

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE GESTÃO E NEGÓCIOS
GRADUAÇÃO EM GESTÃO DA INFORMAÇÃO**

JOHN CLEMMER DE MATTOS JÚNIOR

**INTELIGÊNCIA ARTIFICIAL COMO INTERFACE DE GESTÃO:
um modelo de micro-ERP aplicado ao varejo digital**

**UBERLÂNDIA – MG
2026**

JOHN CLEMMER DE MATTOS JÚNIOR

INTELIGÊNCIA ARTIFICIAL COMO INTERFACE DE GESTÃO:
um modelo de micro-ERP aplicado ao varejo digital

Monografia apresentada ao Curso de Graduação em
Gestão da Informação, da Universidade Federal de
Uberlândia, como exigência parcial para a obtenção
do título de Bacharel.

Orientador: Prof. Dr. José Eduardo Ferreira Lopes

UBERLÂNDIA – MG
2026

RESUMO

O objetivo deste relato técnico-tecnológico foi desenvolver e avaliar uma arquitetura de integração em nuvem, orientada a agentes de inteligência artificial, para a automação do fluxo de dados e gestão de estoque em uma microempresa de varejo digital. Como situação-problema, identificou-se que a sincronização manual de estoque entre múltiplos canais de venda gerava um alto risco de ruptura, cancelamento de pedidos e perda de controle do custo real pela ausência de um cálculo automatizado. A solução desenvolvida consistiu na criação de um micro-ERP invisível, estruturado em um banco de dados relacional em nuvem, integrado por meio de interfaces de comunicação e operado inteiramente por uma interface conversacional em um aplicativo de mensagens via inteligência artificial. Como resultados alcançados, obteve-se a redução do tempo de sincronização de horas para poucos segundos, a mitigação do risco de vendas sem saldo físico e a recuperação do controle financeiro por meio da aplicação automática da regra contábil Primeiro que Entra, Primeiro que Sai, além do aumento da segurança dos dados com o uso de criptografia e restrição de acesso.

Palavras-chave: Gestão de estoque. Micro-ERP. Inteligência artificial. Varejo digital. ChatOps.

ABSTRACT

The objective of this technical-technological report was to develop and evaluate a cloud integration architecture, driven by artificial intelligence agents, for the automation of data flow and inventory management in a digital retail microenterprise. As the problem situation, it was identified that the manual synchronization of inventory across multiple sales channels generated a high risk of stockouts, order cancellations, and loss of real cost control due to the absence of automated calculations. The developed solution consisted of creating an invisible micro-ERP, structured on a cloud-based relational database, integrated through communication interfaces, and operated entirely by a conversational interface within a messaging application via artificial intelligence. As for the results achieved, there was a reduction in synchronization time from hours to a few seconds, the mitigation of the risk of sales without physical stock, and the recovery of financial control through the automatic application of the First-In, First-Out (FIFO) accounting rule, in addition to increased data security through the use of cryptography and access restriction.

Keywords: Inventory management. Micro-ERP. Artificial intelligence. Digital retail. ChatOps.

SUMÁRIO

1 INTRODUÇÃO	5
2 FUNDAMENTAÇÃO TEÓRICA.....	7
2.1 Automação e integração de sistemas em nuvem	7
2.2 Inteligência artificial e interfaces conversacionais	8
2.3 Gestão de estoque e estruturação de dados	9
3 CONTEXTO INVESTIGADO E SITUAÇÃO PROBLEMA	10
4 INTERVENÇÃO ADOTADA	12
4.1 Arquitetura tecnológica e modelagem do banco de dados	13
4.2 Integração de APIs e governança na ingestão de dados	14
4.3 Inteligência artificial como interface de gestão	17
5 RESULTADOS ALCANÇADOS	22
6 CONSIDERAÇÕES FINAIS.....	24
REFERÊNCIAS	26

1 INTRODUÇÃO

O crescimento do varejo digital mudou estruturalmente a forma como as empresas operam e comercializam seus produtos (Laudon; Laudon, 2021). No contexto do segmento de *hardware* e componentes eletrônicos, a estratégia multicanal — que envolve o anúncio simultâneo de produtos em plataformas de terceiros e sites próprios — consolidou-se como um modelo operacional fundamental para a expansão de mercado das microempresas. No entanto, essa atuação cria um grande desafio de gestão da informação para o administrador, que consiste em descobrir como garantir que o estoque físico acompanhe as vendas que acontecem em diferentes canais durante as vinte e quatro horas do dia.

Esse é o cenário de uma microempresa brasileira focada na venda de componentes de impressoras 3D e *hardwares* de informática. A loja opera com um catálogo dinâmico e um estoque único que alimenta várias vitrines virtuais. O grande gargalo da operação atual é que a sincronização desse estoque é feita de forma totalmente manual.

Essa dependência de atualização humana gera problemas graves para a gestão operacional (Slack et al., 2015). Quando uma venda ocorre de madrugada ou aos finais de semana, o produto continua disponível nos outros sites até que o administrador faça a baixa manual no dia seguinte. Isso cria um risco constante de ruptura de estoque — ou seja, realizar vendas sem disponibilidade física em estoque —, o que gera cancelamentos, frustra o cliente e impacta negativamente as métricas de desempenho da conta comercial, podendo resultar na perda de selos de qualificação exigidos pelas plataformas, como o programa "Vendedor Indicado" (Shopee, 2026). Além disso, o modelo manual de planilhas adotado consome tempo excessivo e, por não assegurar a rastreabilidade por lote, impede a atualização consistente e precisa do custo real das mercadorias.

Diante desse cenário, a presente investigação buscou responder de que maneira uma arquitetura de integração em nuvem, baseada em agentes de inteligência artificial, pode automatizar o fluxo de dados para mitigar as falhas de sincronização de estoque e recuperar o controle financeiro em uma microempresa de varejo digital multicanal. Para responder a esse questionamento, o objetivo deste relato técnico-tecnológico foi desenvolver e avaliar uma arquitetura de integração em nuvem, orientada a agentes de inteligência artificial, para a automação do fluxo de dados e gestão de estoque em uma microempresa de varejo digital.

O trabalho está organizado em seis seções. Após esta introdução, a Seção 2 apresenta a fundamentação teórica sobre computação em nuvem, inteligência artificial e gestão de estoque. A Seção 3 detalha o contexto investigado e a situação-problema. A Seção 4 descreve a

intervenção adotada, detalhando a arquitetura tecnológica e a inteligência artificial. A Seção 5 apresenta os resultados alcançados com a implementação. Por fim, a Seção 6 traz as considerações finais do projeto.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Automação e integração de sistemas em nuvem

Vender na internet hoje vai muito além de apenas criar um catálogo *online*. Essa prática exige uma infraestrutura capaz de lidar com muitas informações ao mesmo tempo, sem interrupções ou degradação relevante de desempenho (Laudon; Laudon, 2021). É aqui que a Computação em Nuvem (*Cloud Computing*) entra como a base de toda a operação. Em vez de comprar e manter servidores físicos caros localmente, as empresas agora adotam recursos computacionais e serviços provisionados pela internet (Laudon; Laudon, 2021). Essa alternativa pode ser adequada para micro e pequenas empresas, pois permite o uso de tecnologias avançadas e oferece escalabilidade, com o investimento inicial e contínuo condicionado ao modelo de serviço e à tarifação contratada junto ao provedor.

Porém, possuir infraestrutura em nuvem não resolve o gargalo informacional se os sistemas da empresa não se integrarem (Laudon; Laudon, 2021). Para que o estoque do banco de dados particular se comunique com a vitrine do *marketplace*, é necessária uma interface de integração estruturada. É para isso que servem as APIs (*Application Programming Interfaces*). Um dos padrões amplamente adotados na arquitetura de *software web* é o REST (*Representational State Transfer*), que utiliza o próprio protocolo da internet (HTTP) para permitir que sistemas desenvolvidos em linguagens distintas consigam trocar dados de forma leve e rápida (Masse, 2011; Richardson; Ruby, 2007). Na prática, as APIs funcionam como mecanismos de comunicação entre sistemas: quando um produto é atualizado no banco de dados interno, a API aciona o *marketplace* para alterar o anúncio de forma síncrona.

Além de integrar os sistemas, a rotina de gestão necessita organizar os dados brutos oriundos de fontes variadas, como notas fiscais, relatórios e respostas das próprias APIs (Elmasri; Navathe, 2011). Para estruturar e padronizar essas informações, a engenharia de dados utiliza o conceito de ETL (*Extract, Transform, Load* – Extração, Transformação e Carga). Esse processo consiste em extrair os dados de sua origem, transformá-los (limpando inconsistências e padronizando formatos) e, por fim, carregá-los de forma estruturada dentro de um banco de dados central (Kimball; Ross, 2013). Quando o fluxo de ETL é automatizado, o esforço humano em tarefas repetitivas é consideravelmente reduzido nas etapas processadas. Contudo, a qualidade da informação não decorre apenas da automação, mas depende da implementação rigorosa de regras de validação, tratamento de exceções e monitoramento contínuo, controles essenciais para que a gestão tome decisões baseada em dados consistentes e auditáveis.

2.2 Inteligência artificial e interfaces conversacionais

A Inteligência Artificial (IA) consolidou-se como uma disciplina fundamental na ciência da computação, oferecendo ferramentas práticas para a resolução de problemas complexos do dia a dia (Russell; Norvig, 2021). Nos últimos anos, o mercado de tecnologia acompanhou a rápida evolução dos Grandes Modelos de Linguagem (conhecidos pela sigla LLM, do inglês *Large Language Models*). Esses modelos são treinados com grandes volumes de dados textuais e conseguem interpretar intenções, resumir documentos e até gerar código de programação de forma natural, consolidando a inteligência artificial generativa como uma tendência tecnológica estratégica para os atuais modelos de negócios (Gartner, 2023). No contexto de pequenas empresas, isso significa que a inteligência artificial transcende o mero atendimento ao cliente, podendo ser aplicada para interpretar diretrizes gerenciais e automatizar rotinas administrativas.

O avanço tecnológico significativo para a gestão ocorre com a integração desses modelos de linguagem ao conceito de agentes autônomos (Russell; Norvig, 2021). Diferente de um robô de atendimento tradicional, que apenas fornece respostas textuais pré-programadas, um agente autônomo baseado em IA possui a capacidade de acionar ferramentas do sistema. Na prática, o agente interpreta um comando em linguagem natural enviado pelo usuário e o transforma em uma ação real através de chamadas de função, mecanismo também conhecido como *Tool Calling* (Google, 2026). Dessa forma, se o gestor enviar uma instrução como "zere o estoque do produto X", o agente compreende a intenção, conecta-se ao banco de dados e após a requisição de confirmação executa a baixa do produto, operando de fato como um assistente executivo.

Para que essa interação seja ágil e descentralizada, a engenharia de *software* adotou a prática do *ChatOps*. No modelo *ChatOps*, em vez de obrigar a equipe a utilizar interfaces *web* com múltiplos controles de operação para gerenciar a infraestrutura, o controle do sistema é integrado diretamente aos aplicativos de mensagens corporativos ou de uso diário, como o Telegram (Atlassian, 2026). O uso dessas interfaces conversacionais facilita a interação ao aproveitar um ambiente familiar ao usuário. Contudo, essa prática exige que o robô seja projetado respeitando heurísticas de usabilidade, garantindo que o gestor consiga executar tarefas complexas de forma intuitiva e sem falhas operacionais (Sánchez-Adame et al., 2021).

2.3 Gestão de estoque e estruturação de dados

Gerenciar o estoque é uma das tarefas operacionais mais críticas para o comércio. Produtos parados na prateleira representam capital imobilizado, enquanto a falta de mercadoria frustra o cliente e prejudica o fluxo da produção e logística da loja (Slack et al., 2015). Por isso, manter o registro exato do que entra e do que sai deixou de ser uma tarefa de cadernos para se tornar um componente central da gestão da informação no varejo (Laudon; Laudon, 2021). Quando o administrador sabe exatamente a quantidade de itens que tem em mãos, ele consegue evitar compras desnecessárias e focar nas mercadorias que realmente trazem lucro.

Para calcular esse lucro de forma realista, não basta apenas subtrair as unidades vendidas. O custo das mercadorias muda constantemente devido a promoções de fornecedores ou variações cambiais. Para resolver isso, a contabilidade utiliza um método chamado PEPS, que significa "Primeiro que Entra, Primeiro que Sai" (Iudícibus, 2019). A lógica do PEPS é direta e estruturada. O sistema sempre dá baixa primeiro no custo dos lotes mais antigos que chegaram à loja. Assim, o valor do estoque que sobra reflete os preços mais recentes do mercado. Essa dinâmica permite atribuir custos por ordem de entrada, criando uma rastreabilidade histórica sobre o valor pago em cada lote adquirido.

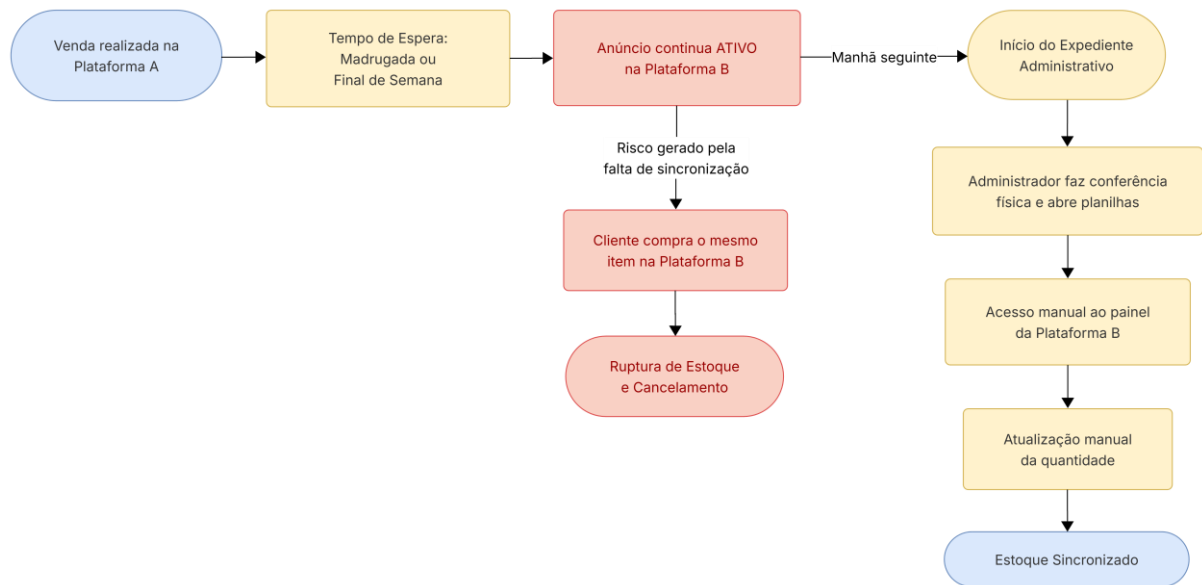
Aplicar essa regra na prática exige uma base tecnológica sólida. É nesse ponto que entram os bancos de dados relacionais. Em vez de usar planilhas sem controles de integridade, versionamento e acesso, a engenharia de dados cria tabelas conectadas para registrar cada entrada e cada venda de forma estruturada (Elmasri; Navathe, 2011). Com a ajuda de rotinas programadas diretamente no núcleo do banco de dados, o sistema lê o histórico e calcula automaticamente o custo do pedido segundo o método PEPS. Essa arquitetura centralizada automatiza a rotina de valoração do estoque e garante que os dados estejam sempre padronizados para alimentar painéis visuais de análise.

3 CONTEXTO INVESTIGADO E SITUAÇÃO PROBLEMA

O presente relato técnico-tecnológico baseia-se na observação empírica da rotina de uma microempresa brasileira, localizada em Uberlândia, que atua no varejo digital de componentes para impressoras 3D e *hardwares* de informática. O diagnóstico operacional foi conduzido pelo próprio autor. A coleta de evidências ocorreu mediante o monitoramento diário dos processos logísticos operacionais durante a execução de um estágio acadêmico de trezentas horas em Gestão da Informação. O modelo de negócios da loja exige que os anúncios sejam distribuídos em diferentes canais e sites de venda para manter um fluxo contínuo de clientes. Atualmente, a operação lida com um catálogo dinâmico. O grande desafio dessa estratégia reside na gestão física das mercadorias. Como existe apenas um estoque centralizado para abastecer todas as vitrines virtuais simultaneamente, qualquer venda efetivada exige uma atualização imediata da quantidade nos demais canais.

Historicamente, essa conferência de estoque era realizada de forma totalmente manual. Como a atualização dependia de o administrador acessar manualmente os painéis do sistema, pedidos ocorridos fora do horário comercial faziam com que o anúncio continuasse ativo nos outros sites até a manhã seguinte. Esse atraso gerava um risco elevado de realizar vendas sem disponibilidade física na prateleira. Quando essa ruptura de estoque acontece, o pedido precisa ser inevitavelmente cancelado. Esse cancelamento frustra o comprador e impacta negativamente as métricas de desempenho da conta. As penalidades previstas nas políticas oficiais da plataforma resultam na perda do selo qualificativo de "Vendedor Indicado" (Shopee, 2026). A Figura 1 ilustra o fluxo desse processo manual e demonstra os pontos exatos onde ocorre a perda de eficiência produtiva.

Figura 1 – Fluxo do processo de controle de estoque antes da intervenção



Fonte: elaborado pelo autor (2026).

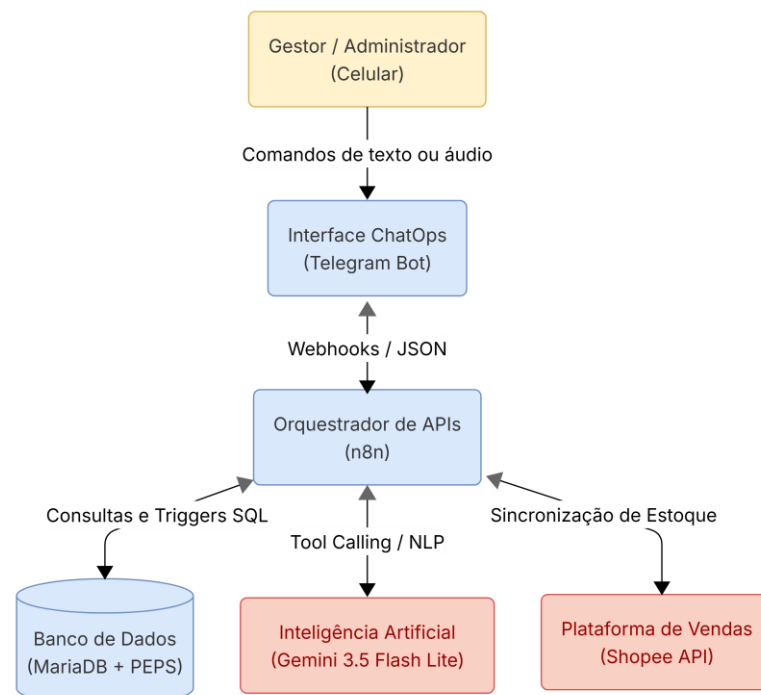
Além do risco de realizar vendas sem mercadoria física, identificou-se uma severa dificuldade no controle financeiro da operação. O comércio de componentes eletrônicos sofre com frequentes variações no preço de custo devido a mudanças cambiais e tributação de importação. Sem um sistema centralizado para registrar a entrada de cada lote e calcular o custo da mercadoria vendida utilizando a regra contábil PEPS (*Primeiro que Entra, Primeiro que Sai*), a empresa perde a visão da sua margem real de lucro. A partir dos registros operacionais e de fluxo de tarefas durante o período monitorado, quantificou-se que a dependência de planilhas soltas e a necessidade de contagem física nas prateleiras demandava cerca de oito horas semanais exclusivas para esse retrabalho administrativo. Além disso, a manipulação recorrente de planilhas contendo dados pessoais de compradores, como nome completo e endereço de entrega, cria um ponto de vulnerabilidade e risco de acesso indevido. Esse modelo não estruturado de tratamento de dados fere diretamente o princípio de segurança e prevenção estabelecido pela Lei Geral de Proteção de Dados Pessoais (Brasil, 2018).

Esse cenário evidencia a urgência de uma reestruturação tecnológica na operação. O problema central a ser superado consiste na substituição do esforço manual e repetitivo por uma arquitetura de dados em nuvem. O sistema necessita ser capaz de centralizar o estoque, automatizar o cálculo financeiro pelo método PEPS e permitir o controle da operação de forma ágil e segura pelo dispositivo móvel do gestor. Foi exatamente com base na necessidade primária de solucionar esses gargalos informacionais que a intervenção tecnológica, detalhada a seguir, foi projetada.

4 INTERVENÇÃO ADOTADA

A intervenção tecnológica consistiu em criar um sistema de gestão enxuto, desenhado para eliminar o trabalho manual de conferência de estoque e controle financeiro. A ideia principal foi construir um ambiente onde o banco de dados da loja, a plataforma de automação e o aplicativo de mensagens no celular funcionassem em conjunto, formando um ecossistema integrado para a automação operacional. A Figura 2 ilustra o desenho geral dessa arquitetura.

Figura 2 – Arquitetura de integração do sistema



Fonte: elaborado pelo autor (2026).

Conforme detalhado no diagrama, o fluxo operacional inicia-se na camada de apresentação, onde o gestor, por meio de um dispositivo móvel, envia comandos em formato de texto ou áudio para o *bot* do Telegram. Essa interface *ChatOps* capta a interação e a transmite via *webhooks* estruturados em JSON para o orquestrador n8n. O orquestrador atua como o núcleo de roteamento (arquitetura *hub-and-spoke*), impedindo que as pontas do sistema se comuniquem diretamente sem validação. A partir desse núcleo, a requisição é ramificada em três integrações principais. Na primeira vertente, o orquestrador aciona a inteligência artificial por meio de processamento de linguagem natural (NLP) e *Tool Calling* para interpretar a intenção do usuário. Na segunda, com a intenção mapeada, o n8n executa consultas e gatilhos SQL no banco de dados MariaDB para persistir os dados e rodar a rotina PEPS. Por fim, a terceira integração conecta-se de forma complementar à API da Shopee para realizar a

sincronização externa do estoque. Essa topologia garante o controle centralizado do tráfego e a segurança das operações.

4.1 Arquitetura tecnológica e modelagem do banco de dados

Para que o sistema ficasse disponível o tempo todo sem depender de um computador ligado fisicamente na empresa, a infraestrutura foi montada na nuvem da Oracle (Oracle, 2026). Em vez de contratar servidores caros, utilizou-se uma máquina virtual leve executando o sistema operacional Ubuntu (Canonical, 2026). Para facilitar a gestão dessa máquina e garantir a segurança, instalou-se o painel de controle CloudPanel (CloudPanel, 2026). Essa ferramenta gerencia a emissão e a renovação automatizada de certificados digitais TLS (*Transport Layer Security*) para a proteção da comunicação externa na camada de transporte. Adicionalmente, o tráfego de rede e o apontamento do domínio foram gerenciados por meio da infraestrutura da Cloudflare (Cloudflare, 2026). Essa ferramenta atua como *proxy* reverso e provê uma camada extra de proteção contra ataques de negação de serviço distribuída (DDoS).

O componente central de persistência e processamento da operação é o banco de dados MariaDB (MariaDB, 2026). Como apontam Elmasri e Navathe (2011), uma boa modelagem de dados evita que informações fiquem duplicadas ou se percam com o tempo. Por isso, substituiu-se o modelo baseado em planilhas e o banco foi desenhado em três grandes blocos de tabelas. Para garantir a integridade referencial, todas as entidades foram amarradas por regras estritas de Chaves Primárias (PK) e Chaves Estrangeiras (FK), conforme detalhado no Quadro 1.

Quadro 1 – Dicionário de dados simplificado

Bloco Lógico	Tabela	Função Principal	Estrutura (PK / FK)
Cadastros	produtos	PK: id_produto	Catálogo de itens comerciais e operacionais.
Cadastros	clientes	PK: id_cliente	Dados de compradores (com versionamento).
Cadastros	fornecedores	PK: id_fornecedor	Dados de distribuidores e fabricantes.
Kits e Embalagens	composicao_kits	PK: id_composicao FK: id_kit, id_peca	Liga produtos simples para formar um Kit.
Kits e Embalagens	mapeamento_embalagens	PK: id_map FK: id_produto, id_caixa	Associa caixas/plásticos a cada venda.
Movimentação	entradas	PK: id_entrada FK: id_produto, id_fornecedor	Registra compras, lotes e custo unitário.
Movimentação	pedidos	PK: id_pedido FK: id_cliente	Registra vendas, status e canal.
Movimentação	itens_pedido	PK: id_item FK: id_pedido, id_produto, id_entrada	Itens vendidos e cálculo do custo PEPS.

Fonte: elaborado pelo autor (2026).

O primeiro bloco cuida dos cadastros básicos de produtos, fornecedores e clientes. O segundo bloco gerencia a montagem de *kits*, permitindo que a venda de um conjunto completo dê baixa automática nas pequenas peças que o formam. O terceiro bloco registra toda a movimentação, como entradas de fornecedores e saídas por vendas.

A grande inovação dessa modelagem foi colocar o controle financeiro para ser executado automaticamente dentro do banco de dados. Em vez de o administrador precisar calcular o custo de cada componente vendido, o sistema foi programado para aplicar a regra do PEPS (*Primeiro que Entra, Primeiro que Sai*) de forma nativa. Toda vez que uma venda é registrada, uma rotina interna do banco (uma *Stored Procedure*) inicia uma transação segura. Essa rotina identifica o lote mais antigo com saldo disponível e vincula o custo exato de aquisição daquele lote ao item vendido no pedido atual. Para garantir a confiabilidade matemática dessa operação, o código foi protegido com mecanismos de bloqueio de concorrência e reversão automática de falhas (*rollback*). Essa estrutura arquitetural impede anomalias graves, como saldos negativos ou baixas duplicadas caso duas vendas ocorram na mesma fração de segundo.

Para facilitar a leitura dessas informações pelo gestor, foram criadas visões de banco de dados (*views*). A *view* de estoque funciona como um radar que soma tudo o que a loja tem e avisa se algum produto está abaixo da quantidade mínima segura para venda. Já a *view* de lucratividade foi desenhada para calcular a margem de contribuição de cada pedido. Essa métrica cruza as tabelas financeiras e aplica uma equação direta, subtraindo as taxas retidas pela plataforma e o custo de aquisição (calculado rigorosamente pelo PEPS) do valor total pago pelo cliente final.

4.2 Integração de APIs e governança na ingestão de dados

Para integrar o banco de dados à internet e ao dispositivo móvel do gestor, adotou-se a plataforma de automação n8n (n8n, 2026). O sistema atua como orquestrador dos fluxos de integração, recebendo os comandos do usuário, traduzindo as intenções para consultas estruturadas de banco de dados e devolvendo a resposta processada.

Esse fluxo de comunicação foi estruturado por meio de *endpoints* denominados *webhooks*. Foram criadas rotas específicas para cada rotina gerencial, como consultar o saldo físico, verificar o lucro por pedido, registrar a entrada de lotes de reposição e acionar a inteligência artificial. Além disso, o orquestrador foi conectado diretamente à API da Shopee (Shopee, 2026). Com essa arquitetura, se uma venda ocorre fora do horário comercial, a API

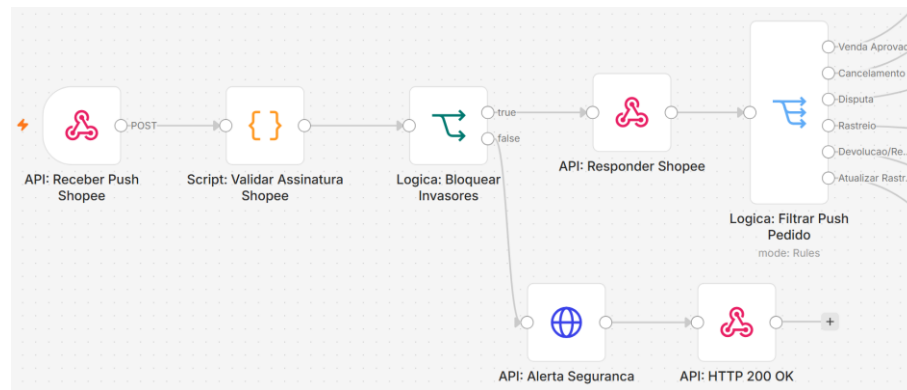
notifica o *webhook* e o sistema processa a baixa no banco de dados. Para garantir a resiliência operacional e evitar falhas de duplicidade, o sistema aplica o princípio da idempotência. A cada notificação recebida, o orquestrador verifica o identificador único do pedido (*Order ID*) no banco de dados antes de executar o desconto físico, garantindo que envios repetidos pela plataforma não causem dupla cobrança no estoque.

Como o sistema opera exposto na internet, foram definidos controles de autenticação e restrição de acesso rigorosos. Para evitar o consumo indevido dos dados por agentes externos, configurou-se uma chave de autenticação nas requisições (*API Key*). Para garantir a integridade da arquitetura, essa credencial de acesso não é codificada abertamente no fluxo visual. Ela permanece isolada nas variáveis de ambiente seguras do servidor. Se um serviço não autorizado tentar disparar o *webhook* sem apresentar essa assinatura exata, a conexão é recusada imediatamente com o código de erro 401 (*Unauthorized*).

A governança do fluxo também foi projetada para mitigar riscos de exposição em consonância com a Lei Geral de Proteção de Dados Pessoais (Brasil, 2018). Ao descontinuar as planilhas manuais, o controle sobre os dados pessoais de clientes tornou-se centralizado e rastreável. A rotina do sistema foi programada em observância ao princípio da minimização, ou seja, extraindo exclusivamente os dados essenciais à emissão da nota fiscal e processamento logístico, impedindo que informações sensíveis de compradores permaneçam armazenadas em arquivos soltos no dispositivo do administrador.

Para garantir a legibilidade e a compreensão técnica da arquitetura, o fluxograma completo de orquestração foi subdividido em blocos operacionais. A Figura 3 apresenta o primeiro estágio do fluxo, focado na ingestão de dados e segurança. Primeiramente, o orquestrador recebe os eventos e executa a triagem de segurança. Para validar a origem da notificação, o nó de autenticação calcula a assinatura criptográfica do pacote de dados utilizando o algoritmo HMAC-SHA256 e a compara com a chave oficial. Apenas após a aprovação nessa barreira matemática o evento segue para a ramificação lógica.

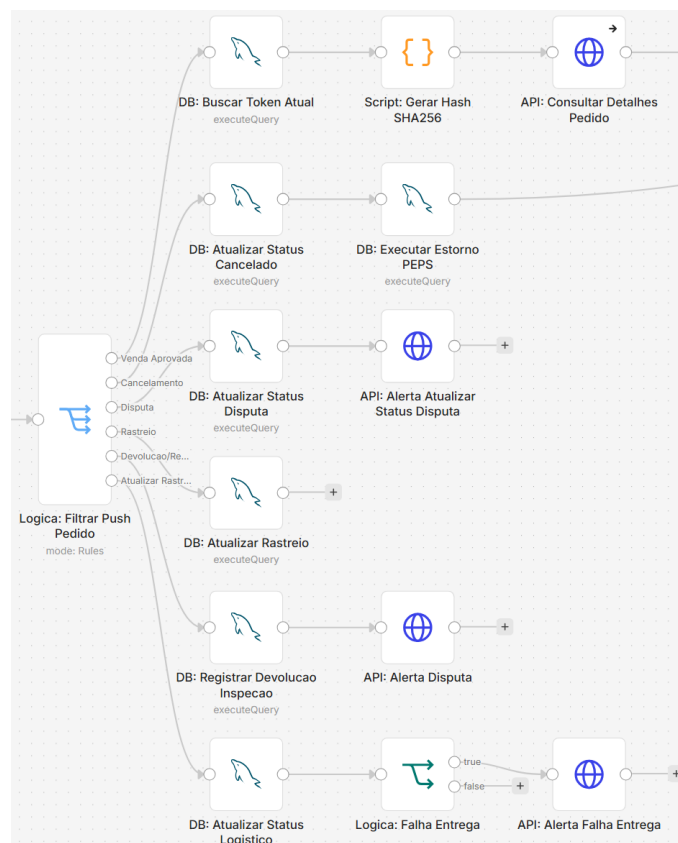
Figura 3 – Estágio de recepção, validação criptográfica e roteamento



Fonte: elaborado pelo autor (2026).

Após a validação, a estrutura de decisões filtra a ação demandada. A Figura 4 ilustra o roteamento voltado para a atualização de status logísticos e operacionais. Nesse estágio, o sistema processa eventos de exceção, como cancelamentos, aberturas de disputa, atualizações de código de rastreio e falhas de entrega. Cada caminho aciona consultas de atualização específicas no banco de dados, garantindo que o painel gerencial reflita a exata situação da encomenda sem intervenção humana.

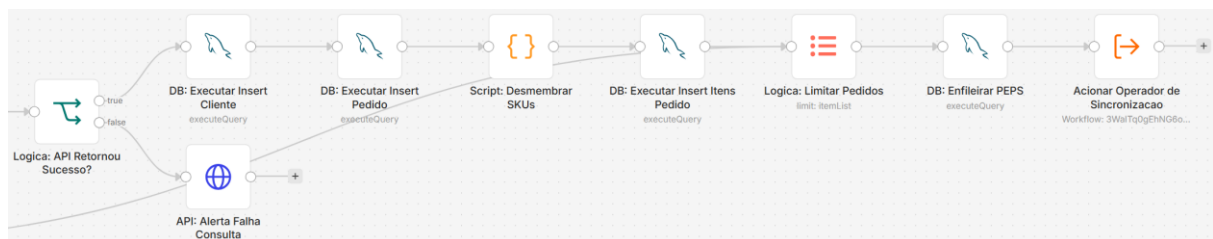
Figura 4 – Estágio de roteamento e atualização de exceções logísticas



Fonte: elaborado pelo autor (2026).

Quando o evento recebido é uma "Venda Aprovada", o sistema aciona a esteira mais crítica da operação, ilustrada na Figura 5. O fluxo realiza a inserção isolada dos dados do cliente e do pedido, desmembra os pacotes caso a venda seja de um *kit* complexo e grava os itens individuais. Ao final dessa esteira, o sistema aciona o nó de enfileiramento lógico do cálculo PEPS e dispara um comando assíncrono para o operador de sincronização, garantindo que a regra contábil seja aplicada.

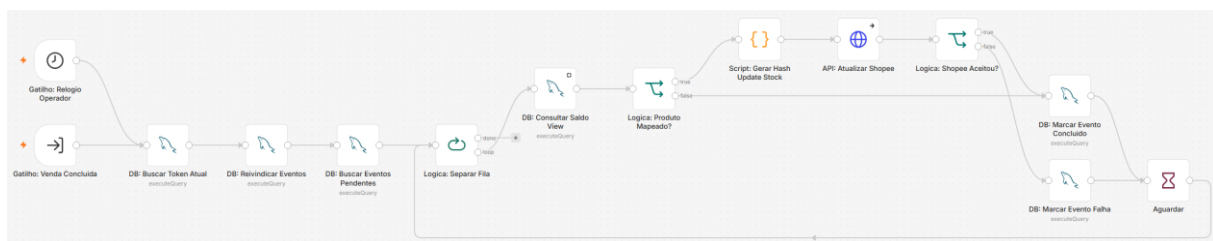
Figura 5 – Estágio de processamento de vendas, gravação de itens e gatilho PEPS



Fonte: elaborado pelo autor (2026).

Por fim, a automação do recálculo de estoque e a comunicação de volta com as plataformas de venda ocorrem em um fluxo dedicado, demonstrado na Figura 6. Esse operador assíncrono funciona como uma fila de processamento contínuo. Ele busca os eventos de venda validados no banco de dados e aciona a infraestrutura da loja para atualizar o saldo disponível. Essa separação de rotinas impede que falhas temporárias de conexão com a plataforma de vendas interrompam a gravação inicial dos pedidos no banco de dados.

Figura 6 – Operador assíncrono de fila para recálculo de estoque e sincronização



Fonte: elaborado pelo autor (2026).

4.3 Inteligência artificial como interface de gestão

A parte final da intervenção foi criar o controle remoto do sistema. A ideia principal era eliminar a necessidade de instalação de aplicativos adicionais ou o acesso a portais *web*

complexos pelo computador. Por isso, a interface de controle foi construída dentro do aplicativo Telegram (Telegram, 2026), adotando o modelo operacional *ChatOps*.

Para implementar o *bot* no Telegram, escreveu-se um código na linguagem Python utilizando a biblioteca nativa *asyncio*. Esse código foi programado de forma assíncrona. Em termos práticos, isso significa que o sistema pode aguardar operações de entrada e saída sem bloquear o fluxo principal de execução (Python Software Foundation, 2023). Os testes de latência da aplicação demonstraram que essa arquitetura concorrente permite responder a múltiplas consultas operacionais com atraso de processamento inferior a dois segundos.

A segurança da interface conversacional também recebeu controle de autorização. O código em Python possui uma validação estrita baseada no número de identificação exclusivo do administrador da empresa. Se qualquer outro usuário encontrar o *bot* no Telegram e tentar enviar um comando, o sistema recusa a interação imediatamente e não executa a requisição. Contudo, reconhece-se que essa lista de identificadores atua como um controle básico de autorização. Esse mecanismo mitiga o acesso público indesejado, mas apresenta limitações na análise de ameaças mais complexas. Caso o dispositivo físico do gestor seja comprometido ou a conta do aplicativo seja sequestrada, o sistema não possui, nesta versão, um segundo fator de autenticação (2FA) nativo para impedir acessos não autorizados de forma absoluta.

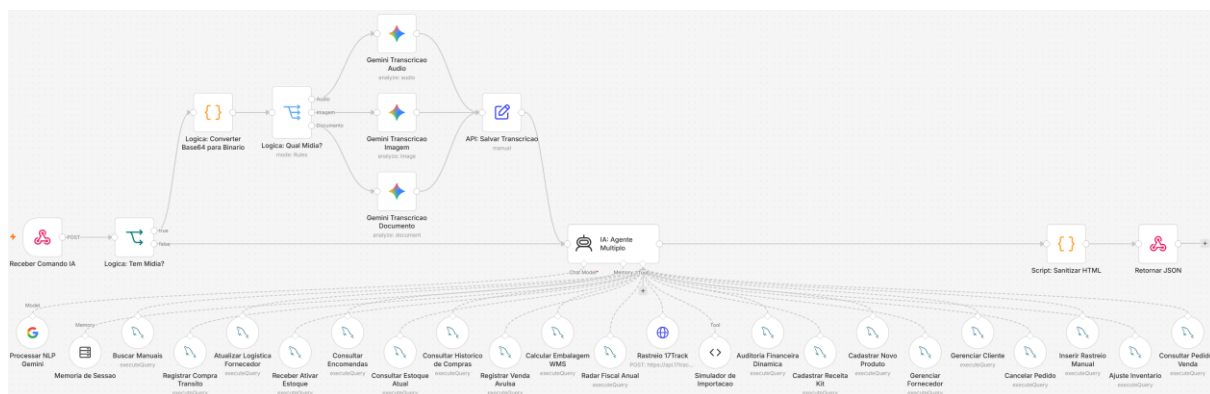
Para tornar o uso intuitivo, a inteligência artificial foi conectada ao fluxo de mensagens com suporte a interações multimodais. O administrador pode navegar por um menu de botões visuais ou enviar comandos diretos em texto, áudio e imagem. O processamento cognitivo dessas requisições é realizado por meio da API oficial do Google AI Studio. Durante a fase de validação deste projeto, utilizou-se o modelo fundacional Gemini 3.5 Flash Lite. Essa versão foi selecionada por atender ao volume de requisições necessário para a operação. A configuração do *endpoint* foi definida com o parâmetro de temperatura estruturado em 0.2. Esse ajuste técnico rigoroso visa reduzir a criatividade do modelo e priorizar respostas puramente determinísticas. Em conformidade com as políticas de privacidade da plataforma, as mídias submetidas para transcrição ou análise não são retidas para treinamento do modelo de base (Google, 2026).

O pacote de dados processado é enviado ao modelo, que interpreta a intenção gerencial e aciona as ferramentas do sistema por meio do conceito de *Tool Calling*. Essa capacidade visual permitiu que o agente autônomo analisasse fotografias enviadas pelo gestor, extraísse as informações comerciais relevantes da imagem e registrasse a entrada de dados diretamente no banco de dados. A autonomia do agente, no entanto, não é irrestrita. Ela é rigorosamente delimitada pelas definições de esquema de dados (*JSON Schema*) fornecidas pelo orquestrador.

Para garantir que o modelo de linguagem não execute comandos arbitrários ou imprevistos, a arquitetura emprega o princípio de restrição por assinatura de funções (*tool declarations*), onde cada ferramenta possui um contrato de parâmetros estrito validado por esquema de tipos. O agente não possui acesso direto à base de dados para escrita livre. Suas permissões operacionais são limitadas a invocar procedimentos armazenados específicos cujos argumentos são rigidamente tipados antes da execução. Adicionalmente, o sistema implementa um controle de fluxo baseado em estado transacional para operações modificadoras. Ações de impacto físico no estoque ou financeiro não são efetivadas na primeira interação do usuário. Elas entram em um estado intermediário de pendência analítica e exigem a recepção de um *token* textual explícito de autorização para que o gatilho final de persistência seja liberado. Em cenários de entradas ambíguas ou dados incompletos fornecidos via áudio, o agente está programado para interromper a execução e retornar uma solicitação de esclarecimento, mitigando o risco de corrupção cadastral por interpretações incorretas.

A Figura 7 apresenta a arquitetura aprimorada do agente de inteligência artificial construída dentro do orquestrador n8n. O fluxograma destaca a roteirização inicial de mídias para transcrição e o nó central do agente conectado a dezessete ferramentas de banco de dados e comunicação. Para mitigar riscos operacionais e prevenir injeções de comandos maliciosos (*prompt injection*), o sistema diferencia consultas de leitura de ações modificadoras. Operações críticas, como o desconto físico no banco de dados, requerem uma etapa de validação humana explícita. O agente prepara os dados e exige que o gestor envie a palavra de confirmação antes que o gatilho de execução final seja acionado.

Figura 7 – Arquitetura do agente autônomo, tratamento de mídia e ferramentas restritas

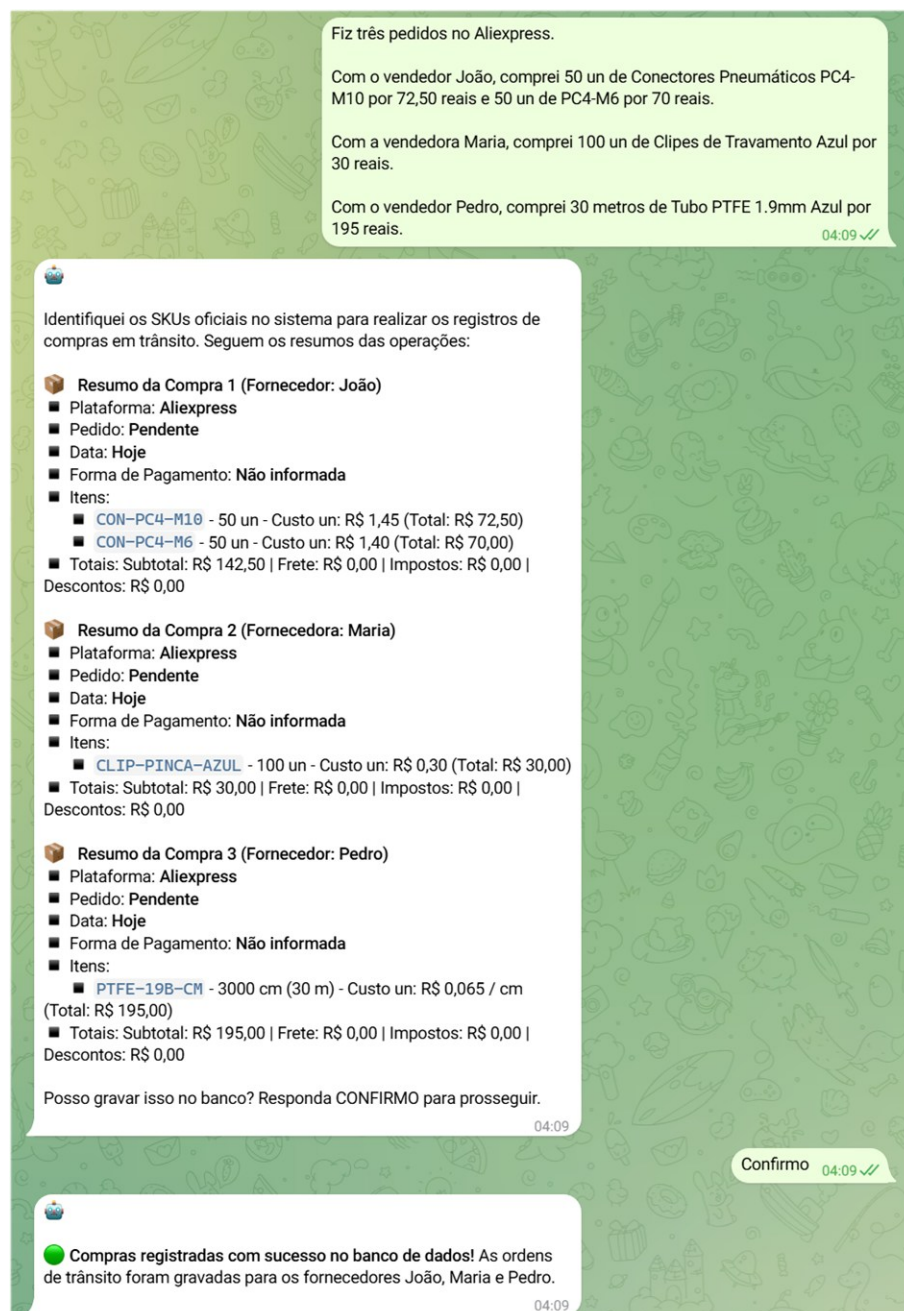


Fonte: elaborado pelo autor (2026).

Na prática, a Figura 8 demonstra o protocolo de validação funcional operando diretamente no aplicativo. É importante ressaltar que os dados comerciais e os nomes exibidos

nas capturas de tela são representações anonimizadas, inseridas estritamente para a execução segura dos testes. O ensaio documentado comprova a capacidade da inteligência artificial em processar entradas com múltiplas variáveis não estruturadas. O agente interpreta o texto bruto, padroniza os códigos oficiais (SKUs) dos produtos, realiza cálculos automáticos precisos (como o rateio do custo unitário e a conversão matemática de metros de tubo para a unidade base em centímetros) e apresenta o resumo estruturado das compras em trânsito para a auditoria e confirmação do gestor.

Figura 8 – Interface ChatOps em funcionamento com validação de dados



Fonte: elaborado pelo autor (2026).

A Figura 9 ilustra a etapa logística subsequente e a modelagem de recálculo dinâmico. Quando o usuário reporta que as encomendas chegaram, o sistema efetiva a ativação das mercadorias no estoque físico. Ao consultar o saldo em seguida, o gestor visualiza instantaneamente não apenas os itens base recém-ativados, mas todos os *kits* de venda complexos que a arquitetura estruturada do banco de dados calculou ser possível montar a partir da combinação dessas novas peças avulsas.

Figura 9 – Interface ChatOps: ativação de estoque e recálculo dinâmico de kits



Fonte: elaborado pelo autor (2026).

5 RESULTADOS ALCANÇADOS

A implantação do sistema gerou impactos operacionais mensuráveis na rotina da microempresa. Ao substituir o modelo baseado em planilhas manuais por um ecossistema integrado em nuvem, foi possível estruturar um fluxo contínuo e auditável de informações para a gestão da loja.

O primeiro ganho expressivo refere-se à sincronização de estoque. O tempo de atualização, que dependia de intervenção manual no dia posterior à venda, passou a ocorrer de forma síncrona por meio do *webhook*, eliminando o intervalo operacional em que o produto permanecia exposto sem saldo físico. Durante o período de monitoramento da intervenção, não foram registradas ocorrências de ruptura por atraso na baixa ou cancelamentos decorrentes de vendas sem estoque físico. Conforme quantificado a partir do fluxo de tarefas, a automação reduziu em quase um turno semanal o tempo dedicado pelo gestor ao retrabalho com planilhas e à contagem manual, liberando cerca de oito horas semanais para atividades estratégicas.

No âmbito financeiro, o cálculo do custo de aquisição tornou-se automatizado. Com as rotinas implementadas no banco de dados, o sistema executa a valoração pelo método PEPS de forma nativa por lote registrado. O indicador financeiro adotado para a análise de desempenho do pedido passou a ser a margem de contribuição, calculada mediante o cruzamento automatizado do preço de venda com as taxas retidas pela plataforma e o custo unitário fornecido pelo PEPS, permitindo uma visualização imediata da rentabilidade sem cálculos manuais.

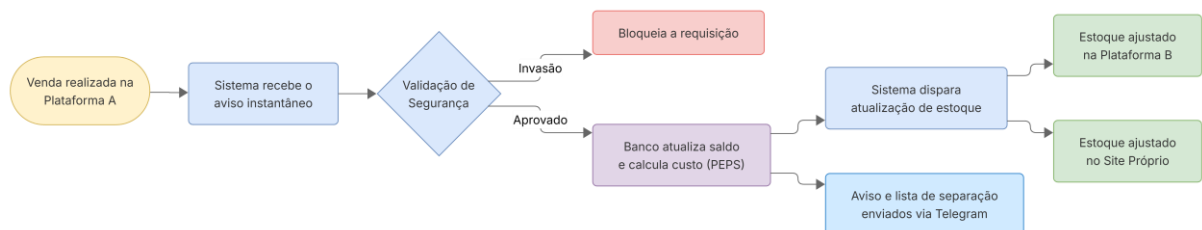
A validação da precisão dessa regra contábil ocorreu por meio de um teste de homologação. Para viabilizar esse ensaio, construiu-se um fluxo de orquestração temporário dedicado exclusivamente à extração de dados em lote. Esse mecanismo conectou-se à plataforma de vendas e importou uma amostragem contendo 126 pedidos históricos reais. Essa carga de informações foi então submetida ao processamento da *Stored Procedure* no banco de dados. Os resultados da valoração de custo gerados pelo sistema foram auditados e cruzados manualmente com os registros financeiros anteriores da empresa. O ensaio demonstrou êxito na atribuição dos custos, comprovando que o banco de dados conseguiu identificar corretamente os lotes mais antigos e descontar os valores exatos de aquisição sem divergências matemáticas. Esse teste de estresse lógico garantiu a confiabilidade necessária para atestar o pleno funcionamento do método PEPS automatizado.

No aspecto logístico, a estruturação de itens agregados permitiu o cálculo automático da disponibilidade de *kits* comerciais a partir da baixa das peças simples, mantendo a acurácia do inventário físico validada por contagens periódicas de amostra.

Em relação à interface conversacional de apoio à gestão no Telegram, o sistema demonstrou eficácia no processamento de entradas não estruturadas. O modelo interpretou comandos, realizou extrações textuais e executou conversões métricas automatizadas, sendo que todas as ações modificadoras foram submetidas à validação e confirmação explícita do usuário antes da gravação definitiva no banco de dados.

Por fim, a operação incorporou camadas adicionais de segurança e governança técnica. O acesso ao painel de orquestração foi protegido por autenticação restrita e isolamento de credenciais em variáveis de ambiente seguras no servidor, mitigando vetores de exposição não autorizada. A adoção de controles técnicos alinhou-se aos princípios de minimização e segurança previstos na Lei Geral de Proteção de Dados Pessoais (Brasil, 2018), centralizando o tratamento de dados essenciais. A Figura 10 ilustra o novo fluxo da informação após a adoção da tecnologia, contrastando com o cenário anterior de vulnerabilidade.

Figura 10 – Fluxo do processo de controle de estoque após a intervenção



Fonte: elaborado pelo autor (2026).

Conforme mapeado no diagrama final, o processo automatizado inicia-se com a recepção instantânea do aviso de venda via *webhook*. O sistema submete o pacote de dados a uma validação de segurança rigorosa. Requisições com assinaturas inválidas ou suspeitas são imediatamente bloqueadas. Em caso de aprovação, o fluxo avança para o banco de dados, que atualiza o saldo e calcula o custo pelo método PEPS de forma nativa. Após essa persistência dos dados, o sistema aciona rotinas assíncronas paralelas. Uma ramificação dispara a atualização de estoque para os demais canais, como a plataforma secundária e o site próprio. A outra ramificação envia o aviso e a lista de separação de mercadorias diretamente para o Telegram do gestor. Esse ciclo integrado elimina o tempo de espera da conferência manual e previne em tempo real o risco de ruptura de prateleira.

6 CONSIDERAÇÕES FINAIS

O objetivo deste relato técnico-tecnológico consistiu em desenvolver e avaliar uma arquitetura de integração em nuvem orientada a agentes de inteligência artificial para a automação do fluxo de dados e gestão de estoque em uma microempresa de varejo digital. A partir da análise dos resultados operacionais durante o período de monitoramento, conclui-se que o objetivo foi atingido dentro do escopo proposto. A intervenção demonstrou a viabilidade técnica da solução no contexto específico da empresa analisada.

A solução construída substituiu um processo manual e sujeito a falhas por um ecossistema autônomo. A arquitetura apresentou viabilidade econômica ao utilizar instâncias de nuvem em camadas gratuitas (*free tier*) e plataformas de orquestração de código aberto, mitigando altos custos de licenciamento. A união do banco de dados relacional com o orquestrador mitigou as ocorrências de ruptura de estoque sistêmico, não havendo registros de vendas sem saldo físico durante o período avaliado. Essa sincronização contínua é fundamental para reduzir os cancelamentos que podem afetar a reputação da loja perante os algoritmos da plataforma de vendas. No aspecto financeiro, a aplicação automática da regra PEPS viabilizou o cálculo rastreável do custo das mercadorias, permitindo a visualização da margem de contribuição de cada pedido de forma exata.

O uso do Telegram como interface conversacional evidenciou que a inteligência artificial pode atuar como uma ferramenta de apoio gerencial eficiente no cenário investigado. O aplicativo de mensagens serviu como um painel de controle capaz de interpretar entradas não estruturadas e acionar gatilhos sem exigir a manipulação direta de planilhas. Esses componentes foram protegidos por restrição de rotas e autenticação. Essa infraestrutura mitiga riscos de vazamento e restringe o acesso não autorizado ao núcleo do sistema nas condições testadas.

Apesar dos resultados operacionais positivos, o desenvolvimento e a implantação da solução apresentaram desafios práticos significativos. A primeira dificuldade consistiu no provisionamento da infraestrutura. A obtenção de instâncias computacionais gratuitas na Oracle Cloud (*Always Free*) foi inviabilizada pela falta de capacidade ociosa nos servidores regionais da provedora. Tentou-se operar rotinas automatizadas de forma contínua por cinco dias na tentativa de alocar uma máquina virtual, mas a abordagem não obteve sucesso. Esse impasse de infraestrutura exigiu a mudança de estratégia, sendo solucionado apenas com o *upgrade* da conta para a modalidade *Pay-As-You-Go*.

Durante a etapa de integração de dados, a escassez de documentação oficial na API da plataforma de vendas exigiu uma engenharia reversa das requisições recebidas. Foi necessário

inspecionar os *payloads* brutos para definir os parâmetros exatos das *queries* de leitura. Adicionalmente, a implementação da camada de segurança e a orquestração apresentaram percalços relevantes. A validação da assinatura criptográfica HMAC-SHA256 gerou falhas frequentes de divergência matemática, exigindo ajustes minuciosos na interpretação do texto bruto pela integração. Do mesmo modo, o isolamento seguro de credenciais exigiu extensos ciclos de depuração para garantir que o orquestrador não conseguisse importar adequadamente os códigos restritos do arquivo de variáveis de ambiente (*.env*).

Quanto à escalabilidade, o volume de processamento atual foi homologado em ambiente de produção sob um fluxo operacional diário e estável. Ainda assim, a infraestrutura foi projetada com arquitetura isolada para suportar aumentos repentinos no volume de pedidos. Para atingir essa resiliência em alta carga, a modelagem empregou técnicas corporativas de desacoplamento de fluxo e processamento assíncrono. Em vez de calcular toda a baixa logística e financeira no momento exato em que a venda ocorre, o *webhook* envia uma confirmação imediata de recebimento para a plataforma externa. Isso evita punições por tempo de resposta excedido (*timeout*). O evento de venda é então inserido em uma fila no banco de dados e o processamento financeiro pelo método PEPS ocorre de forma programada diretamente no núcleo do MariaDB por meio de *Stored Procedures*. Essa estratégia otimiza o consumo de memória do orquestrador e garante que a fila suporte variações massivas de transações sem degradação do serviço.

Como trabalhos futuros para a evolução deste ecossistema, planeja-se o desenvolvimento de um catálogo digital operando com a tecnologia WooCommerce em um domínio próprio. Estrategicamente, devido aos desafios de segurança no comércio eletrônico brasileiro — como fraudes financeiras e logística reversa maliciosa de componentes avariados —, o site não atuará inicialmente como um canal de venda direta. Ele funcionará como um repositório oficial para a disponibilização de manuais, especificações técnicas e suporte, redirecionando o consumidor final para a compra segura nos *marketplaces*. Nessa modelagem de negócios, as taxas retidas pelas plataformas passam a ser tratadas como um custo aceitável de mitigação de riscos operacionais. Além disso, prevê-se a criação de painéis visuais analíticos (*dashboards*) conectados diretamente à base de dados e o aprimoramento da inteligência artificial para realizar a previsão de demanda. Essa última implementação deverá alertar ativamente o gestor sobre o melhor momento para recomprar mercadorias de fornecedores internacionais antes que o estoque de segurança seja atingido.

REFERÊNCIAS

- BRASIL. **Lei nº 13.709, de 14 de agosto de 2018**. Lei Geral de Proteção de Dados Pessoais (LGPD). Diário Oficial da União, Brasília, DF, 15 ago. 2018. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/113709.html. Acesso em: 2 ago. 2026.
- CANONICAL. **Ubuntu OS**. [S. l.]: Canonical, 2026. Disponível em: <https://ubuntu.com>. Acesso em: 2 ago. 2026.
- CLOUDFLARE. **Cloudflare Developers**. São Francisco: Cloudflare, 2026. Disponível em: <https://developers.cloudflare.com>. Acesso em: 2 ago. 2026.
- CLOUDPANEL. **CloudPanel Documentation** [S. l.]: CloudPanel, 2026. Disponível em: <https://www.cloudpanel.io>. Acesso em: 2 ago. 2026.
- ECMA INTERNATIONAL. **The JSON Data Interchange Syntax**. Standard ECMA-404. 2. ed. Genebra: Ecma International, 2017. Disponível em: <https://ecma-international.org/publications-and-standards/standards/ecma-404/>. Acesso em: 2 ago. 2026.
- ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson Addison Wesley, 2011.
- GARTNER. **Top strategic technology trends for 2024: democratized generative ai**. [S. l.]: Gartner, 2023. Disponível em: <https://www.gartner.com/en/newsroom/press-releases/2023-10-16-gartner-identifies-the-top-10-strategic-technology-trends-for-2024>. Acesso em: 7 jun. 2026.
- GOOGLE. **Gemini API Documentation**. Google AI for Developers, 2026. Disponível em: <https://ai.google.dev/gemini-api/docs?hl=pt-br>. Acesso em: 2 ago. 2026.
- IUDÍCIBUS, S. de. **Contabilidade básica**. 12. ed. São Paulo: Atlas, 2019.
- KIMBALL, R.; ROSS, M. **The Data Warehouse Toolkit: the definitive guide to dimensional modeling**. 3. ed. Indianapolis: John Wiley & Sons, 2013.
- LAUDON, K. C.; LAUDON, J. P. **Sistemas de informação gerenciais**. 14. ed. São Paulo: Pearson Education do Brasil, 2021.
- MARIADB FOUNDATION. **MariaDB Server Documentation**. Versão 11.4. [S. l.]: MariaDB Foundation, 2026. Disponível em: <https://mariadb.com/docs/>. Acesso em: 2 ago. 2026.
- MASSE, M. **REST API Design Rulebook**. Sebastopol: O'Reilly Media, 2011.
- MOZILLA DEVELOPER NETWORK. **JavaScript Reference**. [S. l.]: Mozilla, 2026. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso em: 2 ago. 2026.
- N8N. **n8n Documentation: Workflow Automation**. Berlin: n8n, 2026. Disponível em: <https://docs.n8n.io>. Acesso em: 2 ago. 2026.
- ORACLE. **Oracle Cloud Infrastructure Documentation**. Austin: Oracle, 2026. Disponível em: <https://docs.oracle.com/en-us/iaas/Content/home.html>. Acesso em: 2 ago. 2026.

PYTHON SOFTWARE FOUNDATION. **Python Documentation**. Versão 3.12. [S. l.]: Python Software Foundation, 2023. Disponível em: <https://docs.python.org/3/>. Acesso em: 7 jun. 2026.

RUSSELL, S.; NORVIG, P. **Inteligência artificial: uma abordagem moderna**. 4. ed. Rio de Janeiro: GEN LTC, 2021.

SÁNCHEZ-ADAME, L. M. et al. Towards a set of heuristics for evaluating chatbots. **IEEE Latin America Transactions**, [S. l.], v. 19, n. 12, p. 2037-2045, 2021.

SHOPEE. **Shopee Open Platform Documentation**. [S. l.]: Shopee, 2026. Disponível em: <https://open.shopee.com>. Acesso em: 2 ago. 2026.

SLACK, N. et al. **Administração da produção**. 5. ed. São Paulo: Atlas, 2015.

TELEGRAM. **Telegram Bot API**. [S. l.]: Telegram, 2026. Disponível em: <https://core.telegram.org/bots/api>. Acesso em: 2 ago. 2026.

DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL

A elaboração deste relato técnico-tecnológico contou com o uso de Inteligência Artificial Generativa, especificamente o modelo Gemini 3.1 Pro, atuando estritamente como ferramenta de apoio operacional durante o período de desenvolvimento da pesquisa e do sistema (abril a julho de 2026).

A utilização da tecnologia foi direcionada a duas frentes práticas guiadas por comandos de texto (*prompts*) específicos enviados pelo autor. A primeira consistiu no refinamento gramatical e de clareza nas seções de fundamentação teórica. A segunda abrangeu o auxílio na otimização da sintaxe de programação, envolvendo os roteiros em Python para a interface no Telegram e as consultas SQL estruturadas para o banco de dados MariaDB.

Em respeito à integridade acadêmica e à transparência, destaca-se que a IA não gerou análises ou decisões de arquitetura de forma autônoma. A validação do ecossistema e a confirmação de autoria baseiam-se em intervenção e supervisão humana constante. Todas as lógicas sugeridas pela ferramenta foram testadas empiricamente, monitoradas por meio dos *logs* de execução do orquestrador n8n e validadas na prática a partir do processamento de uma amostragem representativa de pedidos em ambiente de produção para auditar o funcionamento do cálculo da regra contábil PEPS. O histórico de versionamento do sistema e os registros de execução das APIs evidenciam a construção ativa por parte do autor, que assume a responsabilidade integral pela arquitetura, veracidade dos dados e resultados apresentados neste documento.