

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
FACULDADE DE ENGENHARIA
ENGENHARIA COMPUTACIONAL

Enedilson Fernandes Júnior

**Streaming de Dados de Sensores Vestíveis: A Construção de um Data
Warehouse Near Real Time**

Juiz de Fora
2026

Enedilson Fernandes Júnior

**Streaming de Dados de Sensores Vestíveis: A Construção de um Data
Warehouse Near Real Time**

Trabalho de conclusão de curso á Engenharia Computacional da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de bacharel em Engenharia Computacional.

Orientador: Prof. Dr. Victor Ströele de Andrade Menezes

Juiz de Fora
2026

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Fernandes Júnior, Enedilson.

Streaming de Dados de Sensores Vestíveis: A Construção de um Data
Warehouse Near Real Time / Enedilson Fernandes Júnior. – 2026.
37 f. : il.

Orientador: Victor Ströele de Andrade Menezes

Trabalho de Conclusão de Curso – Universidade Federal de Juiz de Fora,
Faculdade de Engenharia. Engenharia Computacional, 2026.

1. Data Warehouse. 2. NRT. 3. Sensores Vestíveis. I. MENEZES,
Victor Ströele de Andrade, orient. II. Título.

Enedilson Fernandes Júnior

Streaming de Dados de Sensores Vestíveis: A Construção de um Data Warehouse Near Real Time

Trabalho de conclusão de curso á Engenharia Computacional da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de bacharel em Engenharia Computacional.

Aprovada em 16 de janeiro de 2026

BANCA EXAMINADORA

Prof. Dr. Victor Ströele de Andrade Menezes -
Orientador
Universidade Federal de Juiz de Fora

Prof. Dr. Leonardo Goliatt da Fonseca
Universidade Federal de Juiz de Fora

Prof. Dr. Wagner Antonio Arbex
Universidade Federal de Juiz de Fora

A Deus, aos meus pais e minha família.

AGRADECIMENTOS

Agradeço primeiramente a Deus, que permitiu e me abençoou para cursar Engenharia Computacional, um curso tão concorrido e difícil. Ele quem me deu força e resiliência para continuar a batalha até o fim e concluir com êxito

Agradeço aos meus pais, meus irmãos e minha avó Maria Lúcia e Paulo, um amigo querido, que tanto me apoiaram, incentivaram, torceram e me suportaram nesse longo percurso, um passo muito importante para a minha vida profissional, no qual, não cresci somente em conhecimento, porém, como ser humano.

Agradeço aos meus amigos e professores, que passaram por essa longa jornada comigo, me forjando no que me tornei hoje, auxiliando no que era possível, sempre entregando o melhor que podiam. Facilitando e me fornecendo ferramentas para enxergar o mundo de maneira multidisciplinar, pois todas as áreas do conhecimento podem andar juntas e se beneficiar uma das outras. Esse é um lema que levarei da Engenharia Computacional, a Engenharia é um caminho para resolver inúmeros problemas.

Agradeço à todos os funcionários da Universidade Federal de Juiz de Fora, sejam efetivos ou terceirizados, que sempre ampararam e fizeram minha experiência na graduação, ser agradável e aconchegante. Me permitindo usufruir do que a UFJF nos fornece com muita qualidade e dedicação.

Agradeço ao Prof. Dr. Victor Ströele de Andrade Menezes que ministrou a disciplina de Data Warehouse e aceitou me orientar nesse desafio, me indicando os caminhos para executar este trabalho, o qual espero que tenha um impacto científico relevante que possa gerar bons frutos.

"Sim, coisas grandiosas fez o Senhor por nós, por isso estamos alegres"

Salmo 126:3

"E sabemos que Deus faz todas as coisas cooperarem para o bem daqueles que o amam e que são chamados de acordo com seu propósito."

Romanos 8:28

RESUMO

O envelhecimento populacional e a necessidade de monitoramento clínico contínuo demandam soluções tecnológicas avançadas em saúde digital (*e-health*). Este trabalho apresenta a implementação de um *Data Warehouse Near Real-Time* (NRT) projetado para o processamento de fluxos de dados provenientes de sensores vestíveis (*wearables*). A arquitetura proposta utiliza Apache Kafka para a ingestão de dados, Apache Flink para o processamento em *streaming* com janelamento temporal de 10 segundos, TimescaleDB para o armazenamento otimizado de séries temporais e Grafana para a visualização analítica em tempo real. A solução foi validada com o *dataset* HAR70+, que compreende dados de acelerometria inercial de idosos monitorados em atividades de vida diária, como caminhar, sentar e deitar, ao longo de dois meses. Os resultados demonstraram uma latência de processamento entre 10 e 15 segundos desde a captura do sinal até a disponibilidade visual, o que representa uma redução de aproximadamente 99,9% na latência informacional quando comparada aos modelos tradicionais de ETL em lote (ciclos de 24 horas). A modelagem dimensional implementada permitiu consultas analíticas eficientes sobre o desempenho motor dos usuários, enquanto a Arquitetura Lambda equilibrou o processamento de baixa latência com a persistência histórica de grande volume. O sistema viabiliza o acompanhamento remoto e contínuo da capacidade funcional de idosos, permitindo intervenções rápidas e fornecendo uma infraestrutura robusta para futuros sistemas preditivos baseados em aprendizado de máquina.

Palavras-chave: Data Warehouse; Sensores Vestíveis; Reconhecimento de Atividade Humana; Near Real-Time; Apache Flink; Saúde Digital.

ABSTRACT

Population aging and the need for continuous clinical monitoring demand advanced technological solutions in digital health (*e-health*). This work presents the implementation of a Near Real-Time (NRT) Data Warehouse designed for processing data streams from wearable sensors. The proposed architecture utilizes Apache Kafka for data ingestion, Apache Flink for stream processing with 10-second time windowing, TimescaleDB for optimized time-series storage, and Grafana for real-time analytical visualization. The solution was validated using the HAR70+ dataset, which comprises inertial accelerometry data from elderly individuals monitored during activities of daily living, such as walking, sitting, and lying down, over a two-month period. The results demonstrated a processing latency between 10 and 15 seconds from signal capture to visual availability, representing an approximately 99.9% reduction in informational latency compared to traditional batch ETL models (24-hour cycles). The implemented dimensional modeling enabled efficient analytical queries regarding the users' motor performance, while the Lambda Architecture balanced low-latency processing with reliable high-volume historical persistence. The system enables remote and continuous monitoring of the functional capacity of the elderly, allowing for rapid interventions and providing a robust infrastructure for future predictive systems based on machine learning.

Keywords: Data Warehouse; Wearable Sensors; Human Activity Recognition; Near Real-Time; Apache Flink; Digital Health.

LISTA DE ILUSTRAÇÕES

Figura 2 – Modelo de Dados	24
Figura 2 – Arquitetura Utilizada	26
Figura 3 - Latência de ingestão calculada pela ingestão distribuída intervalo de 60 segundos	29
Figura 4 - Quantidade de atividades registradas por horas na data de 24/03/2021 .	30
Figura 5 - Quantidade de atividades registradas por horas na data de 24/04/2021 .	30
Figura 6 - Quantidade de atividades registradas por horas na data de 27/05/2021 .	31
Figura 7 - Quantidade de atividades registradas por horas no período de 24/03/2021 a 27/05/2021	31

LISTA DE TABELAS

Tabela 1 – Dados da dim_activities	23
--	----

LISTA DE ABREVIATURAS E SIGLAS

AAL	<i>Ambient Assisted Living</i> (Ambientes de Vida Assistida)
BDA	<i>Big Data Analytics</i>
BAN	<i>Body Area Networks</i> (Dispositivos em redes de Área Corporal)
BI	<i>Business Intelligence</i>
CSV	<i>Comma-Separated Values</i>
DW	<i>Data Warehouse</i>
ETL	<i>Extraction, Transformation and Load</i>
IoT	<i>Internet of Things</i> (Internet das Coisas)
LGPD	<i>Lei Geral de Proteção de Dados</i>
NRT	<i>Near Real-Time</i> (Próximo do Tempo Real)
PIR	<i>Passive Infrared Sensor</i>
QoS	<i>Quality of Service</i>
SGBDR	Sistema Gerenciador de Banco de Dados Relacional

SUMÁRIO

1	INTRODUÇÃO	12
2	FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS	14
2.1	Big Data	14
2.2	<i>Data Warehouse</i>	15
2.2.1	NRT: <i>Near Real Time - Próximo do Tempo Real</i>	15
2.3	<i>E-Health</i>	17
2.3.1	Sensores Vestíveis	18
2.4	Trabalhos Relacionados	18
3	MATERIAIS E MÉTODOS	20
3.1	Arquitetura	20
3.1.1	Coleta de Dados de Sensores AAL	21
3.1.2	Ingestão de Dados com Apache Kafka	21
3.2	Processamento de Streaming com Apache Flink (<i>PyFlink</i>)	21
3.3	Armazenamento do Data Warehouse (<i>TimescaleDB/PostgreSQL</i>)	22
3.4	Monitoramento e Visualização com Grafana	24
3.5	Infraestrutura e Ambiente de Desenvolvimento	25
3.6	Configurações do Ambiente de Desenvolvimento	27
3.7	Implementação	27
4	RESULTADOS	29
4.1	Discussão da Implementação	32
4.2	Limitações e Desafios de Implementação	32
4.2.1	Infraestrutura de Hardware e Virtualização	33
4.2.2	Ecossistema de Software e Licenciamento	33
5	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS . .	34
	REFERÊNCIAS	35

1 INTRODUÇÃO

Atualmente, observa-se um expressivo aumento no volume de dados em diversos domínios, como *e-health*, sistemas de Internet das Coisas (IoT), redes sociais, finanças e cadeias de suprimentos. Muitos desses dados não possuem uma estrutura fixa e demandam análises em tempo real ou quase em tempo real, o que impõe novos desafios computacionais. Esse crescimento, impulsionado pelos avanços tecnológicos da última década, foi estimado em um salto de 130 *exabytes* em 2005 para mais de 40.000 *exabytes* em 2020 (1). Esse volume de dados, que, em geral, possui variados formatos e exige processamento em alta velocidade, é conhecido como *Big Data* e tem motivado inúmeras pesquisas voltadas à sua gestão (1).

Diante da necessidade de análise de dados e suporte à tomada de decisão, os *Data Warehouses* (DWs) emergiram como uma solução fundamental. Originados conceitualmente na década de 1970, os DWs configuram-se como sistemas de integração de dados não voláteis voltados ao suporte estratégico. Sua operacionalização ocorre por meio de processos de ETL (*Extraction, Transformation and Load*), que capturam dados brutos e os convertem em conhecimento aplicável por meio de relatórios e *dashboards* (2). Apesar de seus benefícios, sua implementação é intrinsecamente complexa, exigindo estudos aprofundados sobre modelagem de dados e arquiteturas de integração (2).

Tradicionalmente utilizados para consolidar dados históricos, os DWs tornaram-se protagonistas na Inteligência de Negócios (*Business Intelligence* - BI). Entretanto, a necessidade de respostas rápidas aos desafios do *Big Data* impulsionou a busca por processamento em tempo próximo ao real (*Near Real-Time* - NRT). Nesse contexto, o modelo de DW tradicional, baseado em lotes (*batch*) e com latência inerente, mostrou-se inadequado às exigências contemporâneas de respostas ágeis a eventos e de otimização tática (4).

A relevância da computação estendeu-se à área da saúde, onde as novas tecnologias de informação (*e-health*) impulsionaram uma revolução na prestação de cuidados, ampliando o acesso a informações e aprimorando a qualidade do atendimento (5). Nesse cenário, o monitoramento da capacidade funcional — a habilidade de realizar atividades de vida diária de forma autossuficiente — torna-se um indicador crítico de saúde, uma vez que quedas e declínios motores podem reduzir drasticamente a autonomia dos idosos (10). Para mitigar esses riscos, a implementação de sistemas de Reconhecimento de Atividade Humana (*Human Activity Recognition* - HAR) baseados em sensores vestíveis tem se mostrado uma solução de *e-health* eficaz, permitindo a identificação automatizada de movimentos e o envio de alertas rápidos em caso de incidentes (7). Este setor de eletrônicos vestíveis emerge como um segmento distinto na era digital, com projeções de crescimento de até 42 vezes até o ano de 2050, aproximando-se da marca de 2 bilhões de unidades anuais, gerando 3,4 Mt de

emissões equivalentes de CO_2 na atmosfera, causando mais aquecimento(8). Tal expansão impõe desafios que transcendem a eficácia médica, alcançando a sustentabilidade dos sistemas ambientais e sociais devido à ecotoxicidade e ao descarte de resíduos eletrônicos (8). Contudo, a eficácia desses sistemas depende da capacidade de processar grandes fluxos de dados com baixa latência para garantir respostas oportunas (6).

Este trabalho propõe uma arquitetura de *Data Warehouse Near Real-Time* (NRT) projetada para consumir esses fluxos de dados de acelerometria em tempo quase real, preenchendo a lacuna entre a coleta de dados brutos e a análise estratégica necessária para intervenções preventivas. A implementação consiste na ingestão via serviço de mensageria, no processamento dinâmico em *stream* e na carga em banco de dados de séries temporais, permitindo que os sinais dos sensores sejam transformados em indicadores atualizados instantaneamente por meio de janelas temporais de atualização para responder uma pergunta central: **Quais são as atividades mais praticadas pelas pessoas em idade avançada?**

Para alcançar este objetivo, a metodologia estruturou-se em três etapas: (i) Revisão da Literatura; (ii) Desenvolvimento e Implementação da Arquitetura do *pipeline* de dados; e (iii) Análise da viabilidade da proposta.

O trabalho está organizado da seguinte forma: além desta introdução, o Capítulo 2, apresenta a fundamentação teórica, estabelecendo as bases do presente estudo, bem como reúne estudos relacionados pertinente ao uso de DW e atualização de dados em tempo próximo do real; no Capítulo 3, descreve-se a metodologia, detalhando as abordagens e ferramentas utilizadas; o Capítulo 4, apresenta a discussão da solução, com a arquitetura e os componentes do DW proposto para dados AAL; o Capítulo 5, expõe os resultados, incluindo a análise dos achados e a validação da proposta; por fim, no Capítulo 6, são apresentadas as considerações finais, sintetizando as contribuições do trabalho e indicando possibilidades para pesquisas futuras.

2 FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

Neste capítulo, são apresentados os fundamentos teóricos essenciais para o entendimento da pesquisa. Inicia-se com uma contextualização do *Big Data* e dos *Data Warehouses* (DWs) como soluções de suporte à decisão. Na sequência, abordam-se as tecnologias de *streaming* e o paradigma *Near Real-Time* (NRT). O fornecimento de informações precisas sobre o comportamento humano é uma das tarefas centrais da computação ubíqua aplicada à saúde, sendo abordado por duas vertentes: sensores externos e sensores vestíveis (9). A integração desses dispositivos em redes de área corporal (*Body Area Networks* - BANs) permite transformar dados brutos em inteligência contextual para o ecossistema de *e-health* (6), fundamentando a abordagem prática deste estudo. Por fim, apresenta-se uma revisão de trabalhos relacionados que situam a proposta frente ao estado da arte.

2.1 Big Data

A expansão do volume de dados redefiniu a utilização da informação no cenário global. Ao longo das últimas duas décadas de intensa digitalização, emergiu um imenso volume de dados, notável por suas características de Volume, Velocidade, Variedade e Valor, popularmente conhecidas como os "4 Vs do *Big Data*".

Volume refere-se à quantidade massiva de dados, característica fundamental de todo o conceito. O Volume, no contexto do *Big Data*, pode se referir a *terabytes* ou *petabytes* de dados, provenientes de diversas fontes, como os bilhões de curtidas e comentários gerados por uma única plataforma, como o *Facebook* (21).

Velocidade envolve a alta taxa de geração, processamento e análise desses dados, um fenômeno impulsionado pelo avanço da internet e pela capacidade de sistemas inteligentes gerarem informações sem intervenção humana. Esse atributo é fundamental, pois um atraso na transmissão e no processamento pode impactar exponencialmente os sistemas a jusante e comprometer o resultado da análise (21).

Variedade caracteriza a diversidade de tipos de dados, que podem ser estruturados (como em bancos de dados), semiestruturados (como arquivos *.CSV*) ou completamente não estruturados (como fotos e vídeos). Dados não estruturados representam uma parcela significativa (estima-se que seja mais de 80%) e complexa de ser trabalhada, mas são uma presença constante nas bases de dados (21).

Valor, por sua vez, é a característica mais importante, pois sem ela, os demais Vs são considerados "sem sentido"(21). O Valor é o significado obtido do grande volume de dados após a análise. A extração de valor é o que justifica o esforço de processamento e permite que os gestores tomem decisões para impulsionar o negócio, validando a premissa de que "os dados são o novo petróleo"(21).

Dados gerados por transações comerciais, conteúdos de usuários *on-line*, dispositivos de IoT e sensores exemplificam essa diversidade e o rápido ritmo de geração. Conforme discutido por (1), a gestão de *Big Data* exige abordagens complexas e escaláveis, que transcendam o processamento em *batches* e incorporem o processamento em *streaming* (que processa dados em tempo real a partir de modificações pontuais), sendo esta uma perspectiva central para o presente trabalho.

2.2 *Data Warehouse*

Com o surgimento de diversos sistemas em inúmeros setores da sociedade e a consequente expansão do volume de dados, os *Data Warehouses* (DW) consolidaram-se como uma solução integradora de grande interesse para as organizações (2). O DW trabalha com dados não voláteis, que não são alterados no seu modelo estrutural, e tem como propósito fundamental auxiliar na tomada de decisões estratégicas.

Para integrar essas informações, utiliza-se o processo de ETL (*Extraction, Transformation and Load*), que consiste nas seguintes etapas:

- **Extração:** coleta dos dados de diversas fontes, como sistemas ERP, *sites* e outros bancos de dados;
- **Transformação:** limpeza, tratamento e uniformização dos dados conforme as regras de negócio;
- **Carga:** armazenamento dos dados transformados em um banco de dados centralizado (relacional, não relacional ou multidimensional).

Esse fluxo serve como fonte primária para análises e relatórios que geram conhecimento e *insights* estratégicos para a organização.

2.2.1 NRT: *Near Real Time - Próximo do Tempo Real*

A partir da disseminação dos *Data Warehouses* (DWs), as organizações, antes acostumadas a sistemas de bancos de dados relacionais para operações cotidianas, passaram a enxergar o DW como uma ferramenta estratégica indispensável. Apesar de sua implementação ser intrinsecamente complexa e custosa, a capacidade do DW de consolidar o grande volume de dados em indicadores e métricas relevantes para o negócio justificou amplamente o investimento, transformando dados brutos em conhecimento acionável para a tomada de decisão (3).

Contudo, o advento do *Big Data* e a crescente demanda por posturas estratégicas baseadas em dados em tempo hábil impulsionaram uma nova fronteira: a necessidade do processamento de dados em tempo quase real (*Near Real-Time* - NRT). Essa demanda,

uma consequência direta dos avanços tecnológicos, refletiu a busca por uma inteligência de negócios mais ágil, em que a latência dos DWs tradicionais, baseados em processos de lote (*batch*), tornou-se um gargalo significativo (4). Em sistemas tradicionais, a atualização do *data warehouse* é tipicamente realizada em janelas de tempo pré-definidas, como durante a noite, para evitar sobrecarregar os sistemas de origem (14).

Para superar essas limitações, a evolução dos processos de *Extract, Transform, Load* (ETL) para modelos que minimizam a latência, como o ETL NRT, tornou-se fundamental (14, 15). Diferentemente do modelo de lote, o ETL NRT opera em um ciclo contínuo, em que a extração, transformação e carga de dados ocorrem de forma incremental e quase instantânea. Essa abordagem baseia-se em mecanismos que propagam continuamente as mudanças das fontes de dados para o *data warehouse*, em vez de mover grandes volumes de uma única vez (14). O objetivo principal é reduzir a latência e fornecer dados mais atualizados aos usuários finais, garantindo a sua "última versão" (4).

Por exemplo, em uma aplicação que monitora dados de redes móveis, em que bilhões de registros são gerados diariamente, o ETL NRT permite que a gerência acesse relatórios com a situação mais atualizada, em vez de esperar por um relatório defasado de 24 horas, o que permite uma resposta muito mais ágil a eventos importantes (14). Essa abordagem contínua e incremental permite que os dados sejam disponibilizados para análise quase imediatamente, aproximando-se do conceito de tempo real prático (13).

A viabilidade técnica de ambientes NRT em arquiteturas de DW tradicionais, contudo, impõe desafios significativos, especialmente quanto à concorrência entre as cargas de dados e as consultas analíticas. Enquanto processos de ETL convencionais operam em janelas *off-line* prolongadas, a exigência por dados atualizados demanda ciclos de carga frequentes, os quais podem degradar o desempenho das consultas em cerca de 44%, quando executados de forma simultânea (23).

Uma estratégia eficaz para mitigar esse gargalo é o particionamento das tabelas Fato por intervalos de tempo. Essa técnica permite a manutenção e recriação de índices de forma "local", restringindo o processamento apenas às partições mais recentes onde os novos dados são inseridos. Tal abordagem possibilita janelas de atualização extremamente curtas — entre 10 e 20 segundos — sem comprometer o desempenho analítico global do sistema (23).

Para além da infraestrutura de armazenamento, a eficiência da atualização em DWs NRT reside na otimização das operações de junção entre fluxos de *streaming* e dados históricos (*stream-relation join*). Na literatura, algoritmos robustos, como o PCSRJ (*Parallelised Cache-based Stream Relation Join*), propõem o uso de paralelismo e *multi-threading* para executar simultaneamente as fases de sondagem de fluxo e de disco, otimizando o uso da CPU e reduzindo a latência de transformação (24). Adicionalmente, o desempenho do *pipeline* pode ser elevado ao considerar a distribuição enviesada (*skewed*) dos dados,

mantendo em *cache* os registros identificados como frequentes ("quentes") e utilizando estruturas de dados otimizadas para maximizar o volume de registros processados por unidade de tempo (25).

2.3 E-Health

A gestão da saúde tem passado por uma transformação global nos últimos anos, evoluindo de um modelo tradicional focado na doença para um modelo centrado no paciente (*patient-centered model*), alinhado à busca por uma prestação de cuidados de saúde baseada em valor (*value-based healthcare*) (26). Para atender aos requisitos desse novo paradigma e fornecer um cuidado eficaz e de qualidade, são indispensáveis a gestão e a análise do *Big Data* em saúde. Esse cenário demonstra a crescente relevância da *e-health* (saúde eletrônica) e de soluções tecnológicas para aprimorar a comunicação e a qualidade dos cuidados prestados.

O conceito de *Big Data* na área da saúde transcende o mero volume de informações; reside, primordialmente, no processo de criação de valor a partir desses dados (26). No setor de saúde, os fluxos de *Big Data* são notavelmente heterogêneos e diversos. Além de dados clínicos tradicionais, incluem-se informações necessárias para sistemas de monitoramento, como dados biométricos, provenientes de diversos dispositivos que monitoram métricas como peso e pressão, e dados fisiológicos de monitoramento contínuo (26). Ademais, englobam dados gerados pelos próprios pacientes e informações de telemedicina, incluindo dados de sensores e de equipamentos médicos. A complexidade reside na sua diversidade, no volume e na velocidade, o que dificulta a aplicação de métodos tradicionais no gerenciamento de dados não estruturados e exige ferramentas analíticas e tecnologias avançadas.

A implementação da Análise de *Big Data* (*Big Data Analytics* - BDA) no setor de saúde visa, fundamentalmente, melhorar a qualidade dos cuidados, salvar vidas e reduzir custos (26). O potencial reside em extrair regras, padrões e tendências dos dados, possibilitando diagnósticos e tratamentos mais personalizados e, em um contexto clínico, permitindo previsões de longo prazo e o uso de alertas em tempo real para cuidados imediatos. Essa capacidade é importante para o monitoramento de pacientes em casa e na prevenção de hospitalizações desnecessárias (26). Através de técnicas de mineração de dados e análise de processamento em alta velocidade, a BDA capacita as organizações a obter *insights* que apoiam a tomada de decisão baseada em dados (*data-driven decisions*).

Dessa forma, o resultado final do processamento de dados com BDA é uma *data storytelling* apropriada, que contribui para a tomada de decisões com menor risco e maior suporte informacional. O uso de um *pipeline* de *streaming* de dados e da visualização em plataformas como o *Grafana* em projetos de monitoramento, como o desenvolvido neste trabalho, insere-se nessa necessidade de fornecer informações confiáveis e em tempo hábil para a tomada de decisão em ambientes de Vida Assistida (AAL) (26).

2.3.1 Sensores Vestíveis

Diferente das soluções ambientais, que podem sofrer limitações de alcance ou custos elevados de instalação, os sensores vestíveis baseados em acelerômetros e giroscópios permitem o monitoramento ininterrupto da cinemática corporal (9). Dispositivos posicionados em pontos estratégicos ou integrados em *smartphones* garantem uma captura fiel da orientação e aceleração do indivíduo, facilitando a diferenciação entre atividades rotineiras e eventos críticos, como quedas acidentais (7). Esses sensores são fundamentais para o monitoramento assistido, pois preservam a mobilidade do idoso enquanto coletam dados de alta fidelidade para análise funcional (10).

Apesar das vantagens, a implementação de HAR com tecnologias vestíveis enfrenta gargalos técnicos relacionados ao poder de processamento nos dispositivos móveis (6). Sinais brutos de acelerometria possuem pouca utilidade diagnóstica se não forem submetidos a métodos eficientes de extração de características em tempo real (9). Nesse sentido, a utilização de um *pipeline* de *streaming* conectado a um *Data Warehouse* NRT permite que o peso computacional da análise seja transferido para uma infraestrutura escalável. Essa arquitetura garante que a **recência dos dados** (*data recency*) seja mantida para o monitoramento clínico contínuo, permitindo tanto a resposta imediata a emergências quanto a análise histórica de longo prazo sobre a evolução da saúde física do indivíduo.

A eficácia do monitoramento em sistemas baseados em *streaming* reside na capacidade de lidar com a natureza contínua e infinita dos fluxos de dados, naturalmente indexados pela dimensão temporal. Para viabilizar o reconhecimento de atividades (HAR), a literatura propõe uma "versão relaxada" do problema original, que consiste na segmentação das séries temporais em janelas de tempo (*sliding windows*) de comprimento fixo (9). Essa técnica é essencial para a extração de características estatísticas e estruturais, permitindo converter a telemetria bruta em métricas sobre a capacidade funcional do idoso (7). No contexto de um *Data Warehouse* NRT, a configuração dessas janelas é um fator crítico, pois impacta diretamente o equilíbrio entre a carga computacional e a latência da resposta. Ao consolidar esses resultados em um ambiente analítico de baixa latência, o sistema gera uma *data storytelling* eficiente, fornecendo aos cuidadores um suporte informacional robusto para a tomada de decisões táticas e a detecção precoce de declínios físicos (6).

2.4 Trabalhos Relacionados

Esta seção expõe trabalhos do estado da arte relacionados ao *streaming* de dados quase em tempo real em grande volume, em paralelo ao que propõe o presente estudo. A revisão almeja demonstrar diferentes formas de monitoramento ágil e a importância desse paradigma no contexto da saúde digital (*e-health*) e do acompanhamento remoto de indivíduos.

A literatura técnica apresenta diversas abordagens para o monitoramento de idosos: Moreira et al. (7) e Mendes et al. (10) propuseram o desenvolvimento de protótipos vestíveis de baixo custo para detecção de quedas e alertas automatizados. Paralelamente, Lara e Labrador (9, 6) focam na otimização de bibliotecas para o reconhecimento de atividades em dispositivos móveis dentro de redes BAN. No entanto, enquanto os trabalhos citados priorizam a eficácia do dispositivo ou do algoritmo de classificação isoladamente, esta pesquisa se diferencia ao focar na infraestrutura analítica subsequente (DW NRT), demonstrando como o processamento de fluxo e o armazenamento de séries temporais podem reduzir a latência informacional no ecossistema de *e-health*, garantindo a recência dos dados necessária para o suporte à decisão.

Complementando a etapa de aquisição de dados, o trabalho de Madureira (20) explora o uso de sensores integrados em *smartphones* e em dispositivos externos para o monitoramento contínuo da mobilidade. A pesquisa investiga a otimização de frequências de amostragem em acelerômetros e giroscópios, além de adaptar dispositivos vestíveis via **Bluetooth Low Energy** (BLE) para garantir a coleta ininterrupta. Enquanto Madureira (20) concentra-se na otimização da captura diretamente no *hardware* e no sistema móvel, este estudo se diferencia ao focar na infraestrutura de processamento e armazenamento em larga escala, utilizando tecnologias como *Apache Kafka* e *Flink* para construir o armazém de dados analítico.

Avançando para o processamento desses fluxos, Larcher *et al.* (12) propõem um *framework* baseado em eventos voltado para a detecção de padrões biométricos e motores incomuns. Sua arquitetura fundamenta-se no paradigma *Fog-Cloud* e na *Arquitetura Lambda*, utilizando *Apache Kafka* e *Apache Flink* para processar dados de sensores e identificar mudanças na rotina do usuário por meio de *Machine Learning*. Embora compartilhe a base tecnológica de processamento de fluxo, o atual trabalho se distancia ao priorizar a construção de um *pipeline* analítico para um *Data Warehouse*, focando na modelagem dimensional, na persistência otimizada de séries temporais via *TimescaleDB* e na visualização de métricas de acelerometria.

Por fim, no que tange ao estado da arte em arquiteturas de dados, Vilela *et al.* (19) propõem a *Data Magnet*, uma arquitetura voltada para processos de ETL em tempo real fundamentada na reatividade. Validada originalmente no domínio da agricultura digital, a solução utiliza um sistema de publicação/assinatura para coletar dados de fontes heterogêneas. Os autores ressaltam que referências clássicas, como Vassiliadis (14), permanecem consolidadas como pilares teóricos do ETL *Near Real-Time*. Enquanto a *Data Magnet* foca no desacoplamento da extração reativa, este projeto se diferencia ao aplicar o processamento em fluxo contínuo para realizar análises complexas e o janelamento temporal especificamente no monitoramento inercial de idosos, unindo a engenharia de dados à análise funcional da saúde.

3 MATERIAIS E MÉTODOS

Este trabalho propõe um *framework* para o processamento de dados próximo ao tempo real (*Near Real-Time* - NRT) em um *Data Warehouse* (DW), utilizando dados de sensores de Ambientes de Vida Assistida. A metodologia adotou e adaptou conceitos de arquiteturas consolidadas para processamento de *Big Data* e *streaming* de dados, visando viabilizar análises ágeis e a geração de relatórios a partir de um fluxo contínuo de informações.

3.1 Arquitetura

O *framework* proposto inspirou-se fundamentalmente na arquitetura de (16), que visava a integração de dados de sensores AAL de *stream*, e históricos climáticos, em lote, para a consolidação das informações, utilizando SQL como linguagem padrão. De forma análoga, empregou-se a *Arquitetura Lambda*, que se mostrou adequada para combinar o processamento em lote (*batch*) de dados históricos, que podem ser integrados de bases de dados diferentes.

Entretanto, no presente trabalho, tal integração foi realizada com uma porção dos dados selecionados, como uma legenda para as ações que foram representadas no conjunto de dados. O processamento em lote foi combinado com o processamento de *stream* de dados em tempo real, tal camada processa os dados no exato momento em que chegam para fornecer respostas imediatas. Focando na minimização da latência, utilizando funções de baixa complexidade computacional.

A última camada da Arquitetura Lambda, *Serving Layer*, camada de serviço, é integração das informações processadas nas etapas de *batch* e *stream layers*, permitindo a visualização com o contexto completo do que se propõe enxergar através dos dados em tempo hábil.

Esta arquitetura foi particularmente apropriada para cenários de AAL, por sua capacidade de lidar com grandes volumes de dados em tempo real, conferindo ao sistema flexibilidade, escalabilidade, agilidade e alta disponibilidade (17). A abordagem da *Arquitetura Lambda* buscou equilibrar a baixa latência e o alto *throughput* do processamento de *stream* de dados com a abrangência e precisão do processamento em lote, garantindo assim uma visão completa e atualizada dos dados para análises.

Diferentemente do foco principal de (16), que residia no desenvolvimento de um "Processador de Consultas Contínuas" centralizado, a presente pesquisa concentrou-se na construção e integração do *pipeline* completo de dados. Este *pipeline* abrange desde a coleta de dados dos sensores vestíveis até a disponibilização das informações para monitoramento em tempo real por meio de *dashboards*. O objetivo foi demonstrar a viabilidade de um DW NRT funcional para este domínio específico, otimizando o fluxo de transporte,

transformação e armazenamento das informações.

O *pipeline* de dados proposto para o *Data Warehouse Near Real-Time* (DW NRT) foi dividido em etapas lógicas, a fim de garantir o fluxo contínuo e a disponibilidade das informações para análise. Cada etapa foi cuidadosamente selecionada com base em sua eficácia comprovada no processamento de grandes volumes de dados e na gestão de latência, conforme detalhado a seguir.

3.1.1 Coleta de Dados de Sensores AAL

Esta etapa inicial focou na aquisição dos dados brutos gerados pelos sensores em ambientes de AAL, com enfoque nos vestíveis. Consideraram-se todos os tipos de dados típicos desses ambientes, como eventos de movimento, presença, *status* de dispositivos, entre outros. A robustez da coleta é essencial para o sucesso do *pipeline*, pois garante que as informações necessárias sejam capturadas e transmitidas às etapas subsequentes até a visualização final.

3.1.2 Ingestão de Dados com Apache Kafka

Os dados coletados pelos sensores foram ingeridos no *Apache Kafka*. Ele atuou como um *broker* de mensagens distribuído e um cache baseado em disco, garantindo a alta vazão, baixa latência, durabilidade e escalabilidade necessárias para lidar com o volume de dados de *stream* (18).

O Kafka operou sob o modelo *publish-subscribe*, o que viabilizou o desacoplamento entre produtores (sensores) e consumidores (processamento). Além disso, como uma popular plataforma de streaming de eventos, o Kafka foi utilizado para assegurar a integridade e a sequência dos dados em tempo real, operando de forma escalável, elástica e tolerante a falhas (17). Essas características são essenciais para a ingestão de dados em tempo real.

3.2 Processamento de Streaming com Apache Flink (*PyFlink*)

O *Apache Flink*, por meio de sua API em *Python* (**PyFlink**), foi a ferramenta central para o processamento de dados em *stream*. *Jobs* desenvolvidos em **PyFlink** consumiram dados do *Kafka*, realizando operações de transformação, filtragem, agregação (incluindo janelamento) e enriquecimento em tempo quase real. O **Flink SQL** foi utilizado para facilitar as operações de processamento e a integração com dados históricos, consolidando a lógica de negócio diretamente no fluxo de dados.

A escolha do *Apache Flink* justificou-se pela sua capacidade de processamento de *streaming* em baixa latência e alta vazão, sendo particularmente eficaz no tratamento de dados fora de ordem por meio do uso de *watermarks*. Este recurso é utilizado para a

agregação de dados em janelas de tempo, o cálculo de métricas (como médias de aceleração dos sensores) e a extração, em tempo real, da chave de dimensão (`activity_type_key`, (detalhada na seção de Discussão), transformando dados brutos em informações acionáveis. A possibilidade de desenvolvimento em *Python* com *PyFlink* permitiu a prototipagem e a integração ágil com o motor do *Flink*.

Para garantir a robustez e a consistência dos dados em um ambiente de *streaming* contínuo, o *Flink* emprega um algoritmo de *snapshotting* global (ou *checkpointing*). Este mecanismo é fundamental para a funcionalidade de processamento *exactly-once* (exatamente uma vez), garantindo que, em caso de falha de um operador ou do sistema, o *job* possa ser reiniciado a partir de um ponto consistente e conhecido (22).

O processo funciona por meio do envio de **Checkpoint Barriers** da fonte de dados (*Kafka*) que percorrem o *pipeline* de processamento, forçando cada operador a salvar seu estado interno (como o progresso de janelas de agregação e contagens). Ao realizar esse procedimento de forma assíncrona e distribuída, o *Flink* evita a perda ou duplicação de eventos, o que é crítico para manter a integridade das métricas calculadas, como a média da aceleração do sensor. Além disso, a capacidade de gerenciar o estado das operações em diferentes *backends* e a funcionalidade de **queryable state** aprimoraram a tolerância a falhas e a acessibilidade às janelas de tempo de evento, características essenciais para a confiabilidade do DW NRT (22).

3.3 Armazenamento do Data Warehouse (*TimescaleDB/PostgreSQL*)

Os dados processados pelo *Flink* foram armazenados em um banco de dados *PostgreSQL*, utilizando a extensão *TimescaleDB*. Esta escolha foi fundamental por se tratar de um Sistema de Gerenciamento de Banco de Dados Relacional (SGBD-R) otimizado para séries temporais, mantendo a compatibilidade com *SQL* ao estender as funcionalidades do *PostgreSQL* (27).

O *TimescaleDB* otimiza o armazenamento e a consulta de dados de séries temporais (típicos de sensores) por meio do uso de **hypertables** e do particionamento automático de dados, oferecendo desempenho superior para consultas analíticas e históricas (27). A escolha desse banco de dados deu-se por sua capacidade de lidar eficientemente com grandes volumes de dados temporais e por sua compatibilidade nativa com ferramentas de visualização padrão, como o *Grafana*, proporcionando armazenamento eficiente e consulta rápida dos dados de atividade processados, seguindo um modelo dimensional normalizado.

No aspecto da modelagem e persistência, foi implementado um modelo de dados dimensional, composto por uma tabela Fato (*fact_activity_measures*) e uma tabela de dimensão (*dim_activity*), o modelo está representado na Figura 1.

A tabela de dimensão (*dim_activity*) foi criada e populada via script inicial para

armazenar os atributos descritivos das atividades, permitindo a desnormalização das métricas das atividade monitoradas (Tabela 1) no momento da consulta. Seus atributos são:

- **activity_type_key**: Chave Primária (PK) da dimensão, gerada de forma serial (iniciando do número 1).
- **activity_label_id**: Identificador numérico original da atividade no dataset (ID do rótulo da atividade desenvolvida no período).
- **activity_name**: Nome descritivo da atividade (e.g., 'Walking', 'Sitting'), conforme a tabela de mapeamento.

Tabela 1 – Dados da dim_activities

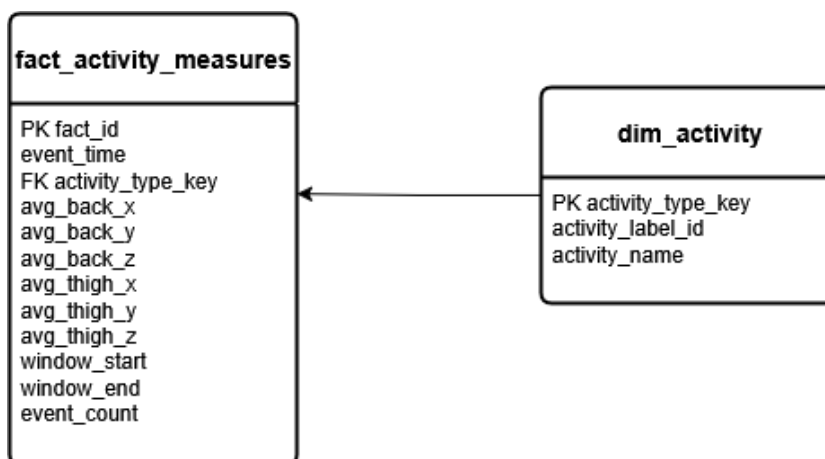
<i>activity_label_id</i>	<i>activity_name</i>
1	walking
3	shuffling
4	stairs (ascending)
5	stairs (descending)
6	standing
7	sitting
8	lying

A tabela Fato (*fact_activity_measures*) armazena as métricas agregadas do processamento em *streaming* realizado pelo Flink, mantendo a granularidade das janelas de tempo. Os atributos da tabela de fatos são:

- **fact_id**: Chave primária de controle da tabela de fatos (PK).
- **activity_type_key**: Chave Estrangeira (FK) que referencia *dim_activity* (*activity_type_key*), estabelecendo a relação entre a métrica agregada e o tipo de atividade a que se refere.
- **window_start**: Marca de tempo que define o início da janela de tempo na qual as métricas foram calculadas.
- **window_end**: Marca de tempo que define o fim da janela de tempo.
- **event_count**: Contagem total de eventos de sensor brutos que ocorreram dentro da janela de tempo definida, indicando o volume de dados processados.
- **avg_back_x**, **avg_back_y**, **avg_back_z**: Média da aceleração no eixo X, Y e Z do sensor localizado nas costas (*back*) do indivíduo, calculada dentro da janela de tempo.

- *avg_thigh_x*, *avg_thigh_y*, *avg_thigh_z*: Média da aceleração no eixo X, Y e Z do sensor localizado na **coxa** (*thigh*) do indivíduo, calculada dentro da janela de tempo.

Figura 1 - Modelo de Dados



Fonte: Autor

3.4 Monitoramento e Visualização com Grafana

Para a camada de apresentação e monitoramento, o *Grafana* foi empregado. *Grafana* é uma ferramenta de análise e monitoramento que se destacou por sua capacidade de se conectar a diversas fontes de dados, incluindo bancos de dados como o *TimescaleDB* (29). Com o suporte do *Grafana*, os usuários podem escrever consultas, visualizar dados por meio de *dashboards* interativos e personalizáveis e até mesmo criar alertas com base nos dados armazenados. Sua flexibilidade permite a exploração e o compartilhamento de dados, facilitando a tomada de decisões e a compreensão de informações relevantes, auxiliando na identificação da causa raiz de incidentes (29).

A escolha do Grafana como ferramenta de visualização e monitoramento de código aberto, com distribuição gratuita, flexível e amplamente utilizada, foi natural pela sua capacidade de criar dashboards interativos e personalizáveis para visualizar as métricas de atividade em tempo real. Além de suas funcionalidades essenciais de visualização, o Grafana oferece recursos robustos para a organização e gestão de dashboards. Ele permite que os dashboards sejam organizados em pastas com diferentes permissões de acesso e edição, adaptando-se a diversos perfis de usuário e às necessidades de operação (30). Essa capacidade é fundamental para organizar a apresentação dos dados para múltiplos públicos e garantir a governança do sistema ao longo do tempo. As capacidades do Grafana de criar visualizações dinâmicas, identificar relações entre os dados e auxiliar na análise da causa raiz de incidentes tornaram-no uma ferramenta ideal para monitorar, em tempo real, o comportamento e o bem-estar em ambientes AAL.

3.5 Infraestrutura e Ambiente de Desenvolvimento

Para garantir a consistência, portabilidade e replicabilidade do ambiente de desenvolvimento e de execução do framework proposto, a tecnologia de contêineres *Docker* foi utilizada. Cada componente do pipeline de dados — incluindo *Apache Kafka*, *Apache Flink*, *TimescaleDB/PostgreSQL* e *Grafana* — foi configurado para operar em contêineres *Docker* isolados.

A orquestração desses serviços foi realizada por meio do *Docker Compose*, permitindo que todo o ambiente fosse inicializado, gerenciado e desativado com facilidade por meio de um único arquivo de configuração. Essa abordagem minimizou a complexidade de instalação e a compatibilidade de dependências, facilitando o desenvolvimento e as futuras demonstrações do sistema. A escolha do *Docker Compose* simplificou a implantação e a interconexão de todos os componentes do pipeline, garantindo um ambiente de desenvolvimento e de teste consistente e isolado. Além disso, garante a portabilidade e a replicabilidade do ambiente, superando as limitações iniciais de *hardware* encontradas no dispositivo local.

O desenvolvimento de *scripts* e *jobs* específicos para o processamento de dados foi feito em *Python*, executados nos respectivos contêineres do *Flink*.

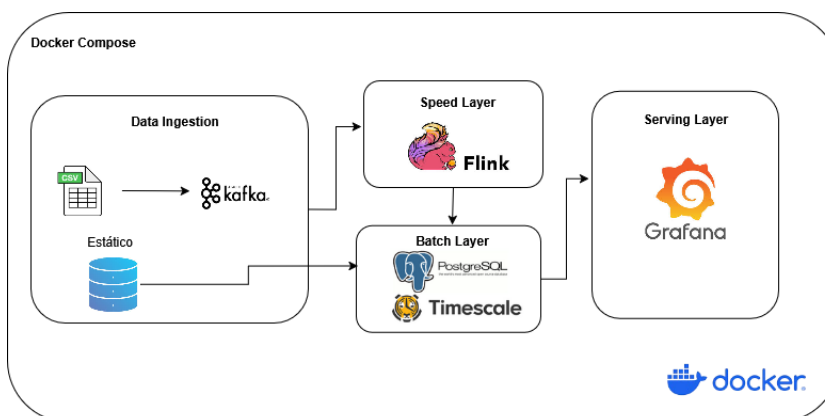
O *framework* foi implementado utilizando o *dataset* denominado HAR70+, disponível no *Kaggle* (32) e descrito em detalhes no artigo de Ustad (31). Este *dataset* representa dados de sensores para o monitoramento de atividades de idosos voluntários.

A representação visual da arquitetura de desenvolvimento é apresentada na Figura 2. Como dito anteriormente, a arquitetura inspira-se na *Arquitetura Lambda*, utilizando o modelo modular para separar as responsabilidades de ingestão, de processamento em tempo real e de persistência.

A estrutura se divide em três camadas principais, responsáveis pela execução do *pipeline* de dados:

- **Data Ingestion:** O *Apache Kafka* atua como o *bus* de mensagens, recebendo o fluxo de dados dos sensores (simulados via *.CSV*) e garantindo a durabilidade e o desacoplamento dos componentes.
- **Speed Layer (Apache Flink):** Esta camada executa o processamento de *stream* de baixa latência. O *Apache Flink* consome os dados do *Kafka*, realiza o enriquecimento, o cálculo de métricas agregadas por janelas de tempo e insere os resultados na camada de persistência.
- **Batch Layer (TimescaleDB):** Utilizado para persistência e armazenamento de longo prazo dos dados processados, o *TimescaleDB* (extensão do *PostgreSQL*) armazena os dados de forma otimizada para séries temporais.

Figura 2 - Arquitetura de Desenvolvimento Modular do Pipeline de Streaming de Dados (DW NRT) para AAL



Fonte: Autor

- **Serving Layer (Grafana):** O *Grafana* conecta-se ao *TimescaleDB* para consultar os dados processados e disponibilizar as informações para o usuário final por meio de *dashboards* interativos NRT, caracterizando a interface de monitoramento e análise.

Para a visualização com *Grafana*, a primeira etapa consistiu na configuração da fonte de dados *PostgreSQL/TimescaleDB*, estabelecendo a conexão direta com o banco de dados. Foram realizados ajustes de conectividade e segurança, como a desabilitação do SSL/TLS (adequada para o ambiente de desenvolvimento local) e a especificação da versão correta do *PostgreSQL* para assegurar a compatibilidade.

A granularidade de tempo no *Grafana* (**Min time interval**) foi ajustada para 10 segundos, correspondendo exatamente à configuração das janelas de agregação processadas pelo *Flink*. Esse alinhamento garantiu a exibição coerente dos dados nos painéis. Por fim, a construção de *queries* SQL, utilizando a função **time_bucket** e a realização de **JOINS** com a tabela **dim_activities** para recuperar o campo **activity_name**, demonstrou a aplicação prática e eficaz do modelo dimensional normalizado nos *dashboards*.

Em síntese, no contexto da *Arquitetura Lambda*, o *Flink* representa a Camada de Velocidade (*Speed Layer*), fundamental para lidar com fluxo contínuo de dados e minimizar a latência. Esta camada processou eficientemente os dados de entrada em tempo limitado, garantindo que a visão em tempo real fosse atualizada prontamente e complementando a camada *Batch* ao lidar com dados que exigiam respostas rápidas (17). O banco de dados é o repositório central do DW NRT, atuando como o componente *Batch* da arquitetura *Lambda* ao persistir o histórico de dados brutos e enriquecidos. Por fim, a integração dos dados com o *Grafana*, ao se conectar diretamente ao *TimescaleDB*, atua como a camada de serviço (*Serving Layer*) da arquitetura proposta, disponibilizando as informações de

forma acessível e acionável aos usuários finais.

3.6 Configurações do Ambiente de Desenvolvimento

Inicialmente, a rotina de produção e consumo dos dados foi executada em um *Ultrabook Dell XPS L321X*, equipado com 4 GB de RAM, processador *Intel Core i5* de 2ª geração e SSD de 128 GB. Entretanto, tal configuração mostrou-se insuficiente para a execução estável do *Docker* com todos os recursos exigidos pelo *pipeline*, criando um desafio técnico por limitações de *hardware*.

Tal impasse foi solucionado com a transposição do experimento para uma configuração *desktop* munida de processador *Intel Core i3-3220* (3.30 GHz), 8 GB de RAM e 128 GB de SSD. Embora o armazenamento possuísse a mesma capacidade da máquina anterior, o fator determinante foi a ampliação da memória principal (RAM), que passou de 4 GB para 8 GB. Este incremento permitiu o particionamento adequado dos recursos entre os serviços alocados na imagem *Docker* construída para a execução da presente metodologia.

A partir desse ajuste na infraestrutura, tornou-se possível avançar com o experimento e validar o processamento contínuo deste grande fluxo de dados.

3.7 Implementação

Esta seção detalha os aspectos práticos da implementação da solução, abordando informações pertinentes ao funcionamento do *pipeline* proposto.

A execução do *pipeline* consistiu em ativar o *container Docker*, onde estão alocados os serviços utilizados nas camadas de ingestão, velocidade e serviços, supracitados, os quais foram instalados utilizando uma imagem, gerando um único ambiente para a integração e versionamento.

Após a constatação do funcionamento de todos os componentes da aplicação, sendo confirmada a saúde do serviço e a estabilidade para realizar cada processo, assegurou-se de que não haveria intercorrências de falhas no momento em que o fluxo está em atividade. O DW é implementado no banco de dados *TimescaleDB*, através de um *script SQL* que contém o **schema** citado na **Modelagem dos Dados**, com as informações estáticas a respeito das atividades e sua descrição presentes. Após a criação do DW, o banco foi conectado ao *Grafana* (acessado pela *web*) para que, a partir do *pipeline*, fossem exibidos os *dashboards* gerados em tempo real, os quais, por motivos de formato de trabalho, são apresentados no formato de instantâneos no capítulo **Resultados**.

Na sequência, o *broker* de mensagens do *Apache Kafka*, utilizando uma interface com o *Python*, foi acionado para produzir mensagens com os dados de sensores, linha a linha, presentes no *dataset* (.CSV). Neste ponto, cada registro é primeiramente lido e desmembrado em seus componentes (campos e valores); este processo de *parsing* permite

que a informação crua seja classificada (como um número, um texto ou um `timestamp`). O registro é então pré-processado: ele é enriquecido com um `timestamp` de evento em tempo real e estruturado em um *payload* JSON que garante a conformidade do esquema com o esperado pelo *Flink*. Este JSON serializado em bytes é transmitido continuamente ao tópico `activity_data_stream`, iniciando assim a ingestão dos dados em *stream*.

A partir do momento em que os dados produzidos pelo *Kafka* se tornam disponíveis, o *Apache Flink* inicia o consumo e processamento contínuo desses dados (*stream processing*). Esse tratamento é realizado por meio de uma função de janelamento (*Windowing*), que agrega os registros em blocos de 10 segundos de tempo de evento. Dentro de cada janela, é calculada a média das coordenadas dos sensores (X, Y, Z), permitindo uma análise agregada das atividades capturadas pelos acelerômetros em intervalos definidos.

Após o tratamento e o janelamento realizados pelo *Flink*, os registros agregados (contendo a média das acelerações) são persistidos no Data Warehouse (DW), especificamente na tabela `fact_activity_measures`. Este armazenamento no DW em tempo quase real possibilita a observação imediata dos resultados do *pipeline*, os quais foram plotados em um gráfico para serem discutidos em detalhes no capítulo **Resultados**.

4 RESULTADOS

Mediante a execução do *pipeline* proposto, o monitoramento das atividades ocorreu em tempo quase real. A utilização do *TimescaleDB* como camada de persistência evitou gargalos de indexação na ingestão dos dados de alta granularidade do *dataset* HAR70+. Com isso, os sinais brutos dos sensores foram processados sem perda de pacotes, gerando indicadores de saúde imediatos.

O volume total de dados é de 1.841.791 eventos de origem, coletados ao longo de 63 dias (cerca de 1.560 horas). A análise quantitativa indica que a estratégia de *roll-up* no *Flink* reduziu o volume total da tabela de fatos para 5.205 registros físicos. Essa proporção de 353 eventos por linha persistida otimiza a infraestrutura e reduz a latência de leitura. Sem essa agregação, a análise clínica de sinais de acelerometria pura seria inviável para um profissional de saúde, devido à densidade dos dados inerciais.

A visualização no *Grafana* permite observar a *data freshness*, superando a latência típica dos modelos *batch*. A inclusão da coluna `data_ingestion_time` foi adotada para diferenciar a latência de origem da latência de processamento, garantindo a rastreabilidade necessária ao cumprimento do *SLA* (*Service Level Agreement*) de tempo real.

A validação deste requisito é apresentada na Figura 3, que exibe a latência, representada como a densidade de ingestão distribuída em uma janela cíclica de 60 segundos. O uso deste mapeamento recorrente permite que todo o tempo de execução seja visualizado em um intervalo fixo, facilitando a identificação de eventuais gargalos periódicos ou padrões de congestionamento no *pipeline*. Com isso, isola-se a variabilidade estatística do atraso (*jitter*) das variações temporais de longa duração, permitindo uma análise precisa da pontualidade e do determinismo do sistema frente aos dados processados.

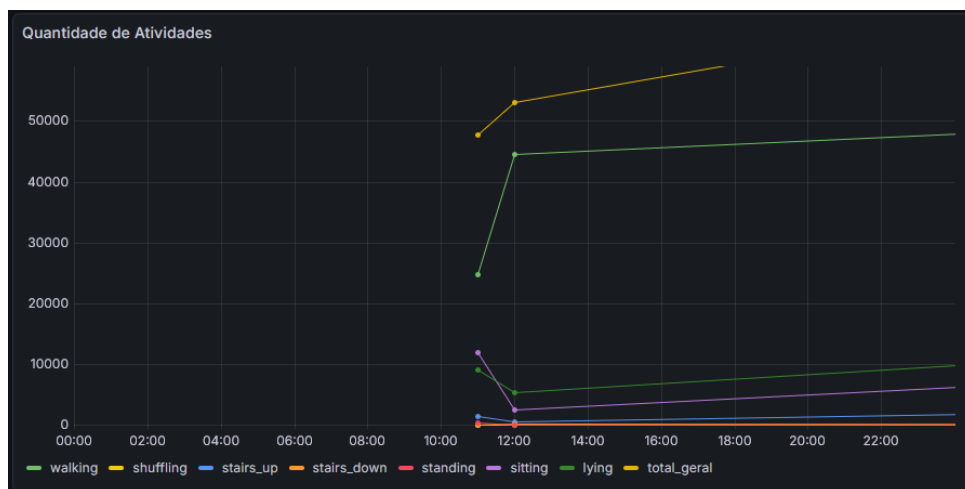
- Figura 3 - Latência de ingestão calculada pela ingestão distribuída intervalo de 60 segundos



Fonte: Autor.

Ao sobrepor múltiplos períodos de um minuto em um único plano visual, torna-se possível identificar padrões repetitivos e atestar a estabilidade do *pipeline*. A distribuição dos pontos ao longo de todo o período analisado demonstra uma latência constante e previsível. A concentração horizontal dos dados na imagem comprova que o sistema mantém o desempenho estável, sem sinais de degradação ou acúmulo de processamento (*backlog*), validando a eficácia da arquitetura para operações em tempo quase real (*near real-time*).

– Figura 4 - Quantidade de atividades registradas por horas na data de 24/03/2021



Fonte: Autor.

– Figura 5 - Quantidade de atividades registradas por horas na data de 24/04/2021



Fonte: Autor.

A análise do comportamento dos dados destaca pontos específicos:

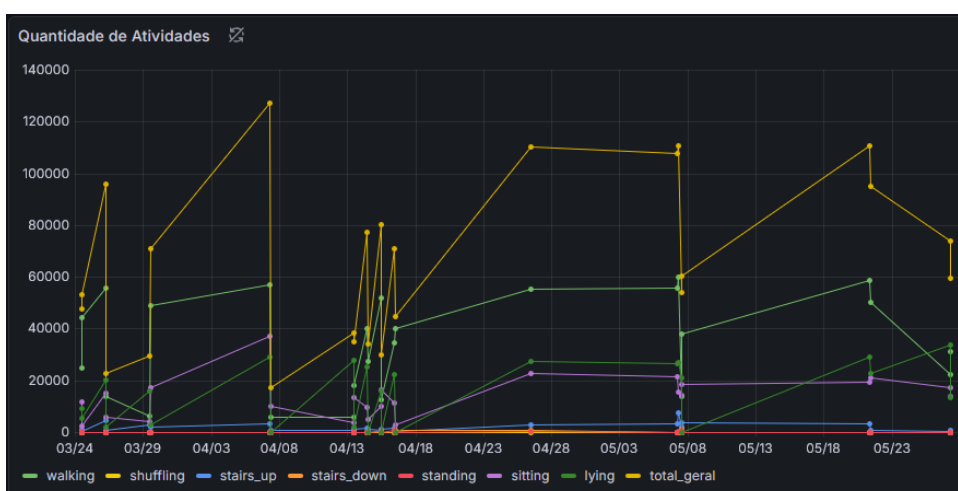
- **Resposta a eventos intensos (Figura 4):** O pico de 50.000 eventos às 11:00 evidencia que o *pipeline* suporta rajadas (*bursts*) de movimentos. Para a gestão

– Figura 6 - Quantidade de atividades registradas por horas na data de 27/05/2021



Fonte: Autor.

– Figura 7 - Quantidade de atividades registradas por horas no período de 24/03/2021 a 27/05/2021



Fonte: Autor.

do cuidado, esse dado identifica os períodos de maior atividade física do idoso, considerando que uma queda deveria ser identificada como um comportamento fora do padrão.

- **Estabilidade operacional (Figura 5):** A progressão linear das atividades em 24/04/2021 evidencia a resiliência do sistema em regime contínuo (24/7). A manutenção desse padrão reflete a autonomia funcional do indivíduo monitorado.
- **Identificação de interrupções (Figura 6):** A queda nos valores às 07:00 de 27/05/2021 demonstra a importância da baixa latência. A percepção imediata desta queda no *dashboard* permite distinguir falhas técnicas de possíveis emergências reais, como quedas ou imobilidade.

- **Tendências históricas (Figura 7):** A base de séries temporais consolidou o histórico de março a maio, permitindo identificar padrões não lineares de comportamento e tendências de longo prazo na mobilidade do idoso.

Em resposta a pergunta central, observou-se uma constância nas atividades, com padrões parecidos, tendo a caminhada como atividade mais praticada, pois, o período observado pelo dataset foi durante a pandemia, no qual, por recomendações de saúde, as pessoas deveriam ficar isoladas em casa, sem saídas desnecessárias. Pelo fato da população idosa se enquadrar nos grupos de risco, a permanência em casa fora percebida nos padrões de atividades.

Dessa forma, o sistema entrega uma ferramenta de engenharia de dados aplicada, reduzindo o tempo de espera típico de processamentos *offline* e garantindo que o dado de saúde mantenha seu valor preventivo.

4.1 Discussão da Implementação

Foi perceptível, o funcionamento da arquitetura proposta, o que suscita a provocação sobre a maneira de escalar o uso de dados vestíveis para mais pessoas, com outros aparatos de mensuração de atividades, elencando como candidatos o smartwatch, o giroscópio entre outros.

Para tal aumento da quantidade de pessoas e dados para o experimento, é necessário compreender as estruturas ou a ausência delas providas dos dados gerados por esses dispositivos de captura, para que o processamento na camada de velocidade seja eficiente para reduzir a latência, um forte objetivo traçado no trabalho, somado à necessidade de verificar o tamanho do intervalo de janelamento, não se limitando a 10 segundos e o aumento da capacidade de memória RAM, para execução do Docker.

Outro ponto, no que tange os dados utilizados, a LGPD não foi uma preocupação direta, pois os dados estão locados no Kaggle que é um acervo de dados públicos para consultas, desafios e promoção de conhecimento nas áreas de Dados e Machine Learning. Contudo, com a expansão do modulo, faz-se notório, estar em conformidade com a legislação vigente para tratamento de dados pessoais, com solicitação e exposição do uso dos dados para pesquisas aos voluntários de experimentações com dados sensíveis, pois, como dados são o novo petróleo, muito valiosos, os dados são pessoais e devem ser utilizados com responsabilidade na área científica.

4.2 Limitações e Desafios de Implementação

Durante o desenvolvimento do experimento foram identificadas limitações de ordem física e logística que exigiram adaptações na infraestrutura e na gestão do ecossistema de ferramentas, conforme descrito a seguir:

4.2.1 Infraestrutura de Hardware e Virtualização

A execução do *pipeline* via *Docker* apresentou desafios significativos em ambientes com recursos limitados. Inicialmente, a rotina de produção e consumo de dados foi executada em um *Ultrabook Dell XPS L321X*, equipado com 4 GB de RAM e um processador *Intel Core i5* de 2ª geração. Entretanto, tal configuração mostrou-se insuficiente para a execução estável do *Docker* com todos os serviços exigidos (*Kafka*, *Flink* e *TimescaleDB*), o que gerou instabilidade devido ao esgotamento de memória.

Este impasse foi solucionado por meio da transposição do experimento para um *desktop* equipado com um processador *Intel Core i3-3220* e 8 GB de RAM. Embora o armazenamento em SSD fosse idêntico ao anterior, o aumento da memória principal foi o fator determinante para o alocamento adequado dos recursos entre os serviços alocados na imagem *Docker*. Tal experiência evidencia que a escalabilidade vertical da memória RAM é um requisito crítico para a estabilidade de arquiteturas de *streaming* em ambientes locais.

4.2.2 Ecossistema de Software e Licenciamento

Para a replicabilidade deste ecossistema, observou-se a necessidade de uma gestão rigorosa das imagens de *containers*. Durante o desenvolvimento, as mudanças de licenciamento nas distribuições do *Kafka* (especificamente as imagens da *Bitnami* após a aquisição pela *Broadcom*) (33) impuseram desafios à portabilidade do projeto.

Para manter a viabilidade da solução em ambientes de código aberto e sem ônus financeiros, foi necessário o uso de versões legadas (*bitnami-legacy*) e uma configuração minuciosa das portas de comunicação. Este cenário ressalta que a manutenção de *pipelines* de dados em *Near Real-Time* depende não apenas da lógica de processamento, mas também da atenção constante à volatilidade das políticas de licenciamento de componentes de terceiros na infraestrutura de *Big Data*.

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

A execução deste trabalho permitiu observar que a principal barreira para o monitoramento de idosos via sensores vestíveis em ambientes de Vida Assistida (AAL) reside na latência do processamento analítico. Ao final deste projeto, foi possível validar um *pipeline* que soluciona o gargalo dos modelos tradicionais de lote, demonstrando que a integração entre *Apache Kafka* e *Apache Flink* é capaz de sustentar o fluxo de mais de 1,8 milhão de registros do *dataset* HAR70+ com a integridade e resiliência necessárias para o processamento de sinais inerciais em tempo real.

O uso do *TimescaleDB* foi o diferencial técnico para garantir que a ingestão contínua do *Flink* não causasse gargalos de indexação. Na prática, a redução do tempo de espera pela informação de um dia inteiro para poucos segundos transforma o monitoramento: o sistema deixa de ser um histórico passivo para se tornar uma ferramenta de acompanhamento em tempo real. Isso permite que uma mudança brusca de comportamento ou uma possível situação de risco seja percebida ainda que haja tempo para intervenção.

O presente trabalho estabelece um fundamento relevante para o *e-health*, ao proporcionar uma infraestrutura que permite que profissionais de saúde monitorem padrões de atividade e identifiquem momentos críticos, maximizando a eficácia das intervenções. Para a replicabilidade deste ecossistema, ressalta-se a necessidade de uma configuração minuciosa da comunicação entre *Kafka* e *Flink*, além da atenção à portabilidade das imagens *Docker* — especialmente diante das recentes mudanças de licenciamento nas distribuições do *Kafka* (como as da *Broadcom*) —, visando manter a viabilidade da solução em ambientes de código aberto e sem ônus financeiros.

Como perspectivas futuras, a arquitetura montada serve de alicerce para a aplicação de modelos de *Machine Learning* diretamente no fluxo de dados (*in-stream*), evoluindo a análise descritiva atual para uma capacidade preditiva de riscos de queda e de outros problemas comportamentais. Conclui-se que a engenharia de dados, quando apoiada em ferramentas de *streaming* e em infraestrutura robustas, é vital para garantir a autonomia e a segurança de idosos em ambientes de vida assistida.

REFERÊNCIAS

- 1 HU, Han et al. Toward scalable systems for big data analytics: A technology tutorial. **IEEE Access**, [S. l.], v. 2, p. 652-687, 2014.
- 2 JAYASHREE, G.; PRIYA, C. Design of visibility for order lifecycle using datawarehouse. **International Journal of Engineering and Advanced Technology**, [S. l.], v. 8, n. 6, p. 4700-4707, 2019.
- 3 OLALERE, Abiodun. Impact of Data Warehouse on Organization Development and Decision making (A Case study of United Bank for Africa and Watchlocker PLC). **International Journal of Research and Scientific Innovation (IJRSI) vol**, [S. l.], v. 10, p. 36-45, 2023.
- 4 BOUAZIZ, Senda; NABLI, Ahlem; GARGOURI, Faiez. From traditional data warehouse to real time data warehouse. In: INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS DESIGN AND APPLICATIONS (ISDA), 16., 2016, Porto, Portugal. **Intelligent Systems Design and Applications: 16th International Conference on Intelligent Systems Design and Applications (ISDA 2016)** held in Porto, Portugal, December 16-18, 2016. Cham: Springer, 2017. p. 467-477.
- 5 KREPS, Gary L.; NEUHAUSER, Linda. New directions in eHealth communication: opportunities and challenges. **Patient Education and Counseling**, [S. l.], v. 78, n. 3, p. 329-336, 2010.
- 6 LARA, Oscar D.; LABRADOR, Miguel A. A mobile platform for real-time human activity recognition. **2012 IEEE Consumer Communications and Networking Conference (CCNC)**, [S. l.], p. 667-671, 2012.
- 7 MOREIRA, Júlia M. P. et al. Elderly Fall Monitoring in Smart Homes Using Wearable Device. **Brazilian Symposium on Multimedia and the Web (WebMedia)**, [S. l.], p. 124-132, 2024.
- 8 YANG, Chuanwang et al. Quantifying the global eco-footprint of wearable healthcare electronics. **Nature**, [S. l.], v. 649, n. 8095, p. 73-82, 2026.
- 9 LARA, Oscar D.; LABRADOR, Miguel A. A survey on human activity recognition using wearable sensors. **IEEE Communications Surveys & Tutorials**, [S. l.], v. 15, n. 3, p. 1192-1209, 2012.
- 10 MENDES, Caroline Mazetto et al. Dispositivo Vestível para Monitoramento de Pessoas Idosas. **Anais do Computer on the Beach**, [S. l.], v. 11, p. 12-13, 2020.
- 11 UDDIN, Md Zia; KHAKSAR, Weria; TORRESEN, Jim. Ambient sensors for elderly care and independent living: a survey. **Sensors**, [S. l.], v. 18, n. 7, p. 2027, 2018.
- 12 LARCHER, Lucas et al. Event-driven framework for detecting unusual patterns in AAL environments. In: IEEE INTERNATIONAL SYMPOSIUM ON COMPUTER-BASED MEDICAL SYSTEMS (CBMS), 33., 2020, [S. l.]. **Anais [...]**. Piscataway: IEEE, 2020. p. 309-314.

- 13 SANTOS, Ricardo Jorge; BERNARDINO, Jorge. Real-time data warehouse loading methodology. In: INTERNATIONAL SYMPOSIUM ON DATABASE ENGINEERING APPLICATIONS, 2008, [S. l.]. **Proceedings of the 2008 International Symposium on Database Engineering Applications**. [S. l.]: [s. n.], 2008. p. 49-58.
- 14 VASSILIADIS, Panos; SIMITSIS, Alkis. Near real time ETL. In: NEW TRENDS IN DATA WAREHOUSING AND DATA ANALYSIS. [S. l.]: Springer, 2008. p. 1-31.
- 15 KAKISH, Kamal; KRAFT, Theresa A. ETL evolution for real-time data warehousing. In: CONFERENCE ON INFORMATION SYSTEMS APPLIED RESEARCH, 2012, [S. l.]. **Proceedings of the Conference on Information Systems Applied Research ISSN**. [S. l.]: [s. n.], 2012. p. 1508.
- 16 AMARÁ, Jefferson et al. Stream and historical data integration using SQL as standard language. In: SIMPÓSIO BRASILEIRO DE BANCO DE DADOS (SBB), 2021, [S. l.]. **Anais [...]**. Porto Alegre: SBC, 2021. p. 193-204
- 17 ALGHAMDI, Rubayyi; BELLAICHE, Martine. An ensemble deep learning based IDS for IoT using Lambda architecture. **Cybersecurity**, [S. l.], v. 6, n. 1, p. 5, 2023.
- 18 KAUR, Jagreet; MANN, Kulwinder Singh. AI based healthcare platform for real time, predictive and prescriptive analytics using reactive programming. **Journal of Physics: Conference Series**, [S. l.], v. 933, n. 1, p. 012010, 2017.
- 19 VILELA, Flávio de Assis et al. A non-intrusive and reactive architecture to support real-time ETL processes in data warehousing environments. **Heliyon**, [S. l.], v. 9, n. 5, e15644, 2023.
- 20 MADUREIRA, João Manuel Cardoso. **Sensors on mobile devices for AAL**. 2013. Dissertação (Mestrado) – Instituto Superior de Engenharia do Porto, Porto, 2013. Disponível em: <https://recipp.ipp.pt/entities/publication/09b7e124-fec8-445b-b885-97488469ed7f>. Acesso em: 26 dez. 2025.
- 21 KESKAR, Vinaya; YADAV, Jyoti; KUMAR, Ajay H. 5V's of Big Data Attributes and their Relevance and Importance across Domains. **International Journal of Innovative Technology and Exploring Engineering**, [S. l.], v. 9, n. 11, p. 154-163, 2020.
- 22 TO, Quoc-Cuong; SOTO, Juan; MARKL, Volker. A survey of state management in big data processing systems. **The VLDB Journal**, [S. l.], v. 27, n. 6, p. 847-872, 2018.
- 23 FERREIRA, Nickerson; MARTINS, Pedro; FURTADO, Pedro. Near real-time with traditional data warehouse architectures: factors and how-to. **Proceedings of the 17th International Database Engineering & Applications Symposium**, [S. l.], p. 68-75, 2013.
- 24 NAEEM, M. Asif et al. Parallelisation of a cache-based stream-relation join for a near-real-time data warehouse. **Electronics**, [S. l.], v. 9, n. 8, p. 1299, 2020.
- 25 HAZRATI, Isa; DANESHPOUR, Negin. Improving Near Real Time Data Warehouse Refreshment. **Signal and Data Processing**, [S. l.], v. 15, n. 2, p. 31-44, 2018.

- 26 BATKO, Kornelia; ŚLEZAK, Andrzej. The use of Big Data Analytics in healthcare. **Journal of Big Data**, [S. l.], v. 9, n. 1, p. 3, 2022.
- 27 ZYMBLER, Mikhail; IVANOVA, Elena. Matrix profile-based approach to industrial sensor data analysis inside RDBMS. **Mathematics**, [S. l.], v. 9, n. 17, p. 2146, 2021.
- 28 KADARINA, TM; PRIAMBODO, R. Monitoring heart rate and SpO2 using Thingsboard IoT platform for mother and child preventive healthcare. **IOP Conference Series: Materials Science and Engineering**, [S. l.], v. 453, n. 1, p. 012028, 2018.
- 29 UDDIN, Sheikh Usman; BAIG, Mirza Jabbar Aziz; IQBAL, Mohammad Tariq. Design and implementation of an open-source SCADA system for a community solar-powered reverse osmosis system. **Sensors**, [S. l.], v. 22, n. 24, p. 9631, 2022.
- 30 BEERMANN, Thomas et al. Implementation of ATLAS Distributed Computing monitoring dashboards using InfluxDB and Grafana. **EPJ Web of Conferences**, [S. l.], v. 245, p. 03031, 2020.
- 31 USTAD, Astrid et al. Validation of an activity type recognition model classifying daily physical behavior in older adults: the HAR70+ model. **Sensors**, [S. l.], v. 23, n. 5, p. 2368, 2023.
- 32 TANWAR, Ansh. **Adult Subjects 70-95 Years Activity Recognition**. [S. l.]: Kaggle, 2021. Disponível em: <https://www.kaggle.com/datasets/anshtanwar/adult-subjects-70-95-years-activity-recognition>. Acesso em: 23 jul. 2025.
- 33 NORTHFLANK. **Bitnami descontinuando imagens gratuitas: etapas de migração e alternativas**. 2024. Disponível em: <https://northflank.com/blog/bitnami-deprecates-free-images-migration-steps-and-alternatives>. Acesso em: 6 jan. 2026.