Faculdades Integradas de Taquara - Faccat
Av. Oscar Martins Rangel, 4.500
Taquara, RS, CEP 95600-000

Curso de Sistemas de Informação

SISTEMA DE INGESTÃO DE DADOS PARA *DATA LAKES*

Juan Vinícius Casagrande Damo
Faculdades Integradas de Taquara - Faccat - Taquara - RS - Brasil
juan.damo@sou.faccat.br

Fernando Lunardelli
Professor Orientador
Faculdades Integradas de Taquara - Faccat - Taquara - RS - Brasil
fernandolunardelli@faccat.br

Resumo

Este artigo apresenta o desenvolvimento de um protótipo de sistema *web*, cujo objetivo principal é facilitar a ingestão de dados para times de engenharia. Este sistema centraliza em uma única interface todo o fluxo de ingestão de dados e, por dispensar a necessidade de desenvolvimentos adicionais e simplificar este processo, pode auxiliar na redução do tempo e custos envolvidos. De forma detalhada, é possível realizar todo o fluxo ETL - *Extract, Transform and Load*, realizando a extração de dados de uma origem, o tratamento e a ingestão. O protótipo abrange o cadastro de uma nova origem, a escolha do tipo de ingestão, sendo elas sob demanda ou agendada, até a saída do dado em formato de arquivo Parquet. Adicionalmente, é possível usar o sistema para ler arquivos Parquet existentes e navegar pelos repositórios de dados.

Palavras-chave: ingestão de dados, fluxo, Parquet.

DATA INGESTION SYSTEM FOR DATA LAKES

Abstract

This article presents the development of a prototype of a web system, whose main objective is to facilitate the ingestion of data for engineering teams. This system centralizes the entire flow of data ingestion in a single interface and, by eliminating the need for additional developments and simplifying this process, it can help to reduce the time and costs involved. In detail, it is possible to perform the entire ETL flow - Extract, Transform and Load, performing data extraction from a source, processing and ingestion. The prototype covers the registration of a new source, the choice of the type of ingestion, whether on demand or scheduled, until the output of the data in Parquet file format. Additionally, you can use the system to read existing Parquet files and browse data repositories..

Keywords: *ingestion, data, flow, Parquet.*

1. INTRODUÇÃO

Na sociedade atual, as organizações buscam informações diariamente para que possam evoluir e expandir, buscando inovações, tecnologias e, principalmente, conhecer e saber o que proporcionar à sociedade (RODRIGUES E BLATTMANN, 2014). A informação em conjunto com a inteligência de negócio traz benefícios como o impulsionamento de estratégias de marketing para uma oferta assertiva e a tomada de decisão baseada em dados, auxiliando para que as organizações possam continuar sendo competitivas, conforme Kondo (2016). Independente da área na qual uma organização está inserida, obter a maior quantidade de informações possíveis, e utilizá-las para o negócio, pode fazer com que estas aumentem sua lucratividade. Estas empresas potencializam seus resultados por meio de dados, aproveitando todas as oportunidades existentes, inserindo informações em inteligências automatizadas, interagindo com clientes, gerando *insights*, aprimorando o desempenho corporativo de forma mais eficiente, prevendo tendências e comportamentos (KONDO, 2016).

Com a necessidade de usar os dados, tornou-se necessário para as empresas possuírem times especializados na área, contendo profissionais como: cientistas de dados, engenheiros de aprendizado de máquina e, dentre outros, os engenheiros de dados (CORRÊA, 2022). Os engenheiros de dados são responsáveis por extrair as informações de uma determinada origem (SCHMID, EICHELBERGER, QIN, 2019), através do processo de ETL - *Extract, Transform and Load*, que inclui a tarefa inicial da extração de dados em uma origem, realiza o tratamento quando necessário e disponibiliza para outros profissionais, mantendo esses dados seguros, acessíveis, organizados e sempre disponíveis (KIMBAL, CASERTA, 2004).

A extração e armazenamento das informações por parte dos engenheiros de dados é realizada através de um processo chamado ingestão de dados. A ingestão de dados consiste no uso de um ambiente muitas vezes complexo, mantido com o uso de uma grande quantidade de ferramentas, pois é necessário atender a todo tipo de origem, sendo estas: bancos de dados, arquivos de log, redes sociais, raspagem de dados em *sites*, dentre outras (MATURANA, ASENJO, 2015). Com um fluxo bem definido, para uma ingestão completa, é necessário processar esses dados extraídos e integrá-los a um *data lake*, local que serve como repositório de armazenamento contendo uma enorme quantidade de dados, sem estrutura ou requisitos não definidos, até que o uso seja exigido (MATURANA, ASENJO, 2015).

O sistema *web* de ingestão de dados para *data lakes* proposto, denominado DIP (*Data*

Ingestion Platform), tem como objetivos diminuir o nível de complexidade de ambiente, unificando todas as ferramentas necessários em uma única plataforma, possibilitando a redução do tempo de desenvolvimento, minimizando custos e a necessidade de grande conhecimento técnico para a ingestão de dados, possibilitando foco na entrega final da informação dentro do repositório de destino.

O protótipo apresentado é baseado em um fluxo de ingestão onde a entrada de dados foi delimitada a um banco de dados relacional MySQL e a saída, delimitada a arquivos do tipo Parquet, que posteriormente são enviados a um sistema de *data lake*.

Como metodologia, foi levada em consideração a experiência obtida trabalhando como um profissional da área, tendo vivenciado a ingestão dos dados e todo seu fluxo e também em conversas informais com outros profissionais. Como pontos norteadores, foram usadas: a compreensão do tipo do dado, compreensão do ambiente e necessidades de *software*, entendimento completo do fluxo e o acompanhamento relacionado ao tempo e custos de toda a ingestão (e a partir da possível redução que este tempo teria se houvesse um processo centralizado e entregue ao profissional de forma mais simples e sem os custos de utilização de uma ferramenta de mercado).

2. REFERENCIAL TEÓRICO

Nesta seção, será apresentado o referencial teórico dos conceitos e *softwares* que estarão presentes no projeto.

2.1 Profissionais da área de dados

As empresas estão diariamente utilizando dados menos estruturados e volumosos, originados de qualquer lugar que seja possível gerar lucro. Bases de dados transacionais, arquivos de texto, planilhas, cliques em *sites* e todos os comportamentos possíveis nas mídias sociais, são alguns exemplos de fontes de dados, que quando devidamente coletadas e analisadas, podem ajudar nos direcionamentos de negócios. Sabendo destas inúmeras oportunidades, as organizações buscam para seus quadros, profissionais capacitados para programar, organizar, manter e analisar essa miríade de dados (DAVENPORT E PATIL, 2022).

De acordo com Massena (2022), o analista de dados é um profissional bem versátil, que tem como principal responsabilidade examinar dados em busca de *insights*. Coletando e

organizando resumos estatísticos e demonstrando através de *dashboards*, que são painéis que permitem relatórios mais visíveis com o uso de gráficos e indicadores, sendo utilizados nas tomadas de decisões. Geralmente esses profissionais têm um bom raciocínio lógico e uma ótima capacidade de interpretação que lhes permitem melhores compreensões acerca dos dados investigados (MASSENA, 2022).

O engenheiro de dados é um profissional muito requisitado nas empresas que utilizam dados, conforme Revelo (2021). Essa afirmação se dá devido a um grande fator: todos os dados utilizados pelos profissionais da área precisam existir, e o engenheiro de dados tem como responsabilidades coletar essas informações, independentemente de sua origem, processá-los e disponibilizá-los em um destino, comumente são usados *data lakes*.

A ingestão de dados é um dos principais trabalhos do engenheiro de dados, visto que é um fluxo baseado em acessar dados em uma origem, podendo ser uma base relacional do tipo MySQL, por exemplo, processá-los (aplicando alguma transformação/evolução) usando uma linguagem como Python e finalmente disponibilizá-los em um destino. Esse destino servirá de origem para os demais profissionais acessarem as informações e realizarem suas implementações (QIN, SCHMID, 2019). Além deste processo, o engenheiro também tem como responsabilidades manter esses dados de forma segura, deixar sempre disponíveis, trabalhar na gestão dos acessos e na disponibilidade (QIN, SCHMID, 2019).

O cientista de dados é um profissional que une estatística, visão de negócio e tecnologia para apresentar *insights,* podendo prever tendências, encontrar problemas, falhas, mostrar necessidades e ajudar a tomar as decisões de negócio que possam gerar valor para a empresa (OLIVEIRA, 2022).

Analista de *Business Intelligence* ou analista de BI, é o profissional que "dá vida" aos dados, estando na inteligência do negócio (REVELO, 2021). Uma das principais funções é realizar a análise das informações, identificando oportunidades de negócio, melhorias nas estratégias e nos processos. Munidos de ferramentas que criam *dashboards*, esses profissionais ajudam na criação de metas e números, auxiliando a nortear a estratégia de negócio (REVELO, 2021).

A área de *Machine Learning* (ML) se deriva da área de inteligência artificial. O profissional desta área tem como umas das maiores responsabilidades o planejamento e desenvolvimento de modelos para criar soluções usando o aprendizado de máquina, utilizando técnicas e modelos preditivos para alavancarem outros sistemas do negócio (REVELO, 2021).

Finalizando, o Engenheiro de Inteligência Artificial precisa ter conhecimento e capacidade de entender o negócio e os desafios que o acompanham. Com os recursos da inteligência artificial, criar soluções para problemas, realizando pesquisas, podendo agilizar processos contínuos (NASCIMENTO, 2021).

2.2 Ingestão de dados

As informações existentes no mundo estão espalhadas em diversos locais tais como *sites*, redes sociais, bancos de dados, arquivo de texto, planilhas dentre outros. Para que os profissionais na área de dados, como o analista de dados ou o engenheiro de aprendizado de máquina possam gerar valor destes dados, eles precisam estar acessíveis em um único repositório (SOUSA, 2021). A ingestão de dados é o processo que coleta informações em sua origem, processa e armazena em um destino, possibilitando a disponibilização para uso geral, conforme Armoogum e Li (2019). Ainda sobre a ingestão, existem duas formas de separarmos os processos. A ingestão sob demanda, aquela que é executada uma única vez, e a ingestão agendada, fluxo que é executado de forma recorrente, agendada (KAVA, 2022).

Como exemplo de ingestão, analistas de *business intelligence* de uma empresa precisam criar gráficos e relatórios conforme as informações de um determinado item da bolsa de valores. Com a técnica de raspagem de dados, é possível coletar informações de *sites* (SOUSA, 2021). Com o uso dessa técnica, e como exemplificação simples de uma ingestão de dados, podem ser coletadas informações de itens em um *site* de comércio eletrônico, estas informações são processadas e são realizados ajustes conforme a necessidade e, por fim, armazenadas em um repositório, onde o acesso ao analista seria disponibilizado.

2.3 Dados

Dado é um termo que caracteriza coisas ou tipos de coisas, podendo possuir ou ter muitos sentidos, caracterizados pelo contexto o qual se encontra. Um número ou uma palavra podem ser considerados dados, sendo uma representação ou um atributo de algo, conforme Arakaki e Arakaki (2020).

2.3.1 Dados estruturados

Dados estruturados possuem uma estrutura pensada com um propósito bem definido.

Assim como uma tabela, que existente em uma base de dados relacional, tem sua estrutura estabelecida, ela não aceita informações diferentes e não tem sua estrutura alterada com frequência. Uma base estruturada pode possuir tabelas, as quais tem colunas e cada uma com um tipo de dado definido, com um número inteiro ou uma data, não podendo ser diferente disso (REDE NACIONAL DE ENSINO E PESQUISA, 2022).

2.3.2 Dados semiestruturados

Os dados semiestruturados possuem um determinado controle e rigidez, porém menor do que os dados estruturados, sem um esquema fixo. Um exemplo é um código em XML, o qual possui *tags* e marcadores para organizar e impor hierarquias nos elementos, campos e registros (REDE NACIONAL DE ENSINO E PESQUISA, 2022).

2.3.3 Dados não estruturados

A composição de 80% dos dados espalhados pelo mundo são não estruturados, não necessitando de estruturas bem definidas, podendo possuir vários elementos com uma organização aberta, com quaisquer tipos e formatos. Exemplos destes dados são fotos, músicas e vídeos (REDE NACIONAL DE ENSINO E PESQUISA, 2022).

2.4 *Data lake*

O termo *data lake* pode ser definido como um local onde ficam armazenados todos os tipos de dados em uma quantidade expressiva, podendo ser dados estruturados, semiestruturados e não estruturados. Os *data lakes* possuem algumas características específicas que são determinadas em três “V”: volume – volume de dados extremamente grande; velocidade – uma taxa de transferência de dados de altíssima velocidade; variedade – possibilidade de armazenar todo e qualquer tipo de dado (MILOSLAVSKAYA, TOLSTOY, 2016).

Algumas literaturas trazem o conceito dos cinco “V”, adicionando: veracidade – a origem de um dado no *data lake* pode ser qualquer uma, porém, uma origem pode não conter dados totalmente corretos ou confiáveis. A precisão das análises realizadas usando os dados vai depender da sua veracidade; valor – considerado um dos “V” mais importantes, as informações existentes em um *data lake* são resultantes de um trabalho complexo, porém,

este trabalho só passa a ter sentido quanto os dados possuem valor. As informações precisam ter uma razão que auxiliem na tomada das decisões e que, principalmente, tragam lucro para a organização (ISHWARAPPA, ANURADHA, 2015).

2.5 Banco de dados

Um banco de dados pode ser definido como uma coleção estruturada de dados, podendo conter quaisquer tipos de dados como números, datas, palavras, uma lista de itens, números com vírgula entre outros. Esses dados são organizados em tabelas e as tabelas, por sua vez, possuem colunas, as quais contém os dados ou registros (MYSQL, 2022).

2.6 *Open Source*

Open Source é uma definição para *softwares* de código aberto o qual qualquer pessoa pode utilizar e realizar modificações, estudos e distribuir gratuitamente para outra pessoa (MYSQL, 2020).

2.7 Tecnologia, Metodologia de Desenvolvimento e Ferramentas

Esta seção traz o referencial teórico referente à metodologia e as tecnologias usadas no projeto de pesquisa.

2.7.1 Github

O Github é uma plataforma na qual profissionais da área de tecnologia podem hospedar seus códigos, obtendo uma certa segurança. É possível criar versões de código, gerar histórico e compartilhar com outros membros para que possam contribuir no desenvolvimento e acessá-lo de qualquer local (GITHUB, 2022).

2.7.2 CSS

CSS é uma linguagem utilizada para estilização de *sites*. Com ela podemos alterar cores e organizar elementos, posicionar em determinados lugares da página ou criar algumas animações como a troca de cor de um botão quando passamos o cursor sobre ele.

Basicamente podemos criar quaisquer estilos (W3SCHOOLS, 2022).

2.7.3 MySQL

O MySQL é um sistema de gerenciamento de banco de dados criado em 1995, responsável por processar os dados, acessá-los, criar inserções, atualizações, exclusões e mostrar estes dados, dentre outras ações, utilizando a linguagem SQL. O sistema pode ser encontrado nas versões de licenciamento GPL (*Open Source*) e comercial (MYSQL, 2020).

2.7.4 Streamlit

Como uma biblioteca de código aberto para Python, o Streamlit é muito usado por profissionais na área de dados devido a sua principal função, fornecer ferramentas para criar aplicações *web* para análise de dados, sem a necessidade de conhecimento aprofundado em desenvolvimento *web* (STREAMLIT, 2022).

Com poucas linhas de códigos podemos implementar a leitura de um arquivo CSV e, a partir dessa leitura, gerar gráficos para a visualização de dados, com filtros diversos, sendo uma ótima ferramenta para auxiliar na tomada de decisão. Há muitos ganhos também no desenvolvimento de códigos para aprendizado de máquina, treinamento de redes neurais e manipulação de dados (STREAMLIT, 2022).

2.7.5 Apache Parquet

O Apache Parquet é um formato de arquivos (.parquet) orientado a colunas. É de código aberto e projetado para o trabalho com grandes volumes de dados, sendo extremamente eficiente na gravação das informações. Os arquivos Parquet possuem uma excelente compactação, extremamente útil para grandes massas de dados, possuindo também um desempenho aprimorado, ajudando na redução de custo de processamento. O formato está disponível para várias linguagens como Python, Java, Scala, C++ dentre outras (APACHE PARQUET, 2022).

2.7.6 Pandas

O Pandas é uma biblioteca *open source* para Python, muito utilizada por

profissionais na área de dados, desenvolvida para análise e manipulação de informações. Com ela é possível, facilmente, realizar a leitura de várias origens como arquivos CSV, JSON, Parquet, TXT entre outros, até bases de dados relacionais. Com a leitura destes dados, existem funções completas de análise, limpeza e formatação de dados, podendo também criar gráficos. O Pandas disponibiliza a escrita de dados em vários destinos como CSV, XLSX, XLS, Parquet dentre outros (PANDAS, 2022).

2.7.7 Python

Lançada em 1991, Python é uma linguagem de programação de alto nível. É uma das linguagens mais utilizadas no mundo devido sua praticidade e a gama de áreas atendidas, desde desenvolvimento *web*, aprendizado de máquina, análise de dados e outros sistemas (PYTHON, 2022).

2.7.8 Método Kanban

Surgido no Japão, na década de 70, criado pela empresa Toyota, o Kanban nasceu para auxiliar no controle da produção dos diferentes modelos de veículos, diminuindo ao máximo o tempo de entrega e atrasos ajudando também na redução de custos. Como um instrumento para o método *Just-in-times*, o kanban serve para a implantação possuindo um conjunto de técnicas capazes de controlar a administração da produção de um determinado produto (SILVA E ANASTÁCIO, 2019).

O método kanban é de simples uso, proposto para eficiência no controle e produção com melhoria contínua. As demandas são divididas em tarefas, cada tarefa é descrita em um cartão, cada cartão possui um nível de urgência o qual é definido pelo dono do produto. As tarefas são divididas em *to-do* (para fazer), *doing* (sendo realizadas) e *done* (prontas). Inicialmente é arrastado uma tarefa de *to-do* para *doing* e, ao ser concluída, arrastada para *done*. Dessa forma as atividades são controladas até serem concluídas, sempre obedecendo uma fila conforme sua criticidade (SILVA E ANASTÁCIO, 2019).

3. SISTEMAS RELACIONADOS

O mercado de ferramentas relacionadas a dados é consideravelmente grande, possuindo muitos *softwares* que podem oferecer vários tipos de entrega para cientistas de

dados, analista de dados, analista de *business intelligence* entre outros. Quando estreitamos esse domínio para produtos voltados à engenharia de dados, a abrangência não é tão grande.

Foram selecionadas algumas ferramentas que realizam a entrega da ingestão de dados:

- Azure Data Factory: ferramenta da Microsoft que fornece integração dos dados e ingestão de modo visual. A proposta é oferecer a construção de um fluxo sem código (arrasta/solta). Além da simplicidade no desenvolvimento, existe a facilidade de integração com outras ferramentas Microsoft (MICROSOFT, 2022).
- AWS Glue: uma plataforma *web* da Amazon, que realiza a entrega da integração de dados de forma visual e por código. Além do serviço e ingestão, o Glue também pode ser usado como catálogo de base de dados quando os dados estão armazenados no *data lake* da Amazon (AMAZON, 2022).
- Oracle GoldenGate: *software* da Oracle que é implementado em um servidor, que pode ser usado para realizar a replicação de dados. O destino da replicação pode ser desde diretórios locais a *data lakes* na nuvem (ORACLE, 2022).
- SAP Data Services: conforme a SAP (2022), o *software* melhora a qualidade dos dados, fornecendo o serviço de integração de dados, realizando a ingestão de bases da própria SAP, ou terceiras, destinando a *data lakes* locais ou em nuvem, trabalhando com processamento paralelo.
- StreamSets: reconhecido na Gartner, a menos tempo que as demais soluções mencionadas, como uma ferramenta de integração de dados. O StreamSets fornece serviços de ingestão de dados automatizados, podendo montar fluxos inteligentes sem restrição de origem ou destino (STREAMSETS, 2022).

4. METODOLOGIA

O desenvolvimento deste trabalho seguiu uma série de etapas, iniciando pela identificação de um problema aplicado a ser abordado e de uma revisão comparativa de ferramentas de mercado que pudessem ser usadas como referencial para resolução deste problema.

Logo após, um conjunto de requisitos foi levantado de forma informal, a partir de conversas com colegas de trabalho, especialistas nas áreas de dados com diferentes graus de experiência, e utilizando a própria vivência e necessidades profissionais como norteadoras. De

posse dos requisitos, um protótipo de sistema de ingestão de dados para *data lakes* foi desenvolvido e posteriormente apresentado novamente para o grupo profissional, anteriormente consultado.

Nesta segunda conversa informal, alguns resultados observados, com o uso do protótipo foram coletados e são apresentados no capítulo seis, assim como comentários e sugestões relevantes foram organizados nos capítulo sete, que trata sobre possíveis aperfeiçoamentos futuros deste trabalho.

4.1. Problema de pesquisa

O conhecimento necessário para realizar a ingestão de dados é obtido pelo engenheiro de dados conforme Maturana e Asenjo (2015). Segundo Oliveira (2020), este tipo de profissional passou a estar menos disponível no mercado devido a grande demanda das empresas e a falta de investimentos para a capacitação na área. Os engenheiros de dados, por estarem escassos e possuírem uma alta procura por todo o mundo, têm um elevado salário, gerando para a empresa, que necessita deste profissional, um alto custo, além de comumente, terem a necessidade de mais de um profissional (OLIVEIRA, 2020).

Aliado ao alto custo e escassez de profissionais, existe a alta complexidade na ingestão de dados e seu fluxo. Atuando como um profissional da área, pertenço a um time de nove engenheiros de dados onde, diariamente, realizamos a integração de dados entre várias origens. O trabalho demanda: entender como são dispostos os dados na origem, desenvolver um conector para acessar esta origem, criar códigos em Python para processar tais dados, montar um ambiente de *cluster* em nuvem para a execução dos códigos, gerir ferramentas que orquestram o fluxo da ingestão, compreender o tipo de saída de dado e saber como disponibilizar no destino. Todo este fluxo, requer bastante tempo de trabalho e grande complexidade para gerir o ambiente e realizar os desenvolvimentos. Por exemplo, um fluxo como o descrito, pode levar em média, de um a três dias para ser construído, multiplicando esse tempo pela quantidade de profissionais, podemos entender que o custo para manter esse tipo de entrega é alto.

Desta maneira, como forma de mitigar os problemas na criação de fluxos de ingestão, este trabalho se propõe a apresentar um protótipo de sistema visando a redução na complexidade e tempo de desenvolvimento, assim como no custo de operação.

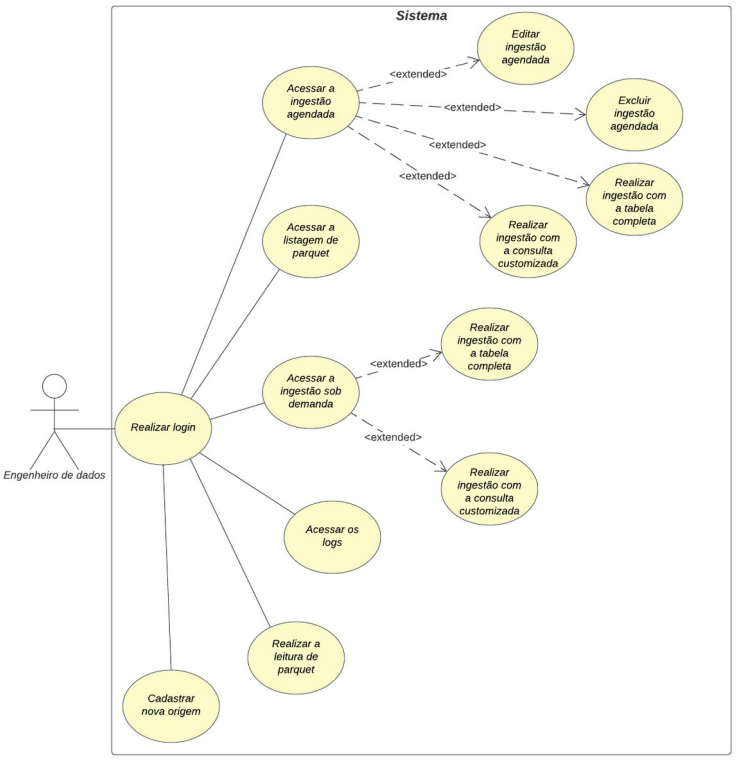
4.2. Requisitos

A análise e identificação de requisitos foram criados como uma lista, a partir da história do usuário, em conversas informais com engenheiros de dados em ambiente profissional e a partir da própria necessidade. Com o levantamento, foram identificados que os seguintes requisitos deveriam existir no protótipo:

- Ingestão de dados sob demanda (*On Demand*): opção para realizar a ingestão uma única vez;
- Ingestão de dados agendada (*Schedule*): operação para realizar a ingestão de forma agendada, quando existe a necessidade de uma rotina;
- Listagem de Parquets (Parquet *List*): utilizado para listar todos os arquivos Parquet existentes no repositório destino, permitindo a navegação e leitura dos registros;
- Logs: tela que contém os logs do sistema para as ações realizadas nas ingestões;
- *Upload* Parquet: opção utilizada para enviar arquivos Parquet para a plataforma, permitindo realizar a leitura dos registros;
- Configuração de origem: opção utilizada para o cadastro de origens das ingestões;
- *Login/logout*: para a possibilidade de restrição de acesso e fluxo de sessão de usuário na plataforma.

Abaixo o diagrama de caso de uso, criado para auxiliar no levantamento dos requisitos funcionais:

Figura 1 – Diagrama de caso de uso



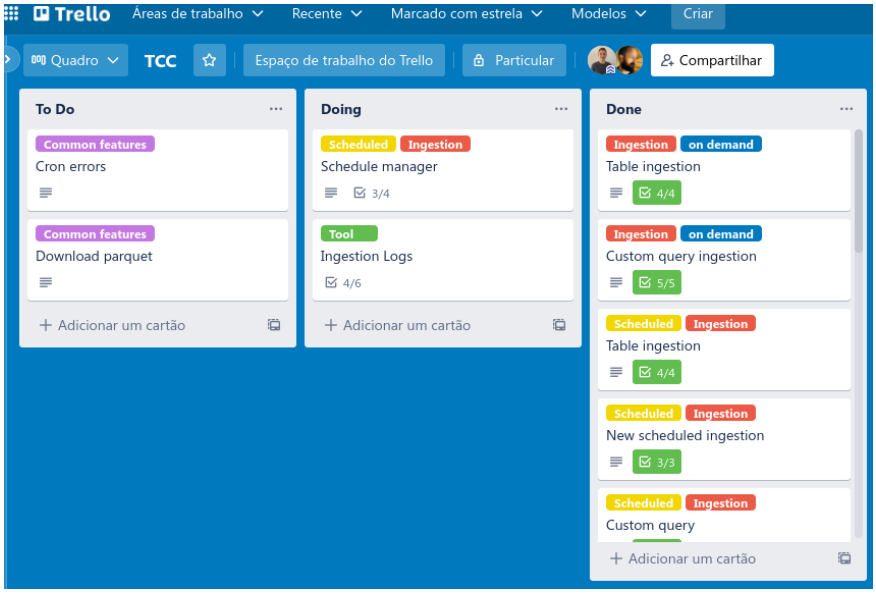
Fonte: Autor (2022).

5. DESENVOLVIMENTO

Com o propósito de resolver as dificuldades impostas pela ingestão de dados, iniciou-se a criação de um protótipo chamado DIP, o qual poderia centralizar todo o fluxo em uma única plataforma, minimizando o nível de complexidade e quantidade de ferramentas necessárias.

Para a organização do desenvolvimento foi utilizada a metodologia kanban, escolhida devido ao dinamismo oferecido, visto que o escopo não era totalmente definido, contendo várias alterações no decorrer do desenvolvimento. As atividades relacionadas foram organizadas utilizando a ferramenta Trello, conforme demonstrado na Figura 2.

Figura 2 – Quadro Kanban usando Trello



Fonte: Autor (2022).

As tarefas foram divididas em grupos e em cada uma das atividades foi definido um nível de urgência e ordem de construção.

Os grupos divididos condizem com as funcionalidades macro levantadas, sendo elas: *Tool* – ferramentas utilizadas no sistema; *scheduled* – ingestão agendada; *configuration* – configurações gerais do sistema; *on demand* – ingestão sob demanda; UX – atividades relacionadas ao design e experiência do usuário; *common features* – *features* comuns de um sistema como o login.

As próximas seções descrevem o processo de desenvolvimento em si, suas ferramentas e métodos, assim como trazem detalhamentos sobre a implementação.

5.1 Descrição geral do desenvolvimento

Conforme a Red Hat (2019), uma IDE (*integrated development environment*) é um ambiente utilizado para o desenvolvimento de *software* e *scripts* em geral. Para o desenvolvimento do protótipo deste estudo, foi utilizado como IDE o Visual Studio Code (VS Code) com uma extensão para Python. Essa IDE foi escolhida devido a facilidade e a experiência de uso.

O protótipo foi desenvolvido na linguagem Python. A escolha da linguagem se dá devido a vários fatores. O primeiro deles é a grande comunidade ativa existente, fazendo com que sempre tenham melhorias, um segundo fator é a gama enorme de bibliotecas disponíveis

para utilização de dados, que somente são possíveis através dessa linguagem, facilitando o desenvolvimento (XIMENES, 2020) e, por fim, a familiaridade com a linguagem.

Corroborando com as afirmações anteriores, duas bibliotecas foram usadas para desenvolvimento do protótipo, o Streamlit e Pandas, bibliotecas que estão disponíveis somente para Python (STREAMLIT, 2022). Focado em facilitar o desenvolvimento *web*, o Streamlit foi usado no DIP para criar toda a interface gráfica como os botões, campos de preenchimento e seleção, barra lateral e etc. O Streamlit tem um ótimo suporte ao Pandas, essa por sua vez foi utilizada para realizar a leitura dos registros contidos na base de dados MySQL, processar os dados em memória e gravá-los no destino em formato Parquet. Além de possuir um ótimo suporte ao Pandas, o Streamlit também facilita o emprego de CSS, usado para estilizar a parte gráfica, permitindo criar a identidade visual do protótipo.

O MySQL foi escolhido para realizar a gestão das informações da plataforma e como origem dos dados, devido a familiaridade com o banco de dados e também por possuir uma comunidade ativa, sendo fácil de obter informações no fórum oficial.

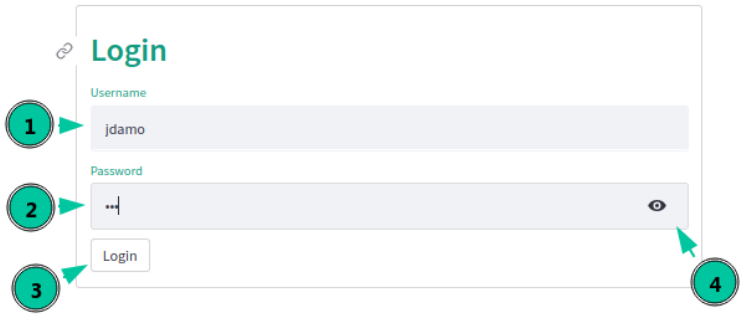
Para controlar os agendamentos do DIP foi utilizado o Cron (sistema de agendamento de tarefas) do Ubuntu 22. A gestão pelo Cron é de fácil implementação e entendimento pelos engenheiros de dados, visto que é a forma que muitas ferramentas na área de dados usam para agendar as atividades. Com o Python ficou fácil e simples gerir os agendamentos das ingestões.

5.2 Detalhamento das funcionalidades

O detalhamento das funcionalidades desenvolvidas segue uma organização de passo-a-passo, para facilitar o entendimento do fluxo de utilização idealizado.

Inicialmente, o usuário deve realizar *login* no protótipo através da seguinte tela:

Figura 3 – Tela de login

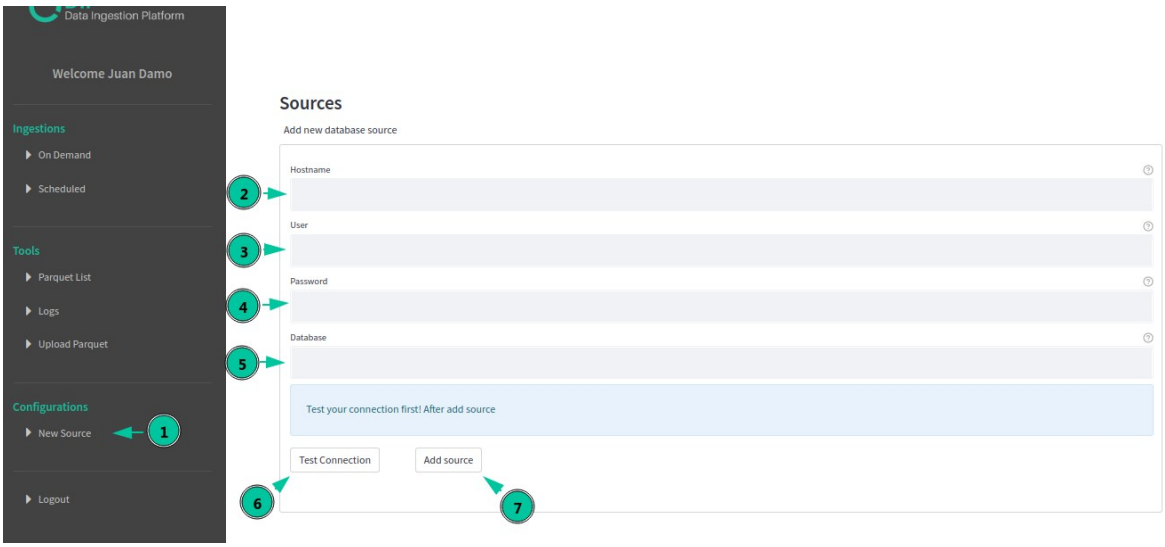


Fonte: Autor (2022).

1. Campo para colocar o usuário;
2. Campo para colocar a senha;
3. Botão para clicar e realizar o *login*;
4. Botão para deixar a senha visível.

A próxima etapa para o uso da ferramenta é cadastrar uma origem, para então iniciar a configuração da primeira parte da ingestão dos dados, conforme tela a seguir:

Figura 4 – Nova origem



Fonte: Autor (2022).

Botão para acessar a tela onde será cadastrada a nova origem para a ingestão;

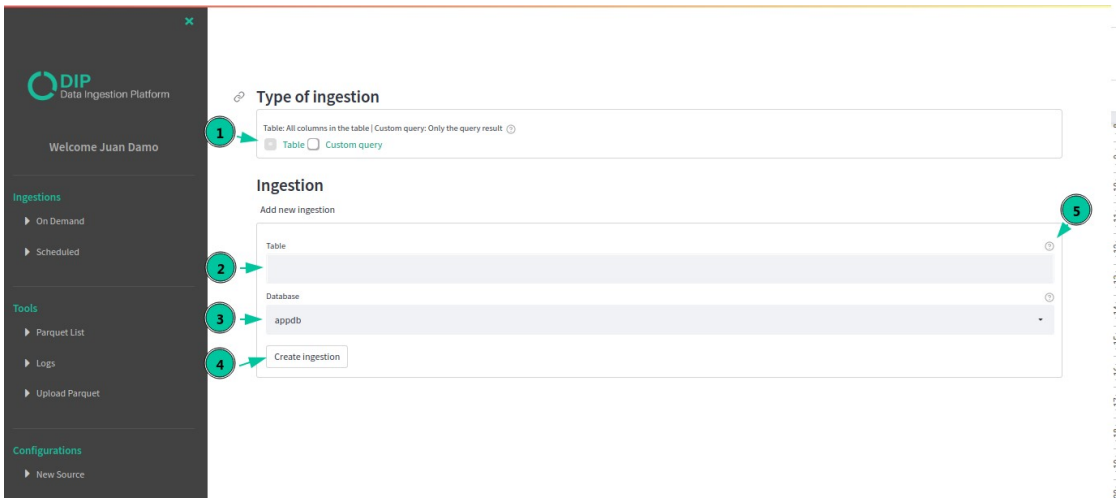
1. Acesso à área de criação de nova base de origem (*new source*);
2. Campo usado para informar o endereço da base de dados;
3. Campo para informar o usuário de acesso a base de origem;
4. Campo onde é informada a senha de acesso a base;
5. Campo para informar o nome da base de dados;
6. Botão para testar a conexão com a base de dados, usando as informações preenchida nos campos anteriores;
7. Botão usado para concluir a adição da origem.

Após adicionar uma origem, é necessário informar um tipo de ingestão que será realizada: sob demanda ou agendada. Primeiramente será demonstrada a tela de ingestão sob

demanda. A ingestão do tipo sob demanda tem dois subtipos: Tabela (*Table*) e Consulta Customizada (*Custom Query*).

A ingestão de subtipo *Table* é demonstrada abaixo:

Figura 5 – Ingestão sob demanda

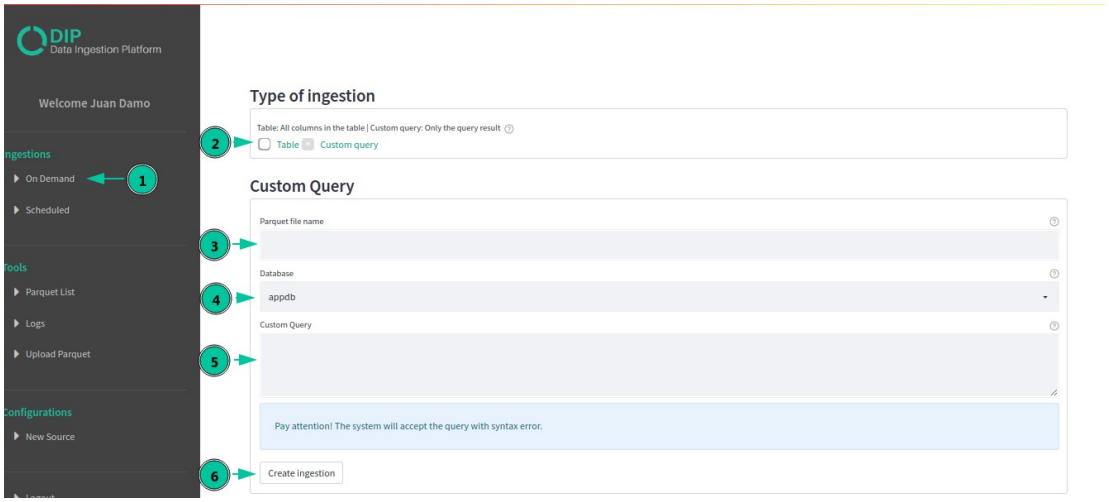


Fonte: Autor (2022).

1. Botão para selecionar o tipo de ingestão;
2. Campo para informar o nome da tabela que será realizada a ingestão;
3. Menu de seleção onde deve ser informada a base de dados onde está a tabela indicada no campo anterior;
4. Botão que irá encerrar o fluxo de configuração, criando a ingestão;
5. Botão de ajuda, que estará em todos os campos, onde será mostrada uma caixa com informações adicionais contextualizadas.

A ingestão de subtipo *Custom Query* é demonstrado abaixo:

Figura 6 – Ingestão sob demanda – consulta customizada

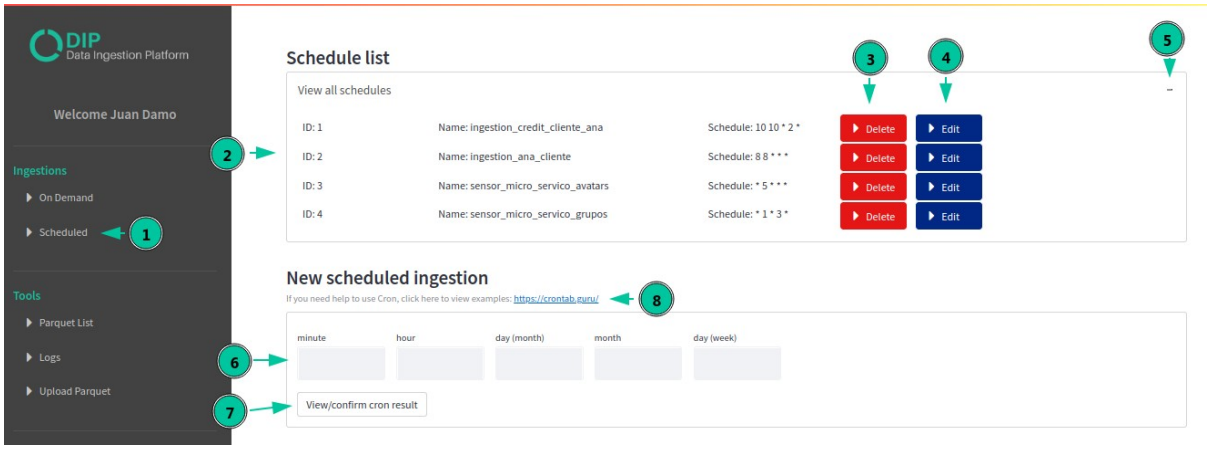


Fonte: Autor (2022).

1. Opção para acessar a tela de ingestão sob demanda (*On Demand*);
2. Botão seletor, abrindo a opção de consulta customizada, em vez de uma tabela completa;
3. Campo para colocar o nome que será integrado ao arquivo Parquet. Como é uma única ingestão, é necessário preencher este campo para poder identificar posteriormente o arquivo;
4. Menu de seleção onde deve ser informada a base de dados onde a consulta anterior será executada;
5. Campo que será escrita a consulta SQL (MySQL);
6. Botão que irá encerrar o fluxo de configuração, criando a ingestão.

A tela de ingestão agendada, que será demonstrada abaixo, será dividida em duas imagens diferentes devido ao tamanho da tela:

Figura 7 – Ingestão agendada

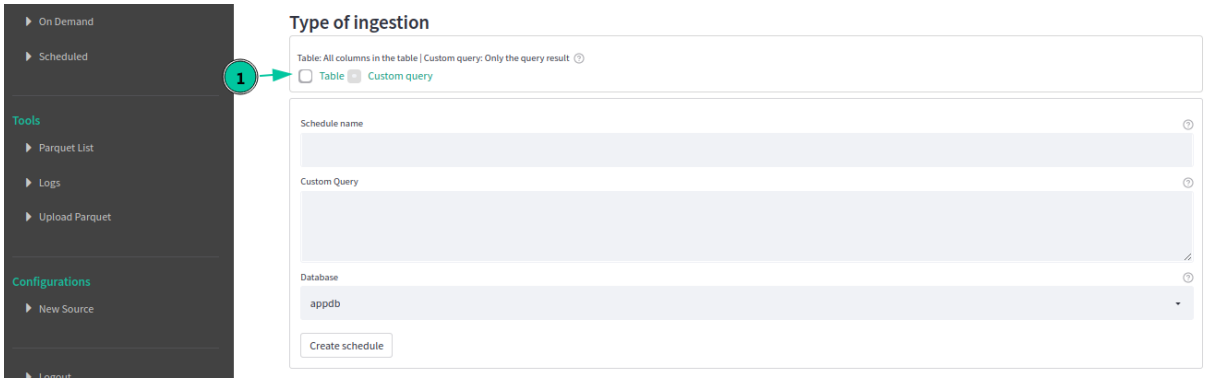


Fonte: Autor (2022).

1. Menu que dará acesso a tela da ingestão agendada (*scheduled*);
2. Menu onde são listadas as ingestões agendadas existentes, contendo o ID da ingestão, o nome e o agendamento;
3. Botão para realizar a exclusão de uma ingestão agendada;
4. Botão para realizar a edição de uma ingestão agendada;
5. Botão para minimizar e expandir a listagem de agendamentos;
6. Campos para preenchimento dos agendamentos, conforme padrão do Cron;
7. Botão para visualizar e confirmar o agendamento;
8. Link externo ao sistema, o qual abre o site oficial da documentação do Cron.

Ainda na tela de ingestão agendada, na segunda parte tem as opções da origem. Nessa tela não serão descritos os detalhes, visto que são os mesmos descritos na figura 5, podendo realizar uma ingestão agendada de uma tabela completa ou de uma consulta customizada.

Figura 8 – Ingestão agendada – consulta customizada



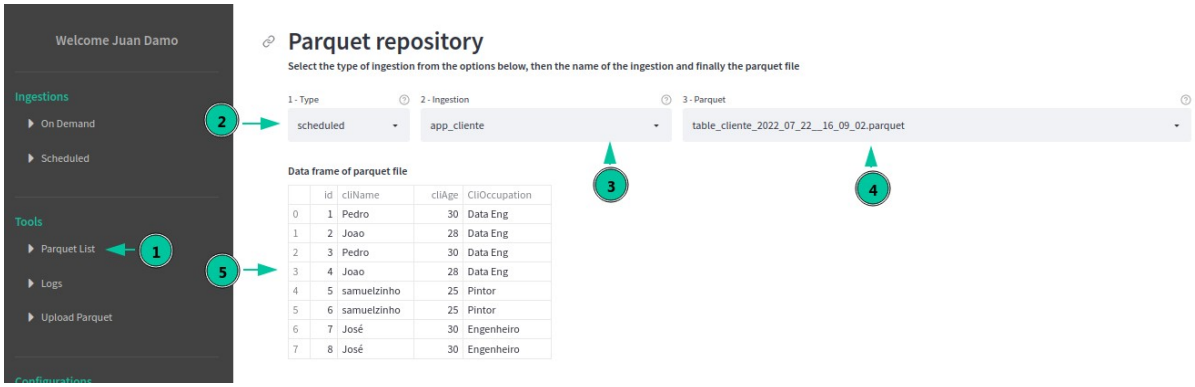
Fonte: Autor (2022).

1. Botão seletor do tipo de ingestão.

Seguindo para as ferramentas do protótipo, serão apresentadas as telas Parquet *List*, *Logs* e *Upload Parquet*, nessa respectiva ordem:

Na tela de Listagem de Parquet (*Parquet List*), é possível ter uma listagem em forma de repositório dos arquivos gerados pelo sistema.

Figura 9 – Parquet *list*



Fonte: Autor (2022).

1. Menu que dará acesso a tela com a listagem dos Parquet;
2. Menu para selecionar o tipo de ingestão que gerou o arquivo desejado;
3. Menu para selecionar o nome da ingestão que gerou o arquivo desejado;
4. Menu para selecionar o arquivo;
5. Tela que mostra os registros do Parquet selecionado.

Já na tela de Logs de ingestão (*Log ingestions*), uma listagem das atividades realizadas no sistema para fins de auditoria.

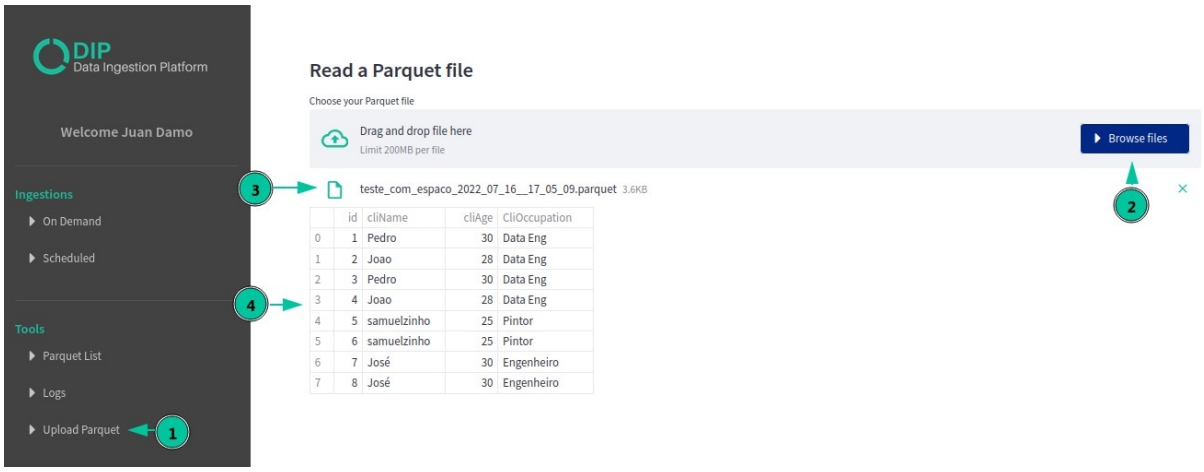
Figura 10 – Logs

Fonte: Autor (2022).

1. Menu que dará acesso a tela com os *logs*;
2. Tabela que mostra informações com o id da ação, o nome da base de dados, o nome da tabela utilizada, seguido pelo usuário que realizou a ação, o tipo de trabalho realizado e, por fim, quando foi executado.

Por fim, na tela de Envio de Parquet (*Upload Parquet*), é possível enviar um arquivo no formato Parquet, para verificar seu conteúdo de forma facilitada.

Figura 11 – Upload parquet



Fonte: Autor (2022).

1. Menu de acesso a tela para realizar o *upload* e leitura de um arquivo Parquet;
2. Botão para realizar a seleção e envio do arquivo Parquet. É possível também arrastar um arquivo para esta área;

3. Informação com o nome do arquivo selecionado;
4. Tabela gerada, mostrando os registros do arquivo enviado.

6. RESULTADOS

A análise dos resultados deste trabalho, foi feita a partir da comparação de um processo de ingestão, realizado da maneira tradicional e com o uso do protótipo criado.

O processo original de ingestão, que tem como origem uma base de dados MySQL de produção, demandou cerca de oito horas de trabalho para sua criação e execução, incluindo toda a complexidade inerente a configuração de ambiente e fluxos de integração do código com a origem, a configuração da orquestração do fluxo, o agendamento e demais etapas da ingestão. A mesma configuração no protótipo levou dois minutos, visto que todo o código que precisava ser desenvolvido em um processo padrão e a orquestração envolvida, já estavam sendo executados pela ferramenta de forma dinâmica, reconhecendo os tipos de dados das tabelas que foram selecionadas.

O protótipo conseguiu entregar no repositório de destino os arquivos Parquet de forma confiável, assim como ocorre nos processos realizados de forma atual. Com esses testes, o acesso ao protótipo foi entregue aos outros oito membros do time de engenharia de dados, para que avaliassem sua utilização. Os testes realizados foram de experiência do usuário, adição de uma nova ingestão e análise da entrega final dos dados no arquivo Parquet. Foram utilizadas também as ferramentas de listagem e *upload* de Parquet.

Os retornos obtidos pelos membros da equipe foram positivos em relação a facilidade do uso e o ganho de tempo ao comparar a adição de novas ingestões com o ambiente atual da empresa. Também entenderam que existe economia devido a redução de tempo que os profissionais tiveram para realizar a demanda, podendo focar em outros projetos, maximizando a qualidade nas entregas e melhorando o tempo de atendimento da lista de demandas a serem realizadas. Um ponto negativo citado, foi a respeito do aumento do tempo de processamento e entrega do arquivo final. Existindo a ingestão configurada em ambos cenários, a execução de tal processo resultou em um tempo superior pelo protótipo, comparado ao fluxo atual.

7. CONCLUSÃO

O protótipo desenvolvido obteve resultados satisfatórios, pois atendeu aos requisitos

levantados e se mostrou uma alternativa viável aos problemas do fluxo de ingestão de dados, identificados inicialmente neste trabalho. O uso do protótipo facilitou a criação de novas ingestões, tanto sob demanda como agendadas, diminuindo o tempo usado para desenvolver o processo atual no ambiente profissional. O time de engenharia compreendeu que a ferramenta mostra atender muito bem na facilidade de uso e redução de custos, visto que o tempo para a criação de novas ingestões é muito menor. Com o tempo extra disponibilizado pela plataforma é possível minimizar a quantidade de profissionais no time.

Como projetos futuros para o DIP, conforme compreendido nos resultados, será necessário analisar a possibilidade de melhoria no código quando usado o Pandas, tentando melhorar o tempo de processamento. Como segunda opção ao processamento, pode ser estudada outra biblioteca ou *software* para o manuseio dos dados. Por fim, o protótipo contempla somente como origem bases MySQL, sendo necessário criar mais conectores, abrangendo uma quantidade maior de origens.

REFERÊNCIAS

AMAZON. **AWS Glue.** Amazon. Disponível em: <https://aws.amazon.com/pt/glue>. Acesso em: 01 Outubro 2022.

APACHE. **Apache Parquet.** Apache. Disponível em: <https://parquet.apache.org/>. Acesso em: 25 Agosto 2022.

ARAKAKI A. C. S.; ARAKAKI F. A. **Dados e metadados: conceitos e relações.** Revista IBICT. v.48, Julho 2021. p. 34-45, set/dez 2020.

ARMOOGUM S., LI X., **Deep Learning and Parallel Computing Environment for Bioengineering Systems.** Arun Kumar Sangaiah. 2019, p. 17-36, ISBN 9780128167182.

DAVENPORT, T. H.; PATIL, D. **Is Data Scientist Still the Sexiest Job of the 21st Century?** Harvard Business Review. Julho 2014. Disponível em: <https://hbr.org/2022/07/is-data-scientist-still-the-sexiest-job-of-the-21st-century>. Acesso em: 23 Agosto 2022.

ESCOLA SUPERIOR DE REDES. **Dados estruturados, não-estruturados e semiestruturados: diferenças e similaridades.** Abril 2022. Disponível em: <https://esr.rnp.br/ciencia-de-dados/dados-estruturados-nao-estruturados-e-semiestruturados/>. Acesso em: 22 Agosto 2022.

FARIA V., et al. **Implantação do Kanban na Linha de Montagem de Sistema e Equipamentos Hidráulicos e Eletromecânicos.** São Paulo: UNESP, Novembro 2006.

GARTNER. **Oracle GoldenGata vs SAP Data Services vs StreamSets DataOps Platform. Gartner Peer Insights.** Disponível em: <https://www.gartner.com/reviews/market/data-integration-tools/compare/product/oracle-goldengate-vs-sap-data-services-vs-streamsets-

dataops-platform>. Acesso em: 01 Outubro 2022.

GITHUB. **Features.** Github. Disponível em: <https://github.com/features>. Acesso em: 25 Agosto 2022.

ISHWARAPPA J., ANURADHA. **A Brief Introduction on Big Data 5Vs Characteristics and Hadoop Technology.** Procedia Computer Science. v. 48, 2015, p. 319-324, ISSN 1877-0509.

KAVA P. **AWS serverless data analytics – arquitetura de referência,** Novembro 2022. Disponível em: <https://aws.amazon.com/pt/blogs/aws-brasil/aws-serverless-data-analytics-arquitetura-de-referencia/>. Acesso em: 8 Dezembro 2022.

KIMBALL R.; CASERTA J. **The data warehouse ETL toolkit: practical techniques for extracting, cleaning, conforming, and delivering data**. 1. ed. 2004, p. 3-25.

KONDO, N. **More and more organisations are collecting and storing vast amounts of data. Yet for all the excitement generated by the potential of this data to transform business models - turning it directly into cold, hard cash can prove difficult,** Janeiro 2016. Disponível em: <https://impact.economist.com/perspectives/technology-innovation/business-data-0>. Acesso em: 23 Agosto 2022.

MARIA, E. S. S. **Raspagem de dados (data scraping): A proteção de bases de dados públicas pela lgpd.** Instituto Avançado de Proteção de Dados. Abril 2021. Disponível em: <https://iapd.org.br/iapd-org-br-raspagem-data-scraping-protecao-dados-lgpd/>. Acesso em: 20 Agosto 2022.

MASSENA, C. **Quem é o analista de dados? Associação Brasileira de Ciências de Dados.** Disponível em: <https://abracd.org/quem-e-o-analista-de-dados/>. Acesso em: 24 Agosto 2022.

MICROSOFT. **Azure Data Factory**. Microsoft. Disponível em: <https://azure.microsoft.com/en-us/products/data-factory/>. Acesso em: 01 Outubro 2022.

MYSQL. **1.2.1 What is MySQL. MySQL.** Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>. Acesso em: 25 Agosto 2022.

NATALIA M., ALEXANDER T. **Big Data, Fast Data and Data Lake Concepts.** Procedia Computer Science. v. 88. 2016. P. 300-305, ISSN 1877-0509.

OLIVEIRA, R. **A escassez de talentos de inteligência e dados.** Associação Brasileira de Ciências de Dados. Disponível em: <https://abracd.org/a-escassez-de-talentos-de-inteligencia-e-dados/>. Acesso em: 02 Outubro 2022.

OLIVEIRA, R. **Big Data ou Data Science? Entenda o que faz cada profissional.** Associação Brasileira de Ciências de Dados. Disponível em: <https://abracd.org/big-data-ou-data-science-entenda-o-que-faz-cada-profissional/>. Acesso em: 24 Agosto 2022.

ORACLE. **GoldenGate**. Oracle. Disponível em: <https://www.oracle.com/br/integration/goldengate/>. Acesso em: 01 Outubro 2022.

PANDAS. **Pandas documentation**. Pandas. Setembro 2019. v. 1. Disponível em: <https://pandas.pydata.org/docs/> Acesso em: 25 Agosto 2022.

PYTHON. **Beginner's Guide to Python**. Setembro 2022. Disponível em: <https://wiki.python.org/moin/BeginnersGuide>. Acesso em: 25 Agosto 2022.

QIN, C.; EICHELBERGER H.; SCHMID,K. **Enactment of adaptation in data stream processing with latency implications -** A systematic literature review. Julho 2019.

RED HAT. **IDE - Ambiente de Desenvolvimento Integrado.** Red Hat. Disponível em: <https://www.redhat.com/pt-br/topics/middleware/what-is-ide>. Acesso em: 22 Agosto 2022.

REVELO. **Carreiras em dados: profissionais mais procurados na área.** Maio 2021. Disponível em: <https://blog.revelo.com.br/carreiras-em-dados/>. Acesso em: 24 Agosto 2022.

RODRIGUES, C.; BLATTMANN, U. **Gestão da informação e a importância do uso de fontes de informação para geração de conhecimento**, Setembro 2014.

SANGAIAH, A. K. **Deep Learning and Parallel Computing Environment for Bioengineering Systems.** 1. ed. Academic Press, 2019.

SAP. **SAP Data Service.** SAP. Disponível em: <https://www.sap.com/products/technology-platform/data-services.html>. Acesso em: 01 Outubro 2022.

SSA GROUP TEAM. **Data professionals: An overview of specialisations and responsibilities.** SSA Group. Novembro 2021. Disponível em: <https://www.ssa.group/blog/data-professionals-an-overview/>. Acesso em: 24 Agosto 2022.

STREAMLIT. **Welcome do Streamlit.** Streamlit. Disponível em: <https://github.com/streamlit/streamlit>. Acesso em: 25 Agosto 2022.

STREAMSETS. **StreamSets Documentation.** Streamsets. Disponível em: <https://docs.streamsets.com/>. Acesso em 01 Outubro 2022.

XIMENES, L. **Python e a Ciência de Dados.** Associação Brasileira de Ciências de Dados. Disponível em: <https://abracd.org/python-e-a-ciencia-de-dados/>. Acesso em: 02 Outubro 2022.

W3 SCHOOLS. **CSS Introduction.** Disponível em: <https://www.w3schools.com/css/css_intro.asp>. Acesso em: 25 Agosto 2022.