

PYMARKET, FERRAMENTA PARA ANÁLISE DE MERCADO

Giovani Almeida de Lima <giovani.lima@rede.ulbra.br>
Christiano Cadoná <cadona@rede.ulbra.br> – Orientador(a)

Universidade Luterana do Brasil (Ulbra) – Curso de Ciência da Computação – Campus Canoas
Avenida Farroupilha, 8001 · Bairro São José · CEP 92425-900 · Canoas/RS

04 de dezembro de 2023

RESUMO

Este artigo descreve toda a etapa para a construção e validação do projeto *PyMarket*, que é o Projeto Tecnológico do curso de Ciência da Computação da Universidade Luterana do Brasil (Ulbra). Este documento apresenta a fundamentação teórica que fornece embasamento e explica a motivação da criação, apresenta alguns casos de ferramentas que tem propostas semelhantes, descreve as tecnologias usadas no projeto e metodologias utilizadas para validar e criar o projeto. A ferramenta *PyMarket* vai possibilitar que os arquivos de dados gerados e não tratados pela Receita Federal Brasileira, sejam baixados, tratados e transformados em insights que possam gerar valor. Vai disponibilizar Dashboards informativos que possam ajudar em possíveis tomadas de decisão e uma API em que se pode obter dados de empresas via CNAE ou CNPJ por qualquer pessoa que tenha vontade de usar a ferramenta.

Palavras-chave: Projeto Tecnológico; Trabalhos Acadêmicos; *PyMarket*; Mercado; Negócios.

ABSTRACT

Title: “*PyMarket, tool for market analysis*”

This article describes the entire stage for the construction and validation of the PyMarket project, which is the Technological Project of the Computer Science course at Universidade Luterana do Brasil (Ulbra). This document presents the theoretical foundation that provides the basis and explains the motivation for creation, presents some tool cases that have similar proposals, describes the technologies used in the project and methodologies used to validate and create the project. The PyMarket tool will enable data files generated and not processed by the Brazilian Federal Revenue Service to be downloaded, processed and transformed into insights that can generate value. It will provide informative Dashboards that can help with decision-making and an API in which company data can be obtained via CNAE or CNPJ by anyone who wants to use the tool.

Key-words: Technological Project; Academic Papers; *Pymarket*; Market; Bussiness;

1 INTRODUÇÃO

Estamos vivendo na era da tecnologia. Cada vez mais o mundo adota a tecnologia, transformando nosso cotidiano. Basta olhar um ou dois séculos para trás e ver o quanto ela influenciou nosso modo de viver, de trabalhar ou de nos relacionarmos. Ela passou do papel de ferramenta de suporte para o papel de protagonista, atuando diretamente em como fazemos as coisas, qual decisão tomamos ou qual caminho seguimos.

Também vivemos na era mais populosa. Em 1900, estimasse que a população era de cerca de 1,6 bilhões. Atualmente, pouco mais de 120 anos, estamos beirando os 8 bilhões. Essa atuação da tecnologia é sem sombra de dúvidas responsável por este crescimento, visto que ela proporciona melhora significativa das condições básicas de vida para os seres humanos, como saúde e educação por exemplo.

Unindo esses dois fatores, temos a globalização, que resumidamente falando, é um fenômeno de integração mundial de diversas áreas, como social, cultural, geográfica e econômica. A globalização existe desde os tempos da Pangeia, embora em pequena escala, mas a tecnologia possibilitou que isso tivesse um alcance maior depois que esse supercontinente se dividiu ao longo dos anos, dando origem aos atuais continentes e mesmo assim as pessoas continuaram indo de um lugar ao outro de diversas maneiras, com as não tão antigas caravelas portuguesas por exemplo, até os mais avançados aviões Boeing.

Agora reflita: Adoção e evolução da tecnologia, crescimento da população e globalização de diversas áreas importantes. Isso gera uma série de implicações para a nossa sociedade, pois causa um efeito de crescimento exponencial: A tecnologia melhora condição básica dos seres humanos, com essa melhora, as

peças vivem mais e a população se reproduz mais. Com mais pessoas, mais mentes pensantes e maior possibilidade de inovações da tecnologia. Com essa inovação, as pessoas, migram, colonizam, se comunicam, se relacionam e globalizam o planeta. Logo, num ambiente que está sempre em constante mudança, existe a necessidade de rápida adaptação às novidades e desafios que surgem. A inovação tecnológica constante, nos fornece as ferramentas necessárias, nos proporcionando cada vez mais vantagens competitivas, seja no desenvolvimento de produtos mais eficientes, na implementação de processos produtivos mais avançados ou na criação de soluções tecnológicas inovadoras para a sociedade. Existem diversos processos e soluções tecnológicas que surgiram durante os anos e dentro deles existe a área de análise de dados. Ela é uma poderosa aliada para as organizações, que conseguem retirar insights valiosos de grandes volumes de dados, tornando-se assim, uma ferramenta de diferencial estratégico, visto que isso efetivamente se torna uma vantagem competitiva contra quem não usa.

Todos os dias milhares e milhares de dados são gerados pelo mundo todo. Estes dados podem ser públicos, privados ou de acesso limitado. Por exemplo, algumas instituições, como órgãos públicos, são obrigadas por lei a disponibilizar uma série de dados para os cidadãos, como o Brasil, que através da lei federal 12527, garante a todos o direito de receber de toda a administração pública, suas informações, seja por interesse particular, coletivo ou geral, desde que não sejam informações sigilosas imprescindíveis à segurança da sociedade e do Estado.

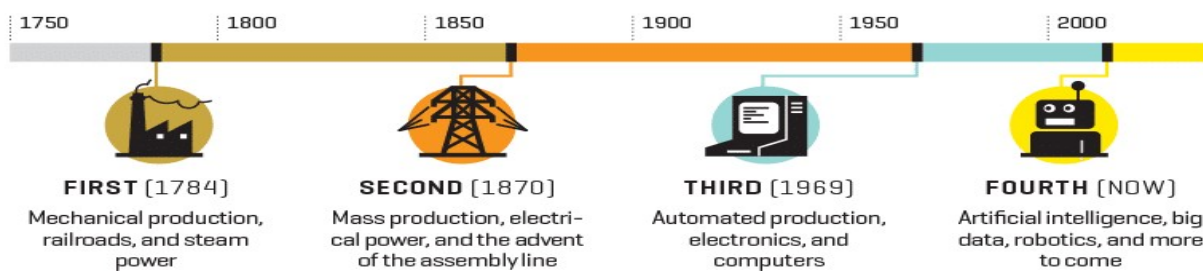
Essas informações são disponibilizadas através do Portal Brasileiro de Dados Abertos, que reúne uma série de dados publicados pelo governo federal. Infelizmente a maioria destes dados contém apenas um compilado de informações que não são tratadas, muito menos analisadas, possivelmente por não ser de interesse público tratá-los ou talvez pelo tamanho dos compilados de dados, o que é fato é que dificulta a visualização por parte do cidadão comum. Uma parte dessas informações são os dados do Cadastro Nacional da Pessoa Jurídica (CNPJ), que é gerenciado pela Receita Federal Brasileira, que fornece informações cadastrais das pessoas jurídicas e de entidades de interesse das administrações tributárias da União, Estados, Distrito Federal e Municípios, que são atualizadas mensalmente segundo o próprio site.

Estes dados são disponibilizados de forma aberta e gratuita e podem gerar valiosos e poderosos insights, porém estão desorganizados e desestruturados. Foi pensando nisso, que surgiu a ideia do PyMarket, que se propõe a fazer o download desses arquivos, extrair, transformar, carregar e disponibilizar uma estrutura, em que esses dados possam ser consultados através de um sistema web, que num primeiro momento, vai trazer algumas informações em formato de gráficos, de porte das empresas de cada setor e de crescimento e declínio de setores. Também se propõe a disponibilizar neste primeiro momento, uma API pública de consulta das informações tratadas, que poderão ser utilizadas por empresas pública ou privadas, cientistas de dados, entusiastas da programação ou qualquer outra pessoa que venha a se interessar por essa informação tratada, visto que este trabalho não é feito pela receita federal brasileira, facilitando assim o acesso a estas informações.

2 FUNDAMENTAÇÃO TEÓRICA

A análise de dados, na sua grande maioria, tem o trabalho de fazer a coleta, organização, interpretação e apresentação de dados que tenham alguma relevância para tomadas de decisão. No mundo contemporâneo, os dados crescem de forma exponencial, ao mesmo passo que a tecnologia por trás de tudo também avança. Mas para chegar até o ponto onde é possível ter a tecnologia para realizar este tipo de trabalho, passamos por diversas revoluções na indústria. Resumidamente, temos a 1ª Revolução Industrial, onde nos foi apresentada a tecnologia de máquinas a vapor, diminuindo drasticamente os trabalhos artesanais. Após, houve a inserção da energia elétrica e a divisão de trabalho ao estilo “Ford”, dando origem a 2ª Revolução Industrial. Por fim, a 3ª Revolução Industrial chegou, com a inserção da tecnologia da informação nos processos. Com todos estes avanços, surgiu recentemente o conceito de indústria 4.0, que segundo Schwab (2016, p. 1) “tudo começou na virada deste século e se baseia na revolução digital. É caracterizada por uma internet muito mais onipresente e móvel, por sensores menores e mais potentes que se tornaram mais baratos, e por inteligência artificial e aprendizado de máquina”. Também segundo Schwab (2016, p. 7) “estamos no início de uma revolução que está mudando fundamentalmente a forma como vivemos, trabalhamos e nos relacionamos um com o outro”. “Diferente das outras revoluções industriais que aconteceram na história, a Indústria 4.0 sucede de maneira muito peculiar devido à sua forma acelerada, abrupta e disruptiva” (ARAUJO et al., 2020). Abaixo, uma imagem que mostra a linha do tempo das revoluções:

Figura 1 – As quatro revoluções industriais.



Fonte: <https://cubienergia.com/wp-content/uploads/2020/05/Timeline-revolução-industrial.png>

De forma sintetizada, podemos concluir que estamos vivendo uma nova era, onde praticamente tudo o que sabemos, fazemos e criamos, muda em passos largos, como por exemplo: A maneira como consumimos produtos e serviços e como nos comunicamos.

Dentro dessa revolução, uma área que se destaca bastante, é a de Big Data, que em síntese, é o processo de análise de uma grande quantidade de dados.

2.1 Big Data

Como descreve o site da CETAX “o termo Big Data nasceu no início da década de 1990, na Nasa, para descrever grandes conjuntos de dados complexos que desafiam os limites computacionais tradicionais de captura, processamento, análise e armazenamento informacional[...]” (CETAX, 2018). Porém o processamento de grandes volumes de dados é um pouco mais antigo que isso, visto que, temos como exemplo, a máquina que Alan Turing construiu com seus companheiros chamada Bombe, que era usada para decifrar as mensagens da máquina Enigma, utilizada pelos alemães na segunda guerra mundial. Percebe-se então que o conceito em si não é novo, porém como o site da UCS descreve “O Big Data só passou a ser realmente difundido a partir de 2005 devido à publicação de um artigo de autoria de Roger Magoulas que trabalhava na companhia O’Reilly Media[...]” (UCS, 2023). Infelizmente, este termo sofreu muitas alterações conforme o tempo foi passando, ganhou definições de vários autores, mas a melhor definição encontrada em minha opinião, é descrita pela Gartner em seu site “Big Data são ativos de informações de alto volume, alta variedade que exigem formas inovadoras e econômicas de processamento de informações que permitem maior percepção, tomada de decisões e automação de processos[...]” (Gartner).

2.2 Dados

Nas palavras de GUIMARAES “Os dados são elementos que constituem a matéria prima da informação. Podemos defini-los, também, como conhecimento bruto, ainda não devidamente tratado para prover insights para uma organização[...]” (GUIMARAES, 2022?). Mas o dado por si só não é relevante. Segundo Shedroff (1994), “Os dados são o produto da descoberta, pesquisa, coleta e criação. É a matéria-prima que encontramos ou criamos, que usamos para construir nossas comunicações.”. Mas para que esses dados sejam transformados em algo de valor, eles precisam ser organizados de uma maneira que faça sentido. Segundo Shedroff (1994), os dados só têm significado para as pessoas se for possível comunicar o seu contexto. Baseado nesses conceitos, podemos afirmar que dados são a matéria prima, a informação não tratada.

2.3 O que é ETL?

ETL é o acrônimo de Extract, Transformation and Load, que consiste em integrar dados de várias fontes, sanitizar estes dados e armazenar em um local centralizado, utilizando ferramentas de ETL (Power BI, Qlik Sense) ou de forma manual, como por exemplo, uma linguagem de programação.

A etapa de Extract é onde os dados são selecionados. Conforme dito anteriormente, os dados podem possuir diversas fontes, sejam elas online, locais, de arquivos de diversos tipos, de banco de dados, entre outros. Nesta etapa, devem ser identificadas as fontes de dados, levando em consideração quais dados devem ser extraídos para que não use espaço de armazenamento desnecessário e não tome tempo de

extração. Esses dados são coletados e armazenados em um ambiente intermediário até que sejam utilizados pela outra etapa, que é a de Transformation.

Na etapa de Transformation, os dados são copiados e submetidos a um tratamento. Geralmente este tratamento serve para resolver problemas de valores nulos, valores vazios, caracteres desconhecidos, remoção de campos desnecessários e todo e qualquer tipo de tratamento que seja necessário para garantir a consistência e integridade destes dados para a etapa de Load.

Por fim, a etapa de Load é o momento em que os dados são de fato carregados em seus respectivos destinos, conforme a necessidade, como um banco de dados, um arquivo CSV, entre outros.

Os principais objetivos do ETL são (KIMBALL, 2004):

- Remover erros e corrigir dados faltantes;
- Assegurar a qualidade dos dados;
- Capturar o fluxo de dados transacionais;
- Ajustar dados de múltiplas origens e usá-los juntos;
- Fornecer estruturas de dados para serem utilizadas por ferramentas pelos analistas responsáveis pelo desenvolvimento;
- Fornecer dados em formato físico para serem usados por ferramentas dos usuários finais.

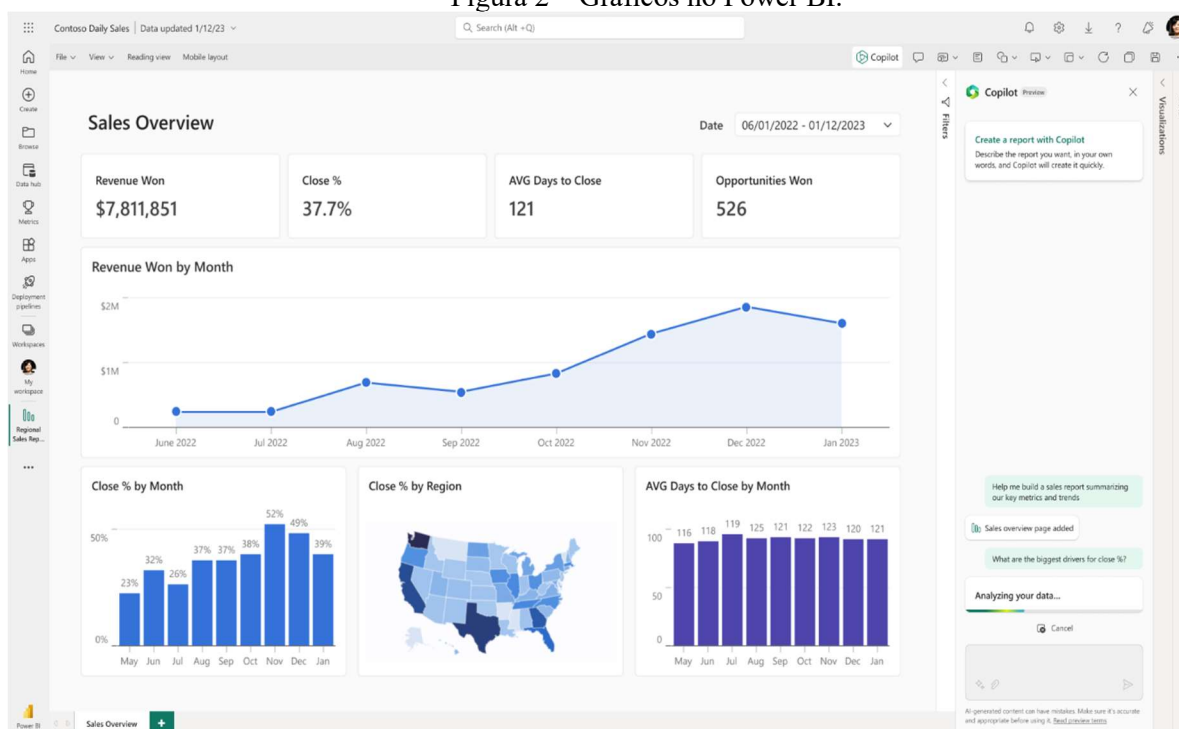
2.4 O que já existe

Existem inúmeras plataformas de análise de dados no mercado, que permitem que o usuário faça análises sem precisar aprender nenhuma linguagem de programação ou extraia dados de alguma fonte.

2.4.1 Power BI

Power BI é uma ferramenta Business Intelligence criada pela Microsoft em 2015, que basicamente tem o objetivo de coletar dados, tratar, organizar e gerar indicadores e relatórios que ajudem na tomada de decisão. Com ele é possível coletar informações de diversas fontes, como banco de dados, arquivos no formato txt, arquivos em formato CSV, entre outros. Um exemplo de gráficos utilizando o Power Bi na imagem abaixo:

Figura 2 – Gráficos no Power BI.



Fonte: Microsoft Learn (2023).

Com essa variedade de fontes de dados, é possível fazer o que chamamos de relacionamentos, onde você tem várias tabelas de lugares diferentes e junta as informações de todas essas tabelas dentro de um relatório. Com esses dados no relatório, é possível transformá-los e prepará-los para que atendam necessidades específicas de utilizando o Power Query, que é uma ferramenta de ETL. Também existe a possibilidade de aplicar filtros para cada campo do relatório, tipos de visualizações, possibilitando a criação de relatórios intuitivos, visuais e dinâmicos.

É possível também suportar um volume muito grande de dados. Diferente do excel por exemplo, que quando você começa a trabalhar com 100 mil linhas a performance cai, no Power Bi é possível suportar milhões de linhas de tabelas.,

Um dos principais diferenciais do Power bi é basicamente você conseguir publicar o seu relatório online. Você pode mandar alguém da sua equipe e essa pessoa vai ter acesso e vai poder interagir com seu relatório.

Por fim, dentro do Power BI, é possível agendar atualizações para suas informações. Ele basicamente vai buscar os bancos de dados, vai ver quais informações novas já entraram desde a última vez que foi atualizado e vai atualizar os relatórios automaticamente sem você precisar fazer absolutamente nada.

2.4.2 Google Analytics

O Google Analytics é uma plataforma que coleta dados em apps e sites para criar relatórios do site que ele está vinculado. Se faz necessário criar um cadastro e em seguida adicionar um código de medição nas páginas do site que o usuário deseja monitorar. Conforme as páginas do site vão sendo acessadas, informações vão sendo coletadas sobre a interação dos usuários do site com ele.

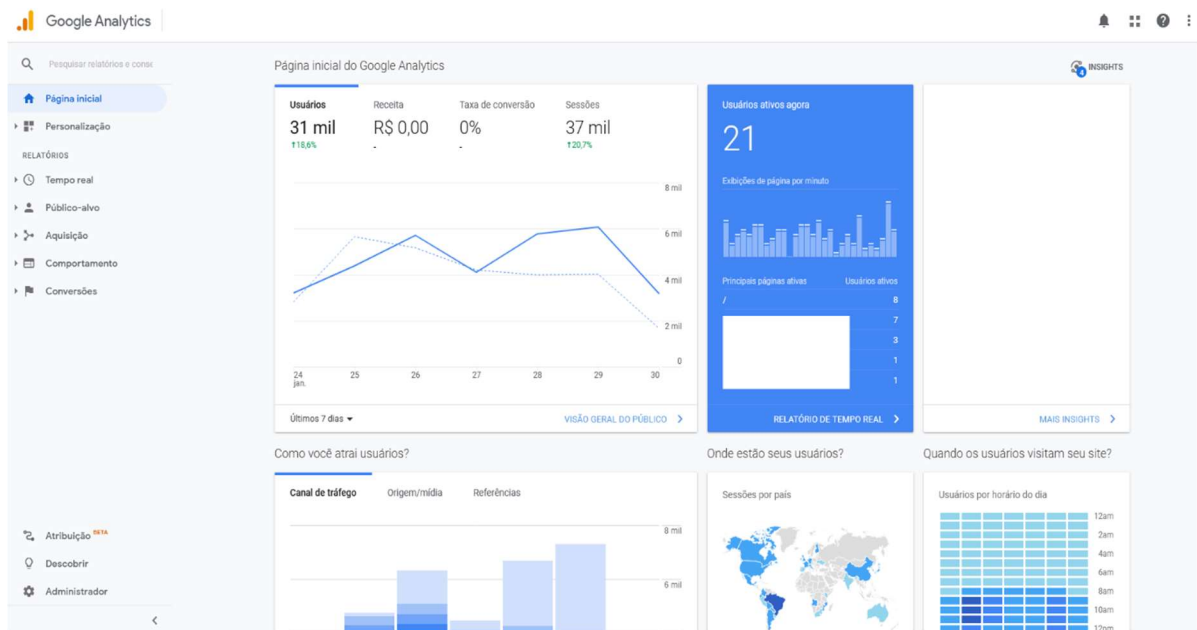
Então, ele vai desde dados mais gerais, como a quantidade de tráfego que seu site possui, quantos usuários você teve em um determinado período, a duração média de uma sessão no site, a taxa de inscrição no site como um todo e vai evoluindo, por exemplo, sabendo mais sobre a persona, qual é a idade das pessoas que acessam o site, qual é o sexo dessas pessoa, onde essas pessoas estão localizadas e a partir disso, formular estratégias mais direcionadas para o público alvo que vai se conhecendo a medida vai analisando a ferramenta.

Além disso, é possível saber o dispositivo que esse usuário está usando para acessar o conteúdo, se foi do computador, celular, qual foi a forma de encontrar o site, se foi através de e-mail, redes sociais, organicamente entre outros.

Outra informação que o Google Analytics proporciona, é saber pouco mais sobre páginas específicas do site e a quantidade de visões que uma página tem comparado com a outra. Quais são as principais páginas de entrada no nosso site? Quais são as principais páginas de saída?

Finalizando, uma das coisas mais importantes que é possível acompanhar, é a performance das metas e os objetivos que o site possui, como por exemplo, se gerar leads, gerar vendas, entre outros. Existe também a possibilidade de criar o famoso funil de vendas dentro do Google Analytics e, a partir disso conseguir ver, por exemplo, as pessoas que chegaram até o carrinho de compras, quantas foram para check out, quantas de fato fecharam uma venda, assim sendo possível otimizar cada etapa se necessário. Abaixo, uma imagem do Google Analytics, mostrando como é a ferramenta de fato:

Figura 3 – Gráficos no Google Analytics.

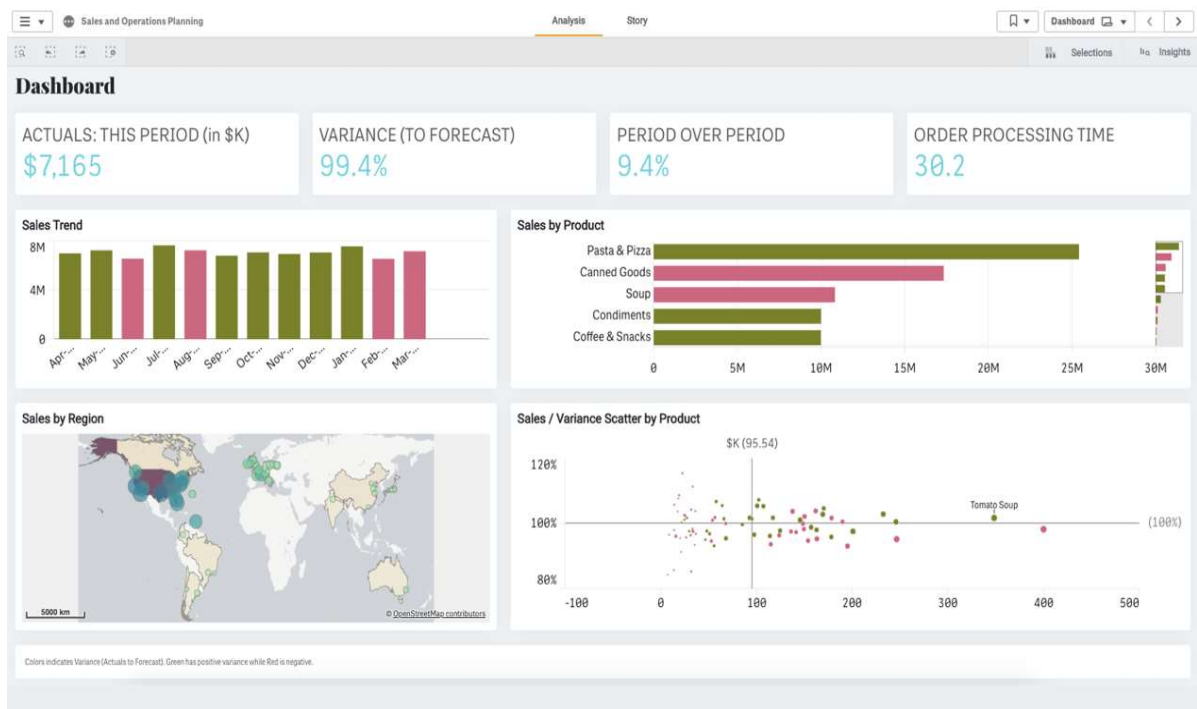


Fonte: Google Analytics (2023).

2.4.3 Qlik Sense

O Qlik Sense é uma ferramenta criada em 2014, que proporciona a visualização de informações, como gráficos, tabelas, gerando poderosos insights utilizando conjunto de dados. No Qlik, existe a possibilidade de criar e compartilhar as visualizações com quem mais se sinta interessado em ver, porém não é possível a colaboração, o que significa que cada pessoa precisa carregar seus próprios dados para colaborar totalmente nos mesmos. Seguindo um perfil mais antiquado, geralmente o Qlik é mais utilizado por quem possui experiência em programação e manipulação de banco de dados, devido a seguir uma abordagem parecida com ferramentas mais antigas como o Tableau, não possuindo uma abordagem tão intuitiva como de outras ferramentas.

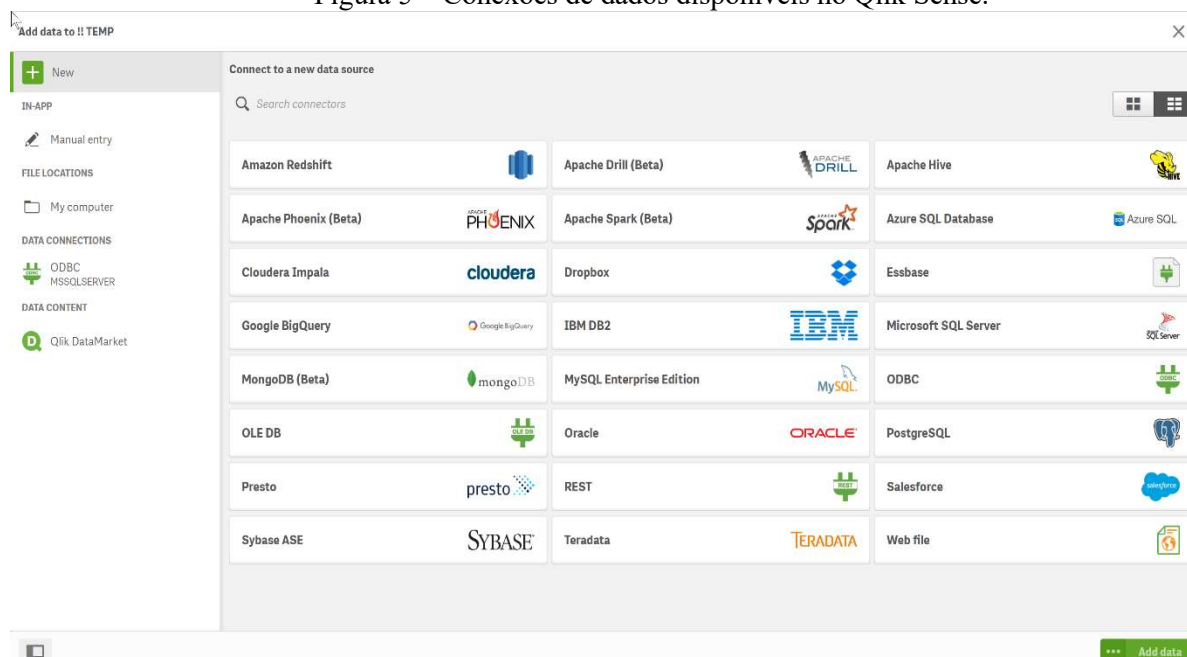
Figura 4 – Gráficos no Qlik Sense.



Fonte: Qlik Sense (2023).

É possível adicionar diversas fontes de dados. Mas é valido destacar que o Qlik tem conexão direta com a Amazon AWS, facilitando possíveis integrações, diferente de outras ferramentas. Também possui conexão direta com a Microsoft Azure, como seu concorrente Power BI, conforme imagem abaixo:

Figura 5 – Conexões de dados disponíveis no Qlik Sense.



Fonte: Qlik Sense (2023).

3 OBJETIVOS

Para realizar este projeto, foram estipulados objetivos para que seja alcançada a excelência no que diz respeito a metodologia, desenvolvimento e estrutura.

3.1 Objetivo geral

Disponibilizar as informações dos CNPJS fornecidas pela Receita Federal Brasileira de forma coesa, ou seja, extraindo, transformado, carregando e disponibilizando de maneira clara e objetiva esse conjunto de informações, visto que atualmente, são fornecidos arquivos de dados sem qualquer tipo de tratamento, trazendo dificuldade na averiguação das informações. Para isso, será criada uma ferramenta capaz de fazer o ETL, gerar alguns indicadores com a informação e disponibilizar a informação para que outros façam as análises que desejam, sejam entusiastas, empresas, órgãos governamentais e afins.

3.2 Objetivos específicos

- Disponibilizar informações em formato de gráficos, de porte das empresas de cada setor e de crescimento e declínio de setores, para que possam servir de ajuda para alguma tomada de decisão, como por exemplo, saber se o momento para abrir uma empresa é propício.
- Possibilitar a qualquer pessoa ou empresa, o acesso as informações de CNPJS que são disponibilizadas pela Receita Federal Brasileira, através da API.
- Possibilitar a visualização de dados históricos de todas as informações que vierem a ser atualizadas pelo processo de ETL.
- Desenvolver toda a estrutura do backend utilizando a linguagem Python, seguindo as melhores práticas de programação e padrões de projeto.
- Criar uma estrutura para disponibilizar a ferramenta, utilizando servidores web e

ferramentas de CI/CD.

4 METODOLOGIA

Para o desenvolvimento do projeto, foi adotada a metodologia ágil Scrum em conjunto com a plataforma ClickUp, que oferece diversos recursos para organização de tarefas e projetos. As tarefas foram organizadas em sprints semanais, utilizando um quadro Kanban para fazer o registro das atividades. As reuniões com o professor orientador, ocorreram de forma presencial e virtual via Meet e contatos prévios via WhatsApp. O versionamento do código foi feito utilizando Git e hospedado na plataforma GitHub.

4.1 Análise de requisitos

Após a definição do tema do projeto junto ao orientador, foi feita a análise de requisitos. Utilizando o método de entrevista de forma virtual via reunião no Google Meet e via Whatsapp com o próprio orientador, foram definidas as principais funcionalidades a serem implementadas no projeto, seguindo o que foi descrito no seminário de andamento e no cronograma. Abaixo, segue o que foi acordado e o que foi definido na análise de requisitos:

- Endpoint para buscar CNAEs
- Endpoint para buscar CNPJs
- Interface web explicando sobre a ferramenta
- Dashboards informativos com análise de tendências em forma de gráfico, mostrando Crescimento e Declínio de setores e Porte das Empresas de Cada Setor.

4.2 Desenvolvimento

A etapa de desenvolvimento buscou seguir rigorosamente a sequência abaixo:

- Desenvolvimento das configurações, classes e objetos.
- Desenvolvimento do Bot para download de arquivos para ETL.
- Desenvolvimento do Backend (API).
- Desenvolvimento do Backend (WEB).
- Desenvolvimento do Frontend (TEMPLATES).
- Revisão das funcionalidades do projeto.

4.3 Validação

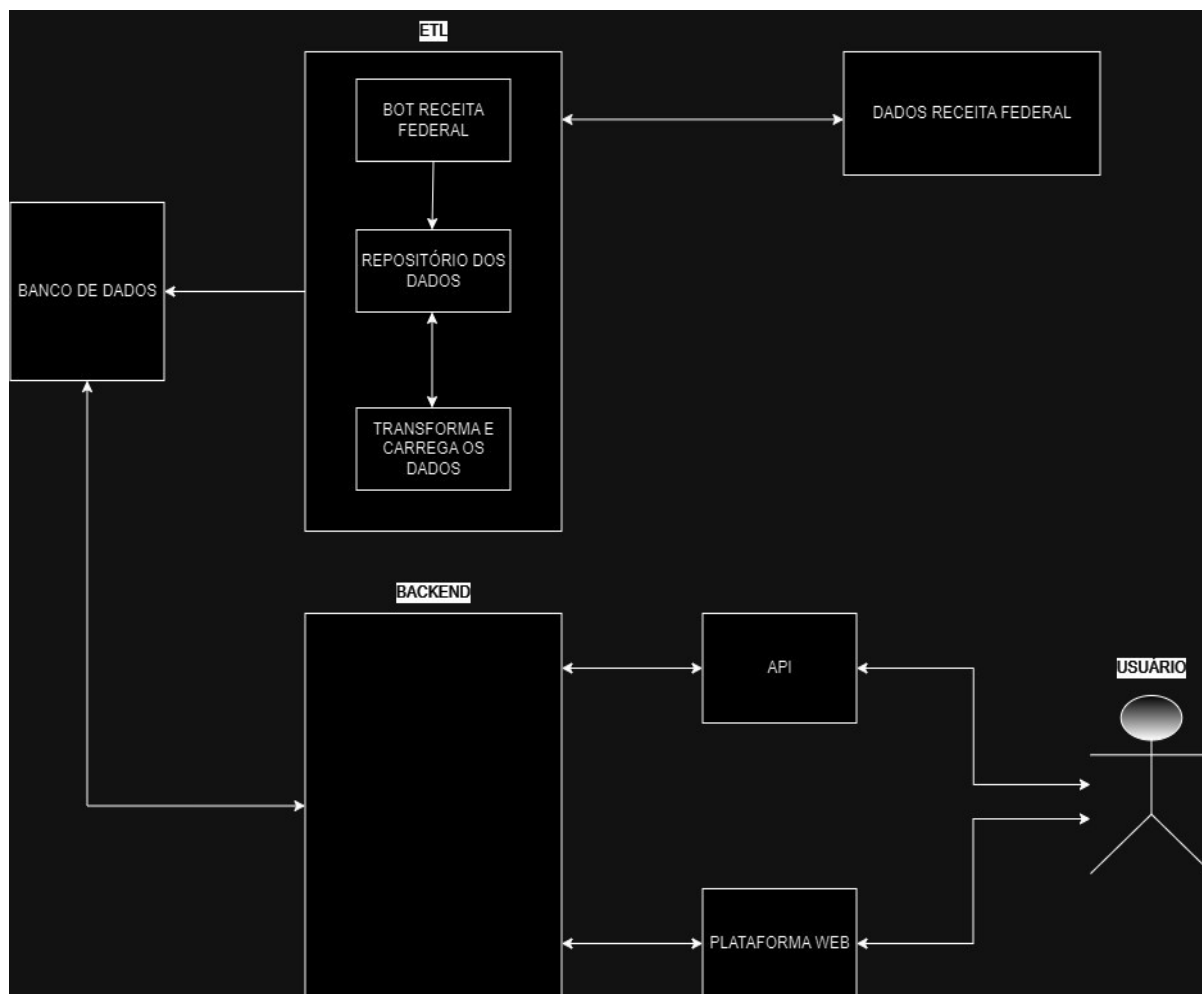
Para a validação, primeiramente a ferramenta seria colocada no ar. Posteriormente seriam convidadas empresas para fazer uso da ferramenta e responder um formulário no Google Forms, onde constariam perguntas relacionadas a experiência do usuário utilizando a ferramenta web e a API e um espaço para opiniões, como por exemplo, possíveis melhorias.

5 FERRAMENTA PYMARKET

O presente Projeto Tecnológico tem como objetivo desenvolver e entregar uma ferramenta que disponibilize uma plataforma web que contenha em um primeiro momento, um dashboard com gráficos de porte das empresas de cada setor, de crescimento e declínio de setores e uma API para consultas de dados de empresas por número de CNAE e CNPJ que possa ser utilizada de forma gratuita por todos que tiverem interesse, como instituições públicas, privadas, desenvolvedores, entusiastas, entre outros.

Para ilustrar o funcionamento completo da ferramenta, abaixo segue uma imagem que mostra o fluxo de trabalho:

Figura 6 – Fluxo de funcionamento da ferramenta PyMarket.



Fonte: Giovani Lima (2023).

O projeto está dividido em diversas partes, explicadas de forma sucinta:

5.1 ETL

Realiza a extração, transformação, e carga dos dados dos arquivos baixados da receita federal no banco de dados. Acessa o site da receita federal e faz o download dos arquivos de dados. Utiliza o módulo Selenium para simular um acesso humano ao site e o módulo Wget, que serve para baixar arquivos que suportam o protocolo http, https ou ftp. Para esse download ser performático, existem funções de download para cada arquivo. Essas funções são disparadas por uma função pai que cria e executa uma Thread de cada uma delas. Utiliza o módulo Zip para ler os arquivos, pois são disponibilizados pela receita federal em formato .ZIP. Após, o utiliza o módulo Pandas que faz a carga de um pedaço dos dados do arquivo e realiza toda a limpeza necessária, como por exemplo, substituição de caracteres nulos por "NC", correção de datas para o formato dd/mm/aaaa, entre outros. Por fim, a informação é inserida no banco de dados.

5.2 Backend

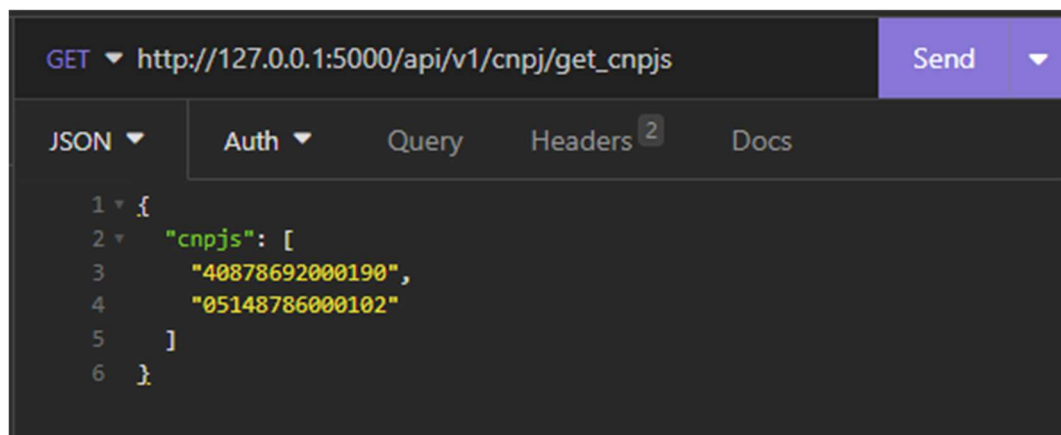
É o responsável por disponibilizar duas das principais funcionalidades, que são a API e a plataforma WEB. Elas podem ser acessadas de maneira indireta, ou seja, não são dependentes uma das outras.

5.2.1 API

A API disponibiliza dois endpoints para consulta de informações da receita federal. Em um endpoint, é possível informar uma lista de CNPJS e receber de retorno as informações de cadastro

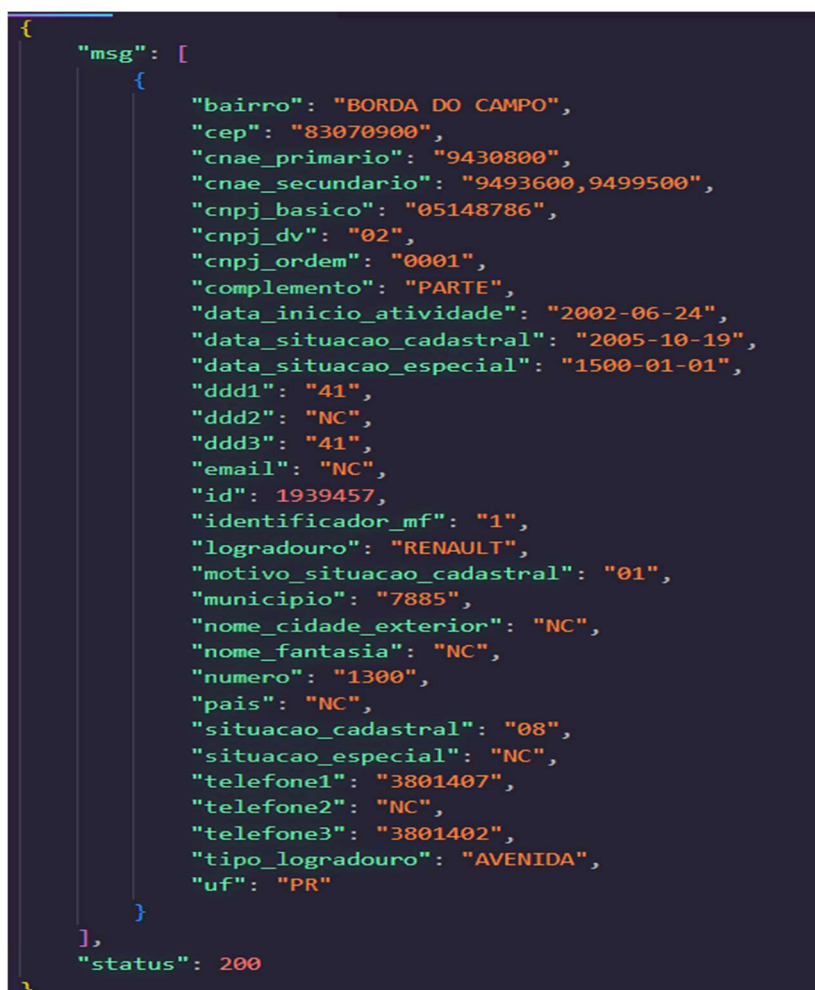
deles, conforme imagens abaixo:

Figura 7 – Parâmetros possíveis para a rota get_cnpj da API PyMarket.



Fonte: Giovani Lima (2023).

Figura 8 – Retorno da rota get_cnpj da API PyMarket.

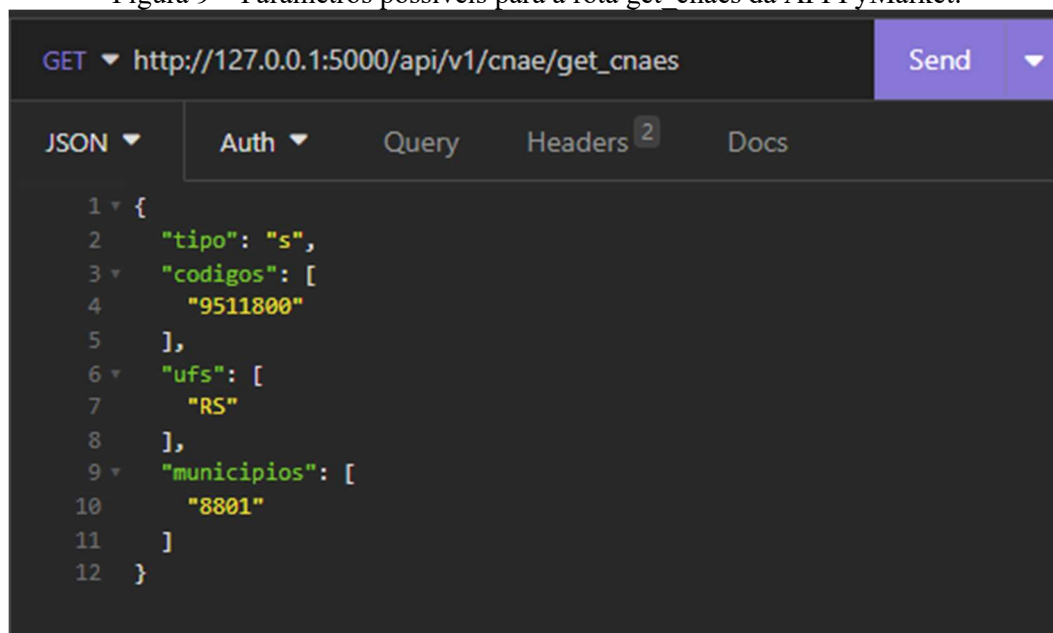


Fonte: Giovani Lima (2023).

No outro endpoint, é possível informar o tipo de CNAE (primário ou secundário), uma lista de códigos de CNAES, uma lista de UFs do Brasil, uma lista de códigos dos municípios do Brasil e receber de retorno todos os CNPJS que estão contemplados na lista de CNAES, juntamente com

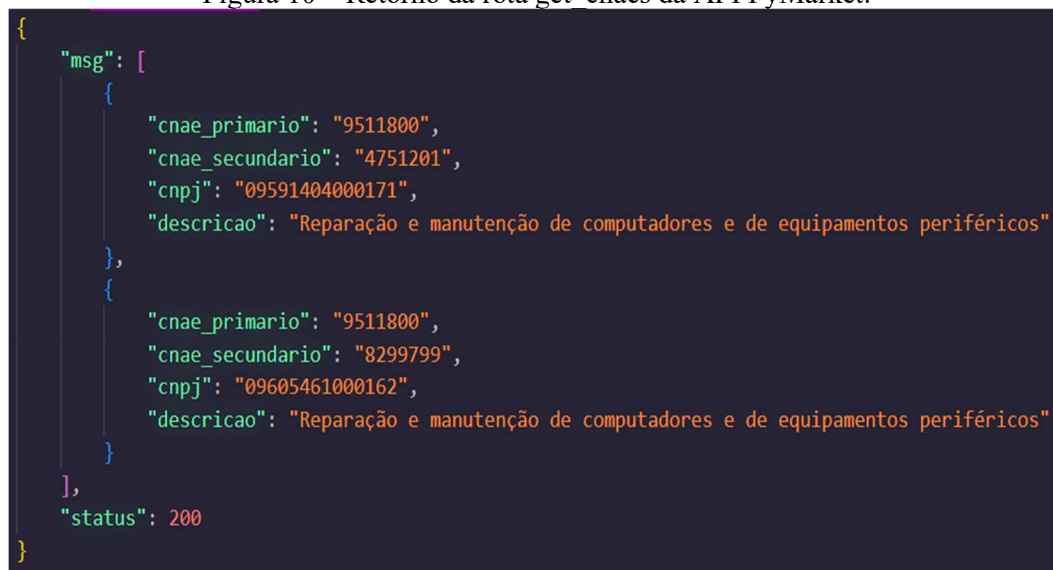
mais algumas informações, conforme imagens abaixo:

Figura 9 – Parâmetros possíveis para a rota get_cnaes da API PyMarket.



Fonte: Giovani Lima (2023).

Figura 10 – Retorno da rota get_cnaes da API PyMarket.



Fonte: Giovani Lima (2023).

5.2.2 Plataforma Web

A plataforma web disponibiliza duas abas navegáveis. Uma aba fornece uma série de instruções sobre o funcionamento da plataforma, conforme imagens abaixo:

Figura 9 – Instruções da plataforma web PyMarket.

Introdução

PyMarket é um projeto tecnológico que tem como objetivo sanitizar os dados de cnpjs disponibilizados pela receita federal e transforma-los em informações que possam gerar valor. O projeto vai disponibilizar inicialmente dois dashboards informando algumas tendências, que são:

- Crescimento e declínio de setores.
- Porte das empresas de cada setor.

Também vai contar com uma API que vai disponibilizar informações a partir de algumas informações que são:

- Tipo de Cnae, Cnaes, ufs e municípios
- Cnpjs

Como funciona

A api conta com 2 rotas GET

- get_cnaes
- get_cnpjs

Na rota GET_CNAES se esperam quatro informações:

- Tipo: Tipo de cnae, se é primário("p") ou secundário("s").
- Códigos: Lista de códigos de Cnaes que se deseja pesquisar.
- Ufs: Lista de Ufs que se deseja pesquisar.
- Municípios: Lista de códigos de municípios que se deseja pesquisar.

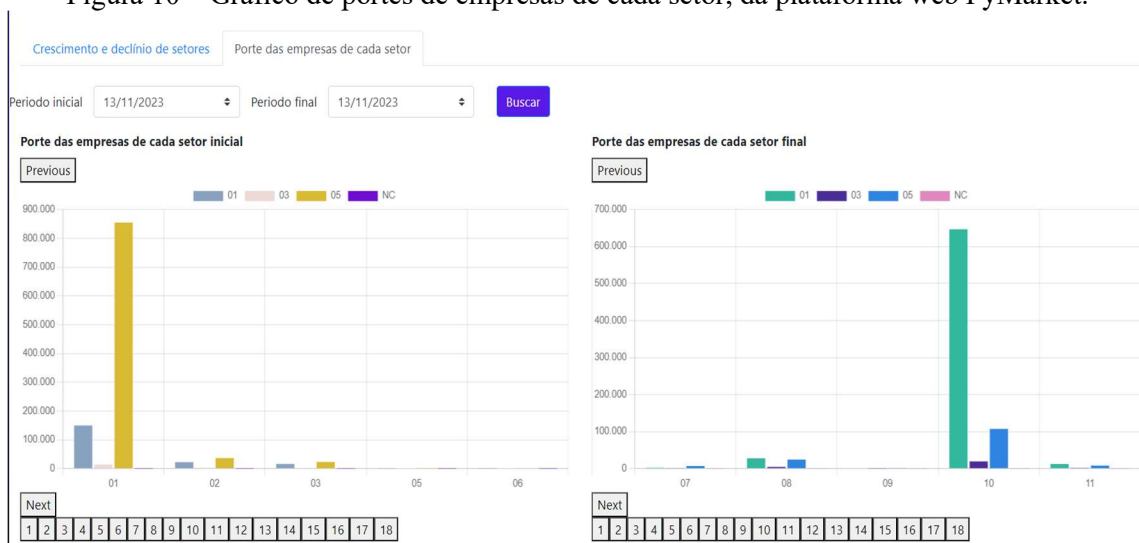
Na rota GET_CNPIJS se espera uma informação:

- Cnpjs: Lista de códigos de cnpjs que se deseja pesquisa.

Fonte: Giovani Lima (2023).

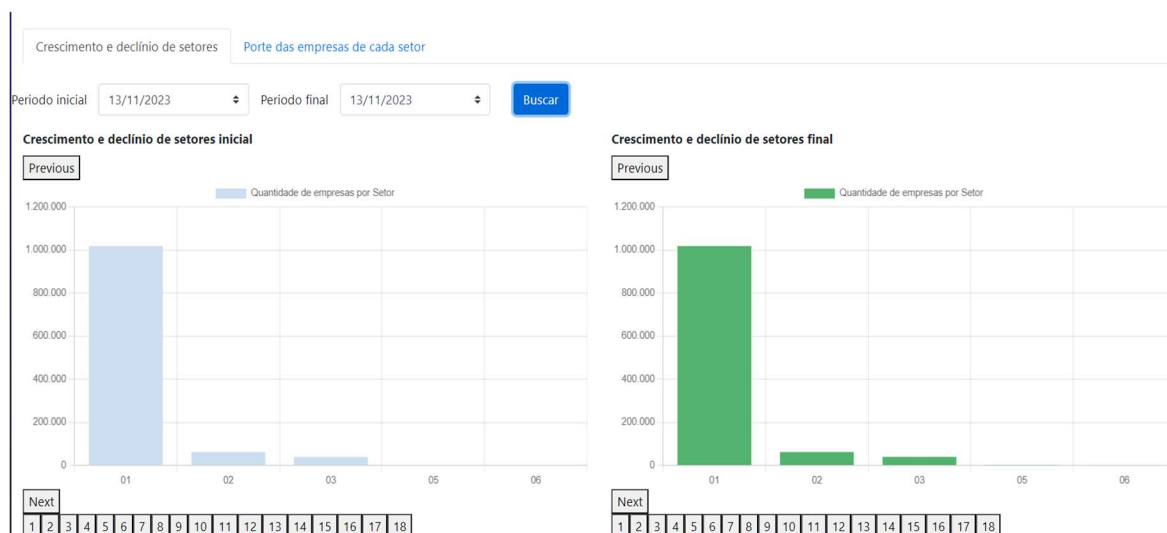
A outra aba, fornece os Dashboards, que contém os gráficos de porte das empresas de cada setor, de crescimento e declínio de setores, conforme as imagens abaixo:

Figura 10 – Gráfico de portes de empresas de cada setor, da plataforma web PyMarket.



Fonte: Giovani Lima (2023).

Figura 11 – Gráfico de crescimento e declínio de setores, da plataforma web PyMarket.



Fonte: Giovani Lima (2023).

5.3 Banco de dados

O banco de dados escolhido foi do Postgresql, pois além de ser robusto e confiável, é um banco de dados gratuito e tem uma documentação completa e de fácil entendimento. É o responsável por armazenar todos os dados disponibilizados pelo ETL e procedures que fazem o pré-processamento das informações que os Dashboards disponibilizam inicialmente, que são:

- Procedure para o dashboard de crescimento e declínio de setores.
- Procedure para o dashboard de portes de empresa por setor.

5.4 Infraestrutura

Todos os recursos serão instalados em um VPS que utilizará o Jenkins como gerenciador. Ele será o responsável por todo o CI/CD do projeto. Também será o responsável pelo disparo de funções agendadas, como por exemplo, a ferramenta de ETL e outras rotinas pertinentes, como por exemplo, backups de banco de dados.

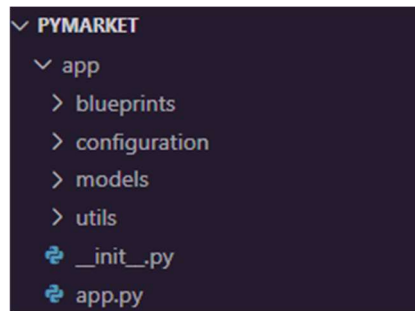
5.5 Estrutura do código

O objetivo central da ferramenta é entregar uma plataforma web que disponibilize dashboards e uma API para consultas de CNAE e CNPJ. Para isso, foi utilizada a linguagem Python e o framework web Flask em conjunto com mais módulos que veremos com maior riqueza de detalhes a seguir.

O código é dividido em diversas partes, iniciando com o arquivo wsgi.py, que é a porta de entrada do projeto. Ele é utilizado pelo Gunicorn, que é um Servidor WSGI. Ele implementa o lado do servidor web para executar aplicativos Python, visto que um servidor web tradicional não entende e nem executa aplicativos Python. No wsgi.py é feita a importação e execução do app desejado.

O app faz uso do conceito de Application Factories do framework Flask para incorporar funcionalidades como se fossem plugins, tornando o app escalável e fácil de manter. Então, é criada uma instancia de Flask e ela é passada por parâmetro para a classe de configuração Configuration, que faz o “encaixe” dos módulos no app, que vai ser importado pelo wsgi e executado pelo Gunicorn. Também está estruturado seguindo o padrão de projeto MVC em conjunto com os conceitos de Service e Repository. Abaixo, uma imagem da estrutura e divisão de pastas do projeto:

Figura 12 – Estrutura do projeto PyMarket

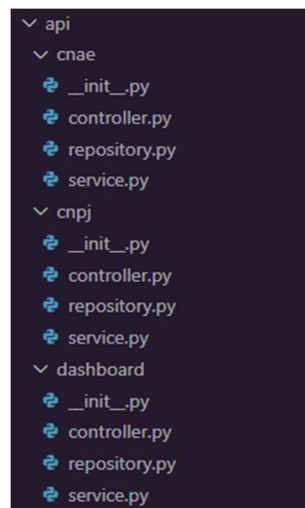


Fonte: Giovani Lima.

Em blueprints, se encontram as funcionalidades de API e WEB.

Em API, estão os códigos para gerar os endpoints de CNAE, CNPJ e DASHBOARDS. Todos têm a mesma estrutura, conforme imagem abaixo:

Figura 13 – Estrutura das funcionalidades da API.



Fonte: Giovani Lima

O controller.py é o responsável por conectar as rotas a determinados services. É ele também que recebe os parâmetros que são passados nas requests e repassa para os services ou recebe alguma função de decorators caso existam, como por exemplo, uma função de proteção de rota. Abaixo uma imagem de como é um controller.py, pois a estrutura é a mesma para todos.

Figura 14 – Funcionalidade de uma controller.

```

controller.py X
app > blueprints > api > cnae > controller.py > ...
1  from app.blueprints.api.cnae.service import CnaeService
2  from flask import request
3
4
5  class CnaeController:
6
7      __slots__ = ("service")
8
9      def __init__(self, service: CnaeService) -> None:
10         self.service = service
11
12     def get_cnaes(self):
13
14         dataset = request.get_json()
15         return self.service.get_cnaes(dataset)
16

```

Fonte: Giovani Lima

Na service.py é onde fica toda a regra de negócio referente ao módulo em que ela se encontra. É também a responsável por acessar o repositório.py, que é o responsável pelo acesso aos dados persistidos da aplicação. Um exemplo de como funciona o service.py, que é estruturado igualmente como todos os outros, se diferenciando apenas no que tange as regras de negócio.

Figura 15 – Funcionalidade de uma service.

```

from app.blueprints.api.cnpj.repository import CnpjRepository
from app.utils.http_response import HttpResponse

class CnpjService():
    __slots__ = ("repository", "http_response")

    def __init__(self, repository: CnpjRepository, http_response: HttpResponse) -> None:
        self.repository = repository
        self.http_response = http_response

    def get_cnpjs(self, dataset):
        sql_where = ""
        len_codigos = len(dataset['cnpjs'])

        for x in range(0, len_codigos):
            sql_where += f"({dataset['cnpjs'][x][0:8]} and "
            sql_where += f"({dataset['cnpjs'][x][8:12]} and "
            sql_where += f"({dataset['cnpjs'][x][12:14]})"

            if x < (len_codigos-1):
                sql_where += " or "

        res = self.repository.get_cnpjs(sql_where)

        if not isinstance(res, Exception):
            return self.http_response("msg", res, 200).http_response()

        else:
            return self.http_response("msg", f"error -> {res}", 500).http_response()

```

Fonte: Giovani Lima

E por fim o repositório.py, que como comentado anteriormente, é responsável pelo acesso aos dados

da aplicação.

Ele geralmente faz a utilização de queries de banco de dados em formato de string. A conexão com o banco de dados é feita via injeção de dependência. Geralmente se utiliza desta forma quando a query é muito complicada e pode ser difícil de ser implementada com um ORM, caso contrário se utiliza a consulta via padrões de ORM. Um exemplo pode ser visto na imagem a seguir:

Figura 16 – Funcionalidade de um repository.

```
repository.py X
app > blueprints > api > cnpj > repository.py > ...
1  from app.utils.database_solution import DatabaseSolution
2
3
4  class CnpjRepository:
5
6      __slots__ = ()
7
8      def get_cnpjs(self, sql_where):
9
10         try:
11             db, schema = DatabaseSolution().get_database, DatabaseSolution().get_schema
12
13             dataset = db.execute_sql(f"""SELECT json_agg(resultados) from
14                                     (select *
15                                      from
16                                         {schema}.tbestabelecimentos t1
17                                         where {sql_where}) as resultados;
18                                     """).fetchall()
19
20             return dataset[0][0]
21
22         except Exception as error:
23             print(str(error))
24             return error
```

Fonte: Giovani Lima

Em WEB estão os códigos para gerar as views da aplicação. Todos possuem praticamente a mesma estrutura, conforme imagem abaixo:

Figura 17 – Estrutura das funcionalidades WEB.

```
app
├── blueprints
│   ├── api
│   └── web
│       ├── dashboard
│       │   ├── static
│       │   ├── templates
│       │   ├── __init__.py
│       │   └── view.py
│       ├── index
│       │   ├── static
│       │   ├── templates
│       │   ├── __init__.py
│       │   └── view.py
│       └── main
│           ├── static
│           ├── templates
│           └── __init__.py
```

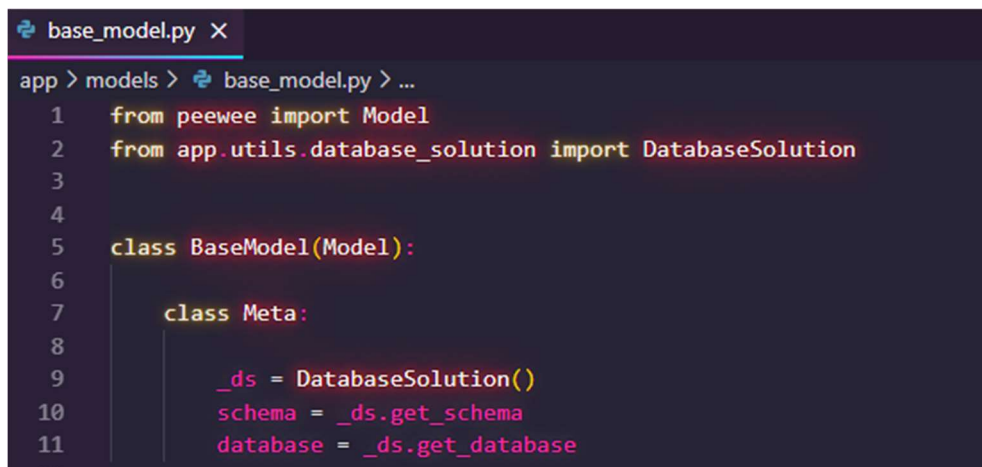
Fonte: Giovani Lima

Na parte de configuration, estão todas as configurações que não fazem parte diretamente do app,

como por exemplo, “encaixe” dos módulos, configuração de conexão com o banco de dados, criação de tabelas, CORS, entre outros

Na parte de models, são todos os modelos ORM da aplicação. Existe um BaseModel, que seria o modelo pai que recebe toda a configuração de banco de dados que vai ser utilizada e os modelos filhos que recebem o modelo pai, conforme imagens a seguir.

Figura 18 – Exemplo de um modelo ORM BaseModel.



```
base_model.py X
app > models > base_model.py > ...
1  from peewee import Model
2  from app.utils.database_solution import DatabaseSolution
3
4
5  class BaseModel(Model):
6
7      class Meta:
8
9          _ds = DatabaseSolution()
10         schema = _ds.get_schema
11         database = _ds.get_database
```

Fonte: Giovani Lima

Figura 19 – Exemplo de um modelo ORM.



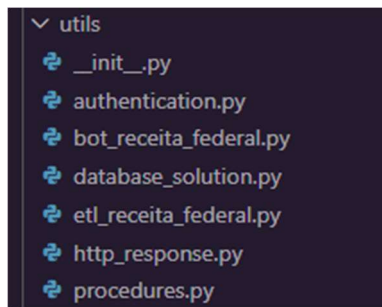
```
update_tabela_geral.py X
app > models > update_tabela_geral.py > ...
1  from playhouse.postgres_ext import PrimaryKeyField, CharField, DateTimeField, JSONField, IntegerField
2  from app.models.base_model import BaseModel
3  import datetime
4
5
6  class UpdateTabelaGeral(BaseModel):
7
8      id = PrimaryKeyField(primary_key=True)
9      tabela = CharField()
10     id_registro_tabela = IntegerField()
11     colunas_afetadas = JSONField()
12     data_hora_modificacao = DateTimeField(default=datetime.datetime.now)
13
14     class Meta:
15         table_name = "tbupdatetabelageral"
```

Fonte: Giovani Lima

Na parte de utils, estão todos os códigos que são do projeto, mas que não impedem a aplicação de funcionar. Geralmente são utilizados pelo projeto de maneira indireta para fazer alguma função muito específica e não recorrente, como por exemplo, o download dos arquivos da receita federal pelo bot_receita_federal.py e o ETL desses arquivos pelo elt_receita_federal.py. Abaixo, segue a estrutura do

utils:

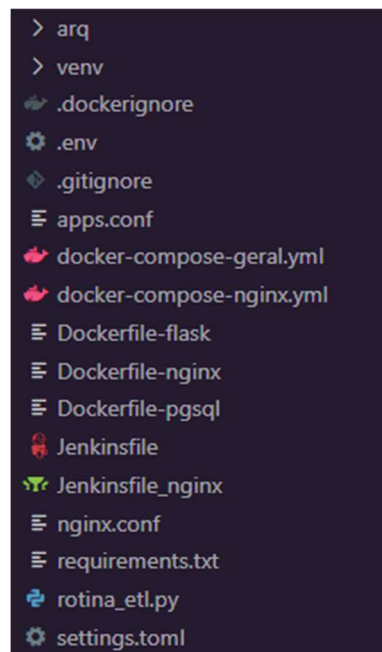
Figura 20 – Estrutura da parte de utils.



Fonte: Giovani Lima

Para finalizar a parte da estrutura, abaixo segue o restante dos arquivos que são mais da parte da infraestrutura do projeto, como containers, configuração de CI/CD, variáveis de ambiente, entre outros, conforme imagem a seguir:

Figura 11 – Estrutura do CI/CD.



Fonte: Giovani Lima

5.6 Regra de uso

Qualquer interessado pode usar a ferramenta, desde que entenda que se trata de algo em fase de testes e que problemas relacionados com a performance e limitações podem ser recorrentes. Os dashboards disponibilizados são para um rápido overview de métricas, logo, para os mesmos não existem regras em particular. A api por outro lado, deverá respeitar algumas regras:

- Todas as keys são obrigatórias.
- No endpoint get_cnaes, a key “tipo”, deverá ser uma string, informando se o cnae é principal(“p”) ou secundário(“s”).
- No endpoint get_cnaes, as keys “codigos”, “ufs” e “municipios” deverão ser listas de

strings, contendo nenhuma([]) ou mais informações(["123456", "78910"]).

- No endpoint `get_cnpjs`, a keys "cnpjs" deverá ser uma lista de string, contendo um(["1234567891011"]) ou mais informações(["1234567891011", "1110987654321"]).

6 METODOLOGIA DE VALIDAÇÃO

Para validação, foram elaboradas algumas perguntas relacionadas a experiência de uso da ferramenta, no que diz respeito a performance da api e a pertinência das informações dos dashboards. Até o momento existem duas empresas do setor financeiro dispostas a fazer a validação: A Empresa de nome fictício "FR" e a empresa de nome fictício "ED". O intuito da validação é verificar questões relacionada ao que o cliente achou do projeto Pymarket, se a os recursos da aplicação tiveram performance satisfatória, se as informações dos gráficos ajudaram em alguma tomada de decisão estratégica, em como foi a experiência do usuário ao utilizar a ferramenta e contribuir de forma textual, com alguma crítica ou sugestão de alguma coisa que pode ser melhorada, mantida ou criada.

A primeira pergunta tem o objetivo saber dos participantes se a ideia do projeto é interessante. O interesse na plataforma é que gera a oportunidade de negócio e crescimento da ferramenta.

A segunda pergunta tem o objetivo de saber sobre a performance da api da ferramenta. A performance faz total diferença, visto que na maioria das vezes os clientes precisam da informação em um tempo muito curto para não perderem oportunidades no mercado.

A terceira tem o objetivo de entender se os dashboards tem alguma pertinência para o negócio das empresas, já que cada segmento de mercado tem uma demanda específica.

A quarta tem o objetivo de saber se a experiência em usar a ferramenta foi satisfatória.

A quinta foca em saber se existem melhorias que podem ser feitas na plataforma.

Essas perguntas foram colocadas em um formulário do Google para serem respondidas e posteriormente se obter insights das respostas.

6.1.1 Sobre a aplicação do formulário

- A validação ocorreu nos dias 19 e 20 de novembro.
- Contou com duas empresas participantes.
- O formulário contém 05 perguntas relacionadas a ferramenta PyMarket.

6.1.2 Sobre o acesso a aplicação

A plataforma PyMarket é uma aplicação pronta para uso em VPS, devido as configurações pré-existentes que foram criadas desde o início do projeto que possibilitaram isso. Porém, por falta de recursos financeiros, não foi possível mantê-la no VPS, devido ao tamanho do banco de dados exceder as capacidades do disco contratado. Foi necessário então, validar a plataforma de maneira local, através de acesso remoto usando a ferramenta Anydesk. Após a validação, os participantes receberam o link do formulário via Whatsapp para deixar suas considerações. As considerações dos participantes se encontram neste [link](#).

7 CONSIDERAÇÕES FINAIS

Conforme já dito nesse documento, a análise de dados é um processo poderoso para qualquer setor. Com ela é possível descobrir informações úteis que trazem vantagens competitivas para diversos segmentos, apoiando na tomada de decisões estratégicas e na geração de insights. Também é um processo que tem diversos níveis de complexidades dependendo do tipo de análise que se deseja. Foram citados também diversos recursos e ferramentas que podem ajudar na análise de dados, mesmo que não se tenha um conhecimento tão grande

em programação.

No decorrer deste documento, foi apresentado um projeto que é uma alternativa para tratar dados que estão sendo disponibilizados e não estão sendo tratados de forma que se possa tirar algum proveito. Durante o decorrer do trabalho, foram passadas por diversas etapas, que foram concluídas com êxito, baseadas na validação que foi proposta.

Existem modificações e melhorias que podem ser feitas no que diz respeito a performance, como por exemplo, no tempo de execução de algumas chamadas da API que podem ser mais performáticas no tempo de resposta, na criação de mais gráficos com diversos outros insights, entre outras.

O projeto foi criado com muita dedicação. Foi literalmente um desafio, onde tive uma série de aprendizados pelo caminho. Sempre reclamei sobre projetos tecnológicos, onde argumentava que muitos eram feitos apenas para terminar o curso. Este com certeza não foi um deles e desejo que ele seja fortemente utilizado. Ele não irá parar no tempo por minha vontade. Enquanto estiver em uso e for de alguma valia, receberá manutenção pela minha pessoa para que continue servindo a todos os interessados.

AGRADECIMENTO(S)

Primeiramente quero agradecer a mim mesmo por enfrentar todos esses anos de faculdade. Só eu sei o trabalho que tive para chegar até a etapa final deste projeto. Foram muitas idas e vindas de trem e ônibus, dias chuvosos andando de motocicleta, problemas mentais e financeiros, assaltos, entre tantas outras dificuldades que não vale a pena citar. Agradecer a minha esposa, que me auxiliou ao máximo, sempre deixando um ambiente propício para que eu pudesse focar nos estudos. Quero agradecer também aos professores que sempre foram solícitos e me passaram todo o conhecimento que puderam, em especial a professora Maria Adelina e os professores Christiano Cadoná e Elgio Schlemer. Agradeço também aos familiares e amigos que me acompanharam e me deram força no processo.

REFERÊNCIAS

SCHWAB, Klaus. The Fourth Industrial Revolution. Disponível em: https://law.unimelb.edu.au/_data/assets/pdf_file/0005/3385454/Schwab-The_Fourth_Industrial_Revolution_Klaus_S.pdf. Acesso em: 19 set 2023.

SHEDROFF, N. Informational Interaction Design: A Unified Field Theory of Design. Disponível em: <https://nathan.com/information-interaction-design-a-unified-field-theory-of-design/>. Acesso em: 17 ago 2023.

PEDRA, D. O que é a Indústria 4.0? Tudo sobre a quarta revolução industrial. Disponível em: <https://www.siteware.com.br/metodologias/o-que-e-industria-4-0/>. Acesso em: 24 set 2023.

ARAUJO, I. C. et al. Indústria 4.0 e seus impactos para o mercado de trabalho. Brazilian Journal of Development, v. 6, n. 4, p. 22326–22342, abr 2020. Disponível em: <https://ojs.brazilianjournals.com.br/ojs/index.php/BRJD/article/view/9370/7915>. Acesso em 24 set 2023.

CETAX, BLOG. Big Data: Tudo o que você precisa saber. Disponível em: <https://cetax.com.br/big-data-tudo-o-que-voce-precisa-saber/>. Acesso em: 24 set 2023.

UCS, BLOG. Big Data: o que é, para que serve, como aplicar e exemplos. Disponível em: <https://ead.ucs.br/blog/big-data#a>. Acesso em: 25 set 2023.

GARTNER, Big Data. Disponível em: <https://www.gartner.com/en/information-technology/glossary/big-data>. Acesso em: 25 set 2023.

GUIMARAES, L. Qual a diferença entre dado e informação? Entenda agora! Disponível em: <https://www.knowsolution.com.br/diferenca-dado-e-informacao/>. Acesso em: 25 set 2023.

DIGITAIS, RESULTADOS. Google Analytics: o que é e como fazer a configuração inicial. Disponível em: <https://resultadosdigitais.com.br/marketing/o-que-e-google-analytics/>. Acesso em 15 nov 2023.

SENSE, QLIK. Analytics moderno em nuvem. Disponível em: <https://www.qlik.com/pt-br/products/qlik->

sense. Acesso em 15 nov 2023.

BI, DOCUMENTAÇÃO DO POWER. Disponível em: <https://learn.microsoft.com/pt-br/power-bi/>. Acesso em 16 nov 2023.

KIMBALL, R.; CASERTA, J. The Data WarehouseETL Toolkit – Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. Disponível em: <https://ia800206.us.archive.org/15/items/2004TheDataWarehouseETLToolkitRalphKimball/2004%20-%20The%20Data%20Warehouse%20ETL%20Toolkit%20%28Ralph%20Kimball%29.pdf>. Acesso em 01 dez 2023.