

Análise e Otimização de *Lead Time* em Processos Ágeis: Estudo de Caso em Empresa de Marketplace

Victor Hugo Oliveira de Amorim¹, Pablo Freire Matos²

¹Discente Superior em BSI, ²Docente de Informática

Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA)
Av. Sérgio Vieira de Mello, 3.150, Zabelê – 45.078-900 – Vitória da Conquista – BA – Brasil

{victor.hugo.amorim.7, pablofmatos}@gmail.com

Abstract. *With the increasing adoption of remote work, the role of flow managers in companies in the technology sector acquires increasingly central relevance. Agile systems, such as Kanban, Scrum and XP, play a fundamental role in leading sectors and improving the performance of their teams, maintaining a delicate balance between requirements, achievements and the well-being of team members. This work proposes to carry out a case study in a marketplace company, consolidated in the online market and which has more than two thousand employees. The objective is to allow personalized analysis of the lead time of data generated by management tools. To this end, the main frameworks that guarantee robust performance and the necessary optimization to deal with data volume consistent with the big data scenario will be explored. The use of the lead time metric not only made it possible to reverse a scenario of customer dissatisfaction due to delays in deliveries, but also facilitated mapping the ideal period for data extraction, helping to avoid excessive costs.*

Resumo. *Com a crescente adoção do trabalho remoto, o papel dos gestores de fluxo em empresas do setor tecnológico adquire uma relevância cada vez mais central. Metodologias ágeis, como Kanban, Scrum e XP, exercem uma função importante ao coordenar atividades dos setores e aprimorar o rendimento de seus times, sustentando um equilíbrio delicado entre requisitos, realizações e o bem-estar dos membros da equipe. Este trabalho se propõe a realizar um estudo de caso em uma empresa de marketplace, consolidada no mercado online e possui mais de dois mil funcionários. O intuito é viabilizar uma análise de lead time personalizada dos dados gerados por ferramentas de gestão. Para esse fim, serão explorados os principais frameworks que asseguram um desempenho robusto e a otimização necessária para lidar com volume de dados condizente com o cenário de big data. A utilização da métrica de lead time não apenas permitiu reverter um cenário de insatisfação por parte dos clientes devido a atrasos nas entregas, mas também facilitou o mapeamento do período ideal para a extração de dados, contribuindo para evitar custos excessivos.*

1 Introdução

Com o advento da transformação digital e a rápida evolução das empresas de tecnologia, a manutenção e gestão de equipes de trabalho tornaram-se desafios significativos, visando alcançar o máximo desempenho possível. Segundo Santos (2023), “a metodologia ágil tem sido amplamente adotada por empresas de diversos setores com resultados positivos”, revolucionando a forma como os negócios são conduzidos. O foco na entrega contínua, iterações ágeis e colaboração intensificada têm gerado resultados impressionantes, o que

impulsiona a eficiência e a capacidade de adaptação das organizações. No entanto, à medida que as empresas se esforçam para otimizar seus processos e maximizar a produtividade, surgem desafios significativos relacionados à gestão e ao aproveitamento dos dados gerados nesse ambiente dinâmico.

É nesse contexto que a engenharia de dados emerge como um pilar para capacitar as organizações a enfrentarem esses desafios de maneira eficaz. A importância da engenharia de dados para *big data* reside no fato de que os dados são considerados a matéria-prima do século 21. Eles podem fornecer *insights* que impulsionam melhorias [Marques 2023]. As análises resultantes da construção do fluxo culminam na tão esperada recompensa. Para avaliar a eficiência do fluxo de entregas dentro de prazos previamente definidos, a métrica de *lead time* demonstra-se eficaz. A vantagem do *lead time* é fornecer informações essenciais para quem faz o gerenciamento de tarefas. Desta forma, é possível identificar gargalo no fluxo de trabalho, entender a capacidade da equipe e melhorar a previsibilidade das entregas, gerando prazos mais precisos [Matos 2022].

Quando o assunto é informações baseadas em dados, alguns obstáculos aparecem no caminho. Independentemente de onde os dados serão extraídos, as principais preocupações serão a tecnologia que será usada para realizar a engenharia, onde os dados serão armazenados e qual a melhor ferramenta para visualizar os resultados das análises. Além disso, cada ramo de negócio contém um tipo específico de problema. A métrica de *lead time* facilita a gestão das entregas empresariais, e será necessário requisitar extrações e análises personalizadas para cada operação. Portanto, empresas que coletam e tratam seus próprios dados vão possuir o poder de modela-los da maneira que melhor lhe atender.

Nesse sentido, este trabalho explora a intersecção entre as metodologias ágeis e a engenharia de dados. Através de um estudo de caso, será examinado como a aplicação da engenharia de dados pode superar os desafios que as empresas enfrentam na gestão e aproveitamento de dados gerados em ambientes ágeis. De modo mais específico, será analisado o *lead time* da extração de dados da *Application Programming Interface* (API) do Jira, a utilização do Python, orquestração do Apache *Airflow* e a integração com o *Amazon Web Services S3* para armazenamento seguro, o que culminará na visualização eficaz dos resultados por meio da ferramenta de análise de dados, Tableau. Ao explorar essa jornada, a engenharia de dados pode se tornar um diferencial estratégico na transformação ágil das empresas de tecnologia.

O artigo organiza-se da seguinte forma: Na Seção 2, serão tratados os fundamentos que baseiam o trabalho; na Seção 3, haverá uma discussão sobre os trabalhos correlatos; na Seção 4, será apresentada a proposta do trabalho. Por fim, na Seção 5, serão destacadas as considerações finais.

2 Conceitos Teóricos

Nesta seção serão tratados os conceitos que fundamentam o trabalho.

2.1 Metodologias Ágeis

As metodologias ágeis são essenciais para empresas que buscam dinamizar e tornar seus processos mais interativos. A metodologia ágil é um conjunto de técnicas e práticas voltadas para acelerar, aprimorar a eficiência e proporcionar flexibilidade à gestão de projetos [Santos 2023]. Inicialmente concebidas no âmbito de Tecnologia da Informação com o propósito de agilizar o desenvolvimento de *softwares*, essas metodologias

atualmente transcendem esse contexto, sendo aplicáveis a negócios de diversos setores e tamanhos. Ao proporcionar uma condução mais simplificada e eficiente dos processos, essas práticas ágeis contribuem para a otimização da produtividade. De acordo com Digital.ai (2022), estima-se que 94% das empresas do setor de *software* relatam pelo menos alguma experiência com desenvolvimento ágil (Figura 1).

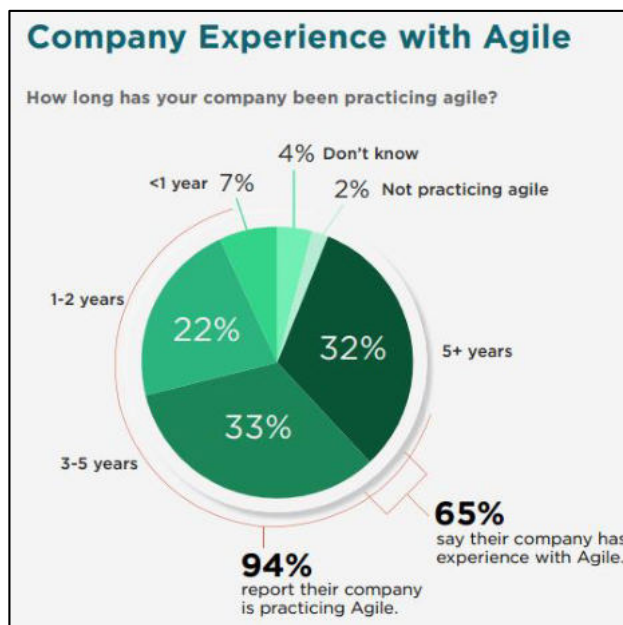


Figura 1. Análise da implementação de metodologias ágeis em 2022. Fonte: Digital.ai (2022)

Dentre os principais benefícios advindos da aplicação das metodologias ágeis na gestão empresarial, destacam-se a melhoria da interação entre equipes, a redução do tempo necessário para a conclusão de projetos e o aumento das oportunidades de correção de potenciais equívocos ao longo do processo [Redação Vivo 2021].

2.2 Engenharia de Dados

A engenharia de dados no contexto de *big data* desempenha um papel crucial na gestão, análise e eficiente utilização dos imensos volumes de dados atualmente disponíveis. Essa prática capacita as organizações a extrair *insights* valiosos, tomar decisões mais embasadas e manter uma posição competitiva em um cenário cada vez mais voltado para dados. Ao empregar técnicas de processamento e análise de dados, a engenharia de dados proporciona benefícios substanciais em diversos setores, que vão desde o financeiro até a indústria, passando por varejo, tecnologia, saúde e outros. Sua importância no cenário atual é inquestionável, e sua relevância só tende a aumentar à medida que a quantidade e complexidade dos dados continuam a expandir [Marques 2023].

2.3 Orquestração

A orquestração de tarefas, exemplificada pelo Apache *Airflow*, engloba a habilidade de coordenar e supervisionar a execução de diversas atividades interdependentes em um ambiente computacional. Essa abordagem viabiliza a definição, a programação e o monitoramento de fluxo de trabalho complexo, no qual tarefas individuais são realizadas de forma sequencial ou paralela, guiadas por condições específicas ou cronogramas predefinidos em Grafo Acíclico Direcionado (DAG, sigla em inglês). O DAG representa

a estrutura principal que delinea o fluxo de dados, comparável a um *pipeline* de dados. Em ambientes empresariais, é comum ter vários DAGs, cada um desempenhando uma função específica e, frequentemente, operando de maneira independente, como exemplificado por "*Pipeline* de Dados A" e "*Pipeline* de Dados B" [Rox Partner 2021].

3 Trabalhos Correlatos

Os trabalhos correlatos abordam aplicações de engenharia de dados em diferentes cenários, a saber: a importância da infraestrutura para garantir qualidade aos dados e a possibilidade de extrações analíticas após tratamentos.

Alves (2023) apresenta benefícios decorrentes da criação e implementação de um *Data Warehouse*. Destaca a importância de tratar e cruzar grandes volumes de dados para a construção de gráficos e indicadores, o que visa aumentar a produtividade e facilitar os níveis de gerenciamento. O autor enfatiza que a efetividade nas áreas gerenciais é ampliada ao eliminar incertezas e economizar tempo na tomada de decisão, tanto presente quanto futura.

Maximiano (2023) destaca a relevância de uma estrutura organizacional eficaz na área de saúde do estado de São Paulo, com enfoque na geração de indicadores a partir dos dados do Cadastro Nacional de Estabelecimentos de Saúde (CNES). O autor ressalta a necessidade de uma equipe técnica dedicada a gerenciar o fluxo de informações, sublinhando os ganhos em eficiência decorrentes de uma infraestrutura de processamento de dados robusta. Os resultados alcançados após a implementação do processo de *Extract, Transform and Load* (ETL) evidenciam a distribuição de profissionais de saúde e estabelecimentos de saúde registrados no CNES. Por meio da análise desses dados, o trabalho examina indicadores quantitativos e qualitativos, oferecendo *insights* sobre a realidade da saúde em diferentes regiões do Estado de São Paulo. A construção de um painel interativo permite aos usuários explorar diferentes cenários, o que fornece uma plataforma para avaliar as necessidades específicas dos municípios paulistas.

Leal (2021) aborda a análise de ferramentas cruciais no contexto da engenharia de dados. O autor destaca as capacidades distintas do Pentaho e do *Airflow* com Python, ressaltando que ambas desempenham efetivamente as funções de extração, orquestração e armazenamento que se propõem a realizar, e até podem executar diferentes etapas do desenvolvimento, contribuindo para resultados mais eficientes. À medida que o conhecimento e a experiência aumentam, o desenvolvedor pode então experimentar o *Airflow*, ganhando maior controle sobre a manipulação dos dados. Isso evidencia a complementaridade dessas ferramentas e como cada uma pode ser aproveitada conforme o nível de habilidade e as necessidades do usuário.

Amaral (2020) aborda a criação de um algoritmo capaz de receber dados coletados de máquinas industriais e armazenados na nuvem. O algoritmo deve realizar transformações e contextualizações necessárias, estruturando esses dados em uma tabela adequada para análise por meio de visualizações. Esse processo viabiliza a obtenção de KPIs (*Key Performance Indicators*), além de possibilitar a criação de representações visuais que permitem ao Gerente de Planta acompanhar a produção.

Conforme evidenciado na Tabela 1, a maioria dos processos em *big data* é executada por meio de estruturas hospedadas na nuvem, alinhando-se ao propósito deste trabalho. Adicionalmente, os processos de ETL formam a base para a implementação de análises, as quais podem ser apresentadas em ferramentas de *Business Intelligence*,

escolhidas livremente de acordo com as necessidades da empresa. Enquanto alguns trabalhos utilizaram o Power BI, o presente trabalho utiliza a ferramenta Tableau.

Tabela 1. Comparação entre trabalhos correlatos.

Autor(es)	Instituição	Foco	Linguagem	Armazenagem	Banco de dados	BI
Trabalho Proposto (2023)	IFBA	Processos de coleta, transferência e consolidação de dados em ferramentas de metodologias ágeis, construindo métricas com ênfase em Lead Time .	Python	AWS	PostgreSQL	Tableau
Alves (2023)	UFES	Desenvolver uma arquitetura de DW que automatize o processo de coleta, transferência e consolidação de dados		Servidor Próprio		Power BI
Maximiano (2023)	UNESP	Desenvolver um pipeline de dados responsável por realizar a coleta e o tratamento do conjunto de informações, possibilitando por meio de dashboards, a análise de dados da área da saúde para o estado de São Paulo		AWS	Redash	
Haefner (2022)	UFES	Uma arquitetura de processamento de dados na nuvem que seja facilmente adaptável a futuras implementações de dados de estações meteorológicas		Servidor Próprio	MariaDB	N/A
Leal (2021)	IFSP	Desenvolver e avaliar diferentes ferramentas open source para a criação de pipeline de Dados para big data , via processo de ELT, com enfoque em ciências humanas .		AWS	MongoDB	Tableau
Amaral (2020)	USP	Desenvolvimento de um algoritmo para transformação de dados industriais com linguagem de programação Python integrado no ambiente de computação em nuvem da Google				

4 Proposta do Trabalho

O presente trabalho fundamenta-se na abordagem de pesquisa do tipo estudo de caso, uma metodologia científica que realiza uma análise aprofundada de uma empresa real de *marketplace*. A pesquisa utiliza uma amostra de dados referente a um período de três meses¹, fornecida pela própria empresa, para conduzir uma investigação fundamentada. As etapas de desenvolvimento deste trabalho serão segmentadas em quatro pontos: Extração, transformação, carregamento e visualização (Figura 2). Cada uma dessas etapas desempenha um papel fundamental na garantia da confiabilidade e na entrega de resultados de alta qualidade ao público interessado.

Após a conclusão da etapa de extração, entra em cena um conjunto de ferramentas de tratamento de dados, assegurando a qualidade e os formatos necessários para os dados. Em seguida, a carga na nuvem será empregada para a criação do *Data Warehouse*, culminando na conexão final com a ferramenta de BI. Esse processo integrado e sequencial visa otimizar a eficiência e a eficácia de cada fase.

¹ Amostra de dados disponível em:

https://docs.google.com/spreadsheets/d/1Woic0LSnIaf_Hn99SHeLJFqZadl_i8K6-_pPVC-CBE/

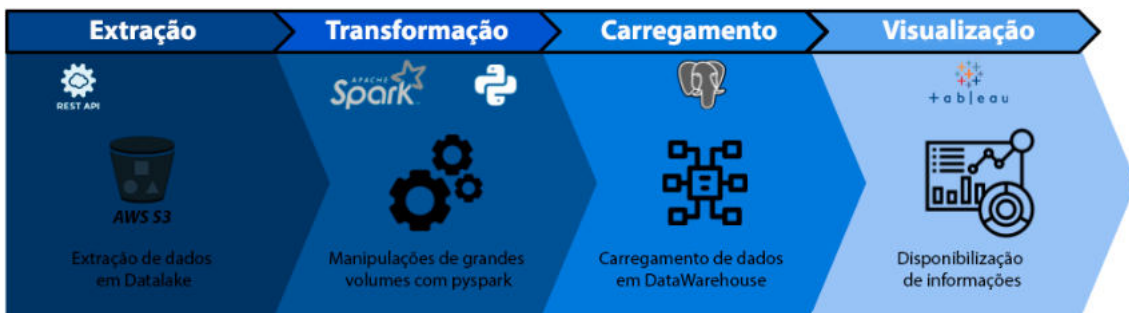


Figura 2. Fluxo de tarefas de desenvolvimento.

4.1 Fonte de Dados

O Jira Software [L3 Software 2023] é uma ferramenta voltada para o trabalho ágil que oferece funcionalidades cruciais para o gerenciamento de projetos ágeis. Neste artigo, os dados resultantes das instruções dos gestores aos desenvolvedores na plataforma Jira serão utilizados como fundamento para as análises pós extração. Todas as atividades programadas serão coletadas, juntamente com seus parâmetros, tais como data de criação, data de conclusão e área responsável. Na Figura 3 é apresentada uma página de exemplo com divisões de tarefas junto a seus status de progresso no Jira. Ele permite a criação e o acompanhamento de tarefas e histórias de usuário, o que facilita o planejamento, a priorização e a distribuição de trabalho em equipes. Além disso, o Jira Software oferece recursos avançados de rastreamento de progresso, como *burndown charts* e *kanban boards*, para monitorar o andamento do projeto. A integração com ferramentas de desenvolvimento, como o GitHub, também torna mais eficiente o fluxo de trabalho de desenvolvimento de *software*. Com sua capacidade de personalização e relatórios detalhados, o Jira Software é uma ferramenta que oferece suporte para equipes ágeis na busca pela entrega de projetos de alta qualidade dentro dos prazos.

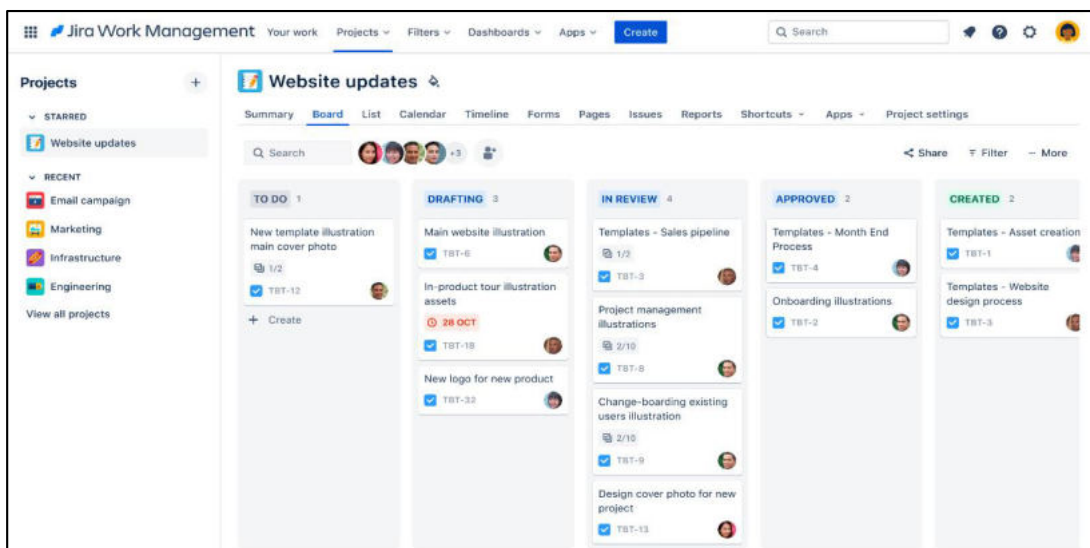


Figura 3. Quadro do Jira com divisão de tarefas. Fonte: Atlassian (2023).

Soma-se a tudo isso o fácil acesso que os envolvidos no processo possuem aos relatórios, que oferecem *insights* em tempo real e mostram o desempenho da equipe durante a realização da *sprint*. Essa funcionalidade permite ajustes pontuais para readequação do projeto.

4.2 Extração de Dados

Na Figura 4 é apresentado um exemplo de código em Python para conexão ao Jira Software. Para estabelecer essa conexão, é fundamental o uso da biblioteca *"requests"* e a inclusão de um *token* de autorização, o qual concede acesso aos recursos necessários. Ao utilizar esses recursos, foi realizada uma extração abrangente das bases de dados disponíveis no Jira. Contudo, é importante ressaltar que essa extração geral não incluiu um tratamento específico de tipagem dos dados. Desta forma, os dados foram obtidos em sua forma bruta, sem conversão para tipos de dados específicos. Essa abordagem inicial de extração permite obter uma visão ampla dos dados, que posteriormente pode ser refinada com tratamento de dados para atender às necessidades específicas de análise ou integração com outras ferramentas.



```
main.py
1 import requests
2 import json
3 import io
4 url="https://your-company.atlassian.net/rest/api/3/search"
5 headers={
6     "Accept": "application/json",
7     "Content-Type": "application/json"
8 }
9
10 query = {
11     'jql': 'project = your-project'
12 }
13
14 response=requests.get(url,headers=headers,params=query,auth=("user@email-example.com", "<your-token>"))
15 data=response.json()
16 lista_issues=data["issues"]
```

Figura 4. Exemplo de requisição ao Jira com Python.

4.3 Tratamento e Armazenamento

Para gerenciar a grande quantidade de dados resultantes da extração bruta da API do Jira, será utilizado o Apache Spark, o qual proporciona suporte à memória, visando otimizar o desempenho de aplicativos voltados para a análise de *big data*. Uma característica destacada é a sua flexibilidade, uma vez que oferece suporte a diversas linguagens de programação, incluindo Scala, Java, Python e SQL [Silva 2023]. A escolha do Spark se deve principalmente à sua eficácia no manuseio de volume massivo de dados, comparado a biblioteca pandas utilizada no Python. Desta forma, tem-se bastante ganho em desempenho, como mostrado na Figura 5. Durante a etapa de extração, os dados brutos foram obtidos no formato JSON, apresentando estruturas agregadas que dificultavam a análise, tornando necessária uma expansão prévia para facilitar a compreensão e utilização desses dados.

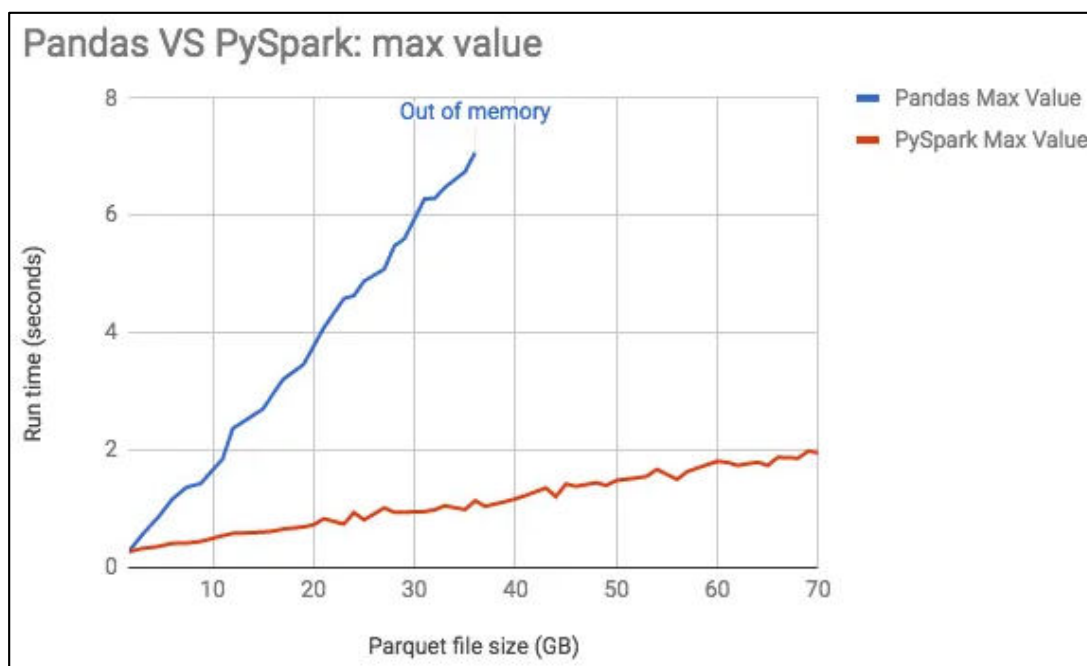


Figura 5. Comparação entre PySpark e o Pandas.

Os dados brutos extraídos do Jira, após o tratamento inicial com o Spark, foram armazenados em um Data Lake hospedado em um Bucket S3 da AWS (Amazon Web Services). Essa escolha proporciona escalabilidade, segurança e durabilidade aos dados. É importante destacar que foram mantidas tanto as versões brutas quanto as versões tratadas dos dados no S3. Isso possibilita flexibilidade para análises futuras e preserva a integridade dos dados originais. Essa abordagem é essencial para garantir que os dados estejam disponíveis para consultas e análises de forma histórica e refinada, sem comprometer a integridade dos dados brutos originais.

4.4 Orquestração e Agendamento

A partir do momento em que o projeto tem maturidade para possuir agendamento de extrações e cargas, deve-se ter claro em mente o tipo de problema de negócio que busque responder. A fase de processamento dos dados pode se ramificar por duas vertentes, sendo elas processamento em *batch* ou *streaming*.

O processamento em *batch* é usado com frequência quando se lida com grande volume de dados. Ele também, por definição, exige que todos os dados necessários para o lote sejam carregados em algum tipo de armazenamento, um banco de dados ou sistema de arquivos para serem processados. Já para o processamento em *streaming*, o fluxo de carga é instantâneo, permitindo análises em tempo real, no entanto, isso pode ser mais custoso por ter como requisito um servidor trabalhando sem interrupções.

O projeto deste trabalho funciona com o modelo de processamento em *batch*, pois apesar de coletar dados de uma plataforma orgânica como o Jira Software, a inserção de novos dados em sua maioria são em horários comerciais, resultado obtido após análise comparando a média de inserções de atividades no Jira entre 2021 à 2022 (Figura 6).

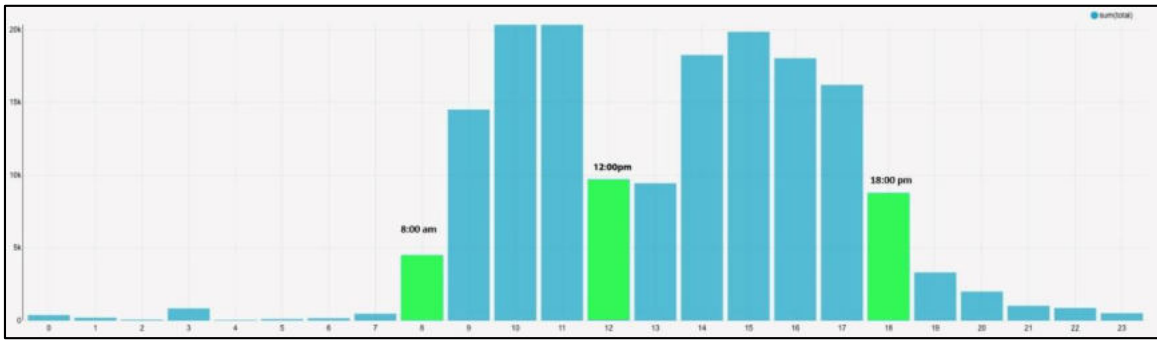


Figura 6. Gráfico de barras com distribuição de dados entre as 24 horas do dia.

A ferramenta *Apache Airflow* realiza o agendamento das extrações em modo *batch* para o seguimento de fluxo do projeto a partir do Grafo Acíclico Direcionado (DAG) (Figura 7). Após análise, janelas com intervalos em horários personalizados foram designadas para armazenamento de cargas atualizadas, além de atender as demandas de negócio da operação. A escolha da ferramenta teve como um dos pontos a facilidade de execução dessas tarefas de forma eficiente e confiável, sem a necessidade de escrever código ou gerenciar *scripts* manualmente. O *Airflow* oferece recursos como planejamento, repetição e monitoramento para facilitar a execução de extrações de dados em *batch*.

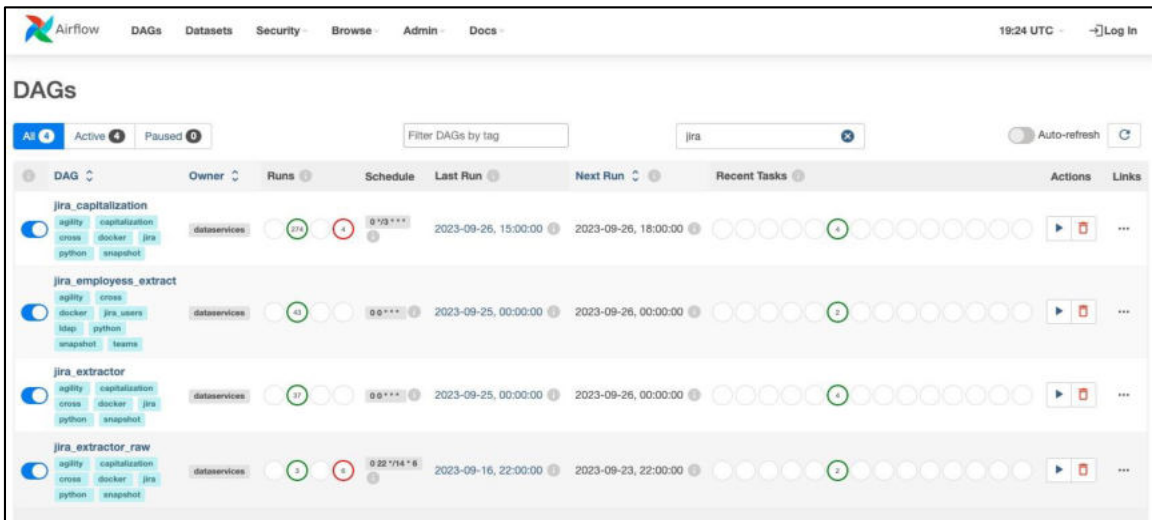


Figura 7. DAGs com tratamento dos dados provenientes do Jira sendo executadas no Apache Airflow.

4.5 Data Warehouse

O *Data Warehouse* (DW) atua como um repositório centralizado e estruturado, podendo ser consolidado e armazenado de maneira organizada. Isso é crucial para garantir a integridade e a consistência dos dados, independentemente de sua origem. Além disso, o DW mantém o histórico de dados ao longo do tempo, permitindo análises históricas, detecção de tendências e conformidade regulatória.

Um fator crucial é que um DW permite a modelagem de dados específica para análise e ligações entre tabelas a partir de chaves primárias, por exemplo "*issue_key*". Isso inclui a criação de esquemas de dados otimizados para consultas analíticas, agregações e relatórios. A estruturação adequada dos dados no *Data Warehouse* facilita

a execução de consultas complexas, o que acelera o processo de análise. Na Figura 8 encontra-se o esquema do DW com as respectivas tabelas: *issues_status*, *issues_metrics* e *organization_map*.

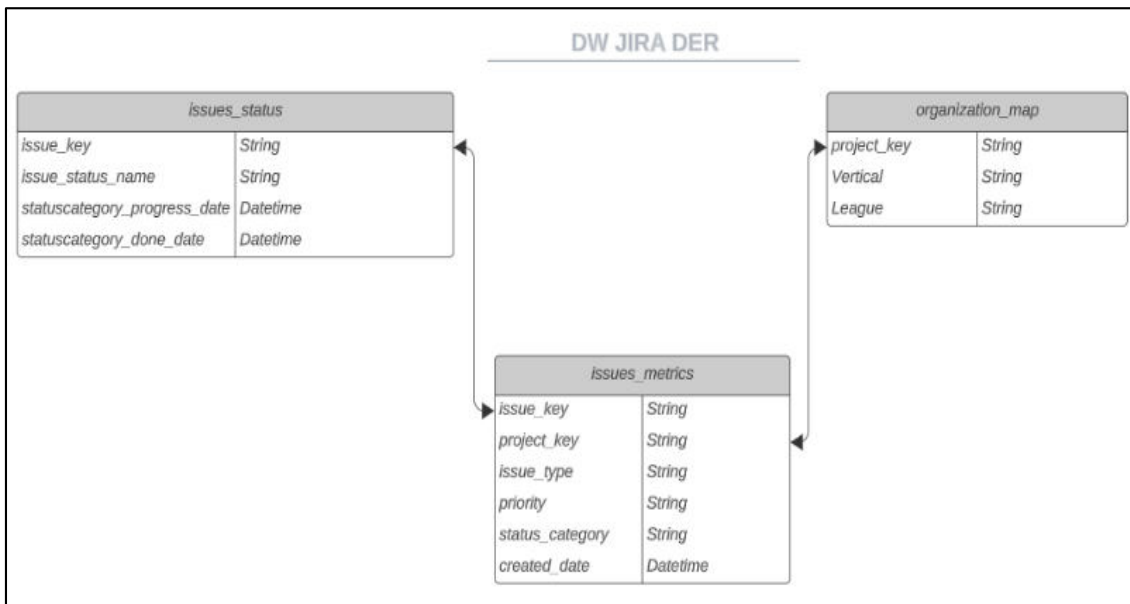


Figura 8. Tabelas criadas formam a estrutura do DW.

A ferramenta de gerenciamento de banco de dados utilizada foi o PostgreSQL, pois oferece um desempenho robusto para consultas SQL. Quando integrado a um *DW*, ele ajuda a melhorar a eficiência das análises, garantindo que as consultas sejam processadas de forma rápida e eficaz. Além disso, o PostgreSQL pode ser escalado para lidar com volume crescente de dados e demandas de análise. A integração torna mais fácil para os analistas executarem consultas analíticas complexas. A capacidade de usar SQL para recuperar dados do *Data Warehouse* por meio do PostgreSQL permite que os analistas extraiam *insights* valiosos, identifiquem padrões e criem relatórios detalhados para a tomada de decisões.

Por fim, a base construída para a leitura de dados se torna compatível com uma variedade de ferramentas de visualização de dados, como Tableau e Power BI. Isso permite que os dados sejam transformados em gráficos, painéis interativos e relatórios visuais que tornam a análise de dados mais acessível e compreensível para toda a organização.

4.6 Visualização dos Dados

Para tornar os dados acessíveis a gestores e interessados, é imperativo empregar ferramentas de *Business Intelligence* (BI). Isso elimina a necessidade de que aqueles sem conhecimento técnico aprofundado tenham que interagir diretamente com linguagens de programação, como o SQL, garantindo uma acessibilidade mais ampla e simplificada aos dados.

O Tableau oferece uma *interface* amigável e intuitiva que permite aos usuários criarem e visualizarem gráficos em *dashboards*, mesmo sem conhecimento avançado em programação. Isso torna a análise de dados acessível a uma variedade de equipes e departamentos dentro da organização. Além disso, é altamente versátil em termos de

integração com várias fontes de dados, incluindo bancos de dados, planilhas, serviços web e muito mais. Isso significa que as organizações podem reunir dados de várias fontes em um único local para análise, incluindo os dados armazenados no *Data Warehouse* construído a partir do PostgreSQL. Na Figura 9 é mostrado o *dashboard* do projeto criado a partir da coleta de dados.²

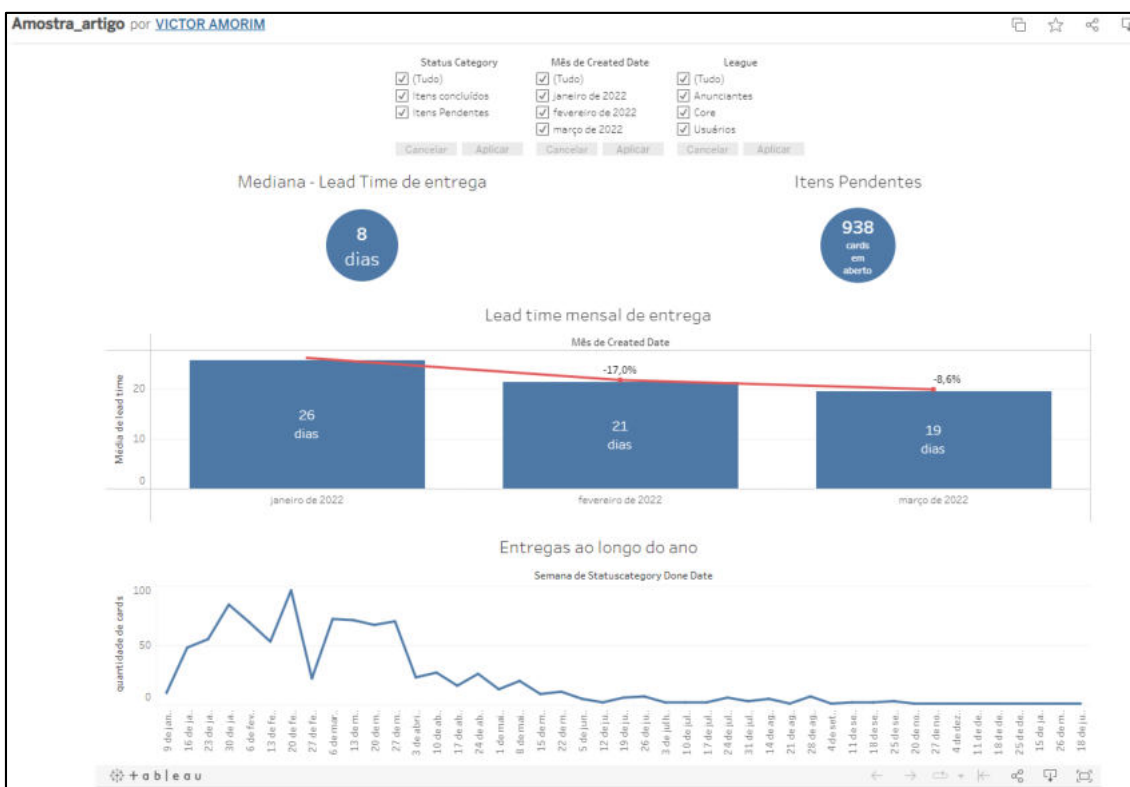


Figura 9. Gráfico de lead time criado na plataforma Tableau.

A partir da coleta de dados, foi possível evidenciar problemas voltados ao fluxo de trabalho interno, a saber: equipes de trabalho sobrecarregadas com tarefas desnecessárias e alto tempo de entrega, além de projetos idênticos que ocupavam mais de uma equipe e que entregavam os mesmos resultados. A capacidade de gestão aumentou gradativamente a cada análise realizada.

5 Considerações Finais

O objetivo do desenvolvimento centrado na engenharia de dados e na geração de métricas, a partir dos dados coletados, é oferecer suporte aos níveis gerenciais. A utilização de ferramenta de coleta de dados para o gerenciamento de demandas internas, prazos e distribuição de tarefas é fundamental para manter um ambiente de trabalho organizado e proporcionar uma visão clara do fluxo de trabalho.

A implementação deste projeto possibilitou uma leitura da métrica de *lead time*, permitindo também comparações retrospectivas com base no histórico da empresa. Além disso, o controle, o armazenamento e a propriedade dos dados tornam-se vantagens

² Visualização de dados disponível em:

https://public.tableau.com/app/profile/victor.amorim/viz/Amostra_artigo_16997967633140/MtricasdeLeadTime

adicionais para a empresa. Pois com uma base de dados relacional estruturada e com ferramentas de análises bem definidas, os pontos positivos serão refletidos em uma boa gestão, colaboradores e clientes satisfeitos com seus prazos de entrega. Já em relação ao ponto negativo, a empresa terá que se preocupar com a mão de obra qualificada para gerir todos os processos, tanto tecnológico quanto de liderança.

Para trabalhos futuros, pretende-se implementar as métricas de cálculo de diagrama de fluxo cumulativo, atuando sobre a carga de fluxo em que cada equipe comporta, e o cálculo de custo por atividade, mapeando o valor agregado em que cada projeto consome. Desta forma, irá beneficiar ainda mais os indicadores de desempenho da empresa.

Referências

Alves, Y. O. (2023) *Design e Implementação de uma Arquitetura de Data Warehouse*. Trabalho de Conclusão de Curso. Universidade Federal de Santa Maria, Santa Maria/RS, <https://repositorio.ufsm.br/handle/1/28249>, [acessado em 15/10/2023].

Amaral, D. R. X. (2020) *Desenvolvimento de Algoritmos com Apache Spark para Tratamento de Dados Industriais para Business Analytics com Tableau*. Trabalho de Conclusão de Curso. Universidade de São Paulo, São Paulo/SP, <https://sistemas.eel.usp.br/bibliotecas/monografias/2020/MEF20006.pdf>, [acessado em 01/11/2023].

Atlassian (2023) "What can I do on a board?", <https://support.atlassian.com/jira-work-management/docs/what-can-i-do-on-a-board/>, [acessado em 01/12/2023].

Digital.Ai (2022), "15th State of Agile Report: Agile adoption accelerates across the enterprise", https://info.digital.ai/rs/981-LQX-968/images/SOA15.pdf?_ga=2.2071153.1396698264.1663000131-2013811604.1663000131, [acessado em 12/12/2023].

L3 Software. (2023) "Jira: o software que se tornou o braço direito dos desenvolvedores". <https://l3software.com.br/blog/software/jira-o-software-que-se-tornou-o-braco-direito-dos-desenvolvedores/>, [acessado em 01/11/2023].

Leal, D. A (2021) *Pentaho, Airflow e Python: Avaliação de Ferramentas para a Criação de Pipeline de Dados*. Instituto Federal de Educação Ciência e Tecnologia de São Paulo, Campinas/SP, <http://pergamum.ifsp.edu.br/pergamumweb/vinculos/000075/0000754b.pdf>, [acessado em 01/10/2023]

Matos E. (2022) "Como reduzir o *lead time* dos processos e aumentar a produtividade na sua empresa", <https://blog.runrun.it/lead-time/#:~:text=A%20import%C3%A2ncia%20do%20lead%20time,gargalos%20no%20fluxo%20de%20trabalho>, [acessado em 01/10/2023].

Marques E. (2023) "Engenharia de Dados para Big Data: O Que é e Qual sua Importância?", https://awari.com.br/engenharia-de-dados-para-big-data-o-que-e-e-qual-sua-importancia/?utm_source=blog&utm_campaign=projeto+blog&utm_medium=Engenharia%20de%20Dados%20para%20Big%20Data:%20O%20Que%20%C3%89%20e%20Qual%20Sua%20Import%C3%A2ncia, [acessado em 01/12/2023].

- Maximiano, C. F. C. (2023). *Uso de Tecnologias de Big Data para Processamento e Análise de Dados da Área da Saúde do Estado de São Paulo*. Trabalho de Conclusão de Curso. Universidade Estadual Paulista Júlio de Mesquita Filho, Sorocaba/SP, <https://repositorio.unesp.br/handle/11449/243963>, [acessado em 01/10/2023].
- Redação Vivo (2021) “Metodologias Ágeis: o que são e como aplicá-las com tecnologia”, <https://vivomeunegocio.com.br/escritorios/innovar/metodologias-ageis/>, [acessado em 01/12/2023].
- Rox Partner (2021) “Conheça o Apache Airflow: a Solução Perfeita para Orquestrar Fluxos de Dados”, <https://roxpartner.com/conheca-o-apache-airflow/>, [acessado em 01/12/2023].
- Santos, C. (2023) "Metodologia Ágil Empresas: Adoção da Metodologia Ágil por Empresas" Awari. https://awari.com.br/metodologia-agil-empresas-adocao-da-metodologia-agil-por-empresas/?utm_source=blog&utm_campaign=projeto+blog&utm_medium=Metodologia%20%C3%81gil%20Empresas:%20Ado%C3%A7%C3%A3o%20da%20metodologia%20%C3%A1gil%20por%20empresas, [acessado em 01/12/2023].
- Silva, G. (2023) "O que é Apache Spark?", <https://coodesh.com/blog/dicionario/o-que-e-apache-spark/>, [acessado em 06/12/2023].

A524a Amorim, Victor Hugo Oliveira de

Análise e otimização de lead time em processos ágeis: estudo de caso em empresa de marketplace. / Victor Hugo Oliveira de Amorim. – Vitória da Conquista-BA : IFBA, 2023.

13 f.il.: color.

Orientador: Prof. Pablo Freire Matos

Trabalho de Conclusão de Curso (Graduação) – Bacharelado em Sistemas de Informação. Instituto Federal de Educação, Ciência e Tecnologia da Bahia - *Campus* Vitória da Conquista - BA, 2023.

1. Estudo de caso. 2. Análise lead time. 3. Empresa de Marketplace.
I. Matos, Pablo Freire. II. Título.

CDD: 621.39