-se planilhas ou softwares de gerenciamento financeiro pessoal, que auxiliam de forma automatizada ao mostrar os rendimentos de

suas aplicações, o que facilita nas tomadas de decisões no dia a dia. Os aplicativos disponíveis na internet, Kinvo e Real Valor, que possuem alta recomendação por investidores experientes, serviram como base de testes a fim de avaliar a necessidade de possíveis melhorias de visualização sobre o rendimento histórico de sua carteira de investimento, atualização automática do valor da cota de cada fundo de forma diária e histórica e o detalhamento individual de cada uma das aplicações.

Para que se possa ter a atualização dos dados diariamente de forma automática, utiliza-se como base API (*Application Programming Interface* ou Interface de Programação de Aplicativos), que são utilizados para conectar diferentes softwares, proporcionam integração entre sistemas de diferentes linguagens de programação de forma ágil e segura. Uma API disponibiliza informações que originam de um banco de dados para que outros softwares possam consultar estas informações disponibilizadas.

Para que seja possível disponibilizar os dados através da API, é preciso encontrar os dados em alguma fonte confiável e segura. Após consultar as informações através da fonte, será preciso inserir no banco de dados para que a API tenha sua eficiência de disponibilizar os dados de acordo com as requisições.

1.1 Problema de pesquisa e/ou hipótese

Ao considerar o contexto descrito surge a questão principal a ser investigada neste trabalho: É possível disponibilizar fundos de investimentos com informações diárias e históricas através de uma API e que se mantenha uma alta performance das consultas e automatização de atualizações diárias dos dados?

1.2 Objetivos

O objetivo principal do trabalho é disponibilizar uma API com informações de fundo de investimentos, com um diferencial por disponibilizar informações históricas. Para que isso seja possível, realizou-se um estudo sobre manipulação de dados no Postgres para que a API tenha uma alta performance e eficiência dos dados.

1.3 Objetivos Específicos

São objetivos específicos desta pesquisa:

- Estudar os principais recursos do banco de dados Postgres;
- Pesquisar recursos e fundamentos sobre o desenvolvimento de APIs para possibilitar a implementação das funcionalidades necessárias;
- Estudar e selecionar fontes confiáveis para consultar os dados de fundos de investimentos;
- Obter conhecimento sobre extração dos dados relacionado a históricos de transações de fundos de investimentos;
- Estudar sobre otimização de bancos de dados e aplicar os mesmos na implementação das funcionalidades da API;
- Realizar testes de desempenho e a validação das funcionalidades da API desenvolvida

1.4 Justificativa da Pesquisa

As aplicações de fundos de investimentos estão cada vez mais simples de se fazer e devido a grande oscilação do mercado e variedade de investimento torna-se de extrema importância o monitoramento dos rendimentos, para ter um maior controle de como estão suas aplicações de acordo com a realidade do mercado. Observa-se que quando trabalhamos com mais de uma corretora ou banco ao mesmo tempo

temos a impossibilidade de analisar de forma rápida a carteira total com todos os investimentos, o que limita o gerenciamento das aplicações por não existir uma integração entre as informações de diferentes fontes, fato que ocorre quando se utiliza plataformas online e controle via planilhas para analisar essas aplicações. Cria-se a necessidade de maiores informações como rentabilidade histórica e rendimentos no caso de ter mais de uma aplicação no mesmo mês, o que torna o valor médio da cota diferente comparado ao de compra.

Observa-se que na atualidade existem diversas APIs ou sistemas que disponibilizam o valor atual de uma cota de fundo de investimento para que se possa fazer integrações com *softwares* terceiros, mas essas informações são limitadas em relação à consulta de dados históricos.

A disponibilidade de uma API tem o grande problema com informações históricas dos fundos de investimentos, devido aos registros serem atualizados diariamente, em conjunto a isso uma carteira de fundos extremamente grande, o que gera uma quantidade de centenas de milhares de registros diários, que demanda a necessidade de uma base de dados sólida além de ser de extrema importância a garantia de que os dados sejam disponibilizados de forma rápida e eficaz. Além disso, o trabalho irá contribuir com conhecimentos sobre o mercado financeiro, com abrangência sobre diversos tipos de investimentos, mas com o enfoque principal sobre a área de fundos de investimentos.

1.5 Estrutura do trabalho

Buscando de forma correta o embasamento para o desenvolvimento deste trabalho, os capítulos foram organizados conforme descrito a seguir.

O primeiro capítulo faz uma introdução sobre a necessidade de analisar os investimentos buscando uma maior rentabilidade, o que gera um grande volume de dados para ter informações históricas de todos os fundos de investimentos. Além disso, também é definido os objetivos do trabalho.

O segundo capítulo faz uma exploração em referências teóricas já existentes, relacionadas com os tópicos abordados no trabalho. Dentre os assuntos, estão relacionados Investimentos, *Big Data*, Extração de dados, Consultas de dados e API.

O terceiro capítulo fala sobre os trabalhos relacionados ao assunto abordado no trabalho, em cada um dele mostra um problema diferente com assuntos paralelos, mas com o tema principal equivalente ao do trabalho apresentado.

No quarto capítulo, apresenta a metodologia utilizada para o estudo, mostra quais métodos foram utilizados e explica detalhadamente o que significa cada método.

O quinto capítulo, mostra os resultados obtidos a partir da execução da aplicação desenvolvida através do estudo feito sobre o tema do trabalho.

O sexto capítulo apresenta as considerações finais deste estudo, analisando todos os resultados alcançados como também as expectativas dos resultados.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apresentar os conceitos relacionados ao trabalho, com base em informações de pesquisa bibliográfica. Inicialmente apresenta-se o conceito de investimentos e seus tipos, conceito de bancos de dados e *Big Data*, que são responsáveis pela parte de armazenamento dos dados. Após tem-se a caracterização e funcionamento da extração de dados e por fim é abordada a disponibilização de dados através de uma API.

2.1 Investimento

Uma das preocupações que o governo tem interesse é o consumo da sociedade e o endividamento consequente das famílias. De acordo com o artigo descrito por Wisniewski (2011), a falta de controle no orçamento financeiro está ligada na maioria das vezes com carência de conhecimentos e de planejamentos financeiro, uma vez que essa falta de informações prejudica a saúde financeira tanto dos adultos que acabam por realizar compras compulsivamente, como também afeta as crianças e jovens que são envolvidos pelas publicidades e acabam por adquirir cada vez mais produtos, o que agrava a saúde financeira familiar.

A ausência da saúde financeira resulta em impactos na qualidade de vida dos consumidores, que devido às dívidas desenvolvem sintomas de estresse, insônia, depressão, entre outros, posto que conforme a pesquisa feita pela Confederação

Nacional do Comércio de Bens, Serviços e Turismo - CNC (2011), cerca de 64% das famílias brasileiras declararam-se como endividadas.

Em vista disso observa-se que o planejamento financeiro é um aspecto muito importante para o equilíbrio familiar, que em conformidade com Nascimento (2015) é necessário mapear os caminhos a serem guiados, coordenar e controlar as ações para atingir seus objetivos, além de formalizar métodos pelas quais as metas devem ser alcançadas. Com esse objetivo deve-se poder se questionar sobre três temas relevantes de como aproveitar as oportunidades de investimentos que o mercado propõe, identificar o grau de endividamento aceitável e determinar a parcela dos lucros aferidos.

O processo de planejamento financeiro inicia com planos financeiros a longo prazo, segundo o mesmo autor, sendo feitos com base em períodos de dois a dez anos ou estratégicos, que guiam a formulação de planos para curto prazo que tem como base em períodos entre um e dois anos.

A captação de recursos no ambiente pessoal para aplicação em ativos busca a geração de dinheiro, rendimento para o bolso, através de aplicações que geram renda e aumento de patrimônio, nas quais não entram imóveis para uso pessoal, mas incluem imóveis em formato de aplicações em ações, investimento, títulos públicos, CDBs, poupança e entre outros. Ainda em concordância com Nascimento (2015) no mercado financeiro existe um leque de produtos financeiros que você tem acesso, sendo elas através de bancos ou corretoras. A seguir serão apresentadas algumas das principais formas de investimento.

2.1.1 Fundos de Investimento

Fundos de Investimentos são considerados como um condomínio composto por investidores, que de acordo com Nave e Castro (2012) possuem uma política de investir de uma forma específica com o estatuto social próprio, em que constam os direitos e deveres dos cotistas, regras referentes ao sistema de cobrança de taxas e aspectos relativos à sua organização social, associados à gestão do patrimônio, administração e custódia da carteira.

No Brasil este segmento surgiu no ano de 1957, como descrito por Nave e Castro (2012) e passou a ser regulado em meados de 1970 pelo Banco Central. Na década de 90, investidores brasileiros começaram a usar os fundos não apenas como proteção de patrimônio contra a inflação, mas também para crescimento em uma época de estabilidade financeira. A partir do século XX, o mercado de fundos passou por um conjunto de reformas liberalizantes, com o objetivo de aumentar a liberdade de gestão e estimular a diferenciação no mercado financeiro.

Os autores Trindade e Malaquias (2015), explicam que os resultados obtidos pelos investidores têm por objetivo a conquista de maior rentabilidade. Existem no mercado uma variedade de categorias de fundos de investimentos para diversos perfis de investidores conforme demonstra algumas no quadro 01 abaixo:

Tabela 01 - Classificação de Fundos ANBIMA

Categoria	**Tipo**
Curto Prazo	Curto Prazo
Referenciado DI	Referenciado DI
Renda Fixa	Renda Fixa
Renda Fixa	Renda Fixa Índices
Multimercados	Multimercados Macro
Multimercados	Multimercados Trading
Ações	Ações IBOVESPA Ativo
Ações	Ações Dividendos
Cambial	Cambial
Previdência	Previdência Renda Fixa

Fonte: Trindade e Malaquias (2015)

A principal diferença entre fundos de renda fixa e renda variável está na variação da rentabilidade segundo Trindade e Malaquias (2015), já que os fundos de

renda fixa proporcionam uma maior segurança em seus recursos aplicados. Os fundos de renda variável são considerados pelo mercado financeiro como voláteis, pois impossibilitam de apresentar a rentabilidade futura do investimento. São compostos na maioria das vezes por aplicações em ações ou títulos, com uma renda variável não se baseando diretamente ao mercado de juros. Nesse sentido, o autor defende essa teoria com as seguintes palavras: “os fundos de renda variável são aqueles cuja composição é, em sua maior parte, de aplicações em ações e/ou títulos e cuja taxa de retorno é variável, portanto, depende de seu desempenho no mercado”. (TRINDADE; MALAQUIAS, 2015, p.81)

O desempenho dos fundos de investimentos pode ser positivo ou negativo, em que este está atrelado ao sucesso das ações, quotas ou títulos que compõem a modalidade, ou seja, o investidor poderá lucrar ou perder dinheiro de acordo com Trindade e Malaquias (2015). Os investidores que almejam ingressar no mercado financeiro, especificamente em fundos de investimentos, devem conhecer a regulamentação vigente e os meios utilizados na administração do fundo, pois os fundos como ações, também conhecido como renda variável, tem-se o requisito de investir no mínimo 67% de seu patrimônio em ações negociadas diretamente na bolsa de valores.

O processo para escolha de um fundo é dividido em duas etapas, consoante os autores Trindade e Malaquias (2015), a primeira corresponde em verificar qual é a carteira mais adequada para o investidor e a segunda em procurar descobrir um gestor que consiga elevar-se ao mercado financeiro, para assim alcançar retornos superior ao do mercado, através de acessos a melhores informações. Analisar os históricos dos fundos de investimentos, te proporciona uma aproximação sobre o funcionamento dos fundos em análise, entender o quanto está rentabilizando o fundo não te trará uma garantia no futuro, mas te dará uma visão e tendência do quão qualitativo e eficiente é. Além disso, analisar os fundos com frequência te proporciona tomadas de decisões para um aumento de rentabilidade ou evitar certas perdas, como o mercado financeiro é bastante flexível, dependendo de situações do momento, poderá afetar definitivamente a rentabilidade atual de certos fundos que constam em sua carteira de ativos comprometidas, desta forma, em diversos momentos é preciso fazer ajustes em sua carteira de aplicações para diminuir ainda mais os riscos.

2.1.2 Caderneta de Poupança

A caderneta de poupança é o investimento mais tradicional de acordo com Nascimento (2015), com características conservadoras, populariza entre os investidores de menor renda. Para fazer uma aplicação na poupança, não é de necessidade ser correntista em uma conta e praticamente todos os bancos têm essa possibilidade, apenas necessita que a pessoa compareça a uma agência bancária portando seu CPF para que possa efetuar a abertura de sua poupança. O risco da aplicação da poupança é extremamente baixo com consequência em baixo retorno de liquidez.

2.1.3 Certificado de depósito bancário (CDB)

O CBD é um título de crédito que os bancos emitem para captar recursos em conformidade com Nascimento (2015), em que é uma obrigação do banco pagar ao investidor ao final do prazo contratado. O prazo varia conforme o recurso disponível do banco em um dia a dezoito meses. CDB por ser um título, pode ter seu resgate antecipadamente do vencimento exposto no momento da contratação, mas com a condição que tenha decorrido o prazo mínimo da aplicação.

O risco do CDB é proporcional à instituição financeira, quanto mais sólida e reconhecida pelo mercado financeiro, este é minimizado pelo seu poder de cumprimento das obrigações. Conforme Nascimento (2015), a vantagem de uma aplicação no CDB é não possuir taxa de administração, logo em alguns casos acaba ultrapassando a rentabilidade dos fundos de renda fixa e DI, porém tem a desvantagem de que por ser uma taxa pré-definida, acaba limitando seus ganhos, algo que por vezes não consegue acompanhar o crescimento da variação do mercado.

2.1.4 Títulos públicos

O mercado financeiro os considera como investimento de baixíssimo risco, pois tem garantia do Tesouro Nacional. Segundo Nascimento (2015), os títulos públicos oferecem retornos significativos e possuem uma quantidade variada de rentabilidade, sendo elas: pós-fixado (remunerado pela taxa básica da economia), prefixadas e indexadas aos índices de preços.

O Tesouro Nacional, de forma a facilitar o processo, conforme Nascimento (2015), criou um site para que as pessoas possam investir no Tesouro Direto de forma direta sem depender de uma corretora, desta forma terá um menor custo que resultará em uma rentabilidade maior. Um investimento no Tesouro Direto possui as seguintes vantagens: segurança, comodidade, liquidez e boa rentabilidade. Se optar por manter o valor aplicado até o dia do vencimento do título, receberá o valor reajustado conforme a rentabilidade informada no momento da compra, mas caso queira realizar um resgate com antecedência, o preço da venda será segundo a procura do mercado.

2.1.5 Análise de investimento

É necessário analisar diversas variáveis de investimento para se poder viabilizar, conforme Nascimento (2015), as variáveis são: risco, retorno, liquidez, custos de operação, taxas de administração e custódia. Porém, além de analisar esses indicadores, é preciso ver se o investimento está adequado com o seu perfil de investidor, se está dentro de suas particularidades como: tolerância a risco, possibilidade de ganhos, idade, estado civil, dependentes e entre outros.

2.1.6 Identificando o perfil de investidor

O perfil de investidor é importante para estipular as categorias de investimentos a serem feitos, definindo os riscos que o mesmo está disposto a aceitar. Com o perfil

de investidor definido, tem-se a capacidade de avaliar as melhores oportunidades para si.

Existem diversas nomenclaturas para os perfis, mas os mais utilizados são três como: conservador, moderado e arrojado. O que diferencia entre as três nomenclaturas é a quantidade da parcela alocada para renda variável, pois quanto maior o risco e longo prazo, maior será direcionado para renda variável.

2.1.7 Variáveis do investimento

Com o perfil de investimento definido, Nascimento (2015) explica que fica mais fácil definir melhores oportunidades de investimento, pois já terá as consciências de riscos que permite que ocorra.

Um investimento tem as seguintes variáveis: retorno, que é o total de ganhos ou perdas de um investimento que ocorreu por um período. Nunca é esperado que este seja negativo, pois se procura sempre retornar ganhos, mas frente a todas as incertezas do mercado financeiro, não podemos garantir que o retorno seja positivo.

Liquidez é o grau da velocidade na conversão do investimento aplicado em dinheiro. O mesmo, tem maior liquidez o quanto mais fácil for a conversão em dinheiro e menor for o prejuízo monetário envolvido nesta transação.

Risco é a hipótese de perda do seu capital investido, quanto maior o risco de perda, também será a oportunidade de obter mais retorno acima do padrão. O risco está ligado à incerteza e volatilidade de determinados ativos no mercado financeiro.

2.2 Extração de Dados (ETL, BI)

Serão apresentados conteúdos de ETL e BI pelo motivo de captação de informações de fontes secundárias e vista a necessidade de passar por processo de

limpeza nos dados consultados da origem no banco de dados para eliminar informações duplicadas, como também arquivos nulos.

A crescente complexidade nos processos de tomada de decisão fez com se tornasse mais importante a utilização de mecanismos de suporte para os gestores e administradores, mais especificamente em áreas que envolvam grande volume de dados.

Os dados devem se converter em informações de forma legível e em tempo útil para que gestores possam utilizá-los em tempo hábil em uma tomada de decisão de acordo com Gomes, Oliveira e Belo (2014). Os mecanismos de decisão podem ser encontrados em sistemas de *Data Warehouse* (SDW), o SDW executa sobre diversas áreas de tratamento de dados, desde um simples processo de extração de dados até uma aplicação de procedimentos de conciliação e limpeza de dados em uma área de estágio de um SDW.

Uma das principais características desse sistema, é a capacidade de manter as informações históricas e de forma organizada conforme Gomes, Oliveira e Belo (2014), porém, a qualidade da informação depende do formato como o sistema é alimentado, portanto, devem ser considerados diversos aspectos relacionados como a angariação, conciliação, garantia de qualidade e padronização dos dados a ser integrado no sistema.

O desenvolvimento de um sistema de povoamento de dados para DW, que segundo Gomes, Oliveira e Belo (2014) é conhecido por *software* de ETL (*Extract, Transform, and Load*). É uma tarefa complexa que envolve em muitos casos recursos sensacionais, entre eles, recursos humanos, financeiros ou materiais, o que estabelece a necessidade de um planejamento de todas as etapas do projeto. No desenvolvimento do ETL, qualquer erro no momento da implementação, mesmo que seja detalhes, poderá gerar consequências desastrosas no SDW. Os cuidados na metodologia do desenvolvimento têm um único objetivo: O sucesso da aplicação do SDW através de uma estratégia minimizando erros que possam ocorrer em cada fase do desenvolvimento, reduzindo custos e o tempo de implementação.

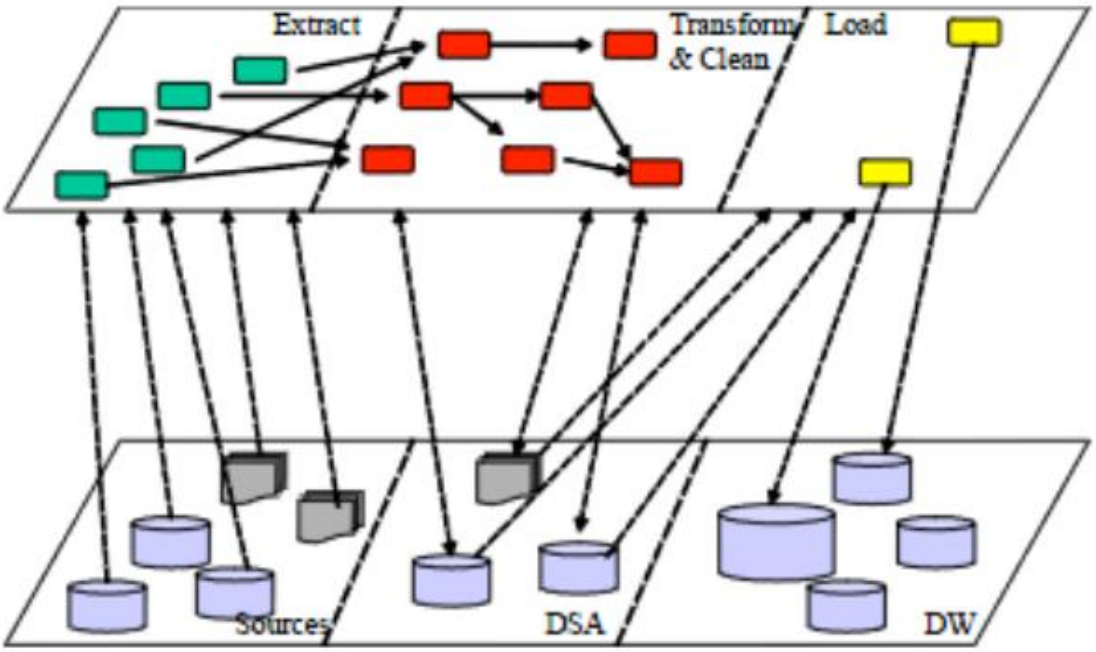
Atualmente, o conjunto existente de ferramentas no mercado para modelação de um sistema de ETL, é muito variado, de acordo com Gomes, Oliveira e Belo (2014),

existem plataformas que cobrem de forma completa todo o ciclo do desenvolvimento. Mas, a maioria dessas ferramentas implementam metodologias próprias, não só para a modelação dos processos, mas também para a representação das tarefas específicas. Essas ferramentas de ETL possuem especificações e construtores muito específicos, faz com que dificulte o processo de aprendizagem e aplicação, além disso, inibem a interpretabilidade entre elas, tornando mais difícil e custosa a migração entre sistemas de ETL.

O processo ETL é destinado à extração e transformação dos dados e finaliza com a inclusão dos mesmos no DW consoante Ferreira et al. (2010). Essa fase é caracterizada por abranger procedimentos de limpeza, integração e transformação dos dados, que conforme o autor, é o processo mais grave e lento na construção de um DW.

A figura 1 descreve de forma geral o processo de ETL. Na camada inferior o armazenamento de dados que são utilizados no processo. No lado esquerdo observa-se os dados originais, que geralmente originam-se de BD ou de ficheiros com formatos heterogêneos como exemplo de um CSV. O DW é representado na parte direita da figura, tem como objetivo armazenar os dados. Por último, na parte superior da figura, tem processo de carregamento dos dados no DW, é realizado através de atividades de cargas.

Figura 1 - Ilustração do processo de ETL



Fonte: Ferreira et al.(2010).

De acordo com Ferreira et al. (2010) o processo de ETL é dividido em três fases cruciais, a extração, transformação e carga dos dados.

A concepção de um processo de ETL incide no mapeamento de atributo dos dados de uma ou várias fontes para os atributos da tabela do DW. É importante salientar que as alterações dos dados não afetam os dados nas fontes de origem, mas sim no momento da extração para o repositório do DW conforme Ferreira et al. (2010).

Logo após o processo de transformação dos dados, ocorre o processo de carga. Neste processo, é mapeado as uniformidades dos resultados extraídos sintáticos e semânticos entre os esquemas, respeita as restrições e integridade, e cria uma visão concretizada de forma unificado dos dados extraídos e transformados. Segundo Ferreira et al. (2010) esse processo é o mais trabalhoso e complexo de se fazer.

Atualmente existem diversas ferramentas que são capazes de executar os três processos, na figura 2 apresenta uma visão geral das evoluções destas ferramentas.

Figura 2 - Gerações de ETL

Ano	Título	Significado
Início de 1990	Codificação manual de ETL	Códigos personalizados escitos à mão
1993-1997	A primeira geração de ferramentas de ETL	Código baseado em ferramentas de ETL
1999-2001	Segunda geração de ferramentas de ETL	Código baseado em ferramentas de ETL
2003-2010	Ferramentas de ETL actualmente	A maioria das ferramentas eficientes

Fonte: Ferreira et al. (2010).

Atualmente, as ferramentas de ETL disponíveis encontram-se preparadas para o processo de extração, transformação e carga. Houve um avanço nestas ferramentas desde 1990. Uma ferramenta de ETL com qualidade, deve ser capaz de integrar com diversos bancos de dados e fazer leituras em vários formatos.

Alguns exemplos de ferramentas são: *Pentaho Data Integration, Data Migrator e PowerCenter.*

2.3 *Big Data*

O *Big Data* é um conceito utilizado para representar grandes volumes de dados e que cada vez mais ganha relevância no momento em que a sociedade se depara com um aumento significativo de informações geradas a cada dia de acordo com Costa (2017). O termo já foi utilizado em vários congressos e eventos especializados de diferentes áreas, mas sem muita precisão quanto a sua definição. Segundo o autor, o termo tem sido utilizado para descrever: I) os fenômenos sociais, como a mudança de comportamento gerada pela informação em tempo real; II) as características específicas dos dados, como grandes volumes gerados rapidamente, e de fontes e 14 formas variadas; III) as técnicas de análises e de armazenamento, que precisam ser pensadas e planejadas de modo a superar limites das técnicas utilizadas hoje em dia; e, ainda, IV) as necessidades tecnológicas, como dispositivos de captura e de processamento. A propagação rápida deste conceito leva a sua ampliação sem um núcleo claro de significado do que é *Big Data*.

De forma geral, o termo *Big Data* aplica-se às informações de vigor que não podem ser tratadas e processadas de forma comum, isso ocorre devido ao grande volume e variedade de dados. *Big Data* é definido como um grupo de dados em que o tamanho está além da habilidade de armazenagem e análise de um *software* de banco de dados típico.

A consolidação do termo *Big Data* enquanto ao tributo dos dados, veio no início do século XX, Costa (2017) afirma que as condições e meios de negócios atuais estão chegando ao limite no conceito de dados convencionais, o que faz surgir novas abordagens. Os desafios de gerenciamento de dados são apresentados por três dimensões, nas quais são: Volume, Velocidade e Variedade, que tem como conhecimento os “3 vês”, que se refere diretamente ao termo *Big Data*.

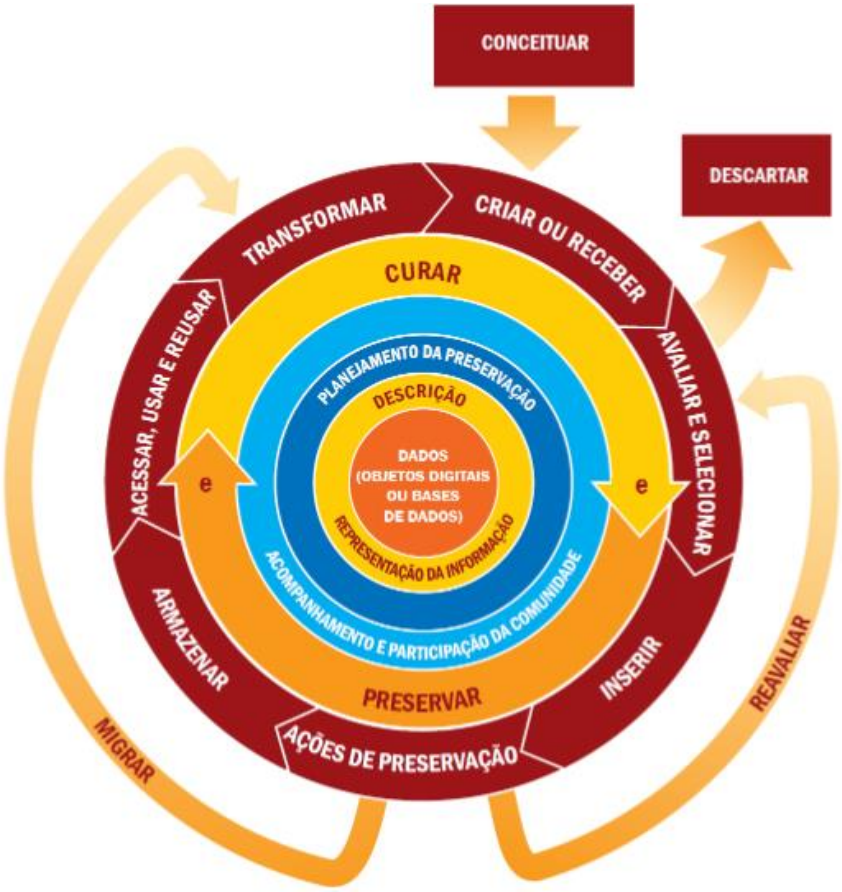
O autor Costa (2017, p.14) esclarece que: “*Big data* é definido como ativos de alto volume, velocidade e variedade de informação que exigem custo-benefício, de formas inovadoras de processamento de informação para maior visibilidade e tomada de decisão”. O desconhecimento desse conceito, a não estruturação e a falta de um padrão dos dados e uso completo das ferramentas de análise ainda é o cenário recorrente na maioria das empresas. Dessa forma, parece muito distante para a maioria dos gestores das organizações a tomada de decisão com as informações certas. De acordo com o autor, somente 65% das organizações são eficientes na disseminação da informação, e desses apenas 4% das organizações usam todos os dados que coletam.

Dadas as características do *Big Data,* de acordo com Carmo (2019), algumas questões importantes afloram:

- ”Como armazenar os dados e metadados em ecossistemas de *Big Data*?
- Como organizar e catalogar os dados e metadados armazenados nesses ecossistemas?
- Como garantir que os dados críticos estejam disponíveis no Big Data para o Processo de Tomada de Decisão?”

Interdisciplinarmente, essas questões almejam algumas competências da Ciência da Informação, introduzindo a Curadoria Digital (FIGURA 3) como elemento importante na definição do ecossistema de *Big Data*.

Figura 3 - Ciclo da Curadoria Digital



Fonte: Rautenberg e Carmo (2019).

A Curadoria Digital é um conceito ligado à autenticidade e à proveniência como também à garantia da qualidade dos dados conforme Carmo (2019). É envolta por boas condutas de planejamento e de gerenciamento de dados. No contexto de *Big Data* com seu ciclo de vida, a Curadoria Digital pode auxiliar em:

- **Conceituar:** É a formalização de documentos que definem as orientações, políticas, requisitos legais e ações de criação, representação, captura, limpeza, avaliação e guarda dos dados.

- **Criar ou Receber:** São ações para criar dados no conceito de *Big Data*. Na criação e recebimento dos dados, deve-se proceder com as políticas de coleta e documentação.

- **Avaliar e Selecionar:** Antes da inserção de novos dados no conceito de *Big Data*, deve-se avaliar os dados quanto a qualidade para inserção.

- **Inserir:** Definido o conjunto íntegro de dados, o passo seguinte é armazenar os dados no *Big Data*, conforme os documentos formalizados.

- **Ação de preservação:** Realiza as ações de garantia na preservação dos dados ao longo do tempo. As ações de preservação são definidas previamente e devem ser ajustadas de modo que os dados continuem autênticos, confiáveis e úteis, mantendo sua integridade.

- **Armazenar:** Ao proteger os dados, deve-se garantir que sejam mantidos seguramente, usando tecnologias apropriadas para o armazenamento no *Big Data*.

- **Acesso, uso e reutilização:** Possibilitar que dados sejam acessíveis pelos usuários. Os controles de autenticação devem ser configurados segundo as políticas inicialmente definidas.

- **Transformar:** Pode de somar ou derivar novos dados a partir dos dados já armazenados no *Big Data*.

- **Descarte:** Os dados podem ser excluídos devido a informações desatualizadas e inválidas ou orientação legal. Normalmente, os dados são extraídos do ambiente de produção e transferidos para um arquivo com possibilidade de recuperação. Em outros casos, eles podem ser excluídos permanentemente por motivos legais que permitem a destruição segura dos dados.

- **Reavaliar:** Pode-se reavaliar uma versão mais recente dos dados não validados anteriormente, conforme os procedimentos formalizados no Conceituar.

- **Migrar:** Deve-se executar ações de migração dos dados para um formato mais atual caso sofra avanços tecnológicos. Desta forma, preservar os dados a longo prazo.

Diante do exposto, mostra-se a necessidade de filtrar os dados fazendo certa limpeza antes da inserção em um *Big Data*, é importante que evite trazer informações

repetidas para evitar um maior volume de dados. Além do processo de inserção, tem como uma prioridade uma infraestrutura de qualidade para manter a disponibilidade dos dados para os usuários.

2.4 Consultas banco de dados relacionais

De acordo com Reis (2019), um banco de dados relacional é organizado em coleções de tabelas bidimensionais, também conhecidas como relações. Desta forma, uma relação é uma forma de organizar os dados em linhas e colunas. O banco de dados relacional permite que as informações fiquem armazenadas de forma estruturada por um padrão pré-definido Brandão (2017).

O aumento no volume de dados a serem manipulados ocasionou o surgimento de diversos sistemas para que seja possível tratar diversas necessidades de banco de dados, entre elas, o controle de acesso aos dados, desempenho e garantia das informações. Os autores Silva et al. (2013) explicam que para facilitar a administração de banco de dados, surgiu o Sistema Gerenciador de Banco de Dados (SGBD). Este é considerado como um conjunto de programas que permite a criação e manutenção de banco de dados por usuários. O meio de acesso ao SGDB é conhecido como *query*, é uma forma de fazer requisições diretamente no banco de dados.

A forma de consulta a informações no banco de dados através da *query* acontece através de uma linguagem de informação conhecida como SQL (*Structured Query Language*). Além de consultas, através do SQL é possível fazer certas manipulações nos dados como inserções, alterações, remoções entre outras conforme Silva et al. (2013).

Com essas requisições feitas através de um sistema junto ao volume de dados, o servidor onde se encontra o banco de dados pode encontrar certas dificuldades em enviar respostas aos clientes em tempo hábil. Para que se evite esse tipo de problema, conforme Vieira e Araújo (2015), existem diversas técnicas para solucioná-las.

Quando é realizado um trabalho de otimização de instruções de SQL, a primeira etapa é a análise do plano de execução, que através das informações que este

fornece, é possível analisar o desempenho da consulta e verificar a necessidade de otimização.

O plano de execução é uma sequência de passos em que o SGBD deverá concluir para depois executar o SQL. Conforme Vieira e Araújo (2015), no plano de execução é exibido em linhas, com detalhamento das etapas que o banco de dados necessitou na execução do SQL com as seguintes informações:

- Ordenação das tabelas referenciadas pela instrução;
- Método de acesso para cada tabela mencionada;
- Método de ligação *join* para as tabelas afetadas nas operações;
- Operações de dados tais como filtro, ordenação ou agregação;
- Custo e cardinalidade da operação;
- Conjunto de tabelas particionadas acessadas;
- Se houve processamento paralelo.

Uma das formas para melhorias no desempenho das consultas de banco de dados é a utilização de índices, de acordo com Vieira e Araújo (2015), além de melhoria no desempenho, gera uma lista de forma ordenada com a posição de cada registro da tabela.

A seguir, será apresentado algumas formas de melhorias no desempenho nas consultas dos dados, mostrará diversas soluções disponíveis no banco de dados através de alterações nas tabelas como também otimizações na própria consulta de banco de dados.

2.4.1 Índice

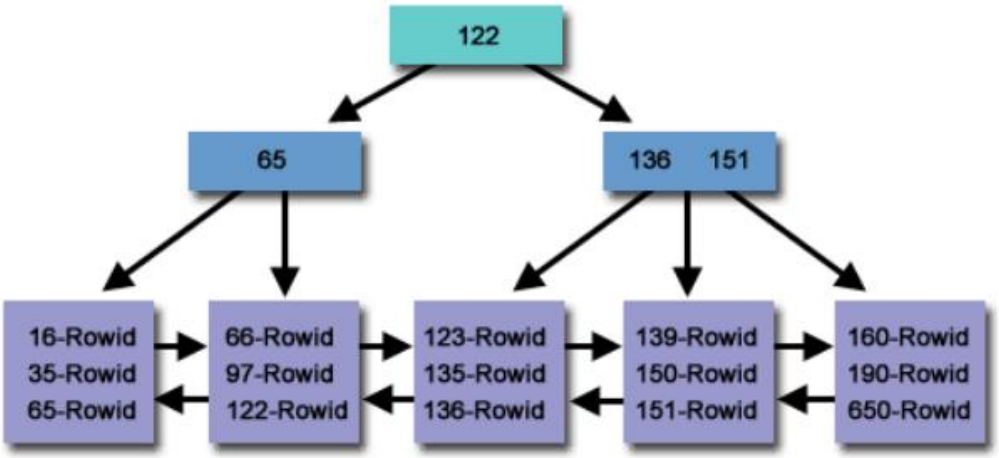
Índice é uma estrutura muito utilizada em banco de dados, mas não se torna obrigatório, segundo Vieira e Araújo (2015), auxilia em melhoria de desempenho nas

consultas SQL, com o índice, gera uma lista de forma ordenada com a posição na tabela de cada registro, desta forma, as ligações entre as tabelas.

2.4.2 Índice B-tree

É considerado como índice padrão, através dele se gerencia os blocos de dados de forma a controlar o ponteiro dentro de cada bloco. Segundo Vieira e Araújo (2015), a criação desse índice é recomendada quando se tem consultas de colunas com alta cardinalidade, como, por exemplo: Colunas com valores únicos, mais utilizados em chaves primárias dentro de uma tabela conforme a figura 4.

Figura 4 - Índice B-Tree



Fonte: Vieira e Araújo (2015).

2.4.3 Índice Bitmap

Mais utilizados em colunas de baixa cardinalidade, não recomendada em colunas na qual se tem alterações nos dados de forma frequente, um exemplo de utilização na figura 5

Figura 5 - Índice Bitmap

	LINHA																		
VALOR	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
MASCULINO	0	0	1	1	1	0	1	0	0	0	0	1	0	1	1	1	0	1	0
FEMININO	1	1	0	0	0	1	0	1	1	1	1	0	1	0	0	0	1	0	1

Fonte: Vieira e Araújo (2015).

2.4.4 Utilização de *functions*, *stored procedures* e *package*

Quando se utiliza uma *stored procedures,* a maioria das regras de negócio estará vinculada no banco de dados. Conforme Vieira e Araújo (2015), é uma boa prática de desenvolvimento separar as camadas dentro da aplicação, na qual cada uma das camadas tem uma função específica, não sendo ideal colocar regras diretamente no banco de dados. Mas ao nível de desempenho, é possível reduzir 55,26% do tempo na execução de um bloco com mil transações através de *stored procedures* comparados o mesmo bloco sendo executado através de uma consulta via SQL.

Uma das grandes vantagens desta rotina, é quando se trabalha com diferentes plataformas e acesso ao mesmo banco de dados, desta forma, independente da linguagem de programação, todas utilizarão os mesmos procedimentos dos dados. Além disso, Bogo (2013) explica que o stored procedures tem acesso diretamente com o banco de dados, desta forma o trabalho é realizado diretamente no servidor apenas retornando o resultado para a aplicação, desta forma há uma grande quantidade de dados que não é preciso enviar para o cliente otimizando o processo.

2.4.7 *View*

Conforme Rodrigue (2019), a técnica de otimização através de *View*, é uma representação de uma tabela de banco de dados, contudo, de forma virtual, é possível conter colunas de diversas tabelas físicas ou até mesmo de outras *Views*. Geralmente,

os dados da *View* não são armazenados dentro do banco de dados, mas quando é chamado através de uma consulta, a View faz uma consulta no banco de dados conforme a consulta solicitada.

2.4.7 View Materializada

A *View* Materializada são objetos de banco de dados que de acordo com Vieira e Araújo (2015), são utilizados para armazenar os resultados de uma instrução de SQL. Tem como utilidade para melhorar o desempenho das consultas no banco de dados mais complexas, como uma consulta com relacionamento entre diversas tabelas e um volume de informações altas.

A *View* Materializada tem a forma de utilização bem semelhante com uma *View* convencional, porém tem uma diferença bem importante entre elas explicam os autores Vieira e Araújo (2015), uma *View* convencional é um objeto criado através de uma instrução determinada como parâmetro e a consulta sempre será executada através da *View*. Já a *View* Materializada, executa a consulta estabelecida como parâmetro apenas na criação e quando há atualização. Os dados contidos dentro da *View* materializada são salvos novamente no banco de dados como se fosse uma nova tabela, mas não ocupa mais espaço no banco de dados. Desta forma traz um ganho no desempenho das aplicações, principalmente de *Data Warehouses* na qual a maior preocupação está no desempenho de novos relatórios e não se preocupando com uma ocupação maior no banco de dados.

2.5 APIs e Web Services

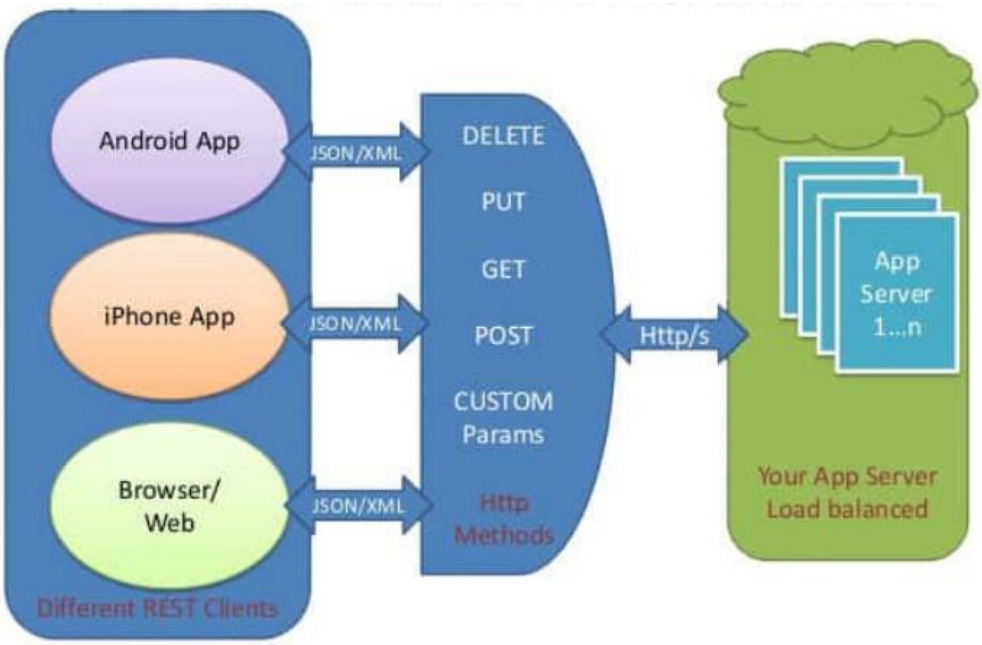
A necessidade de integração de dados entre sistemas de diferentes linguagens e origens é uma necessidade cada vez maior. Com a crescente no uso de smartphones e dados interligados torna-se uma tendência de evolução cada vez maior

com a necessidade de disponibilizar uma comunicação entre servidores *web* e dispositivos móveis, conforme Oliveira (2014).

O autor explica que para ser possível integrar os diferentes sistemas com maior facilidade, a tecnologia de *Web Services* foi criada, desta forma permite disponibilizar formas de integrar diferentes sistemas sem vinculação alguma.

Existem diversos protocolos e padrões denominados para o desenvolvimento de um *Web Services*. Oliveira (2014, p.16) cita alguns exemplos: "*WS-Notification*, WS-Security, WSDL (*Web Services Description Language*) e SOAP (*Simple Object Access Protocol*)". Como a maioria desses *Web Services*, utilizam o protocolo de HTTP para a comunicação externa, pode-se assumir um padrão que utiliza 100% de tudo o que esse protocolo disponibiliza, desta forma nasce um dos padrões mais utilizados na atualidade, conhecido como REST (*Representational State Transfer*). Com a finalidade da simplificação na criação de um *Web Services*, utiliza o protocolo padrão HTTP e aproveita todos os recursos que oferece, tais como interface uniforme e o princípio de endereçamento conforme a figura 6.

Figura 6 - REST API *Architecture*



Fonte: James (2018).

Seguindo a arquitetura REST, tem possibilidade de disponibilizar uma API de forma fácil, em função do uso de um conjunto de modelos de programação que

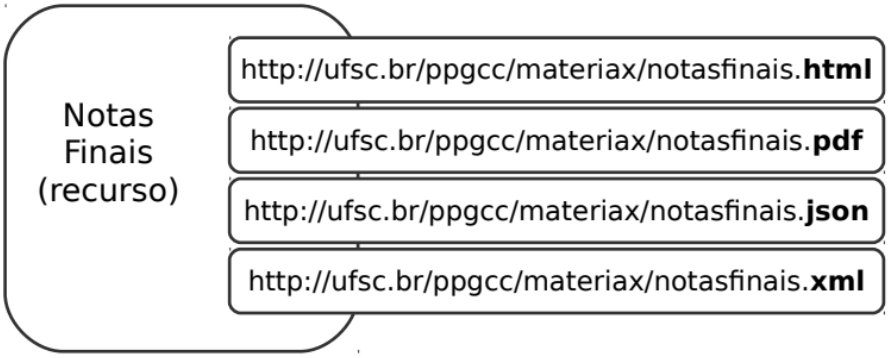
permite a elaboração de aplicativos e a sua utilização de maneira não tão clara para os usuários.

REST é um conjunto de princípios para o desenvolvimento de aplicações Web. usa o padrão de cliente-servidor, na qual as requisições partem do cliente e são respondidas pelo servidor. Segundo Salvadori (2015), esta é uma abordagem leve para o desenvolvimento de *Web Services* e busca simplicidade e baixo acoplamento. Este tem como base que as informações se tornam acessíveis para clientes remotos, que são endereçados por um identificador único, conhecido como URI (*Uniform Resource Identifier*).

Um recurso pode ser identificado por diversas UIRs explica o autor Salvadori (2015), mas uma URI endereça um único recurso. O recurso sempre é acessado através de uma representação, que está sempre associada.

A Figura 08 representa a relação que existe entre um recurso, URIs e diferentes representações. As notas dos alunos representam um recurso e estão disponíveis em formatos (representações) diferentes, nas quais são: HTML, PDF, JSON e XML. Cada representação pode ser acessada pelo meio de uma URI diferente, mesmo que o recurso continue sendo apenas um, não é necessário que exista uma exclusiva para cada representação. Esta pode estar associada a diferentes representações de forma única conforme mostra na figura 7 e 8. Nesse caso, a melhor forma está no resultado da negociação entre cliente e servidor.

Figura 7- Relação entre recursos URIs



Fonte: Salvadori (2015).

Figura 8 - Relação URI única

Fonte: Salvadori (2015).

Um dos princípios do REST é estabelecer uma interface constante entre cliente e servidor, que proporciona uma estrutura simplificada. Uma forma é respeitar a sematologia do protocolo usado pelo *Web Service*. O HTTP é o protocolo mais empregado em *Web Services* REST, e manter a sematologia do protocolo é utilizar de forma adequada os seus verbos e códigos de mensagens. Os verbos mais utilizados de HTTP são:

• GET — Obter a representação de um recurso;

• POST — Criar um novo recurso;

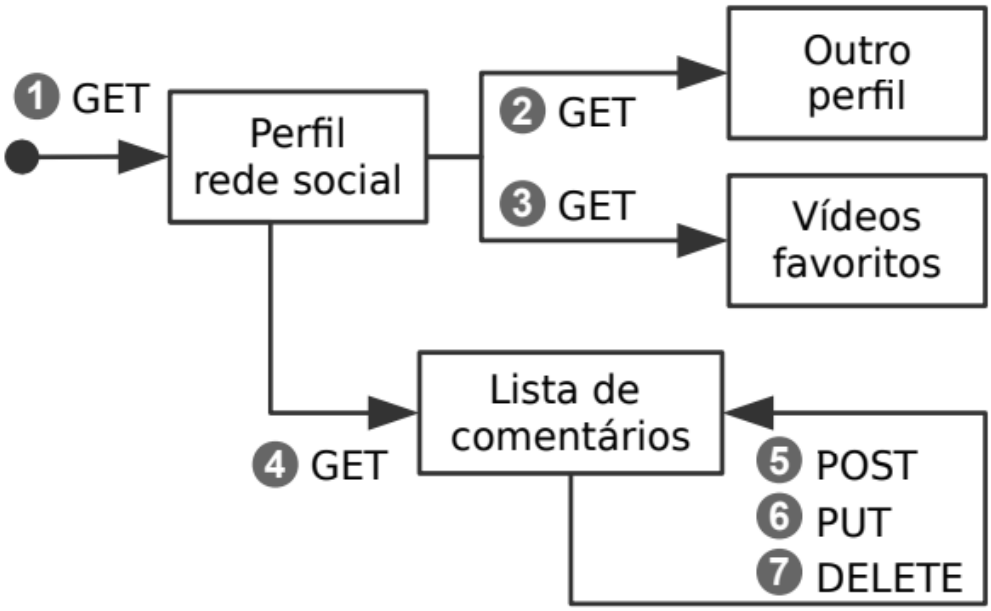
• PUT — Alterar um recurso;

• DELETE — Remover um recurso.

Web Services REST pode ser projetado para interligar recursos, por exemplo, uma página pessoal pode haver links para outras páginas, de forma a interligar os recursos. A figura 9 apresenta o fluxograma de requisições para um determinado recurso. Conforme Salvadori (2015):

> “O ponto de entrada é o recurso Perfil rede social, obtido através de uma requisição GET(1). A partir deste perfil, é possível seguir três caminhos distintos. O primeiro caminho é através de GET(2), onde é possível acessar outro perfil da rede social. O segundo caminho, através de GET(3), é possível obter os vídeos favoritos do perfil. A requisição GET(4) é o caminho que leva a uma lista de comentários. A lista possibilita criar, alterar e remover comentários, através das requisições POST(5), PUT(6) e DELETE(7), respectivamente”.

Figura 9 - Fluxograma de requisições



Fonte: Salvadori (2015).

2.5.1 JSON

Uma das representações mais usadas em *Web Services* REST é o JSON (*JavaScript Object Notation*), é uma forma textual de representar os dados estruturados no formato de chave/valor. A chave é caracterizada como texto, e o valor pode ser declarado como texto, booleano, número, objeto, nulo ou uma sequência ordenada de valores. Conforme o autor, explica com suas palavras: “JSON é muito utilizado no intercâmbio de informações, pois é independente de linguagem de programação, de fácil criação, manipulação e análise”. (SALVADORI, 2015, p.35). A figura 10 mostra um exemplo de arquivo no formato JSON.

Figura 10 - Exemplo de JSON

```
1  {
2     "name": "Maria Rosa",
3     "homepage": "http://mariarosa.com.br/",
4     "image": "http://mariarosa.com.br/maria.png"
5  }
```

Fonte: Salvadori (2015).

A estruturação do JSON é de forma hierárquica composta por uma sequência de vetores, matrizes e objetos de acordo com Flôres (2019). As matrizes têm uma estrutura de forma multidimensional delimitadas pela abertura e fechamento de colchetes, com separação dos elementos delimitados por vírgula, enquanto os vetores, trabalham da mesma forma que a matriz, porém, limitado de forma unidimensional. Por último e da mesma forma que matriz, o objeto tem sua delimitação de abertura e fechamento por chaves, admite múltiplos pares de elementos de nomes e valores e até mesmo outros objetos, conforme a figura 11.

Figura 11 - Exemplo de JSON

```
["RJ", "SP", "MG", "ES"]

Exemplo de Matriz Bidimensional
[
 [1,5],
 [-1,9],
 [1000,0]
]

Exemplo de Objeto
{
 "titulo": "Objeto na Notacao JSON",
 "resumo": "Objetos sao representados em JSON desta forma",
 "ano": 2019,
 "palavras_chave": ["JSON", "semiestruturado", "dados"]
}
```

Fonte: Flôres (2019).

3 TRABALHOS RELACIONADOS

Esta seção apresenta os trabalhos selecionados com tecnologias ou técnicas semelhantes e serviram como base para a teoria do desenvolvimento das aplicações e base para as práticas aplicadas.

3.1 Open Data Processor: Uma ferramenta para a automatização do processo de extração e publicação de Dados Abertos

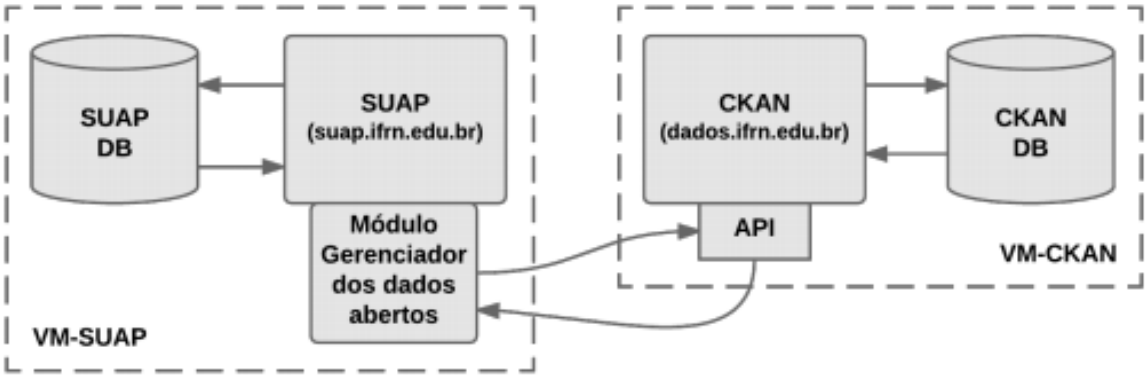
O trabalho apresentado por Vilela (2018) mostra a importância de acesso sobre informações públicas. No artigo escrito pelo autor, mostra que normalmente o processo de extração de dados é feita de forma manual por equipes com competência na área de desenvolvimento.

Neste trabalho o autor visa desenvolver uma ferramenta que facilita e automatiza a extração e publicação de dados abertos e para isso, mostra dois cenários elaborados sobre integração entre SUAP e CKAN.

No primeiro cenário, foi desenvolvido um módulo dentro do SUAP (Sistema Unificado de Administração Pública) que é um sistema corporativo na *web*, desenvolvida pela Diretoria de Gestão e Tecnologia da informação do IFRN, que visa a integração de diversos dados e permite o gerenciamento de conjunto de dados pela interface *web* do sistema. Os desenvolvedores selecionam os dados que serão

publicados e a periodicidade que o mesmos serão atualizados de forma fácil e automáticas, em que serão publicadas no CKAN (*Comprehensive Knowledge Archive Network*), aplicação *web* de catalogação de dados que mantém uma descrição dos dados armazenados, quem é o dono e se são livremente disponíveis. Com essa abordagem, Vilela (2018) explica que elimina a necessidade de uma implementação de um novo serviço para *web*, o que permite uma maior agilidade na publicação de novos conjuntos de dados. Por outro lado, uma desvantagem apontada é a constatação que o processo de extração pode fazer com que o servidor fique mais lento, prejudicando o funcionamento do sistema, além de funcionar apenas no contexto de SUAP. Na figura 12 abaixo, mostra o funcionamento de um SUAP.

Figura 12 - Gerenciamento de SUAP

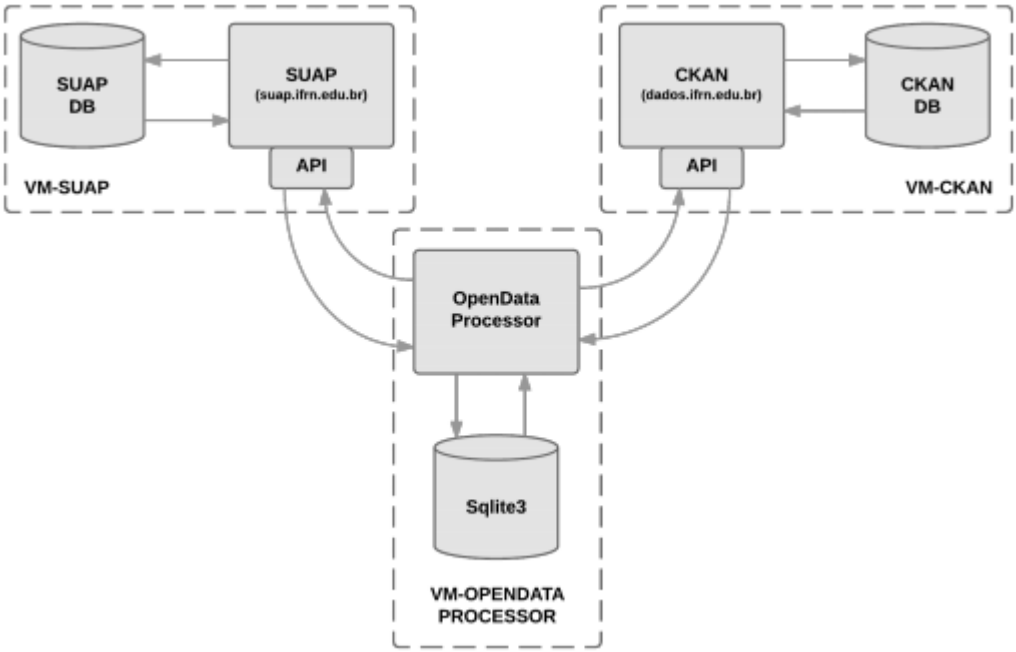


Fonte: Vilela (2018).

No cenário 2, conforme a figura 13, o autor do artigo explica sobre a aplicação separada que é responsável pela extração dos dados através de uma API do SUAP, pelo processamento e pelo carregamento dos dados diretamente no CKAN. No trabalho citado foi desenvolvido uma aplicação separada do SUAP e do CKAN, nesta aplicação, a equipe de desenvolvimento deverá cadastrar as organizações, grupos e conjunto de dados, além disso, cadastrar as URLs dos serviços *web* disponibilizados pela API já existente no SUAP e o intervalo do período que deverão ser coletados, processados e carregados no CKAN. Neste cenário, a equipe de desenvolvimento deverá desenvolver e expor um novo serviço *web* na API, que tem como benefício o processamento de extração, publicação e atualização dos dados sejam feitas de forma

isolada do servidor de produção do SUAP, trazendo uma maior performance para o servidor durante a execução deste processo.

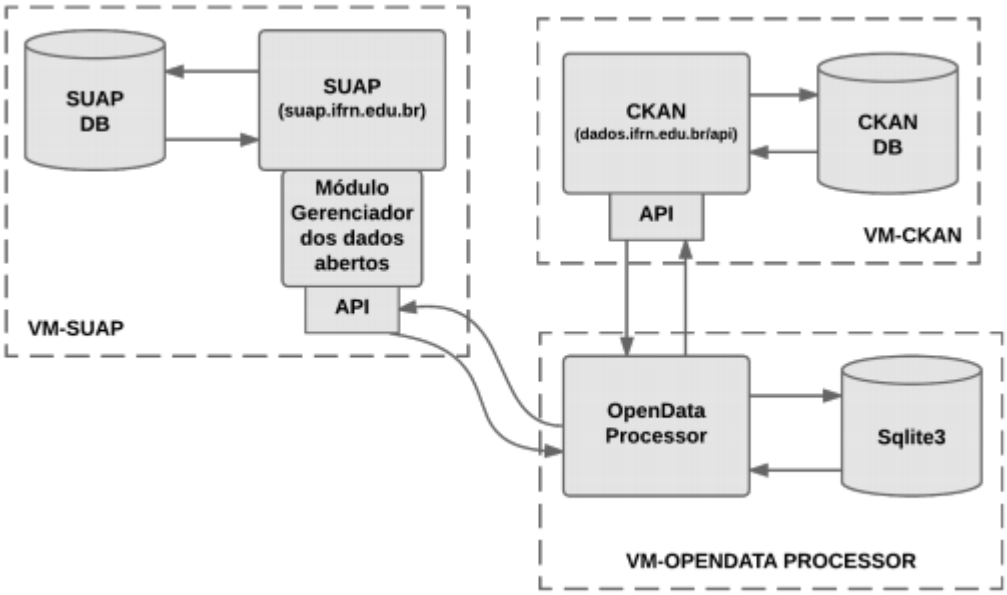
Figura 13 - Extração dos dados através de uma API do SUAP



Fonte: Vilela (2018).

No terceiro e último cenário, ilustrado na Figura 14, foram aproveitadas as vantagens de cada um dos cenários anteriores, é mostrado no primeiro cenário e a aplicação responsável pela extração, publicação e atualização no segundo cenário. Assim, os responsáveis continuam utilizando a interface *web* a fim de definir os dados que serão abertos, de aspecto automático os serviços *web* são criados e publicados através de uma API utilizada de forma exclusiva para disponibilizar as informações. Após ter os serviços expressos pela API, a aplicação incubida pela coleta dos dados consomem cada serviço, cadastra os dados no CKAN e agenda a extração de acordo com o horário definido no conjunto de dados.

Figura 14 - União das vantagens dos cenários



Fonte: Vilela (2018).

Com os resultados apresentados pelo trabalho citado, foi possível concluir que para atender um maior número de instituições públicas, seria necessário a criação de uma ferramenta de forma genérica isolada do ambiente de produção, que permita a extração das informações via API ou banco de dados, que com base no cenário 2, foi desenvolvida uma aplicação web.

No artigo apresentado Vilela (2018) também mostra a importância da separação dos processos entre a extração e publicação dos dados, nele, desenvolveu cenários que trazem diversos ganhos de desempenho e garantia para auxiliar nesses processos, pois, quando se trata em grande quantidade de informações e uma faixa de atualização frequente, se não cuidar de certos detalhes acabará trazendo gargalos ao servidor. Também deixou claro que para que possa atender todas as instituições públicas, é necessária uma ferramenta que trabalhe de forma genérica.

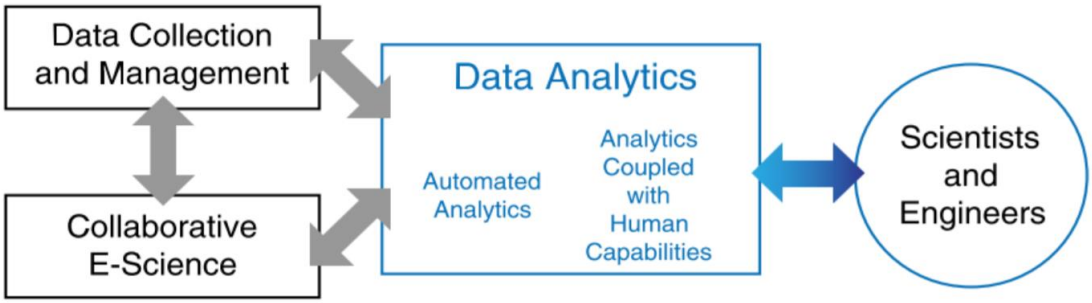
3.2 Inovação dos negócios por meio da análise de Big Data.

O trabalho produzido por Nascimento et al. (2017), mostra temas relacionados à *Big Data*, Plataforma de inovação e *Big Data Analytics*. Nele os autores explicam

que as empresas precisam estar preparadas estrategicamente para a inovação do uso eficiente da grande quantidade de dados disponíveis, de forma que a isso está relacionado diretamente com o *Big Data*. O trabalho está voltado para os avanços de debates relacionados ao tema e o uso prático de ferramentas como um diferencial.

Conforme Nascimento et al. (2017), explica que a inovação se torna cada vez mais decisiva devida a mineração efetiva de dados com *Big Data*, uma vez em que ela possibilita extrair ideias inovadoras e aprofundar conhecimentos de diversas áreas. Para separar as informações necessárias e identificar o que se torna relevante e viável, são necessários muito tempo e esforço neste processo e geralmente é preciso criar uma infraestrutura analítica relacionando a mesma com vários itens de informações estratégicas em direção ao objetivo conforme a figura 15 apresentada pelo autor.

Figura 15 - Análise de dados Big Data



Fonte: Nascimento at el. (2017).

A exploração e análise dos dados através do *Big Data*, mais conhecido como *Analytics*, é feita através de diferentes ferramentas, baseia-se em análises preditivas, mineração de dados, estatística, inteligência artificial entre outras. Segundo Nascimento et al. (2017), uma das principais relevância é a capacidade de converter grande volume de dados em informações importantes para as organizações de forma que gere uma vantagem competitiva comparada às demais, desta forma, *Big Data* e *Big Data Analytics* tem sido usada para descrever conjunto de dados e técnicas analíticas. O uso dos dados acontece em aplicações grandes com tamanho de dados de *terabytes* para *hexabytes*, que ocasiona formas complexas que requerem um *software* avançado, armazenamento e gerenciamento dos dados e eficiência nas visualizações das informações.

A importância do *Big Data* na atualidade é deixada explícita por Nascimento et al. (2017), devido ao aumento das informações e volume de dados, que torna inábil a inserção de informações de forma convencional antes de serem tratadas. Também informa a importância do tempo de trabalho para tratar os dados antes de armazenar, trazendo uma maior qualidade nos registros para o banco de dados.

Portanto, é possível concluir, mediante os dados levantados no trabalho citado, que, de acordo com Nascimento et al. (2017), devido à evolução das tecnologias de informação de TI, o volume, variedade e velocidade passa a se tornar cada vez maior e eficiente. Além disso, desde que se utilizou o termo *Data Warehouse*, utiliza-se o relacionamento junto às soluções de *Business Intelligence* (BI) de forma crescente nas empresas.

3.3 Fundamentos e aplicação do Big Data: como tratar informações em uma sociedade de yottabytes.

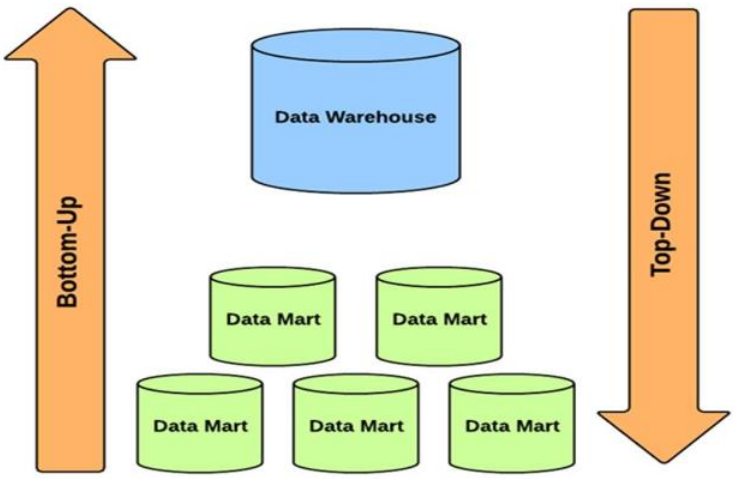
Uma questão que cientistas da área de computação tentam resolver há algum tempo é a de administrar uma quantidade de dados de grande volume a todo o momento. Devido a isso, buscam respostas sobre quais áreas o *Big Data* é utilizado. Segundo Caldas e Silva (2016), a necessidade de processar dados de forma rápida e eficaz é cada vez maior. Por isso o objetivo do trabalho citado é tratar de analisar o *Big Data* e mostrar os recursos tecnológicos envolvidos, considerando uma evolução para os próximos anos devido ao crescimento imparável na produção dos dados, que demanda uma extração de informações, pois, na maior parte das vezes são armazenados dados sem um objetivo, o que cria a necessidade de evoluir em tecnologia mais ampla.

Um *Data Warehouse* (DW) tem apenas duas operações básicas, uma delas de inserção de dados iniciais e incremental e a segunda que é a consulta ao repositório de dados através da leitura. Com essas características o DW se torna diferente de sistemas de gerenciamento de banco de dados (SGBD) tradicionais. Conforme o autor, não é preciso se preocupar com a questão de concorrência de dados, pois, os

dados ficam registrados no DW até enquanto são necessários, após perder sua função, acabam sendo descartados (CALDAS e SILVA 2016).

Os *Data Marts* (DM) são *Data Warehouse* orientados a departamentos, como de uma organização, sendo dividida como (comercial, administrativo, financeiro, etc.), desta forma, cada departamento será considerado como um DM, transformando o conjunto de DM em um DW. De acordo com Caldas e Silva (2016), após o DW ser implementado por diversos DM, para a realizar um caminho inverso de um DM, alimentando os DM separadamente afirmando-se que os DM surgem de duas formas, sendo elas *Top-down* e *Bottom-up* conforme a figura 16 abaixo:

Figura 16 - Top Down e Bottom Up



Fonte: Caldas e Silva (2016).

De acordo com Caldas e Silva (2016), a expressão *Big Data* está se tornando cada vez mais popular, mas não se tem de forma clara o real significado, aplicabilidade e finalidade. Tem como principal foco pela escala acelerada com uma quantidade de volume cada vez maior. Quando se fala neste conceito imagina-se um grande volume de informações a serem analisados, porém, o mesmo não se limita a isto, tendo a base de cinco “V’s” para conhecimento, sendo elas: Volume, velocidade, variedade, veracidade e valor.

Neste trabalho apresentado, mostra-se de forma mais abrangente o significado de *Big Data*, deixando claro que não se trata apenas de grande volume de dados, além disto, mostra diversas tecnologias a serem implementadas para utilizar o *Big*

Data. Podemos concluir que *Big Data* está sendo cada vez mais utilizados para inúmeras soluções, mediante os dados levantados no trabalho citado, esses fundamentos permitem que seja possível armazenar quase tudo que seja veiculado nos meios eletrônicos, sendo dados estruturados e não estruturados como imagens, vídeos, áudio e documentos mostrando a importância de utilizar o Big Data dentro de sua empresa trazendo uma melhor qualidade no gerenciamento dos dados.

3.4 Aumentando a acessibilidade aos sistemas de Big Data por meio de um API de serviços comuns.

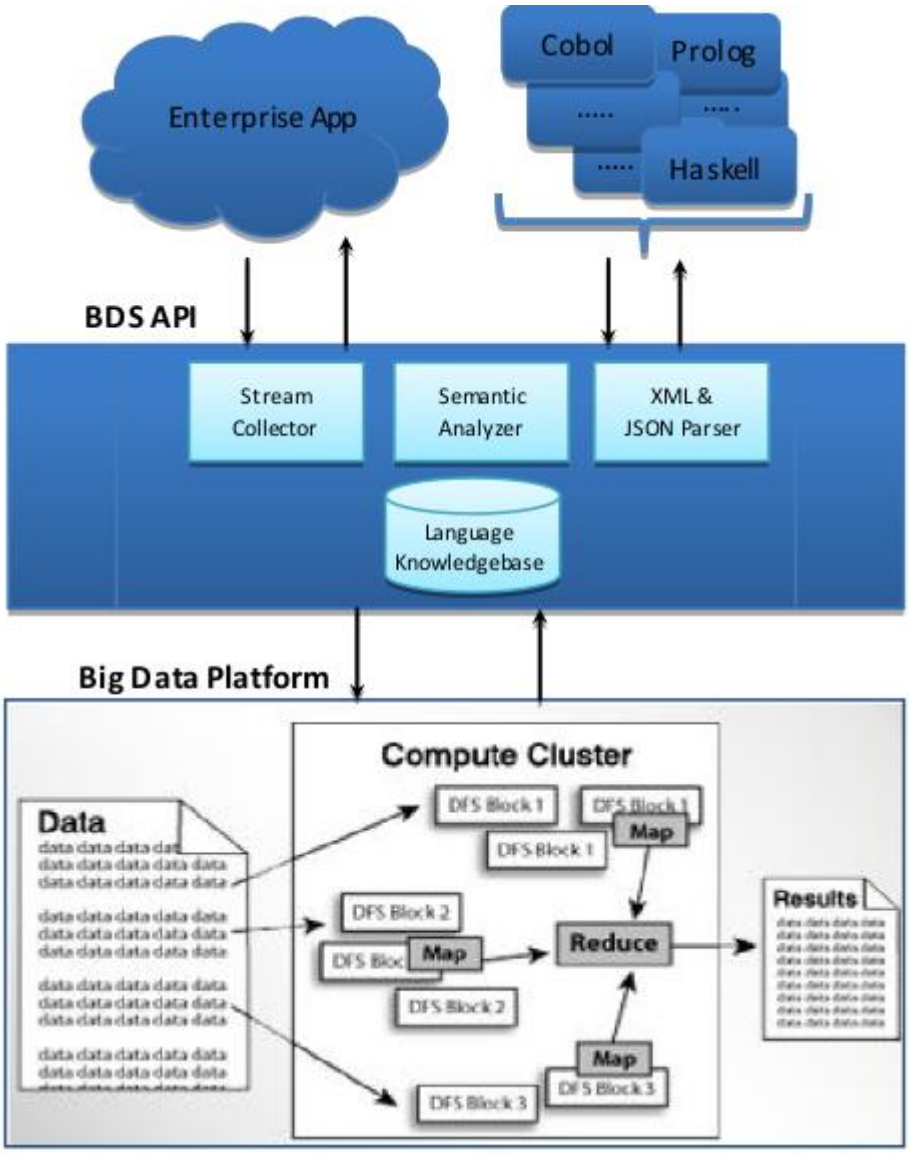
O trabalho desenvolvido por Malcolm et al. (2014) aborda a importância da infraestrutura junto ao *Big Data*, explica sobre o padrão de tecnologia das grandes empresas como *Amazon*, *Apple*, *Facebook*, *Twitter*, *Netflix* e *Google* para que possam coletar e analisar grandes volumes de dados, visto que não seria possível com um padrão de arquitetura convencional.

Uma API (Interface de Programação de Aplicativo) é composta por regras e instruções, recursos que *softwares* terceiros poderão acessar conforme a disponibilidade da mesma. Esta é formada por um conjunto de métodos e chamadas que ditam o uso dos métodos disponibilizados conforme Malcolm et al. (2014) mostram no trabalho.

Uma das formas de facilitar a manipulação de dados é a utilização de uma BDS API (*Big Data Services*), que é independente da plataforma de linguagem, o que permite que se conecte com qualquer serviço, além de ser focada principalmente em processamento, comunicação, armazenamento e compartilhamento. Permite o cruzamento do processamento através de gerenciamento de fluxos e que seja dividido os dados de processamento em vários clusters.

A comunicação entre a API e serviços terceiros ocorrem através de *Extensible Markup Language* (XML) e *JavaScript Object Notation Análise* (JSON), um exemplo mostra na figura 17 abaixo.

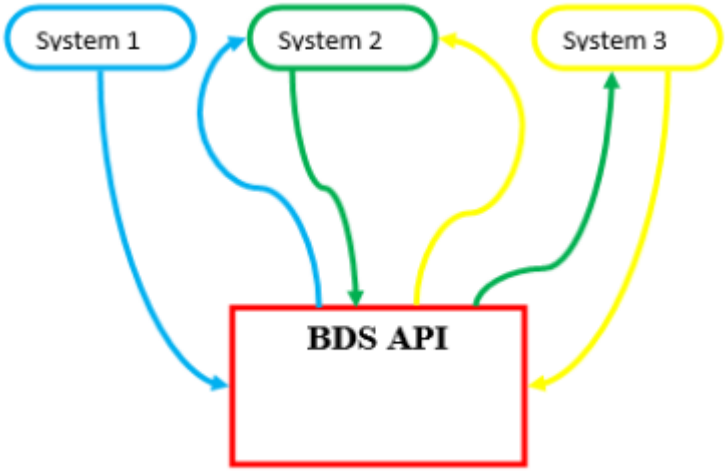
Figura 17 - BDS API Function.



Fonte: Malcolm et al. (2014).

Na parte de comunicação entre diversos aplicativos, os aplicativos terceiros o autor explica que independente da linguagem e tecnologia, a chamada da API retorna os dados solicitados através de JSON e XML, que com esses dados o *software* terceiro poderá tratar os dados de forma que achar melhor. Um exemplo dessa chamada encontra-se na figura 18.

Figura 18 - Sistema de comunicação usando BDS-API



Fonte: Malcolm et al. (2014).

Neste trabalho apresentado pelos autores tem-se de forma clara que para a aplicação de uma API, não precisa se preocupar com tais tecnologias a serem utilizadas por terceiros, mas sim em disponibilizar os dados através de XML ou JSON.

Também mostra que dependendo da quantidade de dados, é melhor a utilização de diversos clusters para garantir uma boa infraestrutura e garantia dos dados como também a utilização do componente API BDS. Pois, o autor conclui que a utilização de diversos clusters faz com que distribua os processos de forma adequada, como também a utilização da API BDS que contribui para um melhor gerenciamento do fluxo das solicitações externas. Mediante os dados levantados no trabalho citado, a infraestrutura adequada é de extrema importância para garantir a integridade das comunicações dentre sistemas externos.

3.5 Modelo UML de uma API padrão para aplicativo de computação em desenvolvimento em nuvem

O artigo proposto por Escalera e Chávez (2012) propõe um padrão de interface de programação de aplicativos (API) utilizado para aplicativos comuns na nuvem. Tem como principal objetivo a facilidade de portabilidade no desenvolvimento em nuvens,

que inclui análise, *design* e implementação com UML (Linguagem de modelagem unificada).

Para isso, é preciso identificar os serviços comuns fornecidos pelas nuvens que tem como objetivo resolver problemas e atender certos requisitos do usuário. Neste artigo, o autor propõe agrupar as necessidades de oito serviços mais gerais para aplicações *web*, sendo elas: Serviço de cotas, armazenamento persistente, gerenciamento de banco de dados, segurança, comunicação, gerenciamento de usuários, acesso ao e-mail e processamento de dados.

A API foi implementada em uma versão abstrata usando o modelo de *design*, ou seja, ainda não foram implementados os serviços mas, sim as classes abstratas para moldar os serviços. Esta cresce junto com o modelo de *design*, diversas classes vão sendo adicionadas e trazem as funções necessárias para a API, que, de forma abstrata não contém código que realmente lida com os recursos, chamando-se de SPI (Serviço Interface do Provedor).

Os autores explicam que o resultado deste projeto é uma API, um *software* como diversos no mercado, assim como seu processo de desenvolvimento, mas com a essência de facilitar o desenvolvimento de aplicativos em nuvem, logo a importância de se projetá-la com qualidade.

Nas conclusões dos autores do trabalho, mostra-se as possibilidades e capacidades de manter um padrão de projeto para desenvolvimento de diversas API, como exemplos utilizados, existem várias opções que utilizam o mesmo padrão, com uma boa documentação, desenvolvimento e fácil manutenção no código. Além disso, ressaltam as importâncias de projetar o crescimento do projeto, através de um planejamento de versões que trará uma boa visão de plataforma, que contribui com pontos positivos para os usuários.

Um projeto é bem desenvolvido para uma organização, por possibilitar planejar seu projeto e desenvolver com qualidade e padronização, que trarão um melhor produto e crescimento.

Tabela 2 - Comparativos Trabalhos Relacionados

Trabalho Relacionado	**Tema**	**Tecnologia**	**Objetivo**
Open Data Processor: Uma ferramenta para a automatização do processo de extração e publicação de Dados Abertos	Desenvolvimento de uma ferramente para facilitar a integração e disponibilização de dados públicos	API	Disponibilizar os dados através de uma API
Inovação dos negócios por meio da análise de Big Data	Uso prático do *Big Data* como um diferencial	*Big Data*	Converter um grande volume de dados em informações importantes
Fundamentos e aplicação do Big Data: como tratar informações em uma sociedade de yottabytes	Significado de *Big Data*	*Big Data*	Administrar uma quantidade de dados de grande volume a todo o momento
Aumentando a acessibilidade aos sistemas de Big Data por meio de um API de serviços comuns.	Facilitar a manipulação de dados é a utilização de uma BDS API (Big Data Services)	API	Foco em processamento, comunicação, armazenamento e compartilhamento
Modelo UML de uma API padrão para aplicativo de computação em desenvolvimento em nuvem	Padrão de interface de programação de aplicativos (API)	API	facilidade de portabilidade no desenvolvimento em nuvens, que inclui análise, design e implementação com UML

Fonte: Autor (2021)

De acordo com a tabela 2, os trabalhos relacionados tem seus objetivos diferentes mas que tem algum problema em comum entre eles, e relacionado ao tema do meu trabalho, completa todas as áreas necessárias para a aplicação do estudo. Contém estudo relacionado à busca de dados públicos na *web*, mostra o significado de *Big Data* e a utilização de uma API, que, ao utilizar todos esses temas, é possível pôr em prática o trabalho desenvolvido.

‘

4 MATERIAIS E MÉTODOS

4.1 Pesquisa quanto aos métodos científicos

Apresenta-se neste capítulo, os métodos utilizados no desenvolvimento do trabalho, tendo como base os conhecimentos teóricos.

A sessão será subdividida em quatro grupos, sendo elas: Pesquisa enquanto à natureza, Pesquisa enquanto ao Modo de abordagem, Pesquisa enquanto aos Objetivos e Pesquisa enquanto aos Procedimentos técnicos.

4.1.1 Pesquisa quanto à Natureza

A pesquisa aplicada apresenta diversos pontos diretamente com a pesquisa, depende de descobertas e enriquece com o desenvolvimento. Conforme Gil (2019), tem como característica o interesse na aplicação, utilização e consequências dos conhecimentos práticos. Prodanov e Freitas (2013) levam o contexto da pesquisa aplicada para o mesmo sentido, além disso, afirmam que esse método de pesquisa é utilizado para soluções de problemas específicos.

Por se tratar de um trabalho que envolverá a prática do desenvolvimento de uma aplicação, é importante o uso da pesquisa aplicada para a implementação do

software, ou seja, todo o tema estudado será implementado para chegar em uma certa conclusão.

4.1.2 Pesquisa quanto a Abordagem

O trabalho apresentará uma estratégia qualitativa. Segundo o autor Mascarenhas (2012), a abordagem qualitativa não é formada por etapas engessadas, tem a liberdade para que o pesquisador possa desenhar seu estudo de forma mais adequada, mas sempre deverá manter a estrutura do trabalho de forma coerente e sólida. Segundo Prodanov e Freitas (2013), a abordagem qualitativa tem como objetivo usar a visão positiva para o desenvolvimento, podendo usar suas ideias e estudos para enfatizar os resultados, diferente do modo quantitativo na qual são informações que precisam ser medidas e precisas.

A estratégia qualitativa será essencial para entender os dados dos fundos de investimentos e tratar de uma melhor forma, não se limitando a dados de resultados matemáticos, mas sim em conteúdo e valorização através dos dados.

4.1.3 Pesquisa quanto aos Objetivos

De acordo com Prodanov e Freitas (2013), a pesquisa exploratória possui um planejamento flexível, permite o estudo do tema em diversos aspectos. Nele, envolve levantamento bibliográfico, entrevistas com pessoas que tiveram experiências com o problema pesquisado e análise de exemplos. Já o autor Gil (2019), leva para o lado em que a pesquisa exploratória tem como principal objetivo em desenvolver, esclarecer e modificar ideias e conceitos. Em todos os tipos de pesquisa, esta é a que apresenta a menor rigidez no planejamento. Além disso, a pesquisa exploratória consiste na primeira etapa de uma investigação mais ampla, torna-se necessário quando se tem um tema escolhido bastante genérico o que torna necessário o esclarecimento e delimitação.

Para o trabalho apresentado, é importante o uso da pesquisa exploratória como escopo inicial do projeto de estudo, por se tratar de aspectos tecnológicos e esta haver uma grande abrangência sobre o planejamento e desenvolvimento, porém com o desenvolvimento do projeto e da API tem-se a natureza aplicada.

4.1.4 Pesquisa quanto aos Procedimentos Técnicos

O trabalho utilizará os procedimentos técnicos Pesquisa Bibliográfica e Experimental.

O autor Mascarenhas (2018) explica que a pesquisa bibliográfica concentra-se em estudos de livros, artigos, dicionários e enciclopédias, estudos em fontes que tem um resultado final já aplicado, segundo ele, esse método tem muita vantagem por fazer pesquisas em fontes ricas em quantidade de informações, porém quando se trata de pesquisa de amostra de dados, o desenvolvimento com esse método se torna limitado. Prodanov e Freitas (2013) leva o assunto sobre Pesquisa Bibliográfica para o mesmo caminho, mas salienta que precisa-se tomar cuidado de onde está buscando as informações, pois precisa ter uma confiabilidade na fonte que está buscando, na qual é necessário verificar a veracidade do conteúdo, o que não seria necessário para a pesquisa documental. Já o método experimental, segundo Gil (2012), concentra em submeter objetos de estudos sobre influência de certos fatores, em condições contidas pelo investigador, para observar o resultado que produz no objeto. Conforme Prodanov e Freitas (2013), não seria exagero considerarmos que parte dos conhecimentos alcançados nos últimos três séculos, deve-se ao emprego do método experimental.

Será utilizado a pesquisa bibliográfica neste trabalho devido às fontes que será usada nas pesquisas, por tratar de assuntos que envolvem tanto tecnologia como também a área financeira, necessita de confiabilidade nas fontes dos dados. Já o método experimental, visa por alcançarmos informações além do que está escrito, ser preciso mexer nas variáveis dos dados obtidos para encontrar novos resultados.

4.2 Tecnologias

Como o principal desafio está relacionado com o banco de dados e a alta disponibilidade, as tecnologias utilizadas para o desenvolvimento será algo gratuito e bastante consolidada. Para isso, será desenvolvido com a linguagem de programação Java junto com a biblioteca Spring Framework. Além disto, foi escolhido o Maven para auxiliar na automação do projeto, por ter diversos plugins, facilitará no desenvolvimento do projeto.

Como foco principal, o banco de dados escolhido foi o Postgres, devido ao tamanho e qualidade do banco e por se tratar de um banco totalmente gratuito, não existe nenhuma limitação relacionada a volume de dados ou limitação de usuários.

Por último, foi escolhido como ferramenta para fazer as requisições da API, será utilizado o Postman, por também se tratar de uma ferramenta totalmente gratuita e muito semelhante a qualquer outra disponível da internet.

4.2.1 Banco de dados PostgreSQL

PostgreSQL é um sistema de gerenciamento de banco de dados, conforme Secco (2017), é um banco de código aberto avançado, teve início de sua implementação em 1986 e passou por diversas atualizações. Este sistema pode ser usado e distribuído por qualquer pessoa para finalidades privados, comerciais ou acadêmicos. O postgreSQL é capaz de suportar quase toda parte de um padrão SQL, oferece características modernas como: consultas complexas, controle de concorrência, chaves estrangeiras, e integridade transacional. Além disto, com o PostgreSQL, é possível programar pelo próprio usuário de várias maneiras como: adicionar tipos de dados, funções, funções agregadas, operadores e índice de linguagens.

‘

4.2.2 Linguagem de programação Java

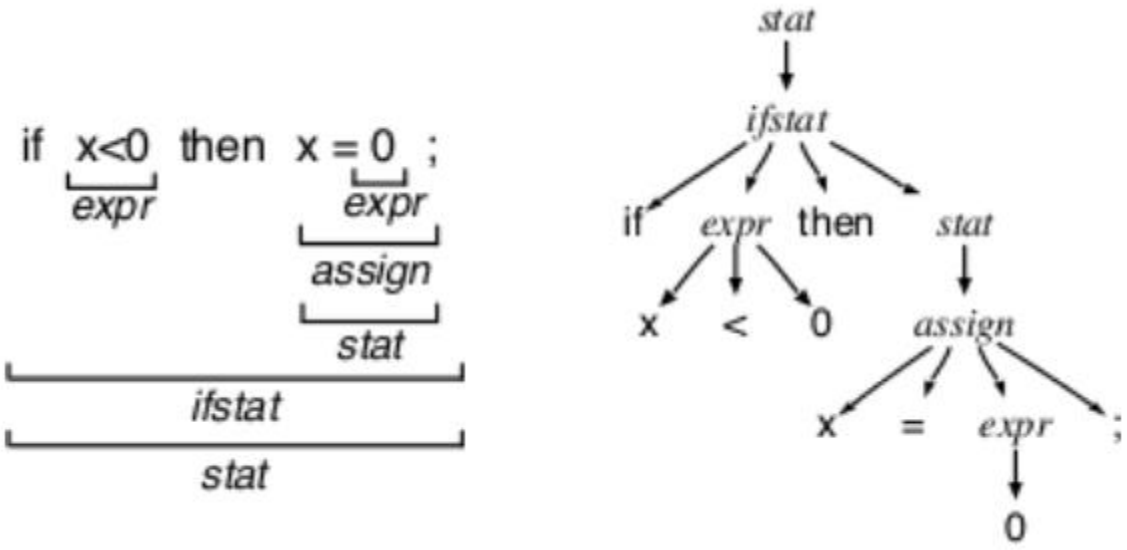
Java é uma linguagem de programação orientada a objetos, feita para ser independente de plataforma, o desenvolvimento em Java faz com que seja necessária uma máquina virtual (JVM) para executar. Conforme Cavalcanti e Almeida (2016), devido à JVM, faz com que possa ser executada em diversos ambientes computacionais, apenas como requisito a JVM instalada para rodar.

Com o passar do tempo, o Java passou por diversas atualizações e em 2004 passou a ser uma linguagem de programação pública, segundo Cavalcanti e Almeida (2016), foi nomeada como Java 5.0. Na qual obteve uma mudança de bastante significância para quem desenvolvia utilizando a linguagem, incluindo bibliotecas mais eficientes.

Como qualquer linguagem de programação, é composta por sentenças válidas que contém regras que são definidas e respeitadas, como explica Cavalcanti e Almeida (2016), é semelhante a uma gramática em qualquer idioma. Pois, são compostas por frases na qual cada frase é é definida com um conjunto de símbolos e sub-frases que caracterizam a linguagem utilizada.

No momento em que símbolos são agrupados de forma correta, a linguagem será convertida em instruções de máquina, de forma similar como o inglês e português, na qual o vocabulário é composto por verbos, nomes entre outros, de acordo com Cavalcanti e Almeida (2016), a linguagem na programação não é diferente, a linguagem possui diferentes símbolos com diversas regras para que se possa efetuar uma comunicação com o computador, podem ser variáveis, operadores e entre outros. Pode ser identificado uma sequência como a seguinte expressão “if x<0 then x = 0”. na figura 19 mostra uma representação de operadores.

Figura 19 - Fluxograma sentenças de linguagem



Fonte: Cavalcanti e Almeida (2016).

Na figura 19, mostra o primeiro estágio na qual é realizada um agrupamento de classes que compõem uma sentença e um segundo estágio preservando a hierarquia de classes e símbolos definidos na gramática da linguagem Cavalcanti e Almeida (2016).

4.2.3 Postman

Postman é uma ferramenta que auxilia na documentação nas requisições feitas na API. Conforme Versiani (2020), com a utilização do Postman, facilitará o uso de API construindo solicitações mais eficientes sendo possível analisar as respostas enviadas pela API. Um exemplo da utilização de Postman é uma solicitação na API de dados da empresa através do Postman pela instrução GET, caso fosse fazer um teste através de programação, seria necessário diversas linhas de programação para fazer a requisição na API como também uma interface visual para poder ver os dados retornados. Já na com o Postman, o que precisa fazer para testar a requisição seria apenas seguir alguns passos de configuração conforme os passos abaixo:

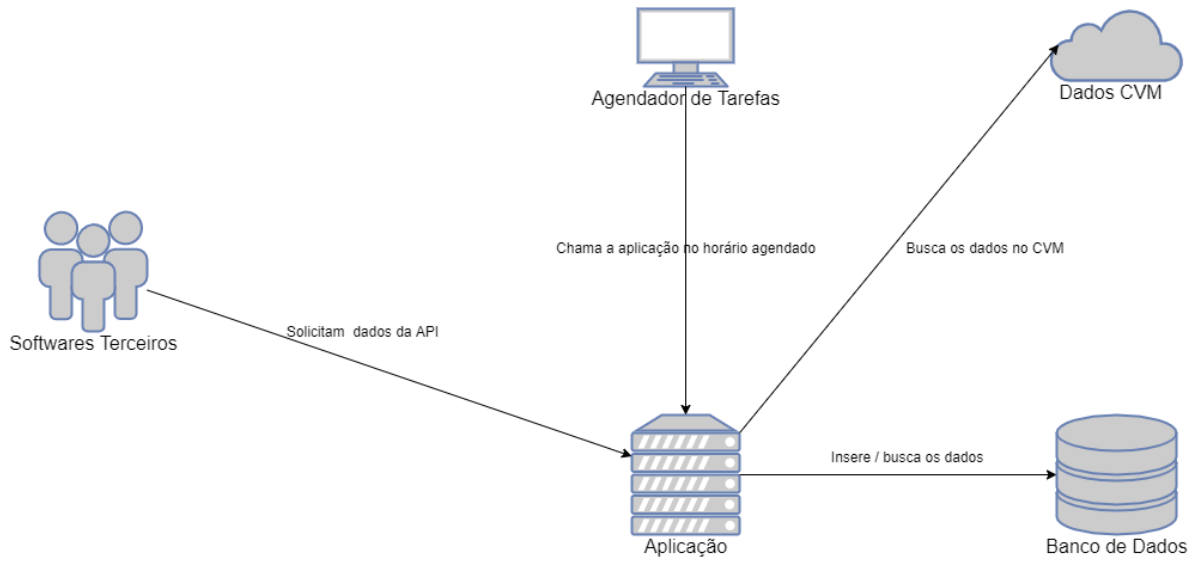
- Obter a conexão da rota à barra de endereços;
- Selecionar o método GET;
- Digitar a chave da API;

- Especificar o formato de resposta (JSON, XML entre outros);
- Enviar a chamada.

4.3 Desenvolvimento

Nesta seção serão apresentados os fundamentos e etapas do desenvolvimento das aplicações. Na figura 20, demonstra o diagrama de processo do funcionamento do trabalho.

Figura 20 - Diagrama de processo



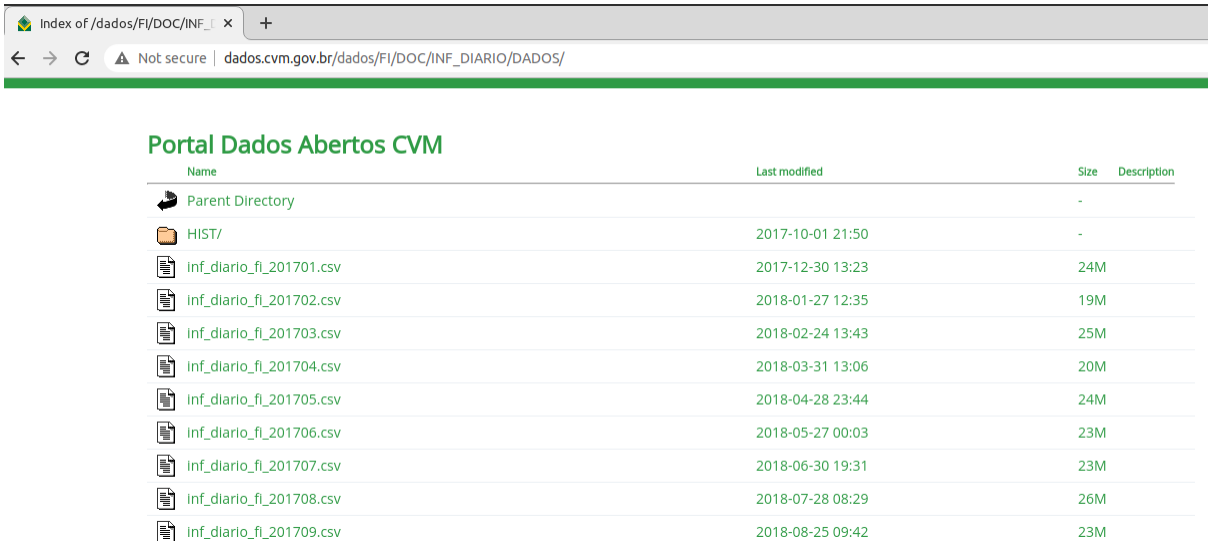
Fonte: Do autor (2020).

O trabalho consiste na criação de uma API, nela, terá registros de informações dos fundos de investimentos com atualizações diárias. Ao fazer uma consulta na API, será retornado os dados em formato de JSON de acordo com as delimitações das consultas.

Para que tenha os dados na aplicação e disponibilize os dados para os terceiros através da API, é preciso encontrar essas informações em algum local com fonte confiável que tenha os registros dos fundos para que se possa armazenar no banco de dados.

A fonte de dados que será utilizada para buscar as informações dos fundos de investimentos será no próprio site do governo DADOS CVM (Comissão de Valores Mobiliários), conforme figura 21, que é uma fonte oficial do governo utilizada para deixar os dados acessíveis. Nele se encontram apenas dados abertos, não contendo informações e dados privados. Os dados de fundos de investimentos sofrem uma atualização diária, contendo as informações com dois dias de atraso.

Figura 21 - Dados diários CVM



Fonte: CVM(2021).

Após buscar as informações no site do governo, foi inserido no banco de dados para manter os registros a todo momento, os dados sofrerão alterações/ novas inserções todos os dias no período da manhã, na qual todos os dias terão registros de informações sobre os fundos atualizadas.

O gatilho da busca das informações no site do governo acontece através do agendador do sistema operacional na qual chama a aplicação para fazer o processo de busca e inserção no período configurado no agendador.

4.3.1 Consultas através da API

A API disponibilizada tem disponível algumas consultas como:

- Cota de um determinado fundo para um determinado dia;
- Informações cadastrais de um determinado fundo.

4.4 Testes e análise dos resultados

Durante o desenvolvimento foram avaliados dados de extrato dos fundos de investimentos disponibilizados pelo site do governo para avaliar o padrão e estrutura dos dados para a modelagem do banco de dados. A imagem mostrada na figura 22, mostra o arquivo no formato CSV com os registros históricos diários de cada fundo de investimento avaliado. Na avaliação dos dados históricos, foi identificado que cada fundo de investimento tem o seu respectivo CNPJ que se repete para cada data de competência.

Figura 22 - Resultado dados fundos de investimento

CNPJ_FUNDO	DT_COMPTC	VL_TOTAL	VL_QUOTA	VL_PATRIM_LIQ	CAPTC_DIA	RESG_DIA	NR_COTST
00.017.024/0001-53	01/06/2020	1123668.44	27.473.249.900.000	1118313.83	0.00	0.00	1
00.017.024/0001-53	02/06/2020	1123796.81	27.474.873.700.000	1118379.93	0.00	0.00	1
00.017.024/0001-53	03/06/2020	1123923.29	27.476.474.500.000	1118445.09	0.00	0.00	1
00.017.024/0001-53	04/06/2020	1124051.70	27.478.028.300.000	1118508.34	0.00	0.00	1
00.017.024/0001-53	05/06/2020	1123870.79	27.479.629.800.000	1118573.53	0.00	0.00	1
00.017.024/0001-53	08/06/2020	1123999.18	27.481.230.300.000	1118638.68	0.00	0.00	1
00.017.024/0001-53	09/06/2020	1124127.59	27.482.831.600.000	1118703.86	0.00	0.00	1
00.017.024/0001-53	10/06/2020	1124256.02	27.484.433.600.000	1118769.07	0.00	0.00	1
00.017.024/0001-53	12/06/2020	1124384.47	27.486.035.600.000	1118834.28	0.00	0.00	1
00.017.024/0001-53	15/06/2020	1115774.06	27.487.637.800.000	1115869.04	0.00	3030.46	1
00.017.024/0001-53	16/06/2020	1121539.54	27.489.174.400.000	1115931.42	0.00	0.00	1
00.017.024/0001-53	17/06/2020	1121667.68	27.490.774.400.000	1115996.37	0.00	0.00	1
00.017.024/0001-53	18/06/2020	1121795.83	27.492.374.100.000	1116061.31	0.00	0.00	1
00.017.024/0001-53	19/06/2020	1121891.42	27.493.172.400.000	1116093.72	0.00	0.00	1
00.017.024/0001-53	22/06/2020	1121367.31	27.494.026.200.000	1116128.38	0.00	0.00	1
00.017.024/0001-53	23/06/2020	1121458.08	27.501.006.100.000	1116411.73	0.00	0.00	1
00.017.024/0001-53	24/06/2020	1121554.61	27.501.632.500.000	1116437.16	0.00	0.00	1
00.017.024/0001-53	25/06/2020	1121612.50	27.502.426.500.000	1116469.39	0.00	0.00	1
00.017.024/0001-53	26/06/2020	1121708.08	27.503.079.300.000	1116495.89	0.00	0.00	1
00.017.024/0001-53	29/06/2020	1121794.58	27.503.876.900.000	1116528.27	0.00	0.00	1
00.017.024/0001-53	30/06/2020	1121892.09	27.504.556.500.000	1116555.86	0.00	0.00	1
00.068.305/0001-35	01/06/2020	52139566.09	27.293.878.000.000	52118344.60	24116.09	29093.84	7638
00.068.305/0001-35	02/06/2020	52162343.91	27.297.560.000.000	52152672.02	37140.00	9843.97	7638
00.068.305/0001-35	03/06/2020	52145672.94	27.300.141.000.000	52133032.33	15445.12	40016.83	7639
00.068.305/0001-35	04/06/2020	51938774.48	27.304.749.000.000	51923006.33	3550.00	222374.74	7638
00.068.305/0001-35	05/06/2020	51946649.15	27.305.199.000.000	51927925.70	7200.00	3136.38	7636

Fonte: Do autor (2021).

Após avaliar os padrões históricos dos fundos de investimento, foi criado um banco de dados e suas tabelas para que fosse possível armazenar os dados

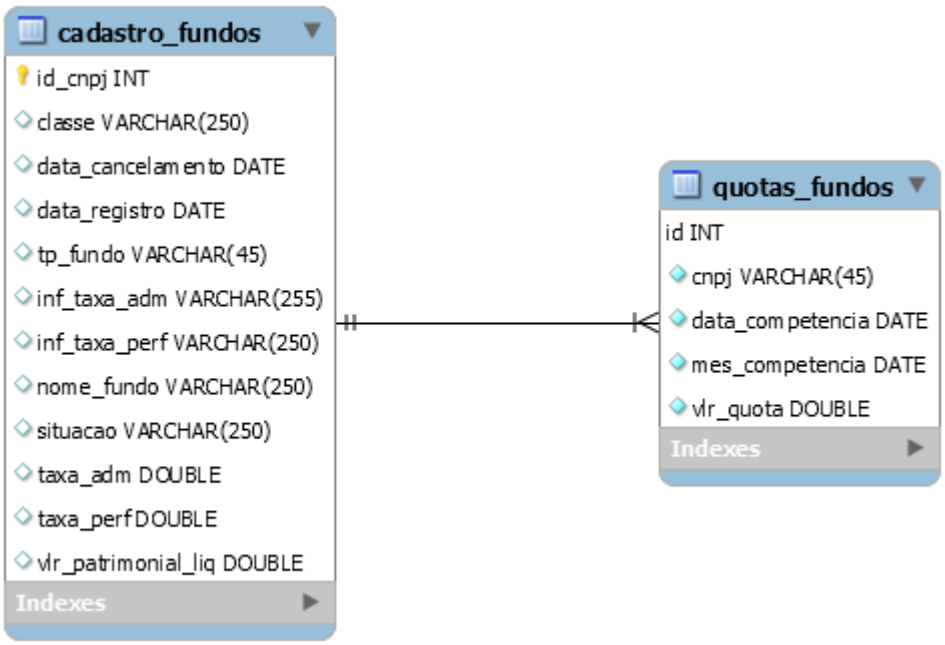
consultados no site do governo e também fazer diferentes testes de desempenho, o modelo do banco utilizado para os testes se encontra na figura 23. Na versão final dos testes do banco de dados, foi utilizado na tabela de *quotas_fundos* (dados históricos de fundos) com uma chave primária ID de forma sequencial e o CNPJ para junção com a tabela de cadastro de fundos para buscar informações cadastrais do fundo.

Na figura 23, foi modelado o banco de dados pensando na separação dos dados cadastrais dos fundos de investimentos na tabela *cadastro_fundos,* nessa tabela, tem informações como o CNPJ (chave primária), nome do fundo, data da criação do fundo, o tipo do fundo, taxa de administração e performance entre outros.

Já na tabela *quotas_fundos*, tem como objetivo cadastrar as movimentações diárias de cada fundo, devido essa modelagem, não foi preciso gravar as informações cadastrais dos fundos, evitando um maior volume de dados repetidos dentro de uma única tabela, isso é possível por ter como chave principal na tabela *quotas_fundos* um ID sequencial e o CNPJ que tem ligação direta com a tabela *cadastro_fundos*. Além disso, tem os campos da data da movimentação, mês de competência e o valor da quota que é a informação principal para saber qual é o valor do fundo no determinado dia.

Esse modelo apresentado na figura 23, também foi pensado no conceito posto pelo autor Costa (2017, p.14) sobre o tema de *Big Data*, visto que tem como o objetivo de alto volume, velocidade e variedade de informação.

Figura 23 - Modelo banco de dados



Fonte: Do autor (2020).

A seguir à criação do banco de dados, foi feita uma aplicação para buscar os dados do site do governo de forma automática, ler o arquivo CSV e inserir no banco de dados. Nesse processo foram inseridos dados de extrato de fundos de um mês fechado, levando em consideração o mês de junho de 2020 na qual foram inseridos 371142 registros de transações de fundos de investimentos na tabela de *quotas_fundos*. O processo de inserção dos dados foram de um bloco único e teve um tempo de inserção bastante agravado, devido a grande quantidade de inserções, acarretou travamentos na aplicação e no banco de dados, desta forma, é preciso avaliar processos de inserções em uma quantidade de blocos para evitar o grande processamento no banco e possíveis problemas com interrupções.

Após o processo de inserção, nos testes de consulta buscando o valor da cota de um determinado fundo e de um determinado dia utilizando nas cláusulas where ID (CNPJ) e data_competencia, teve o retorno das informações em menos de 1 segundo, um tempo que pode ser melhorado, isso aconteceu devido a montagem das chaves com as principais informações utilizadas nas cláusulas *where* que faz com que o sistema de banco de dados transforme em índice melhorando a performance do banco de dados.

‘

Por último, foi feito uma aplicação para disponibilização dos dados que puderam ser consultados através de uma chamada GET, teste realizado através de uma ferramenta chamada Postman, percebe-se que o retorno dos dados utilizado na consulta foi com o tempo paralelo ao da consulta, visto que o processo entre ter os dados na consulta e transformar em JSON não impactou em tempo, teste inicial e limitado a localhost, na qual não podemos deixar de pensar que gargalos em rede como tempo de respostas e múltiplas requisições afetarão o tempo de respostas em uma evolução do trabalho.

5. RESULTADOS PRELIMINARES OU ESPERADOS

Neste capítulo serão apresentados os resultados obtidos que incluem a implementação da API, demonstração dos recursos disponíveis como também o desempenho das informações buscadas com um grande volume de informações gravados no banco de dados.

5.1 Análise dos resultados

Neste tópico, serão apresentados os resultados obtidos inicialmente pela busca das informações através do site do governo, seguindo com a inserção das informações no banco de dados, posteriormente a API e os testes de consultas analisando o desempenho na aplicação para retornar os dados obtidos com um grande volume de dados gravados no banco de dados.

5.1.1 Dados de inserção cadastro de Fundos de Investimentos

Em primeiro momento, foram avaliado os dados cadastrais dos fundos de investimentos para entender a padronização dos dados para que fosse possível fazer a inserção deles. Nessa etapa, foi preciso analisar os dados de acordo com os

arquivos disponibilizados no site do governo, entender a estrutura dos dados e o que foi preciso tratar de informações para que fosse possível desenvolver uma aplicação em cima dessas informações.

Ao analisar os dados, percebeu-se que o mesmo CNPJ constava mais de um registro no cadastro, o que acabava afetando a ideia de utilizar o CNPJ como chave primária no banco de dados na tabela *cadastro_fundos*. Depois de testar essas situações, percebeu-se que em diversos momentos o mesmo fundo tinha sido cancelado e sido criado um novo, não havendo a necessidade de inserir os cadastros cancelados para o banco de dados conforme figura 24

Figura 24 - Cadastros duplicados cancelados CNPJ 01.603.578/0001-03

	B	C	D	E	F	G	H
1	CNPJ_FUNDO	DENOM_SOCIAL	DT_REG	DT_CONS	CD_CV	DT_CANC	SIT
5845	01.603.578/0001-03	ICATU VANGUARDA XI FUNDO DE INVESTIMENTO REND	23/03/2005	10/01/1997	34924		EM FUNCIONAMENTO NORMAL
51568	01.603.578/0001-03	ICATU VANGUARDA XI FUNDO DE INVESTIMENTO REND	30/04/2003	10/01/1997	9202	11/03/2005	CANCELADA

Fonte: Autor (2021).

Além disso, identificou-se que existe outro motivo de duplicidade de registros para alguns CNPJ, levando em consideração um exemplo como o CNPJ 01.392.020/0001-18, percebeu-se que um mesmo fundo pode haver diversos gestores, mas que no cenário do trabalho, não é levado em consideração os gestores de cada fundo, desta forma, é preciso ignorar essas duplicações devido aos gestores sem cadastrar o mesmo fundo duas vezes, conforme a figura 25.

Figura 25 - Cadastros duplicados gestores CNPJ 01.392.020/0001-18

B	C	D	E	F	G	H	I	J
CNPJ_FUNDO	DENOM_SOCIAL	DT_REG	DT_CONS	CD_CV	DT_CANC	SIT	CPF_CNPJ_GESTOR	GESTOR
01.392.020/0001-18	BRADESCO FUNDO DE INVESTI	09/04/2005	30/09/1996	52043		EM FUNCIO	51.990.695/0001-37	BRADESCO VIDA E PREVIDÊNCIA S.A.
01.392.020/0001-18	BRADESCO FUNDO DE INVESTI	09/04/2005	30/09/1996	52043		EM FUNCIO	62.375.134/0001-44	BRAM - BRADESCO ASSET MANAGEM

Fonte: Autor (2021)

5.1.2 Dados de Inserção Quotas Fundos

Nessa etapa, foi necessário analisar os dados do arquivo do site do governo relacionado às quotas de fundos de investimentos, foi preciso avaliar a padronização

dos dados registrados como também o comportamento de cada período para identificar diferenças e possíveis problemas de formatação de dados.

Ao analisar os registros das quotas de fundos, percebeu-se que não existe uma padronização fixa nos arquivos, na qual dependendo do mês do arquivo, a ordem das colunas podem sofrer alterações.

Figura 26 - Quotas de fundos de Janeiro/2020

	A	B	C	D	E	F	G	H
1	CNPJ_FUNDO	DT_COMP	VL_TOTAL	VL_QUOTA	VL_PATRIM_LIQ	CAPTC_DIA	RESG_DIA	NR_COT
2	00.017.024/0001-53	02/01/2020	1132491.66	27.225.023.000.000	1123583.00	0.00	0.00	1
3	00.017.024/0001-53	03/01/2020	1132685.12	27.224.496.000.000	1123561.25	0.00	0.00	1
4	00.017.024/0001-53	06/01/2020	1132881.43	27.225.563.600.000	1123605.31	0.00	0.00	1
5	00.017.024/0001-53	07/01/2020	1133076.85	27.226.700.700.000	1123652.24	0.00	0.00	1
6	00.017.024/0001-53	08/01/2020	1132948.59	27.227.815.800.000	1123698.26	0.00	0.00	1
7	00.017.024/0001-53	09/01/2020	1133142.13	27.228.907.600.000	1123743.32	0.00	0.00	1
8	00.017.024/0001-53	10/01/2020	1132395.91	27.229.977.200.000	1123787.46	0.00	0.00	1
9	00.017.024/0001-53	13/01/2020	1129159.92	27.231.043.300.000	1120401.09	0.00	3430.37	1
10	00.017.024/0001-53	14/01/2020	1129352.82	27.232.123.900.000	1120445.55	0.00	0.00	1
11	00.017.024/0001-53	15/01/2020	1129545.75	27.233.181.900.000	1120489.08	0.00	0.00	1
12	00.017.024/0001-53	16/01/2020	1129739.65	27.234.240.100.000	1120532.62	0.00	0.00	1
13	00.017.024/0001-53	17/01/2020	1129932.64	27.235.322.400.000	1120577.15	0.00	0.00	1
14	00.017.024/0001-53	20/01/2020	1126010.73	27.236.382.800.000	1120620.78	0.00	0.00	1
15	00.017.024/0001-53	21/01/2020	1126203.08	27.238.676.700.000	1120715.16	0.00	0.00	1

Fonte: Autor (2021).

Ao analisar a figura 26, percebemos que a coluna A da planilha é *CNPJ_FUNDO* e a coluna B *DT_COMP*.

Figura 27 - Quotas de fundos de Abril/2020

	A	B	C	D	E	F	G	H	I
1	TP_FUNDO	CNPJ_FUNDO	DT_COMPTC	VL_TOTAL	VL_QUOTA	VL_PATRIM_LIQ	CAPTC_DIA	RESG_DIA	NR_COTST
2	FI	00.017.024/0001-53	01/04/2020	1127984.12	27.380.980.800.000	1121661.69	0.00	0.00	1
3	FI	00.017.024/0001-53	02/04/2020	1128125.49	27.383.353.300.000	1121758.88	0.00	0.00	1
4	FI	00.017.024/0001-53	03/04/2020	1128288.77	27.385.179.200.000	1121833.68	0.00	0.00	1
5	FI	00.017.024/0001-53	06/04/2020	1128434.93	27.387.583.000.000	1121932.15	0.00	0.00	1
6	FI	00.017.024/0001-53	07/04/2020	1128235.55	27.389.568.800.000	1122013.50	0.00	0.00	1
7	FI	00.017.024/0001-53	08/04/2020	1128400.75	27.391.345.000.000	1122086.26	0.00	0.00	1
8	FI	00.017.024/0001-53	09/04/2020	1127600.46	27.393.795.800.000	1122186.66	0.00	0.00	1
9	FI	00.017.024/0001-53	13/04/2020	1123677.49	27.395.616.700.000	1118176.96	0.00	4084.29	1
10	FI	00.017.024/0001-53	14/04/2020	1123836.33	27.397.967.500.000	1118272.91	0.00	0.00	1
11	FI	00.017.024/0001-53	15/04/2020	1123993.28	27.400.272.400.000	1118366.99	0.00	0.00	1
12	FI	00.017.024/0001-53	16/04/2020	1124154.08	27.402.531.900.000	1118459.21	0.00	0.00	1
13	FI	00.017.024/0001-53	17/04/2020	1124314.91	27.404.884.400.000	1118555.23	0.00	0.00	1
14	FI	00.017.024/0001-53	20/04/2020	1123789.43	27.407.237.600.000	1118651.28	0.00	0.00	1
15	FI	00.017.024/0001-53	22/04/2020	1123939.71	27.409.572.000.000	1118746.56	0.00	0.00	1
16	FI	00.017.024/0001-53	23/04/2020	1124093.83	27.411.666.500.000	1118832.05	0.00	0.00	1
17	FI	00.017.024/0001-53	24/04/2020	1124250.83	27.413.855.600.000	1118921.40	0.00	0.00	1
18	FI	00.017.024/0001-53	27/04/2020	1124376.53	27.415.975.400.000	1119007.92	0.00	0.00	1
19	FI	00.017.024/0001-53	28/04/2020	1124528.78	27.418.258.600.000	1119101.11	0.00	0.00	1

Fonte: Autor (2021).

Ao analisar a figura 27, percebemos que a coluna A é *TP_FUNDO*, a coluna B *CNPJ_FUNDO* e a coluna C *DT_COMP*, ao comparar a figurar 26 e figura 27,

percebemos que houve um deslocamento de informações para a direita depois de um determinado mês.

Desta forma, foi preciso verificar como estavam as estruturas dos arquivos antes de fazer a inserção dos dados para evitar problemas de inserção devido às diferenças estruturais dos arquivos.

5.1.3 Gerador FI - Cadastro de Fundos

Nessa etapa dos resultados, umas das principais partes do trabalho, envolveu todas as regras relacionadas a limpeza, integração e transformação dos dados, nesta etapa, foi considerado toda a teoria estudada no item extração de Dados (ETL, BI) conforme Gomes, Oliveira e Belo (2014).

Mesmo que não tenha sido utilizado uma ferramenta terceira como um exemplo mostrado por Ferreira et al.(2010) o *Pentaho Data Integration* no tópico extração de Dados (ETL, BI) para fazer a parte de extração e integração dos dados, foi desenvolvida uma aplicação específica para extrair os dados do site do governo na qual passou por um processo de análise, limpeza e tratamento dos dados para depois inserir no banco de dados já modelado no tópico anterior.

Nesta etapa do projeto, o desenvolvimento incluiu a criação de uma aplicação denominada Gerador FI. Nesta aplicação, tem como objetivo inserir os dados consultados no site da CVM no banco de dados. A aplicação foi desenvolvida em Java com o *framework* Hibernate, cujo nela tem uma classe com toda a regra da carga dos dados com as respectivas validações e tratamentos necessários.

Na aplicação Gerador FI, foi implementado um método denominado *gerarCadFundos* na qual tem como objetivo gerar os dados cadastrais dos fundos de investimentos, nela, faz a busca em um *link* específico da CVM em que tem um arquivo de CSV com os dados cadastrais dos fundos de investimentos com as informações atualizadas. É preciso baixar o arquivo, armazenar em uma pasta da aplicação para depois fazer leitura linha por linha e gravar os dados no banco de dados conforme parte da figura 28 mostra.

Figura 28 - Inserção cadastro de fundos

```
URL url = new URL("http://dados.cvm.gov.br/dados/FI/CAD/DADOS/cad_fi.csv");
File file = new File("C:\\Users\\AndreiMileto\\Documents\\NetBeansProjects"
        + "\\GeradorFI\\libs\\Fundos\\Cadastro FI\\cad_fi.csv");
ReadableByteChannel rbc = Channels.newChannel(url.openStream());
FileOutputStream fos = new FileOutputStream(file);
fos.getChannel().transferFrom(rbc, 0, Long.MAX_VALUE);
fos.close();
rbc.close();

// imprimir na tela
BufferedReader reader = new BufferedReader(new InputStreamReader
(new FileInputStream("C:\\Users\\AndreiMileto\\Documents\\NetBeansProjects"
        + "\\GeradorFI\\libs\\Fundos\\Cadastro FI\\cad_fi.csv")));
String linha = null;
CadFIControl cadDAO = new CadFIControl();
linha = reader.readLine();
ArrayList<CadFi> listCads = new ArrayList();
while ((linha = reader.readLine()) != null) {
    CadFi cadFI = new CadFi();
    String[] dadosUsuario = linha.split(VIRGULA);
    if (dadosUsuario[1] != "CNPJ_FUNDO") {
        if (!dadosUsuario[7].equalsIgnoreCase("Cancelada")) {
            int retorno = 1;

            //verificar se o CNPJ já foi inserido
            for (int i = 0; i < listCads.size(); i++) {
                if (listCads.get(i).getIdCnpj().equals(dadosUsuario[1].repla
                    retorno = 2;
                }
            }
```

Fonte: Autor (2021)

No método *gerarCadFundos* mostrado na figura 28, inicia fazendo a busca em um *link* específico no site do governo, grava em uma pasta, depois inicia o processo de leitura linha por linha para fazer uma validação se o fundo da linha de leitura não está cancelado, além disso, verifica se esse fundo de investimento já foi cadastrado anteriormente para evitar problema de duplicações de registros. O processo de inserção dos cadastros dos fundos gerou uma quantidade de 26363 registros com um tempo de inserção dos dados de 06:15.

5.1.4 Gerador FI - Cadastro Quotas Fundos

Nesta etapa dos resultados, da mesma forma que a etapa anterior, uma das principais etapas do projeto, faz a leitura dos arquivos no site do governo, trata as informações e insere no banco de dados, nesta etapa a leitura é feita no arquivo de quotas de fundos de investimentos na qual existe um arquivo para cada mês que atualiza as informações de forma diária.

A inserção dos dados de Quotas de Fundos, funciona através do método *gerarQuotasFundos* da mesma aplicação, cujo nome é Gerador FI.

Nesse método, existe uma semelhança de funcionalidade comparada ao *gerarCadFundos*, porém existe algumas mudanças como o endereço de onde busca os dados, utiliza uma periodicidade dinâmica de acordo com a necessidade e também conta com campos diferentes no arquivo devido à necessidade do negócio conforme figura 29.

Figura 29 - Cadastro Quotas Fundos

```
URL url = new URL("http://dados.cvm.gov.br/dados/FI/DOC/INF_DIARIO/DADOS/"
        + "inf_diario_fi_" + periodo + ".csv");

File file = new File("C:\\Users\\AndreiMileto\\Documents\\NetBeansProjects"
        + "\\GeradorFI\\libs\\fundos\\quotas\\" + periodo
        + "\\inf_diario_fi_" + periodo + ".csv");
ReadableByteChannel rbc = Channels.newChannel(url.openStream());
FileOutputStream fos = new FileOutputStream(file);
fos.getChannel().transferFrom(rbc, 0, Long.MAX_VALUE);
fos.close();
rbc.close();

BufferedReader reader = new BufferedReader(new InputStreamReader
(new FileInputStream("C:\\Users\\AndreiMileto\\Documents\\NetBeansProjects"
        + "\\GeradorFI\\libs\\fundos\\quotas\\"
        + periodo + "\\inf_diario_fi_" + periodo + ".csv")));
String linha = null;
QuotaFundosControl quotasDAO = new QuotaFundosControl();
linha = reader.readLine();
String[] dadosUsuarioTest = linha.split(VIRGULA);

//verifica padronização do arquivo
if (dadosUsuarioTest[0].equals("CNPJ_FUNDO")) {
    while ((linha = reader.readLine()) != null) {
        QuotasFundos quotas = new QuotasFundos();
        String[] dadosUsuario = linha.split(VIRGULA);
        if (dadosUsuario[0] != "CNPJ_FUNDO") {
            quotas.setCnpj((dadosUsuario[0].replace(".", "").replace("/", "").rep
            DateFormat formatter2;
            Date data;
            formatter2 = new SimpleDateFormat("yyyy-MM-dd");
            data = formatter2.parse(dadosUsuario[1]);
            quotas.setMesCompetencia(mesCompetencia(data.getMonth() + 1));
            try {
                DateFormat formatter;
                Date date;
                formatter = new SimpleDateFormat("yyyy-MM-dd");
                date = formatter.parse(dadosUsuario[1]);
                quotas.setDataCompetencia(date);
            } catch (Exception e) {
            }

            quotas.setVlrQuota(Double.parseDouble(dadosUsuario[3]));
            quotasDAO.salvar(quotas);
        }
    }
```

Fonte: Autor (2021).

Como é possível visualizar na figura 29, a aplicação faz a consulta no link da CVM e armazena em uma pasta conforme o período passado por parâmetro, após isso, faz o processo de inserção segundo os dados consultados na pasta linha a linha.

Para uma inserção dos dados relativos a um período de um mês, foi usado como parâmetro o período 04/2021, que gerou uma quantidade de 406915 registros com um tempo de inserção de 1:55.

Avaliando o tempo de inserção x a quantidade de registros, percebe-se que o tempo para a inserção das quotas para cada registro é bem menor comparada aos cadastros dos fundos 1:55 x 6:15. Essa discrepância ocorre pela existência de algumas validações nos cadastros de fundos devido às duplicidades que existem nos cadastros de fundos, não sendo possível fazer um comparativo de tempo de inserção para cada registro inserido.

Após a inserção de um período, foi feita uma inclusão de todas as quotas de 2020 de uma única vez. Para isso, foi realizada a chamada para o método *gerarQuotasFundos* 12 vezes passando o parâmetro de cada período (mês).

O tempo de execução de todo o período de 2020 foi de 17:03, gerando uma quantidade de 4491242 registros para 2020. Com todo o histórico de 2020, percebeu-se que a quantidade mensal se torna bem dinâmica, faz com que tenha uma quantidade diversificada entre um período e outro, gerando uma diferença de mais de 100 mil registros comparando o mês com menor quantidade de registros, comparado com o mês com maior quantidade de registros conforme o gráfico 1.

Gráfico 1 - Histórico de registros do ano de 2020

Fonte: Autor (2021)

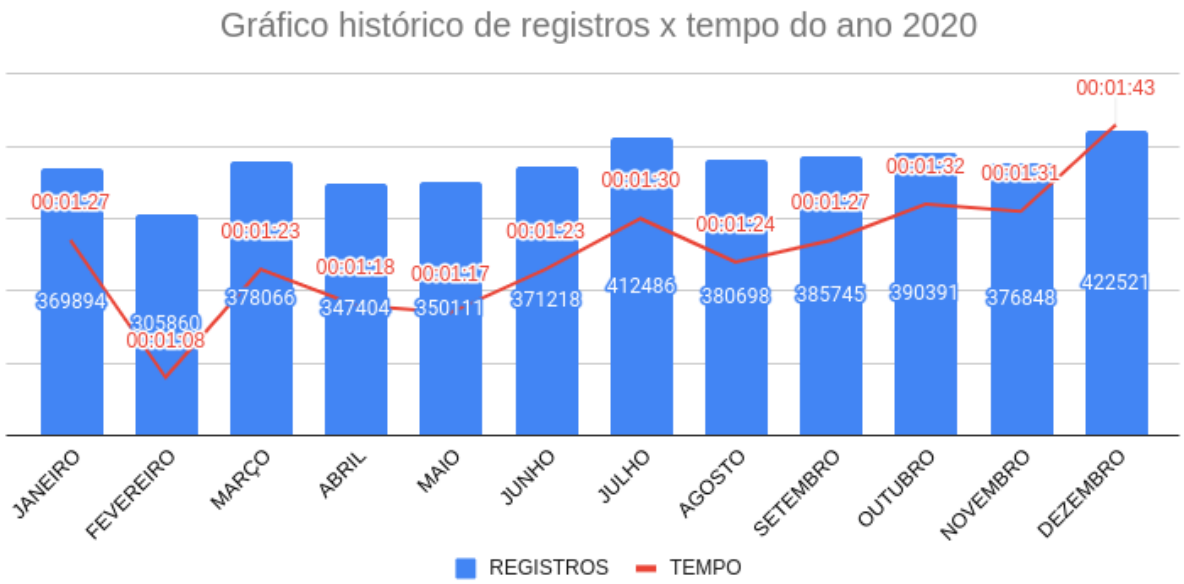
Além disso, é possível analisar o tempo de inserção de cada período conforme o gráfico 02. Percebe-se que o tempo de inserção varia muito comparado entre os meses, isso ocorre devido à quantidade de registros que variam entre os períodos.

Gráfico 2 - Histórico de tempo de inserção do ano de 2020

Fonte: Autor (2021)

Ao comparar a quantidade de registros x tempo de inserção de cada mês no gráfico 3, percebe-se que o tempo de inserção tem a mesma variância, comparado a quantidade de registros, gerando um tempo maior quando se tem uma quantidade maior de inserção conforme cada período.

Gráfico 3 - Comparativo de quantidade de registros x tempo de inserção - Dados de 2020



Fonte: Autor (2021)

5.1.5 Chamada API

Foi desenvolvida também uma nova aplicação com a necessidade de disponibilizar os dados já inseridos pela aplicação Gerador FI, disponibilizando as informações dos fundos para que *softwares* terceiros possam consultar as informações, independente da tecnologia e linguagem utilizada. Neste segundo *software* desenvolvido, chama-se api-rest.

Neste tópico, foi feito estudo com base nos conceitos APIs e *Web Services*, foi preciso entender como funcionam os protocolos HTTP e as formas de buscas na aplicação através de aplicações externas. Além disso, o foco principal foi analisar o tempo de respostas entre o retorno dos dados através da API comparado ao banco de dados, ressaltando que com o grande volume de dados inseridos, foi priorizada a performance dos resultados para garantir uma eficácia na proposta do trabalho.

Após a inserção do volume de dados conforme o item anterior, foi feita a chamada da API para analisar o tempo de resposta se haverá lentidão no retorno das

informações, conforme a figura 30, a consulta foi feita na tabela *quotas_fundos* fazendo um *left join* com a tabela *cadastro_fundos* comparando o CNPJ.

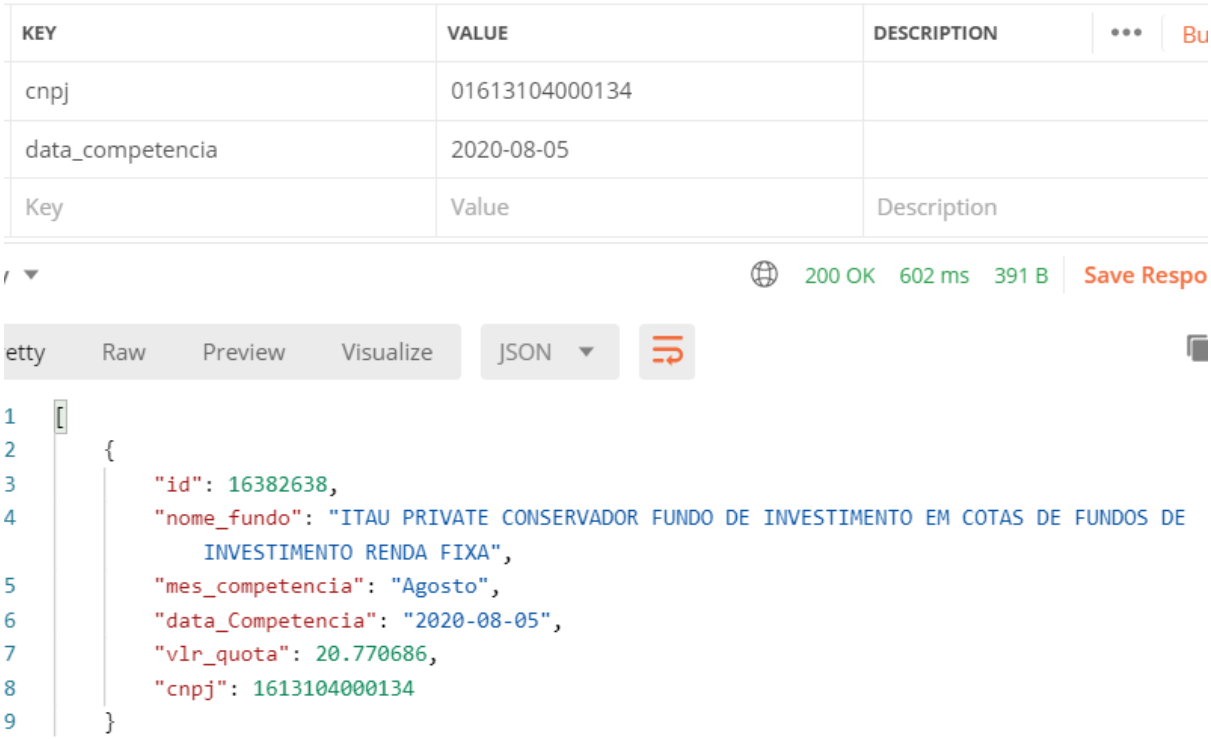
Figura 30 - Consulta SQL Aplicação API-REST

```
public interface QuotasFundosRepository extends JpaRepository<QuotasFundos, Long>{

    CadFI findById(long id_cnpj);

    @Query( value= "SELECT QFI.* "
            + ", CAD.NOME_FUNDO "
            + " FROM QUOTAS_FUNDOS QFI "
             + " LEFT JOIN CAD_FI CAD ON (QFI.CNPJ = CAD.ID_CNPJ) "
            + "WHERE QFI.CNPJ = :cnpj AND QFI.DATA_COMPETENCIA = :data_competencia"  , nativeQ

    public Collection<QuotasFundosDTO> findByQuotaFundoDia(@Param("cnpj") String id_cnpj, @Par

}
```

Fonte: Autor (2021)

Para a chamada na API, utilizou-se o *software* Postman para utilizar a função da API independente da consulta via banco de dados utilizando a função GET, e através desse *software*, foi considerado um CNPJ e uma data de competência para retornar o valor da quota do fundo do dia especificado conforme figura 31.

Figura 31 - Chamada API via Postman



Fonte: Autor (2021)

Ao analisar a chamada da API, percebeu-se que o tempo de resposta foi muito rápido, levou apenas 602ms para retornar o valor da quota de um determinado período. Ao comparar com uma consulta diretamente no banco de dados, obteve aproximadamente o mesmo tempo de resposta, demorou cerca de 581ms conforme figura 32.

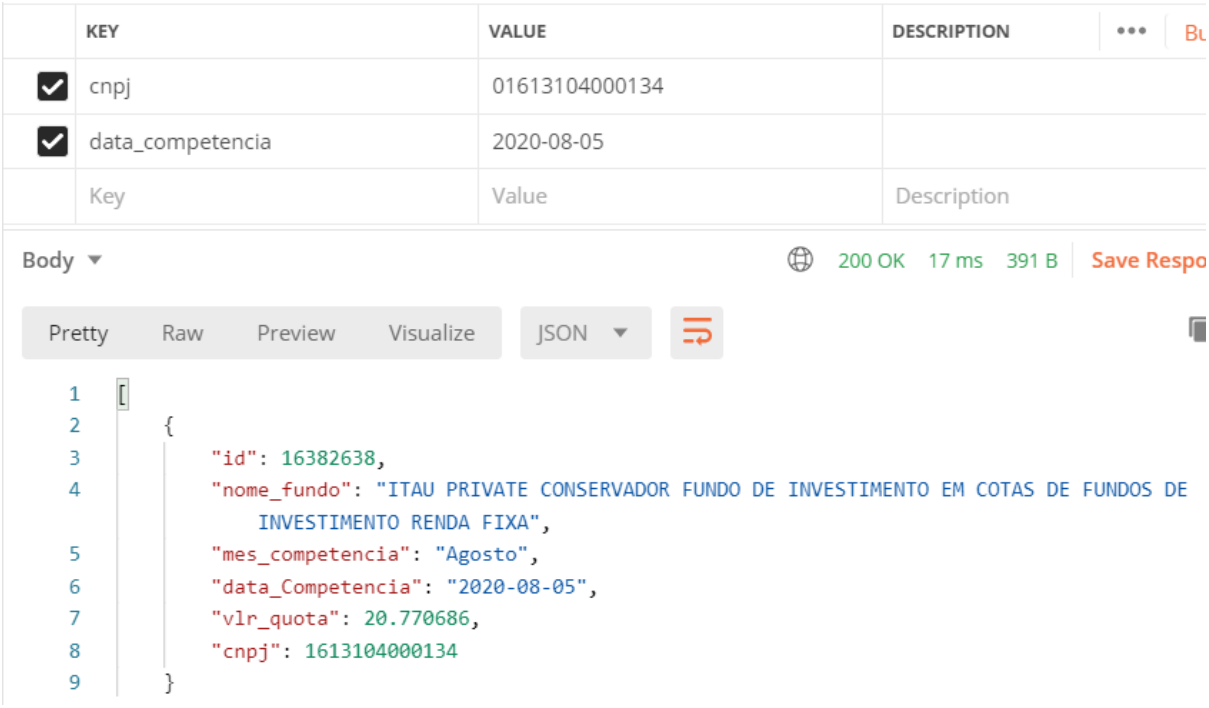
Figura 32 - Consulta através do banco de dados

Fonte: Autor (2021).

Apesar de o retorno das consultas ser rápido e eficaz, foi feita uma melhoria no banco de dados para tentar aprimorar esse tempo de retorno das consultas.

De acordo com os conceitos relacionados a consultas banco de dados relacionais, foi criado um índice na tabela de *quotas_fundos* na coluna CNPJ em que é o campo principal no comparativo *where* das consultas, e após a criação deste índice, houve um ganho de desempenho de aproximadamente 3500%, uma melhoria de extrema importância para o tema do trabalho, pois dos 600ms em que já é considerado rápido, o tempo de resposta baixou para apenas 17ms na consulta através da API conforme figura 33.

Figura 33 - Consulta através da API após criação de índice.



Fonte: Autor (2021)

Já na consulta através do banco de dados, o tempo de resposta resultou um pouco maior comparado ao da API após a aplicação do índice, mas de qualquer forma, se tornou mais rápida comparada anterior a melhoria, conforme a figura 34.

Figura 34 - Consulta através do banco de dados após criação de índice

Fonte: Autor (2021).

6 CONCLUSÕES

Primordialmente foi definido como objetivo principal de pesquisa a construção de uma API com dados históricos de fundos de investimentos diários, com garantia da disponibilização dos dados para consultados com uma alta performance, embora houvesse uma grande quantia de registros no banco de dados.

O propósito se cumpriu através do desenvolvimento de duas aplicações, a primeira, a busca de informações dos fundos no site do governo, com tratamento das informações e inserção em um banco de dados. A segunda caracterizou-se por disponibilizar os dados para consulta através da busca chamada GET. Para a realização dos testes necessários na API utilizou-se a ferramenta terceira: Postman, em que é possível fazer consultas através de parâmetros pré-definidos de forma dinâmica.

No decorrer do trabalho percebeu-se que com o uso de um campo principal na tabela do banco de dado: CNPJ do fundo junto ao índice se obteve uma significativa melhora no desempenho das consultas, com redução de tempo, cerca de 600ms para 17ms, fato que se manteria considerável mesmo com um número maior de registos (histórico maior).

Como acréscimo para futuros estudos, novas áreas no âmbito de investimentos, como, implementações dos valores de ações da IBOVESPA, fundos imobiliários e valores de taxas básicas utilizadas no Brasil (Poupança e Selic), resultariam em um aumento da utilidade da API. Além da viabilização da hospedagem das aplicações na *web* para a realização de testes sem limitação da rede local em comparativo ao desempenho com a rede externa.

‘

REFERÊNCIAS

Alaska Asset Management: Relatório mensal Maio – 2021. Disponível em: https://www.alaska-asset.com.br/pdf/Fundos/BLACK_INSTITUCIONAL.pdf. Acesso em: 03 de Jun. 2021.

ALVARENGA, Laura, Entenda os impactos da queda na taxa Selic e como ela afeta a economia, 2020. Disponível em: https://www.jornalcontabil.com.br/entenda-os-impactos-da-queda-na-taxa-selic-e-como-ela-afeta-a-economia/. Acesso em: 10 de Set. 2020.

BOGO, Felipe, Melhores práticas para o desenvolvimento de stored procedures em PL/SQL, **UCS**, Caxias do Sul, Brasil, 2013. Disponível em: https://repositorio.ucs.br/handle/11338/1494. Acesso em: 28 de Out. 2020.

BRANDÃO, Robson, O que é um Banco de Dados Relacional ?, **Insiders**, 2017. Disponível em: **https://www.datainsiders.com.br/post/o-que-e-um-banco-de-dados-relacional**. Acesso em: 13 de Out. 2020.

CALDAS,Silva; SILVA, Claudino, Fundamentos e aplicação do Big Data: como tratar informações em uma sociedade de yottabytes, **UFMG,** Belo Horizonte, Brasil, V. 3, n. 1, 2016. Disponível em: **https://periodicos.ufmg.br/index.php/revistarbu/article/view/3086.** Acesso em: 10 de Set. 2020.

CAVALCANTI, Tiago G; ALMEIDA, Vinicius C. Caracterização do uso de construções da linguagem Java em projetos Open-Source, **UNB**, Brasília, Brasil, 2016. Disponível em: https://bdm.unb.br/handle/10483/15733. Acesso em: 20 de Out. 2020.

CVM. Disponível em: http://dados.cvm.gov.br/about. Acesso em: 20 de Out. 2020.

DA COSTA, Garcia, Visualização de dados e sua importância na era do Big Data, **Universidade Federal do Rio Grande do Sul**, Porto Alegre, Brasil, 2017. Disponível em: https://www.lume.ufrgs.br/handle/10183/169109. Acesso em: 13 de Ago. 2020.

ESCALERA, Miguel F. P; CHÁVEZ, Miguel A. L. Modelo UML de uma API padrão para desenvolvimento de aplicativos de computação em nuvem, **IEEE Xplore**, Puebla, Mexico, 2012. Disponível em: https://ieeexplore.ieee.org/abstract/document/6421112. Acesso em: 17 de Set. 2020.

FERREIRA, João; MIRANDA, Miguel; ABELHA, António; MACHADO, José, O Processo ETL em Sistemas Data Warehouse, **INForum**, Braga, Portugal, 10 de Set. 2020. Disponível em: **http://inforum.org.pt/INForum2010/papers/sistemas-inteligentes/Paper080.pdf.** Acesso em: 11 de Ago. 2020.

FLÔRES, Fabiano N. Um processo para a geração automatizada de instâncias de esquemas JSON a partir de consultas SQL, **Universidade Federal de Santa Maria**, Santa Maria, Brasil, 2019. Disponível em: **https://repositorio.ufsm.br/bitstream/handle/1/18773/DIS_PPGCC_2019_FLORES_FABIANO.pdf?sequence=1&isAllowed=y**. Acesso em: 15 de Ago. 2020.

GIL, Antonio C. **Métodos e Técnicas de Pesquisa Social**, 7. ed. São Paulo, Brasil, 2019.

GOMES, Cláudia; OLIVEIRA, Bruno; BELO Orlando, Especificação do comportamento de padrões de ETL. **Atas da Conferência da Associação Portuguesa de Sistemas de Informação**, Évora, Portugal, v. 14, n. 122, 2014. ISSN 2183-489X. Disponível em: http://capsi.apsi.pt/index.php/capsi/article/view/250. Acesso em: 11 de Ago. 2020.

Icatu Vanguarda Pré-Fixados: Relatório mensal Maio – 2021. Disponível em: http://www.icatuvanguarda.com.br/pt/fundos/15. Acesso em: 03 de Jun. 2021.

Índices econômicos. Disponível em: http://www.yahii.com.br/tr.html. Acesso em: 10 de Set. 2020.

JAMES, Geordy, Creating a simple REST API in PHP, **Shareurcodes**, 2020. Disponível em: https://shareurcodes.com/blog/creating%20a%20simple%20rest%20api%20in%20php. Acesso em: 15 de Out. 2020.

Malcolm et al. Aumentando a acessibilidade aos sistemas de Big Data por meio de um API de serviços comuns, **IEEE Xplore**, Washington, Estados Unidos, 2014. Disponível em: **https://ieeexplore.ieee.org/abstract/document/7004319**. Acesso em: 15 de Set. 2020.

MASCARENHAS, Sidnei A. **Metodologia Científica**, 1. ed. São Paulo, Brasil, 2012.

MASCARENHAS, Sidnei A. **Metodologia Científica**, 2. ed. São Paulo, Brasil, 2018.

NASCIMENTO et al. Inovação nos negócios por meio da Análise de Big Data, **VI SINGEP**, São Paulo, Brasil, 2017. ISSN 2317-8302. Disponível em: **https://singep.org.br/6singep/resultado/333.pdf** Acesso em: 20 de Out. 2020.

NASCIMENTO, José A. R. Planejamento financeiro pessoal, **Revista de Ciências Gerenciais**, Santa Terezinha, Brasil, v. 15, n. 22, 2015. ISSN 1415-6571. Disponível em: **https://revista.pgsskroton.com/index.php/rcger/article/view/2101**. Acesso em: 14 de Ago. 2020.

OLIVEIRA, Paulo H. C. Desenvolvimento de um gerador de API REST seguindo os principais padrões de arquitetura, **UNIVEM**, Marília, Brasil, 2015. Disponível em: https://aberto.univem.edu.br/handle/11077/1008. Acesso em: 13 de Out. 2020.
Poupança: entenda como funciona o rendimento e saiba quando deixar de lado. Disponível em: https://www.infomoney.com.br/guias/poupanca/. Acesso em: 10 de Set. 2020.

PRODANOV, Cleber C; FREITAS, Ernani C. **Metodologia do Trabalho Científico**, 2. ed. Novo Hamburgo, Brasil, 2013.

RAUTENBERG, Sandro; DO CARMO, Viviurka, Big data e ciência de dados: complementaridade conceitual no processo de tomada de decisão, **Brazilian Journal of Information Science: research trends**, Marília, Brasil, v. 13, n. 1, 2019. ISSN 1981-1640. Disponível em: https://revistas.marilia.unesp.br/index.php/bjis/article/view/8315. Acesso em: 13 de Ago. 2020.

REIS, Fábio, O que é um Banco de Dados Relacional, 2019. Disponível em: **http://www.bosontreinamentos.com.br/bancos-de-dados/o-que-e-um-banco-de-dados-relacional/**. Acesso em: 20 de Out. 2020.

SALVADORI, Ivan L. Desenvolvimento de web APIs RESTFUL semÂnticas baseadas em JSON, **UFSC**, Florianópolis, Brasil, 2015. Disponível em: **https://repositorio.ufsc.br/handle/123456789/132469.** Acesso em: 28 de Set. 2020.

SECCO, Ricardo R. Análise comparativa entre o banco de dados cassandra (modelo NoSQL) e o PostgreSQL (modelo relacional) em duas diferentes organizações empresariais, **Unoeste**, Presidente Prudente, Brasil, v. 8, n. 2, 2017. ISSN 2178-8332. Disponível em: http://revistas.unoeste.br/index.php/ce/article/view/1324. Acesso em: 26 de Out. 2020.

TRINDADE, João A. de S.; MALAQUIAS, Rodrigo F. Análise de desempenho de fundos de investimento de renda fixa e renda variável. **Revista de Auditoria Governança e Contabilidade**, v. 3, n. 5, 2015. ISSN 2317-0484. Disponível em: **http://fucamp.edu.br/editora/index.php/ragc/article/view/546**. Acesso em: 10 de Ago. 2020.

VERSANI, Rafael. O que é o Postman?, **Enotas**, 2020. Disponível em: https://enotas.com.br/blog/postman/. Acesso em: 28 de Out. 2020.

VIEIRA, Dutra; ARAÚJO, Pereira, Otimização em banco de dados Oracle através do uso de Views materializadas, **SEER**, v. 2, n. 2, 2015. Disponível em: https://seer.cesjf.br/index.php/cesi/article/view/521. Acesso em: 15 de Set. 2020.

VILELA, Allyson B. C. B. OpenData Processor: Uma ferramenta para a automatização do processo de extração e publicação de Dados Abertos, **ufrn**, Natal , Brasil, 2018. Disponível em: https://repositorio.ufrn.br/jspui/bitstream/123456789/26014/1/OpenDataprocessorferramenta_Vilela_2018.pdf. Acesso em: 13 Out. de 2020.

WISNIEWSKI, Marina L. G. A importância da educação financeira na gestão das finanças pessoais: Uma ênfase na popularização do mercado de capitais brasileiro. **Revista Intersaberes**, Curitiba, Brasil, v. 6, n. 11, 2011. ISSN 1809-7286. Disponível em: **https://www.uninter.com/intersaberes/index.php/revista/article/view/32/17**. Acesso em: 10 de Ago. 2020.