

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

José Lucas Ferreira de Lima

**De voltas rápidas a decisões rápidas: o processo
de coleta e tratamento de dados em tempo real
na Fórmula 1**

Uberlândia, Brasil

2025

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

José Lucas Ferreira de Lima

De voltas rápidas a decisões rápidas: o processo de coleta e tratamento de dados em tempo real na Fórmula 1

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Sistemas de Informação.

Orientador: Ronaldo Castro de Oliveira

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2025

José Lucas Ferreira de Lima

De voltas rápidas a decisões rápidas: o processo de coleta e tratamento de dados em tempo real na Fórmula 1

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Sistemas de Informação.

Trabalho aprovado. Uberlândia, Brasil, 01 de novembro de 2016:

Ronaldo Castro de Oliveira
Orientador

Professor

Professor

Uberlândia, Brasil
2025

*Dedico este trabalho aos meus pais, pelo apoio incondicional em toda a minha trajetória;
aos meus amigos, que foram pilares essenciais durante minha jornada acadêmica; e à
minha avó, que nos deixou enquanto eu construía este trabalho, mas cuja memória
permanece viva em meu coração.*

Resumo

AJUSTAR AO FINAL DA ESCRITA DO TRABALHO

A Fórmula 1 é um ambiente altamente tecnológico onde a análise de dados em tempo real desempenha um papel crucial no desempenho das equipes, deixando a competitividade do esporte cada vez mais alta. Este trabalho explora a aplicação de **Big Data** na Fórmula 1, abordando tanto os conceitos teóricos quanto uma implementação prática baseada na telemetria do jogo **F1 23**.

O objetivo principal é demonstrar como a coleta e análise de dados podem fornecer insights valiosos para a tomada de decisões estratégicas durante uma corrida. Para isso, foi desenvolvida uma aplicação em **Python** que se conecta ao jogo pela rede local, estabelece um **socket** com este, recebe os dados de telemetria via **UDP**, processa as informações, e envia os dados por mensageria utilizando **Kafka**. Os resultados são exibidos em um **dashboard interativo** conectado a essa fila, permitindo análises em tempo real durante a corrida. A aplicação foi testada em um ambiente controlado, e os resultados obtidos são discutidos em detalhes.

A metodologia adotada inclui o estudo dos princípios do **Big Data** aplicados à F1, bem como o desenvolvimento da pipeline de dados para captura e visualização dos dados do jogo. A implementação prática busca simular a experiência real das equipes de F1 no uso de dados para análise de desempenho, ou seja, a visão do engenheiro de corrida durante a prova.

Os resultados esperados incluem a demonstração do impacto do Big Data na performance das equipes e a validação do potencial da tecnologia utilizada para fins de simulação e aprendizado. Este trabalho contribui para a compreensão do uso da telemetria na F1 e reforça a importância da análise de dados em cenários de alta performance.

Palavras-chave: big data, formula 1, engenharia de dados, python, pipeline.

Lista de ilustrações

Figura 1 – Arquitetura Proposta da Pipeline de Dados	15
Figura 2 – Isso é o que aparece no sumário	24

Lista de tabelas

Lista de abreviaturas e siglas

Fig.	Area of the i^{th} component
UDP	User Datagram Protocol
ETL	Extract, Transform, Load
CFD	Computational Fluid Dynamics

Sumário

1	INTRODUÇÃO	10
1.1	Contextualização e Motivação	10
1.2	Problema de Pesquisa	11
1.3	Objetivos	12
1.3.1	Objetivo Geral	12
1.3.2	Objetivos Específicos	12
1.4	Justificativa	13
1.5	Metodologia do Trabalho	14
1.5.1	Tipo e Abordagem da Pesquisa	14
1.5.2	Desenho da Solução: Arquitetura da Pipeline	14
1.5.3	Ferramentas e Tecnologias Utilizadas	16
1.5.4	Procedimentos de Desenvolvimento	16
1.5.5	Estratégias de Avaliação e Validação	17
1.6	Estrutura do Trabalho	17
2	REVISÃO BIBLIOGRÁFICA	19
2.1	Engenharia de dados	19
2.2	Análise de dados	19
2.3	Conceitos fundamentais de dados em tempo real (streaming)	20
2.4	Extração, Transformação e Carga (ETL)	21
2.5	Telemetria de dados em tempo real	21
2.6	Tecnologias Utilizadas	22
2.6.1	Python	22
2.6.2	Kafka	22
2.6.3	Docker	23
2.6.4	InfluxDB	23
2.7	Trabalhos Relacionados	23
3	DESENVOLVIMENTO	24
3.1	Arquitetura da Solução	24
3.2	Coleta de Dados via Telemetria	24
3.3	Processamento e Armazenamento de Dados	24
3.4	Visualização dos Dados	24
3.5	Implementação e Testes	24
4	ANÁLISE DOS RESULTADOS	25

4.1	Validação da Coleta e Processamento	25
4.2	Desempenho e Eficiência da Aplicação	25
4.3	Limitações e Melhorias Futuras	25
5	CONCLUSÃO E TRABALHOS FUTUROS	26
5.1	Revisão dos Objetivos e Contribuições	26
5.2	Desafios Encontrados	26
5.3	Aplicações Futuras	26
	REFERÊNCIAS	27
	APÊNDICES	29
	APÊNDICE A – QUISQUE LIBERO JUSTO	30
	APÊNDICE B – COISAS QUE FIZ E QUE ACHEI INTERESSANTE MAS NÃO TANTO PARA ENTRAR NO CORPO DO TEXTO	31
	ANEXOS	32
	ANEXO A – EU SEMPRE QUIS APRENDER LATIM	33
	ANEXO B – COISAS QUE EU NÃO FIZ MAS QUE ACHEI INTE- RESSANTE O SUFICIENTE PARA COLOCAR AQUI	34
	ANEXO C – FUSCE FACILISIS LACINIA DUI	35

1 Introdução

1.1 Contextualização e Motivação

De acordo com um artigo publicado pela [Mercedes \(s.d.\)](#), cada carro de Fórmula 1 é equipado com mais de 250 sensores categorizados em controle, instrumentação e monitoramento, que coletam dados vitais de pressão, temperatura, inércia e deslocamento de todos os sistemas do veículo. Durante um fim de semana de corrida, o volume total de dados gerados por um único carro, incluindo telemetria ao vivo (cerca de 30 MB por volta), vídeo e informações auxiliares, excede 1 terabyte, podendo duplicar ou triplicar após o pós-processamento. Adicionalmente, as fábricas contribuem com vasta quantidade de dados provenientes de dinamômetros, simuladores e túneis de vento, criando um ecossistema de informação que exige a transferência de até 11 terabytes entre a pista e as sedes em Brackley e Brixworth, com telemetria ao vivo chegando em meros 10 milissegundos em corridas europeias. Essa imensa e contínua geração de dados sublinha a natureza intensiva em dados da Fórmula 1 moderna.

No cenário altamente competitivo da Fórmula 1, a premissa fundamental é que a ciência de dados, e por extensão, a otimização da performance, não pode existir sem um fluxo contínuo e confiável de informações. A limitação drástica do tempo de testes em pista impõe uma pressão imensa para a coleta de dados precisos e completos na primeira tentativa, pois não há oportunidade para repetições, como aponta a [Mercedes \(s.d.\)](#). Essa característica ressalta a importância crítica de uma infraestrutura de dados robusta, onde, como detalhado por [Msakamali \(2024\)](#), a análise de dados é fundamental para aprimorar o desempenho e embasar decisões estratégicas em tempo real. Uma vez coletados do carro, os dados são sincronizados, criptografados e transmitidos via telemetria para a garagem, garantindo que engenheiros e pilotos tenham acesso a informações consistentes e seguras. O uso de software especializado, como o ATLAS da McLaren Applied, tanto em tempo real quanto para análises retrospectivas, demonstra a dedicação contínua à extração de insights valiosos, seja para comparar o desempenho dos pilotos, analisar dados de frenagem e velocidades de curva, ou para identificar oportunidades que tornarão o carro e o piloto ainda mais rápidos. A enorme quantidade de dados, embora desafiadora, é também uma oportunidade de engenharia para priorizar e analisar as informações corretas, acelerando o aprendizado e a evolução das equipes. Assim, a existência de uma pipeline de coleta e processamento de dados em tempo real não é apenas uma conveniência, mas uma necessidade estratégica e operacional para sustentar a competitividade na Fórmula 1.

A competitividade na Fórmula 1 moderna é diretamente proporcional à confia-

bilidade dos dados, pois a ciência e a tomada de decisão dependem intrinsecamente da qualidade e integridade da informação. Dados imprecisos ou inconsistentes comprometem insights e decisões estratégicas, como enfatizado por [Suman \(2025\)](#). Em um ambiente de alta performance e tempo crítico, a latência é inaceitável; a transformação de vastos volumes de telemetria em conhecimento acionável em milissegundos exige uma arquitetura de dados eficiente. O processamento em tempo real, segundo o artigo do [Richman \(2025\)](#), é crucial para insights imediatos e ações ágeis, permitindo a detecção e correção de anomalias. Assim, torna-se imperativa uma pipeline robusta de coleta, transmissão e processamento em tempo real, capaz de assegurar a integridade e prontidão dos dados do sensor até o engenheiro, garantindo resultados críveis e reproduzíveis ([SCISURE, 2025](#)).

Diante da complexidade e da criticidade dos desafios impostos pela volumosa geração e pela necessidade de processamento em tempo real dos dados na Fórmula 1, este trabalho propõe uma solução prática e inovadora. Nosso estudo se dedicará a explorar e aplicar diversas técnicas de ETL (Extract, Transform, Load) para construir uma pipeline robusta, tolerante a falhas e escalável de coleta e processamento de dados. O diferencial reside na utilização de ferramentas gratuitas e open-source, demonstrando que é possível desenvolver soluções de alta performance e segurança sem depender de softwares proprietários ou de alto custo. O objetivo é evidenciar como essas tecnologias podem efetivamente resolver o problema da aquisição, tratamento e disponibilização dos terabytes de dados gerados, garantindo que as equipes de Fórmula 1 possam extrair insights valiosos e tomar decisões ainda mais assertivas em um ambiente onde cada milissegundo conta.

1.2 Problema de Pesquisa

A Fórmula 1 contemporânea, impulsionada por uma colossal geração de dados – com mais de 250 sensores em cada carro produzindo terabytes de informações cruciais por fim de semana de corrida –, demanda uma capacidade sem precedentes de coleta, processamento e análise em tempo real para a otimização contínua do desempenho das equipes. Diante desse cenário de alta complexidade e criticidade temporal, surge o seguinte problema de pesquisa:

- Como é possível projetar e implementar uma pipeline de dados robusta, tolerante a falhas e escalável, utilizando tecnologias modernas, preferencialmente open-source, que garanta a extração, transporte (com mínima latência, integridade e segurança), transformação de valores brutos em formatos legíveis e a visualização efetiva de terabytes de dados provenientes de múltiplos sensores veiculares da Fórmula 1, tudo isso no menor espaço de tempo possível, para subsidiar decisões estratégicas e operacionais das equipes?

Este trabalho se propõe a responder a essa questão fundamental, explorando a aplicação de técnicas de ETL (Extract, Transform, Load) e arquiteturas de processamento

de stream de dados, visando demonstrar a viabilidade de construir uma solução eficiente e segura para o ecossistema de dados da Fórmula 1.

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo geral deste trabalho é projetar e implementar uma pipeline de dados em larga escala, robusta e eficiente, utilizando exclusivamente softwares de código aberto, para simular o processo completo de coleta e tratamento de dados em tempo real no contexto da Fórmula 1. Esta pipeline servirá como um alicerce fundamental para futuras aplicações de ciência de dados e inteligência artificial, demonstrando a viabilidade de transformar volumes massivos de dados brutos em informações acionáveis para otimizar a performance.

1.3.2 Objetivos Específicos

Alguns objetivos específicos deste trabalho incluem:

- **Analisar os Conceitos de ETL e Processamento de Dados em Tempo Real:** Investigar e compreender os principais conceitos de Extração, Transformação e Carga (ETL), bem como as metodologias de processamento de dados em tempo real (streaming), focando em sua aplicabilidade e desafios em ambientes de alta velocidade e volumetria de dados, como a Fórmula 1.
- **Desenvolver um Módulo de Coleta e Pré-processamento de Telemetria:** Criar uma aplicação em Python capaz de estabelecer conexão com uma fonte simulada de dados de telemetria da Fórmula 1 (neste caso, o jogo F1 23), realizar a extração e o pré-processamento inicial desses dados (transformação de valores binários brutos para formatos legíveis), e encaminhá-los para um sistema de mensageria.
- **Implementar uma Arquitetura de Mensageria e Processamento de Streams:** Configurar e integrar um sistema de mensageria (message broker) para garantir a transmissão eficiente e tolerante a falhas dos dados, e desenvolver módulos de processamento que consumam esses dados em tempo real, aplicando transformações e enriquecimentos necessários.
- **Visualizar Dados de Telemetria em Tempo Real:** Construir um dashboard interativo que receba os dados processados e os exiba graficamente em tempo real, permitindo a visualização clara de métricas cruciais de desempenho veicular, validando a funcionalidade e a latência da pipeline.

- **Avaliar a Eficiência e a Robustez da Pipeline:** Realizar testes controlados na aplicação desenvolvida para analisar o desempenho da pipeline em termos de latência, vazão (throughput), integridade dos dados e tolerância a falhas, utilizando métricas quantitativas.
- **Discutir Implicações e Aplicações Futuras da Ciência de Dados:** Apresentar como os dados coletados e processados podem subsidiar a tomada de decisões estratégicas por engenheiros na Fórmula 1 e propor futuras melhorias e extensões para a pipeline desenvolvida, incluindo potencial integração com modelos de ciência de dados e inteligência artificial.

1.4 Justificativa

A Fórmula 1, mais do que um esporte de alta velocidade, consolidou-se como um dos ecossistemas mais avançados para o estudo e aplicação da engenharia e ciência de dados. Cada carro é, em essência, um laboratório de inovação em tempo real, equipado para gerar uma quantidade colossal de informações – com dados de telemetria que podem chegar a 1,5 terabytes por fim de semana de corrida ([LUZICH, 2025](#)). Conforme evidenciado por [Msakamali \(2024\)](#), a análise de dados na F1 é fundamental para aprimorar o desempenho na corrida, fornecendo uma base sólida para a tomada de decisões estratégicas em tempo real. Nesse ambiente, a tecnologia de dados não é apenas um suporte, mas um diferencial estratégico crucial, tornando a F1 um campo de estudo exemplar para desafios contemporâneos em processamento de streaming e engenharia de dados de alta performance.

Apesar do vasto potencial dos dados na Fórmula 1, sua exploração plena é barrada por desafios técnicos inerentes ao Big Data, especialmente a velocidade e o volume com que são gerados. Cada milissegundo de atraso na aquisição ou processamento de dados pode comprometer uma decisão estratégica, tornando a baixa latência um requisito não negociável. Conforme a [Mercedes \(s.d.\)](#) destaca, a transmissão de telemetria em tempo real, embora impressionantemente rápida (10 milissegundos em corridas europeias), ainda lida com a complexidade de milhões de pontos de dados por segundo. Além da velocidade, a variedade dos dados provenientes de centenas de sensores exige sistemas que possam integrar e harmonizar diferentes formatos e estruturas. Tais complexidades impõem a necessidade crítica de pipelines de dados robustas, eficientes e seguras. Essas pipelines não apenas garantem a coleta e o transporte ininterrupto das informações, mas também realizam as transformações necessárias para converter dados brutos em insights acionáveis, salvaguardando a integridade e a segurança da informação desde o carro até a tomada de decisão final.

Para além do espetáculo e da competitividade da pista, os avanços tecnológicos e

as soluções desenvolvidas na Fórmula 1, particularmente no que tange à engenharia de dados e processamento em tempo real, possuem um impacto e uma aplicabilidade notáveis em diversos outros setores. Os desafios de gerenciar volume massivo, alta velocidade, variedade de dados e a necessidade de baixa latência – inerentes ao ambiente da F1 – são, em essência, os mesmos enfrentados pela indústria moderna, especialmente no contexto da Indústria 4.0 (SINGH; SINGH; SINGH, 2021). Setores como a manufatura, por exemplo, demandam a capacidade de coletar e processar dados de sensores em tempo real para fins de manutenção preditiva, otimização de linhas de produção e controle de qualidade, onde cada milissegundo na detecção de anomalias pode significar uma economia substancial ou a prevenção de falhas catastróficas (WIJESINGHE; ABEYSINGHE; SAMARANAYAKE, 2024). Dessa forma, o estudo aprofundado dos processos de coleta e tratamento de dados em tempo real na Fórmula 1 não apenas revela as inovações de ponta da categoria, mas também serve como um modelo prático e um campo de testes para demonstrar como a engenharia de dados e o Big Data podem ser aliados estratégicos fundamentais na tomada de decisões em cenários de alta performance e complexidade industrial.

1.5 Metodologia do Trabalho

Esta seção detalha o planejamento e a execução do desenvolvimento da pipeline de dados em tempo real para simular o processo de coleta e tratamento de telemetria na Fórmula 1.

1.5.1 Tipo e Abordagem da Pesquisa

O presente trabalho classifica-se como uma pesquisa de natureza **aplicada**, pois visa desenvolver e implementar uma solução prática para um problema real na engenharia de dados. A abordagem é predominantemente **quantitativa** e **experimental**, centrada na construção, validação e análise de desempenho de uma arquitetura de pipeline de dados em tempo real, utilizando métricas objetivas como latência e *throughput*.

1.5.2 Desenho da Solução: Arquitetura da Pipeline

Para atingir os objetivos propostos, será desenvolvida uma arquitetura de pipeline de dados *end-to-end* conforme ilustrado na Figura 1.

A arquitetura proposta compreende os seguintes módulos e fluxo de dados:

- **Fonte de Dados:** O jogo **F1 23** será utilizado como a fonte primária para simular a geração de dados de telemetria em tempo real, transmitindo-os via protocolo **UDP** (**User Datagram Protocol**).

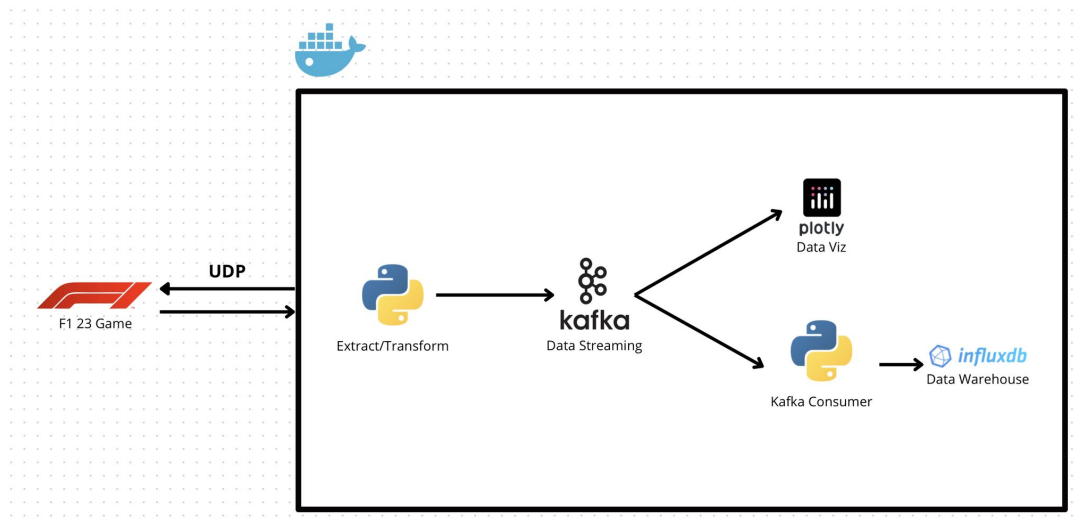


Figura 1 – Arquitetura Proposta da Pipeline de Dados

- **Módulo de Extração/Transformação (Python):** Uma aplicação desenvolvida em **Python** será responsável por capturar os dados UDP brutos do jogo. Este módulo realizará a etapa de **Extração**, lendo os pacotes de dados, e a primeira fase de **Transformação**, convertendo os valores binários recebidos em formatos legíveis (inteiros e decimais) e estruturando-os para processamento.
- **Mensageria (Kafka):** Após a extração e transformação inicial, os dados serão publicados em tópicos do **Apache Kafka**, servindo como uma plataforma de *streaming* de dados. O Kafka garantirá a transmissão eficiente, tolerante a falhas e escalável dos dados em tempo real.
- **Consumo e Persistência (Kafka Consumer / InfluxDB):** Um segundo módulo em **Python** atuará como **Kafka Consumer**, lendo os dados dos tópicos do Kafka. Este consumidor será responsável por carregar os dados em um **InfluxDB**, que funcionará como um *Data Warehouse* especializado em séries temporais, permitindo o armazenamento e a recuperação eficiente de grandes volumes de dados de telemetria.
- **Visualização (Plotly/Dash):** Para a visualização em tempo real, um *dashboard* interativo será desenvolvido utilizando a biblioteca **Dash** (baseada em Plotly). Este *dashboard* consumirá os dados diretamente dos tópicos do Kafka, exibindo as métricas processadas em tempo real e fornecendo uma interface visual para acompanhamento e análise.

1.5.3 Ferramentas e Tecnologias Utilizadas

Para a implementação da pipeline, serão utilizadas as seguintes tecnologias de código aberto:

- **Linguagem de Programação:** Python (para os módulos de extração/transformação e consumo/persistência).
- **Orquestração de Contêineres:** Docker ou Podman (para empacotar e gerenciar os serviços da pipeline, como Kafka e InfluxDB).
- **Plataforma de Streaming de Dados:** Apache Kafka.
- **Banco de Dados de Séries Temporais:** InfluxDB.
- **Framework para Dashboards Interativos:** Dash (com Plotly para visualização de dados).
- **Fonte de Dados de Telemetria:** Jogo F1 23.

1.5.4 Procedimentos de Desenvolvimento

A construção da pipeline seguirá as seguintes etapas interconectadas:

1. **Configuração do Ambiente:** Preparação do ambiente de desenvolvimento, incluindo a instalação de Docker/Podman e a configuração dos contêineres para Kafka e InfluxDB.
2. **Módulo de Extração e Transformação:** Desenvolvimento do script Python para capturar dados UDP do F1 23, realizar o *parsing* inicial dos pacotes de telemetria e publicar os dados processados no Kafka.
3. **Módulo de Persistência:** Implementação do script Python `Kafka Consumer` que consumirá os dados do Kafka e os armazenará no InfluxDB.
4. **Módulo de Visualização:** Desenvolvimento do *dashboard* interativo com Dash/Plotly, configurando-o para consumir dados em tempo real diretamente do Kafka e exibir as métricas de telemetria relevantes.
5. **Integração e Testes Iniciais:** Conexão e validação de todos os módulos da pipeline em conjunto para assegurar o fluxo de dados contínuo e correto.

1.5.5 Estratégias de Avaliação e Validação

A robustez e a eficiência da pipeline serão avaliadas através de testes sistemáticos, focando nas seguintes métricas e aspectos:

- **Tempo de Processamento (Latência):** Medição do tempo desde a geração do dado no jogo F1 23 até sua visualização no *dashboard* e persistência no InfluxDB.
- **Taxa de Perda de Pacotes:** Verificação da integridade do fluxo de dados para identificar qualquer perda de informação durante a transmissão e o processamento.
- **Volume/Stress:** Testes de carga para avaliar o desempenho da pipeline sob diferentes volumes de dados, simulando cenários de alta demanda.
- **Integridade dos Dados:** Validação da consistência e precisão dos dados após as etapas de transformação, comparando-os com os dados brutos da fonte.
- **Funcionalidade do Dashboard:** Verificação da correta exibição e atualização das métricas no *dashboard*, garantindo sua usabilidade para a análise de dados em tempo real.
- **Diferentes Condições de Pista:** Simulação de cenários variados no jogo F1 23 para testar a adaptabilidade da pipeline a diferentes tipos e volumes de dados gerados (ex: chuva, trechos de alta/baixa velocidade).

1.6 Estrutura do Trabalho

Este trabalho está organizado em cinco capítulos principais, cada um abordando um aspecto fundamental da pesquisa sobre o processo de coleta e tratamento de dados em tempo real na Fórmula 1, desde a contextualização teórica e a metodologia de engenharia de dados até a análise dos resultados obtidos.

No **Capítulo 1 - Introdução**, são apresentados o tema da pesquisa, a contextualização e motivação para o estudo, que destaca a evolução da Fórmula 1 para um esporte intensivo em dados, bem como a formulação do problema de pesquisa. Além disso, são definidos os objetivos geral e específicos, a justificativa para a escolha do tema e a relevância do estudo. Também são descritos a metodologia utilizada e a própria estrutura do trabalho.

O **Capítulo 2 - Revisão Bibliográfica** traz um embasamento teórico sobre os conceitos fundamentais de engenharia de dados, processamento de dados em tempo real (streaming) e Extração, Transformação e Carga (ETL), e sua aplicação no contexto da Fórmula 1. São abordados temas como a telemetria e a coleta massiva de dados em

tempo real na categoria, além das principais tecnologias utilizadas na construção de pipelines de dados, incluindo Python, Apache Kafka, Docker/Podman e InfluxDB. Este capítulo também inclui uma revisão de trabalhos relacionados e uma discussão sobre a indispensabilidade da engenharia de dados para a otimização da performance na F1.

No **Capítulo 3 - Desenvolvimento**, são detalhados os aspectos técnicos e práticos do estudo. Este capítulo apresenta a arquitetura da solução proposta para a pipeline de dados, os métodos específicos de extração de dados via telemetria simulada do jogo F1 23, e os processos de transformação, mensageria e armazenamento dessas informações. Além disso, aborda-se a implementação do dashboard interativo para visualização dos dados em tempo real e os testes realizados para validação prática da solução.

O **Capítulo 4 - Análise dos Resultados** apresenta a avaliação detalhada da pipeline desenvolvida, discutindo seu desempenho e eficiência em termos de latência, throughput, integridade dos dados e robustez sob diferentes condições. Também são analisadas as limitações do estudo e possíveis melhorias futuras que podem ser implementadas para aprimorar os processos investigados.

Por fim, o **Capítulo 5 - Conclusão e Trabalhos Futuros** revisita os objetivos propostos no início do trabalho e discute as principais contribuições da pesquisa para a área de engenharia de dados em tempo real. São destacados os desafios encontrados ao longo do estudo e sugeridas aplicações futuras para os conceitos e tecnologias exploradas, considerando possíveis avanços na área e a expansão da pipeline.

2 Revisão Bibliográfica

2.1 Engenharia de dados

A Engenharia de Dados é uma disciplina fundamental no cenário atual da informação, atuando como a espinha dorsal para a criação de valor a partir de grandes volumes de dados. Ela se concentra no projeto, construção, otimização e manutenção de infraestruturas e sistemas para a coleta, processamento, armazenamento e disponibilização de dados de forma eficiente e confiável (REIS; HOUSLEY, 2022). Esta área abrange uma gama de atividades que vão desde a ingestão e limpeza de dados até a sua transformação e modelagem, preparando-os para análises subsequentes e para o consumo por analistas, cientistas de dados e algoritmos de machine learning. Seu objetivo primordial é garantir a qualidade, integridade, segurança e acessibilidade dos dados, permitindo que as organizações derivem insights precisos e automatizem processos decisórios.

No contexto da Fórmula 1, a Engenharia de Dados assume um papel ainda mais crítico, dada a exigência por dados em tempo real e de alta fidelidade. Ela é indispensável para a criação de pipelines robustas e confiáveis que extraiam terabytes de dados brutos dos sensores veiculares, os transformem em informações úteis e os armazenem de forma escalável e segura. Essa infraestrutura de dados assegura que engenheiros e estrategistas no pit wall e na fábrica possam acessar as informações necessárias com mínima latência, possibilitando tomadas de decisão estratégicas e operacionais em frações de segundo durante as sessões de treino, qualificação e corrida. Além disso, a Engenharia de Dados é a responsável por mitigar desafios como perda de pacotes, corrupção de dados e inconsistências, garantindo a integridade e a confiabilidade que são cruciais para a vantagem competitiva na F1.

2.2 Análise de dados

A análise de dados é uma etapa fundamental para a extração de insights e tomada de decisões estratégicas. Ela envolve a coleta, processamento, modelagem e interpretação de dados para identificar padrões, tendências e correlações relevantes, envolvendo massivamente a utilização de técnicas de estatística, matemática avançada, machine learning e visualização de dados. Na Fórmula 1, a análise de dados é utilizada para monitorar o desempenho dos carros, simular cenários de corrida, prever resultados e otimizar estratégias de corrida.

Ao longo deste trabalho, iremos discutir sobre a sua importância ao final do fluxo

dos dados, afinal, de nada adianta coletar, processar e armazenar os dados se não conseguirmos extrair valor deles. Para isso, utilizaremos técnicas de visualização de dados, como análise exploratória, gráficos e dashboards, para facilitar a interpretação dos dados e a tomada de decisões.

2.3 Conceitos fundamentais de dados em tempo real (streaming)

No panorama atual da ciência de dados, a capacidade de processar informações em tempo hábil tornou-se um diferencial competitivo crucial para organizações e sistemas complexos. Tradicionalmente, o processamento de dados era predominantemente realizado em modo de lote (*batch processing*), onde grandes volumes de dados são coletados e processados periodicamente. Embora eficiente para tarefas que não exigem imediatismo, essa abordagem apresenta limitações significativas em cenários dinâmicos e de alta volatilidade (HASHEM et al., 2015).

É nesse contexto que o processamento de dados em tempo real (*real-time data processing*), ou *streaming* de dados, emerge como uma abordagem indispensável. O *streaming* de dados refere-se à metodologia de processamento de informações à medida que elas são geradas ou chegam, de forma contínua e com mínima latência (XU et al., 2018). Diferentemente do processamento em lote, que opera sobre conjuntos de dados estáticos e finitos, o processamento de *streaming* lida com fluxos de dados contínuos e potencialmente ilimitados.

As principais características que definem o processamento de *streaming* incluem:

- **Baixa Latência:** A informação é processada e disponibilizada com o menor atraso possível, frequentemente na ordem de milissegundos, o que é vital para tomadas de decisão ágeis.
- **Processamento Contínuo:** Os dados são consumidos e processados incessantemente, sem interrupções ou a necessidade de acumular grandes volumes antes da análise.
- **Alta Velocidade e Volume:** Os sistemas de *streaming* são projetados para gerenciar e analisar grandes volumes de dados que chegam em taxas extremamente elevadas.

A relevância do processamento de *streaming* é particularmente acentuada em domínios como a Fórmula 1. Dada a criticidade do tempo em decisões estratégicas – como ajustes de performance veicular, gestão de pneus ou respostas a incidentes em pista – a análise instantânea dos dados de telemetria provenientes de centenas de sensores é fundamental. A capacidade de processar e reagir a esses fluxos contínuos de dados permite

que as equipes otimizem o desempenho em frações de segundo, transformando a agilidade informacional em vantagem competitiva (MERCEDES, s.d.).

2.4 Extração, Transformação e Carga (ETL)

Segundo Seenivasan (2023), estes trabalhos são responsáveis pela extração de dados de diversas fontes (no nosso caso, os sensores do carro), transformação desses dados para torná-los mais consistentes e úteis, e carregamento desses dados em um sistema de armazenamento (como um banco de dados ou *data warehouse*).

Como apontado pela Mercedes (s.d.), os carros possuem em torno de 250 sensores em sua construção, o que exige a construção de uma pipeline de dados robusta e eficiente para coletar, processar e armazenar esses dados em tempo real. Mehmood e Anees (2020) sugere que a construção de uma pipeline de ETL em tempo real é bem desafiadora, vide a própria natureza do dado em respeito ao seu volume, variedade, velocidade, volatilidade, variabilidade, veracidade e valor - todos esses desafios serão amplamente encontrados aqui, já que os sensores disparam diferentes tipos de dados em formatos diferentes, num grande volume e numa velocidade altíssima.

Sendo assim, chega-se à conclusão de que a implementação das melhores práticas de ETL tornam esse processo muito mais simples de ser construído e permite que as equipes atinjam maior vantagem competitiva.

2.5 Telemetria de dados em tempo real

O grande desafio de todas as equipes até aqui: como coletar gigabytes de dados de diversos formatos em uma rede privada, compartilhada apenas entre as equipes, de forma segura, eficiente e no menor espaço de tempo possível, e mais, sem deixar esse dado ser corrompido ou perdido? O começo de tudo é a telemetria, que é a tecnologia que permite a coleta, transmissão e análise de dados em tempo real.

Todos os sensores do carro monitoram os componentes atrelados a si e, a cada milissegundo ou segundo, enviam esses dados para o paddock através de uma conexão com um computador disponibilizado aos engenheiros de equipe. Esses dados são enviados através de uma conexão UDP, que é um protocolo de comunicação rápido e eficiente, ideal para a transmissão de dados em tempo real.

Assim que os dados chegam ao computador, uma pipeline de dados é acionada, que é responsável por processar, armazenar e analisar esses dados. Esse passo é necessário pois os dados chegam em um formato de binário, que precisa ser tratado e transformado em informações úteis para os engenheiros - uma dessas transformações é justamente converter os dados binários para inteiros e números decimais.

A pipeline é composta por diversas etapas, como a coleta, transformação e carregamento dos dados. Com os dados já tratados, eles são enviados para um sistema de mensageria, que é responsável por transmitir esses dados para o dashboard, que é a interface gráfica que os engenheiros utilizam para analisar o desempenho do carro em tempo real.

Além disso, podemos, também, armazenar esses dados tratados em um banco, seja ele relacional ou não, para futuras análises e consultas. Isso é importante pois, com o passar do tempo, podemos identificar padrões e tendências que não seriam possíveis de serem identificados em tempo real. Outra possibilidade seria justamente usar esses dados para treinar um modelo de machine learning, que pode prever o desempenho do carro em diferentes cenários, ou até mesmo simular corridas.

Com base na telemetria, os engenheiros conseguem pensar em estratégias, realizar ajustes no carro e testar o desempenho dessas modificações. Ele também consegue monitorar o desempenho do piloto, analisando sua concentração, batimentos cardíacos e pressão sanguínea, o que pode ajudar outros membros da equipe a fazer melhorias no treino físico do piloto, além de ajustar o carro de acordo com suas necessidades, aumentando o conforto para que o piloto possa se concentrar apenas em pilotar.

2.6 Tecnologias Utilizadas

2.6.1 Python

Python é uma linguagem de programação de alto nível, interpretada, de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte. Foi lançada por Guido van Rossum em 1991 com o objetivo de ser uma linguagem de fácil leitura e aprendizado, com uma sintaxe limpa e clara, diferentemente de C, que inspirou sua criação. Atualmente, é uma das linguagens mais populares do mundo, com uma comunidade ativa e uma vasta quantidade de bibliotecas, tanto para desenvolvimento web, científico, como para automação de tarefas.

Neste trabalho, iremos utilizá-la para desenvolver a aplicação que se conecta ao jogo, recebe os dados de telemetria, processa esses dados e envia para a mensageria, além do dashboard final que exibirá os dados tratados. Sua escolha se deu pela facilidade de uso, vasta quantidade de bibliotecas disponíveis e por ser uma linguagem de alto nível, o que facilita o desenvolvimento e manutenção do código.

2.6.2 Kafka

Apache Kafka é uma plataforma de mensageria distribuída, que permite a publicação e subscrição de streams de dados em tempo real. Foi criada pela LinkedIn em 2011

e, atualmente, é mantida pela Apache Software Foundation. Sua arquitetura é baseada em tópicos, partições e consumidores, o que permite a escalabilidade e tolerância a falhas. Além disso, é altamente eficiente, sendo capaz de lidar com milhões de mensagens por segundo.

Seu uso neste trabalho será extremamente importante visto a quantidade de dados que serão gerados e a necessidade de transmiti-los em tempo real para o dashboard sem perda de informações. Nestes casos, o Kafka é a melhor opção, pois garante a entrega das mensagens e a escalabilidade necessária para lidar com a quantidade de dados gerados.

2.6.3 Docker

Docker é uma plataforma de código aberto que facilita a criação, implantação e execução de aplicativos em contêineres. Contêineres permitem que um desenvolvedor empacote uma aplicação com todas as partes necessárias, como bibliotecas e outras dependências, e a envie como uma única imagem. Com essa ferramenta, podemos colocar uma série de serviços em contêineres, como o Kafka, o dashboard e o banco de dados, e executá-los juntos, garantindo que todos os serviços estejam disponíveis e funcionando corretamente, facilitando a execução e o gerenciamento da aplicação, além de garantir que a aplicação funcione da mesma forma em diferentes ambientes.

Neste trabalho, sua utilização se dá pela sua leveza, facilidade de uso e interconectividade entre as ferramentas presentes no contêiner, o que facilita a execução e o gerenciamento da aplicação.

2.6.4 InfluxDB

InfluxDB é um banco de dados de séries temporais de código aberto, otimizado para armazenar e consultar grandes volumes de dados de séries temporais em tempo real. Ele foi criado pela InfluxData em 2013 e, atualmente, é mantido pela mesma. Sua arquitetura é baseada em séries temporais, pontos de dados e tags, o que permite a rápida inserção e consulta de dados de séries temporais. Além disso, é altamente escalável, sendo capaz de lidar com milhões de pontos de dados por segundo.

Sua utilização neste trabalho se dá pela necessidade de armazenar os dados tratados para futuras análises, consultas e alimentar modelos de inteligência artificial, além de garantir a integridade e a disponibilidade dos dados gerados.

2.7 Trabalhos Relacionados

Ver com o Ronaldo depois para entender o que podemos colocar aqui.

3 Desenvolvimento

Um ou mais capítulos (por exemplo um para testes)

3.1 Arquitetura da Solução

3.2 Coleta de Dados via Telemetria

3.3 Processamento e Armazenamento de Dados

3.4 Visualização dos Dados

3.5 Implementação e Testes

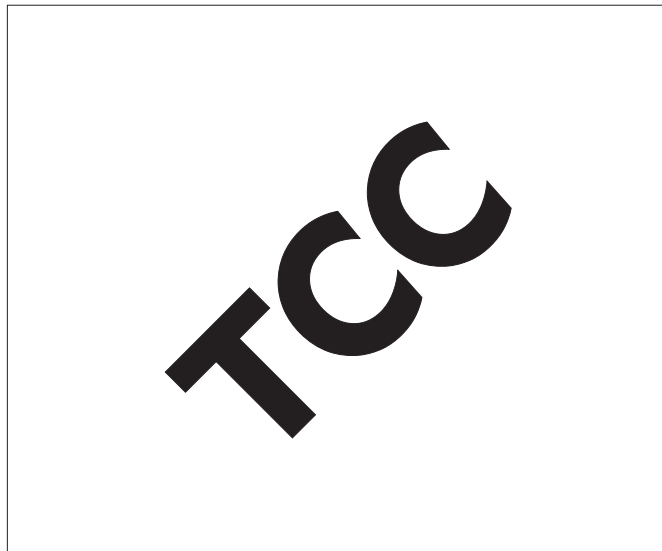


Figura 2 – Imagem de exemplo.

4 Análise dos Resultados

4.1 Validação da Coleta e Processamento

4.2 Desempenho e Eficiência da Aplicação

4.3 Limitações e Melhorias Futuras

5 Conclusão e Trabalhos Futuros

E daí?

5.1 Revisão dos Objetivos e Contribuições

5.2 Desafios Encontrados

5.3 Aplicações Futuras

Referências

- HASHEM, I. A. T.; YAQOOB, I.; ANUAR, N. B. M.; MOKHTAR, S.; GANI, A.; KHAN, S. U. The rise of “big data” on cloud computing: Review and open research issues. **Information Systems**, Elsevier, v. 47, p. 98–115, 2015. Citado na página 20.
- LUZICH, M. **Data-Driven Dominance: How Real-Time Analytics Are Reshaping F1 Race Strategy**. 2025. <<https://michaelluzich.com/data-driven-dominance-how-real-time-analytics-are-reshaping-f1-race-strategy/>>. Citado na página 13.
- MEHMOOD, E.; ANEES, T. Challenges and solutions for processing real-time big data stream: A systematic literature review. **IEEE Access**, v. 8, p. 119123–119143, 2020. Citado na página 21.
- MERCEDES. What is big data? introduction, history, types, characteristics, examples and jobs. s.d. Citado 3 vezes nas páginas 10, 13 e 21.
- MSAKAMALI. **F1 Data Analysis and Tactical Insights: Exploring Formula 1 Race Performance Strategies**. 2024. <https://www.theseus.fi/bitstream/handle/10024/856650/Msakamali_Baraka.pdf?sequence=6>. Citado 2 vezes nas páginas 10 e 13.
- REIS, J.; HOUSLEY, M. **Fundamentals of Data Engineering: Plan and Build Robust Data Systems**. [S.l.]: O'Reilly Media, 2022. Citado na página 19.
- RICHMAN, J. **What Is Real-Time Processing (In-depth Guide For Beginners)**. 2025. <<https://estuary.dev/blog/what-is-real-time-processing/#:~:text=Real%2Dtime%20data%20processing%20refers,available%20for%20use%20almost%20instantly.>> Citado na página 11.
- SCISURE. **Data Integrity in Labs: Why It is Essential and How to Achieve It**. 2025. <https://www.scisure.com/blog/data-integrity-in-labs-why-its-essential-and-how-to-achieve-it>. Citado na página 11.
- SEENIVASAN, D. Etl (extract, transform, load) best practices. **International Journal of Computer Trends and Technology**, v. 71, p. 40–44, 01 2023. Citado na página 21.
- SINGH, V. K.; SINGH, S. K.; SINGH, A. K. Explore big data analytics applications and opportunities: A review. **Applied Sciences**, v. 11, n. 4, p. 157, 2021. Citado na página 14.
- SUMAN, S. **The Importance of data quality in decision-making**. 2025. <https://www.expresscomputer.in/guest-blogs/the-importance-of-data-quality-in-decision-making/125738/>. Citado na página 11.
- WIJESINGHE, T. M. H. S. B.; ABEYSINGHE, A. M. H. R.; SAMARANAYAKE, V. S. G. P. Big data for predictive maintenance in industry 4.0: Enhancing operational efficiency and equipment reliability. **International Journal of Computer Applications Technology (IJCAT)**, v. 13, n. 10, p. 1003–1010, 2024. Citado na página 14.

XU, J.; ZHANG, J.; LI, J.; LI, G. Real-time big data stream processing: A survey. **Future Generation Computer Systems**, Elsevier, v. 82, p. 681–692, 2018. Citado na página [20](#).

Apêndices

APÊNDICE A – Quisque libero justo

Quisque facilisis auctor sapien. Pellentesque gravida hendrerit lectus. Mauris rutrum sodales sapien. Fusce hendrerit sem vel lorem. Integer pellentesque massa vel augue. Integer elit tortor, feugiat quis, sagittis et, ornare non, lacus. Vestibulum posuere pellentesque eros. Quisque venenatis ipsum dictum nulla. Aliquam quis quam non metus eleifend interdum. Nam eget sapien ac mauris malesuada adipiscing. Etiam eleifend neque sed quam. Nulla facilisi. Proin a ligula. Sed id dui eu nibh egestas tincidunt. Suspendisse arcu.

APÊNDICE B – Coisas que fiz e que achei interessante mas não tanto para entrar no corpo do texto

Nunc velit. Nullam elit sapien, eleifend eu, commodo nec, semper sit amet, elit. Nulla lectus risus, condimentum ut, laoreet eget, viverra nec, odio. Proin lobortis. Curabitur dictum arcu vel wisi. Cras id nulla venenatis tortor congue ultrices. Pellentesque eget pede. Sed eleifend sagittis elit. Nam sed tellus sit amet lectus ullamcorper tristique. Mauris enim sem, tristique eu, accumsan at, scelerisque vulputate, neque. Quisque lacus. Donec et ipsum sit amet elit nonummy aliquet. Sed viverra nisl at sem. Nam diam. Mauris ut dolor. Curabitur ornare tortor cursus velit.

Morbi tincidunt posuere arcu. Cras venenatis est vitae dolor. Vivamus scelerisque semper mi. Donec ipsum arcu, consequat scelerisque, viverra id, dictum at, metus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut pede sem, tempus ut, porttitor bibendum, molestie eu, elit. Suspendisse potenti. Sed id lectus sit amet purus faucibus vehicula. Praesent sed sem non dui pharetra interdum. Nam viverra ultrices magna.

Aenean laoreet aliquam orci. Nunc interdum elementum urna. Quisque erat. Nullam tempor neque. Maecenas velit nibh, scelerisque a, consequat ut, viverra in, enim. Duis magna. Donec odio neque, tristique et, tincidunt eu, rhoncus ac, nunc. Mauris malesuada malesuada elit. Etiam lacus mauris, pretium vel, blandit in, ultricies id, libero. Phasellus bibendum erat ut diam. In congue imperdiet lectus.

Anexos

ANEXO A – Eu sempre quis aprender latim

Sed mattis, erat sit amet gravida malesuada, elit augue egestas diam, tempus scelerisque nunc nisl vitae libero. Sed consequat feugiat massa. Nunc porta, eros in eleifend varius, erat leo rutrum dui, non convallis lectus orci ut nibh. Sed lorem massa, nonummy quis, egestas id, condimentum at, nisl. Maecenas at nibh. Aliquam et augue at nunc pellentesque ullamcorper. Duis nisl nibh, laoreet suscipit, convallis ut, rutrum id, enim. Phasellus odio. Nulla nulla elit, molestie non, scelerisque at, vestibulum eu, nulla. Ut odio nisl, facilisis id, mollis et, scelerisque nec, enim. Aenean sem leo, pellentesque sit amet, scelerisque sit amet, vehicula pellentesque, sapien.

ANEXO B – Coisas que eu não fiz mas que achei interessante o suficiente para colocar aqui

Sed consequat tellus et tortor. Ut tempor laoreet quam. Nullam id wisi a libero tristique semper. Nullam nisl massa, rutrum ut, egestas semper, mollis id, leo. Nulla ac massa eu risus blandit mattis. Mauris ut nunc. In hac habitasse platea dictumst. Aliquam eget tortor. Quisque dapibus pede in erat. Nunc enim. In dui nulla, commodo at, consectetur nec, malesuada nec, elit. Aliquam ornare tellus eu urna. Sed nec metus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

ANEXO C – Fusce facilisis lacinia dui

Phasellus id magna. Duis malesuada interdum arcu. Integer metus. Morbi pulvinar pellentesque mi. Suspendisse sed est eu magna molestie egestas. Quisque mi lorem, pulvinar eget, egestas quis, luctus at, ante. Proin auctor vehicula purus. Fusce ac nisl aliquam ante hendrerit pellentesque. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Morbi wisi. Etiam arcu mauris, facilisis sed, eleifend non, nonummy ut, pede. Cras ut lacus tempor metus mollis placerat. Vivamus eu tortor vel metus interdum malesuada.