v7.n11.2026

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

PREVISIO: DESENVOLVIMENTO DE UM SISTEMA DE APOIO À DECISÃO PARA GESTÃO DE ESTOQUES NO VAREJO COM CLASSIFICAÇÃO ABC E MODELAGEM PREDITIVA

PREVISIO: DEVELOPMENT OF A DECISION SUPPORT SYSTEM FOR RETAIL INVENTORY MANAGEMENT USING ABC CLASSIFICATION AND PREDICTIVE MODELING

PREVISIO: DESARROLLO DE UN SISTEMA DE APOYO A LA DECISIÓN PARA LA GESTIÓN DE INVENTARIOS MINORISTAS MEDIANTE CLASIFICACIÓN ABC Y MODELADO PREDICTIVO

Carlos Eduardo Lujan[1]; Felipe Diniz Dallilo[2]

e7119393
https://doi.org/10.47820/recima21.v7i11.9393

PUBLICADO: 11/2026

RESUMO

A gestão de estoques representa um desafio estratégico para pequenos varejistas, especialmente no equilíbrio entre a disponibilidade de mercadorias e o capital imobilizado. Este trabalho tem como objetivo desenvolver a plataforma Previsio, um sistema de apoio à decisão para gestão de inventário baseado em classificação ABC e modelagem preditiva. O protótipo foi implementado em Python, utilizando Regressão Linear para estimar a demanda e combinando essa estimativa com regras de estoque, tempo de entrega (lead time) e histórico de custos; o MongoDB Atlas realiza a persistência dos dados e o Streamlit compõe a interface. A validação funcional foi realizada com uma base sintética de 24 meses contendo 160 produtos, 56.766 registros de vendas e 125 notas fiscais de entrada. A classificação ABC identificou 38 itens na classe A, responsáveis por 78,21% do faturamento simulado, 40 na classe B e 82 na classe C. No módulo de ressuprimento, a previsão de demanda é combinada ao estoque atual, ao estoque de segurança e ao lead time para calcular o ponto de ressuprimento e apoiar a sugestão de compra. Os resultados demonstram a viabilidade técnica do protótipo como camada de apoio à decisão; contudo, a validação ocorreu exclusivamente em ambiente computacional simulado, com avaliação retrospectiva preliminar da precisão preditiva, mas sem mensuração de ganhos operacionais reais. Nesse cenário, verificou-se o funcionamento integrado dos módulos do protótipo. A integração automática com ERPs de terceiros permanece como trabalho futuro.

PALAVRAS-CHAVE: Classificação ABC. Gestão de estoques. Modelagem preditiva. MongoDB. Streamlit.

ABSTRACT

Inventory management represents a strategic challenge for small retailers, especially in balancing merchandise availability and tied-up capital. This study aims to develop the Previsio platform, a decision support system for inventory management based on ABC classification and predictive modeling. The prototype was implemented in Python, using Linear Regression to estimate demand and combining this estimate with inventory rules, lead time, and cost history; MongoDB Atlas handles data persistence and Streamlit composes the interface. Functional validation was performed with a 24-month synthetic dataset containing 160 products, 56,766 sales records, and 125 purchase invoices. The ABC classification identified 38 items in class A, accounting for 78.21% of simulated revenue, 40 in class B, and 82 in class C. In the replenishment module, demand forecasting is combined with current stock, safety stock, and lead time to calculate the reorder point and support purchase recommendations.

[1] Universidade de Araraquara, Graduando em Sistemas de Informação.
[2] Universidade de São Paulo, Doutor em Engenharia de Software.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

The results demonstrate the technical feasibility of the prototype as a decision support layer; however, validation occurred exclusively in a simulated computational environment, with a preliminary retrospective evaluation of predictive accuracy, but without measurement of real operational gains. In this scenario, the integrated operation of the prototype's modules was verified. Automatic integration with third-party ERPs remains as future work.

KEYWORDS: *ABC classification. Inventory management. Predictive modeling. MongoDB. Streamlit.*

RESUMEN
La gestión de inventarios supone un desafío estratégico para pequeños minoristas: equilibrar la disponibilidad de mercancías y el capital inmovilizado. Este trabajo tiene como objetivo desarrollar la plataforma Previsio, un sistema de apoyo a la decisión para la gestión de inventario basado en la clasificación ABC y el modelado predictivo. El prototipo se implementó en Python, utilizando Regresión Lineal para estimar la demanda y combinando esta estimación con reglas de inventario, tiempo de reposición (lead time) e historial de costos; MongoDB Atlas realiza la persistencia de los datos y Streamlit compone la interfaz. La validación funcional se realizó con una base sintética de 24 meses que contiene 160 productos, 56.766 registros de ventas y 125 facturas de entrada. La clasificación ABC identificó 38 artículos en la clase A, responsables del 78,21% de la facturación simulada, 40 en la clase B y 82 en la clase C. En el módulo de reabastecimiento, la previsión de demanda se combina con el inventario actual, el stock de seguridad y el tiempo de reposición para calcular el punto de reabastecimiento y apoyar la sugerencia de compra. Los resultados demuestran la viabilidad técnica del prototipo como capa de apoyo a la decisión; sin embargo, la validación ocurrió exclusivamente en un entorno computacional simulado, con una evaluación retrospectiva preliminar de la precisión predictiva, pero sin medición de ganancias operativas reales. En este escenario, se verificó el funcionamiento integrado de los módulos del prototipo. La integración automática con ERPs de terceros permanece como trabajo futuro.

PALABRAS CLAVE: *Clasificación ABC. Gestión de inventarios. Modelado predictivo. MongoDB. Streamlit.*

INTRODUÇÃO

A gestão de estoques é um desafio estratégico para pequenos varejistas brasileiros. Segundo o Relatório Executivo GEM Brasil 2024, cerca de 46,9 milhões de brasileiros participavam de negócios iniciais ou estabelecidos, evidenciando a diversidade empreendedora (ANEGEPE; SEBRAE, 2024). Diferenças de escolaridade, renda e maturidade dos negócios reforçam a necessidade de ferramentas claras e acessíveis. Um desafio central é equilibrar capital imobilizado e disponibilidade de produtos. Dias (2010) considera o estoque um “mal imprescindível”: sua falta interrompe vendas e afeta clientes, enquanto o excesso reduz a rotatividade, dificulta a armazenagem e compromete recursos financeiros. O controle empírico ou manual favorece erros, rupturas e acúmulo de itens de baixo giro.

A classificação de produtos, especialmente pela Curva ABC, pode amenizar esses problemas. Esse método “[...] permite identificar aqueles itens que justificam atenção e tratamento adequados quanto à sua administração. Obtém-se a curva ABC através da

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

ordenação dos itens conforme a sua importância relativa" (Dias, 2010, p. 69). Contudo, a classificação estática não contempla integralmente a dinâmica da demanda; por isso, o trabalho associa a priorização financeira a um modelo preditivo baseado no histórico de vendas.

Técnicas de aprendizagem de máquina podem apoiar a estimativa de demanda a partir de séries históricas. Ermelindo (2025) demonstra o emprego de regressão na previsão de demanda e no planejamento de ressuprimento. No Previsio, a Regressão Linear estima apenas a tendência da quantidade vendida; a recomendação de compra é construída com regras de estoque, *lead time*, custos históricos e cotações.

Este trabalho tem como objetivo geral desenvolver o Previsio, sistema de apoio à decisão para gestão de inventário que processa históricos de vendas, estoque e compras em uma base MongoDB configurável. Para isso, foram desenvolvidas rotinas em Python de conexão, organização, consulta e análise dos dados. A integração automática com bancos de dados de ERPs de terceiros, originalmente prevista como camada de ETL, permanece como possibilidade futura e não integra a versão atualmente validada.

O trabalho se justifica pela necessidade de tornar técnicas de gestão de estoques e análise de dados acessíveis a pequenos varejistas, independentemente da familiaridade do gestor com estatística ou ferramentas de *Business Intelligence*. O perfil heterogêneo dos empreendedores brasileiros apresentado pelo GEM Brasil 2024 reforça a importância de soluções que simplifiquem a interpretação das informações. O Previsio busca apresentar esses fundamentos de maneira simples, interativa e objetiva.

Diante dos desafios à sustentabilidade econômica de micro, pequenos e médios empreendedores, uma logística de estoque eficiente é relevante para a permanência e a competitividade no mercado. Conforme descreve Inácio (2016, p. 14):

> [...] a Logística orienta que nos dias atuais, para se visualizar a diferença entre uma empresa vencedora e uma perdedora deve-se vincular a Administração da Logística aplicada pelos seus administradores. E esta diferença irá refletir não só na redução de seus custos, como também na satisfação de seus clientes.

Nesse contexto, o Previsio busca apoiar decisões de compra pela análise de dados de estoque e vendas, sem pressupor reduções comprovadas de custos ou capital imobilizado.

Como problema de pesquisa, busca-se responder à seguinte questão: Como uma ferramenta de apoio à decisão baseada em análise de vendas, classificação ABC e previsão de demanda pode auxiliar pequenos varejistas na gestão de estoques e na elaboração de recomendações de compra?

A hipótese é de que o Previsio, como camada de apoio à decisão baseada em Python e MongoDB, pode automatizar os cálculos de Regressão Linear, Curva ABC e ressuprimento e

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

apresentá-los por meio da interface Streamlit. Espera-se que essa organização facilite decisões de compra mais fundamentadas; entretanto, os efeitos sobre o capital imobilizado e o espaço físico dependem de validação posterior em ambiente varejista real.

O procedimento técnico baseia-se em experimentação computacional com protótipo funcional. A validação utilizou uma base sintética no MongoDB Atlas, permitindo observar a classificação ABC, as previsões de demanda, o ponto de ressuprimento e as sugestões de compra do sistema. Nesta etapa, não foi realizado estudo de caso com uma empresa varejista real.

O público-alvo compreende micro e pequenas empresas do varejo. Para a validação técnica, desenvolveu-se um cenário computacional sintético de *pet shop*, utilizando um script Python para gerar 730 dias de operações simuladas. Foram definidos perfis de giro, probabilidades de venda, variações sazonais de demanda e custos, níveis de estoque e prazos de fornecimento. A geração empregou a semente aleatória 42 para favorecer a reprodutibilidade, embora o período simulado dependa da data de execução. A base reúne registros de produtos, fornecedores, vendas, movimentações de estoque, notas fiscais e cotações, permitindo testes funcionais dos algoritmos sem representar operações comerciais reais.

1. FUNDAMENTAÇÃO TEÓRICA

A revisão fundamenta as decisões de projeto do Previsio ao relacionar Sistemas de Apoio à Decisão (SAD), gestão de inventário, Python, Streamlit, Curva ABC e modelagem preditiva aos módulos implementados, do processamento dos dados à apresentação de resultados.

1.1. Sistemas de Apoio à Decisão (SAD)

A eficácia de um Sistema de Informação não depende apenas dos bastidores (*back-end*), mas também da transformação de dados operacionais em informações úteis à decisão. Segundo O'Brien e Marakas (2007), um Sistema de Apoio à Decisão (SAD) combina dados, modelos e recursos de interação para auxiliar o gestor na análise de problemas. Assim, o SAD não substitui o responsável pela decisão, mas organiza evidências e cenários para escolhas mais fundamentadas.

No Previsio, esse conceito é aplicado pela separação entre a camada de dados, os módulos analíticos e a camada de apresentação. Os registros de vendas, estoque, fornecedores, notas fiscais e cotações permanecem no MongoDB; as rotinas em Python calculam classificação, demanda e ressuprimento; e o Streamlit apresenta os resultados por meio de indicadores, filtros,

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

gráficos e recomendações. Essa organização permite ao usuário visualizar os resultados sem interpretar diretamente consultas ao banco ou operações matemáticas.

1.2. Processamento de Dados de Inventário

A gestão de estoques, na perspectiva de Sistemas de Informação, envolve variáveis críticas para a saúde financeira da organização. Dias (2010) conceitua o estoque como um "mal imprescindível", exigindo algoritmos capazes de equilibrar a disponibilidade de itens e o capital imobilizado. A eficiência logística mencionada por Inácio (2016) depende da qualidade do processamento desses dados, transformando registros brutos em indicadores de desempenho que auxiliam na redução de custos operacionais e na melhoria do nível de serviço.

O Previsio utiliza MongoDB Atlas, banco NoSQL orientado a documentos. Seu modelo flexível admite campos e tipos distintos em uma coleção (MongoDB, s. d.), organizando produtos, vendas, estoque, fornecedores e notas fiscais. Entretanto, a interoperabilidade requer estruturas compatíveis, pois a conversão automática de esquemas de ERPs ainda não foi implementada.

1.3. Python

A linguagem Python dispõe de bibliotecas voltadas à computação científica e numérica, à análise e ao processamento de dados (Python Software Foundation, s. d.). No Previsio, constitui o núcleo lógico do protótipo: os módulos separam acesso aos dados, classificação ABC, previsão de demanda, inteligência de compra e interface, facilitando a organização e manutenção do código.

A biblioteca Pandas organiza vendas em estruturas tabulares, agrupa registros por produto, data, mês e fornecedor e calcula médias e valores acumulados. O NumPy auxilia operações numéricas, como somatórios de demanda e projeção de estoque. Já o Scikit-Learn fornece a classe LinearRegression para estimar a tendência temporal da quantidade vendida, sem decidir isoladamente quanto ou quando comprar, pois essa decisão depende de outras etapas do motor de ressuprimento (Scikit-learn Developers, s. d.).

1.4. Streamlit

O *framework* Streamlit permite criar aplicações web interativas a partir de scripts Python (Streamlit, 2026). No Previsio, forma a camada de apresentação do SAD e reúne os módulos de Ressuprimento, Visão Geral e Curva ABC, Dados, Vendas e Histórico, Notas Fiscais de Entrada e gerenciamento de acessos. Os resultados do *back-end* são exibidos em painéis de controle (*dashboards*), filtros, tabelas, métricas e mensagens de apoio à decisão.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Como as interações reexecutam o script, o sistema utiliza memória temporária (*cache*) para evitar consultas repetidas ao MongoDB. A interface recebe parâmetros de histórico, horizonte de previsão e produtos exibidos, enquanto os cálculos permanecem nos módulos Python.

1.5. Algoritmos de Priorização: A Lógica da Curva ABC

No Previsio, a Curva ABC é calculada dinamicamente a partir do faturamento histórico de cada produto. O algoritmo agrupa os registros de vendas por produto, soma o campo de valor total, ordena os itens em ordem decrescente de faturamento e calcula o percentual acumulado de participação no faturamento global. A classificação implementada atribui classe A aos itens até 80% do faturamento acumulado, classe B aos itens acima de 80% e até 95%, e classe C aos demais itens.

Essa lógica está de acordo com o princípio de priorização financeira associado à Curva ABC. Conforme discutido por Ribeiro, Lopes e Correia (2020), a técnica permite diferenciar itens segundo sua importância relativa e apoiar políticas de controle de estoque. No protótipo, a classe ABC é utilizada como informação de priorização em filtros, gráficos e tabelas, enquanto os cálculos de previsão e ressuprimento continuam disponíveis para os produtos das três classes.

2. METODOLOGIA

2.1. Caracterização da pesquisa

A pesquisa bibliográfica acompanhou o desenvolvimento do Previsio, concebido como camada de análise, não como ERP completo. O usuário informa URL e nome da base MongoDB, e a aplicação consulta coleções com estrutura compatível. Conectores automáticos para bancos de ERPs de terceiros permanecem como trabalho futuro.

Esta pesquisa classifica-se como aplicada, pois objetiva a geração de um produto tecnológico, denominado software Previsio, voltado à solução de um problema prático do setor varejista. A abordagem é quantitativa, baseada no desenvolvimento tecnológico e em experimentação computacional com testes funcionais em ambiente simulado, pois utiliza dados numéricos de vendas, estoque e custos para executar cálculos, observar o comportamento do protótipo e verificar sua viabilidade funcional (Gil, 2022).

2.2. Modelagem da demanda e regras de ressuprimento

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

O motor analítico utiliza Python e a biblioteca Scikit-Learn para estimar a demanda diária por Regressão Linear sobre o histórico de quantidades vendidas. A decisão de ressuprimento é complementada por cálculos de estoque, *lead time* e nível-alvo, enquanto o histórico de notas fiscais e cotações é utilizado para indicar um período de compra potencialmente mais vantajoso segundo o histórico de custos e um fornecedor de referência. Dessa forma, o Previsio transforma cálculos e dados operacionais em indicadores e recomendações organizadas para apoio à decisão.

No Previsio, a Regressão Linear representa apenas a primeira etapa quantitativa do processo de apoio à compra. Sua função é estimar a tendência da quantidade vendida ao longo do tempo. A recomendação final não é obtida diretamente pela reta: o resultado da regressão é posteriormente combinado com estoque atual, estoque mínimo, estoque de segurança, estoque máximo, *lead time* do fornecedor, histórico de custos de entrada e cotações. Assim, a expressão "previsão de demanda" é mais adequada ao papel da Regressão Linear no sistema do que "previsão de compra".

As vendas são filtradas por período e agrupadas diariamente; datas sem movimentação recebem zero para manter a continuidade temporal. No modelo adotado, a quantidade diária vendida constitui a variável dependente, enquanto a posição temporal representa a variável independente. Quando a série diária construída possui menos de dois períodos ou apresenta consumo acumulado nulo, utiliza-se a média histórica como alternativa ao ajuste da Regressão Linear. Havendo dados suficientes, usa-se a função linear simples descrita por Westphal et al. (2023):

$$y = \beta_0 + \beta_1 x \qquad (1)$$

Onde y representa a quantidade diária estimada, x representa a posição cronológica do dia analisado, β0 corresponde ao intercepto e β1 ao coeficiente angular que expressa a tendência de crescimento ou redução do consumo. Após a geração dos valores futuros, estimativas negativas de demanda são ajustadas para zero, pois não possuem significado operacional. Se a projeção resultante for integralmente nula apesar da existência de consumo histórico positivo, o sistema volta a utilizar a média diária observada.

O preenchimento dos dias sem movimentação com quantidade zero permite manter a continuidade temporal da série, mas não diferencia ausência efetiva de demanda, indisponibilidade do produto ou falhas de registro. Dessa forma, produtos com demanda intermitente são representados por históricos com períodos sem vendas, sem a aplicação de um modelo específico para esse comportamento. A implementação também não realiza tratamento individualizado de valores atípicos nem considera diretamente promoções ou fatores sazonais

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

como variáveis explicativas da Regressão Linear, que utiliza apenas a posição temporal. Embora a base sintética contemple sazonalidade na geração das vendas, essa característica não é incorporada explicitamente ao modelo preditivo, constituindo uma limitação a ser considerada na interpretação das estimativas.

Para fins de exemplificação matemática, considera-se uma série simplificada de quatro períodos, em que $x1 = 1$, $x2 = 2$, $x3 = 3$ e $x4 = 4$. O exemplo tem finalidade didática para demonstrar a interpretação da reta; no sistema real, o cálculo é executado sobre a série diária construída a partir do histórico selecionado.

Caso o ajuste produza, para esse exemplo, β0 = 120 unidades e β1 = 15 unidades por período, a equação preditiva é consolidada conforme demonstrado na Equação 2:

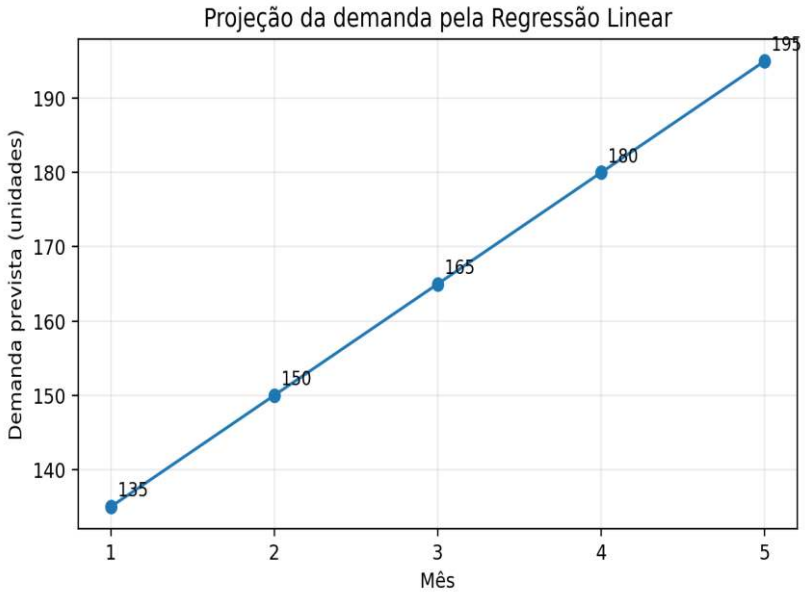
$y = 120 + 15x$ (2)

Para estimar a demanda do quinto período absoluto ($x = 5$), aplica-se a mesma função:

$y = 120 + 15(5) = 195$ unidades (3)

Dessa forma, as 195 unidades representam somente a demanda estimada para o período do exemplo. Esse valor não corresponde, por si só, a uma ordem de compra. No código do Previsio, a demanda é projetada diariamente para o horizonte escolhido e depois acumulada conforme a necessidade de cada cálculo operacional.

Figura 1. Projeção da demanda mensal baseada em regressão linear simples



Fonte: elaboração própria (2026).

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

A etapa seguinte transforma a estimativa de demanda em risco de reposição. O sistema soma a demanda prevista para os dias correspondentes ao *lead time* do fornecedor e acrescenta o estoque de segurança. Essa relação determina o ponto de ressuprimento, seguindo a lógica de proteção contra a demanda que ocorrerá enquanto o novo pedido ainda estiver em trânsito (Dias, 2010):

PR = DLT + ES (4)

Em que PR é o ponto de ressuprimento, DLT é a demanda acumulada prevista durante o *lead time* e ES é o estoque de segurança. Em seguida, o sistema calcula o estoque projetado subtraindo cumulativamente a demanda prevista do estoque atual. Com isso, identifica o primeiro dia em que o saldo alcança o ponto de ressuprimento e o primeiro dia em que pode ocorrer ruptura. O cálculo pode ser representado de forma simplificada por:

$E_t = \max(0, E_0 - \Sigma \hat{y}_i)$ (5)

Os parâmetros de estoque mínimo, estoque máximo e estoque de segurança são obtidos dos respectivos campos cadastrados para cada produto no MongoDB, não sendo estimados pela Regressão Linear. Na base sintética, esses valores foram inicialmente definidos pelo gerador `seed_petshop.py`, conforme os perfis de giro estabelecidos. O estoque mínimo foi calculado considerando 70% do estoque de segurança, respeitando o limite mínimo de cinco unidades. Já o ponto de ressuprimento é recalculado dinamicamente pelo módulo `forecasting.py`, utilizando a demanda prevista durante o prazo de entrega e o estoque de segurança cadastrado. O *lead time* é obtido do cadastro do fornecedor, sendo utilizado o valor padrão de cinco dias quando essa informação não está disponível. Na implementação atual, esse prazo é tratado como fixo para cada cálculo, sem modelagem de suas possíveis variações, e os pedidos de compra pendentes de recebimento não são considerados explicitamente na projeção do saldo disponível.

A partir desses limites, o produto é classificado como CRÍTICO quando o estoque atual está no nível ou abaixo do estoque de segurança; como ALERTA quando já atingiu o ponto de ressuprimento ou deve alcançá-lo dentro do horizonte analisado; e como NORMAL quando o saldo projetado permanece acima desses limites. Portanto, a data operacional de compra é determinada pelo risco de estoque, não apenas pela tendência da regressão.

O histórico de notas fiscais é agrupado por mês para identificar, como heurística, o menor custo médio observado. Notas e cotações também permitem comparar fornecedores por família de produtos. Essas indicações não garantem os mesmos preços futuros, pois dependem de promoções, inflação, condições comerciais, quantidades adquiridas e prazos de entrega. Os preços de venda alimentam a Curva ABC, enquanto os custos apoiam a inteligência de compra.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

A data sugerida combina as duas análises. Se a próxima ocorrência do mês historicamente mais barato acontecer antes da data em que o estoque alcançará o limite operacional, esse período de compra potencialmente mais vantajoso pode ser escolhido. Caso a janela de menor custo ocorra depois do risco de reposição, o algoritmo prioriza evitar ruptura e mantém a data operacional. Desse modo, a recomendação considera simultaneamente tendência de consumo, disponibilidade de estoque e histórico de custos.

Por fim, o nível-alvo de reposição é definido pelo maior valor entre estoque máximo, ponto de ressuprimento, estoque mínimo e estoque de segurança. Para calcular a quantidade, o sistema estima quanto estoque restará na data de recebimento, considerando os dias até o pedido e o *lead time*, e repõe somente a diferença necessária até o nível-alvo:

$$Q = \text{ceil}(\max(0, N_alvo - E_recebimento)) \quad (6)$$

A adoção do maior valor entre os parâmetros de estoque busca impedir que o nível-alvo de reposição seja inferior à necessidade estimada durante o período de abastecimento. Entretanto, essa regra não estabelece o estoque máximo cadastrado como um limite obrigatório. Assim, quando o ponto de ressuprimento calculado supera o estoque máximo, o sistema pode definir um nível-alvo superior ao limite informado. A implementação atual não realiza uma validação específica dessa incompatibilidade nem considera diretamente restrições de capacidade de armazenamento ou orçamento disponível. Dessa forma, o aperfeiçoamento da validação dos parâmetros, o tratamento de pedidos pendentes e a consideração de restrições operacionais constituem possibilidades de desenvolvimento futuro.

Esse encadeamento mostra que o comportamento final do Previsio não é puramente linear. A Regressão Linear fornece uma tendência de demanda auditável, mas a decisão apresentada ao usuário resulta da combinação desse valor com regras de inventário e com a análise histórica de custos e fornecedores. Embora tenham sido realizados testes quantitativos preliminares de precisão preditiva, a utilização de dados sintéticos e as limitações identificadas nas regras de previsão e ressuprimento ainda exigem validação com históricos comerciais reais e avaliação de diferentes modelos e condições operacionais.

2.3. Geração e estruturação da base sintética

Para a validação funcional do Previsio, foi desenvolvido o módulo seed_petshop.py, responsável pela geração automatizada de uma base de dados sintéticos destinada à simulação de operações comerciais de um estabelecimento varejista do segmento *pet shop*. O script foi implementado em Python, utilizando as bibliotecas random e datetime para gerar operações

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

simuladas a partir de parâmetros predefinidos de demanda, sazonalidade, preços e estoques, com posterior armazenamento dos registros no MongoDB Atlas.

A base foi estruturada a partir de oito famílias de produtos, com vinte variações por família, totalizando 160 produtos. O histórico de vendas foi gerado para um período de 730 dias, correspondente a aproximadamente 24 meses de operações simuladas, contendo 56.766 registros de vendas e 125 notas fiscais de entrada. Para favorecer a reprodutibilidade, foi utilizada a instrução random.seed(42), que estabelece uma sequência fixa de números pseudoaleatórios. Entretanto, como a data inicial é calculada a partir de datetime.now(), a reprodução exata depende também da fixação da data de referência e da versão do código.

A simulação utiliza três perfis predefinidos de giro, identificados como A, B e C, com probabilidades diárias iniciais de venda de 78%, 42% e 16%, respectivamente. As quantidades por venda variam entre 2 e 7 unidades no perfil A, 1 e 4 no perfil B e 1 e 2 no perfil C. Esses perfis orientam a geração artificial da demanda e não correspondem necessariamente às classes ABC posteriormente calculadas pelo sistema, cuja classificação considera o faturamento acumulado.

A ocorrência de vendas é determinada por um cálculo probabilístico que combina o perfil de giro, um fator de sazonalidade mensal associado à família do produto e uma variação aleatória uniforme entre 0,75 e 1,18. A probabilidade resultante é limitada a 96%, e a quantidade vendida é sorteada dentro do intervalo previsto para cada perfil, conforme apresentado no Código 1.

Código 1. Simulação probabilística das vendas, trecho do módulo seed_petshop.py

```
fator_mes = fator_demanda_mensal( data_venda_base.month, prod["familia_produto"])
prob_venda = min( 0.96, prod["prob_venda"] * fator_mes * random.uniform(0.75, 1.18))

if random.random() > prob_venda:
    continue
quantidade = random.randint(
    prod["qtd_min"], prod["qtd_max"])
```

Fonte: elaboração própria (2026).

Conforme o Código 1, o fator mensal modifica a probabilidade de comercialização de acordo com a categoria do produto. Para produtos antipulgas, por exemplo, o gerador aplica um fator de 1,35 nos meses de setembro a fevereiro e de 0,85 nos demais meses, simulando

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

períodos de maior e menor procura. A função `random.random()` determina probabilisticamente a ocorrência da venda, enquanto `random.randint()` define a quantidade comercializada. Esse procedimento permite construir históricos com diferentes frequências de movimentação e oscilações sazonais.

Além da demanda, o gerador simula variações nos custos de aquisição por meio da função `fator_custo_mensal()`, que estabelece multiplicadores específicos para determinados meses do ano, conforme demonstrado no Código 2.

Código 2. Parâmetros de sazonalidade dos custos, trecho do módulo seed_petshop.py

```
def fator_custo_mensal(mes):
    descontos = {
        1: 0.86,
        2: 0.94,
        6: 0.95,
        7: 0.96,
        11: 0.92,
        12: 1.10,
    }
return descontos.get(mes, 1.0)
```

Fonte: elaboração própria (2026).

No Código 2, janeiro recebe um multiplicador de 0,86, representando uma redução simulada de 14% sobre o custo-base, enquanto dezembro utiliza 1,10, correspondente a um acréscimo de 10%. Nos meses sem multiplicador específico, utiliza-se o valor 1,0. O custo final também considera o fator de preço do fornecedor e uma variação aleatória uniforme entre 0,97 e 1,04. Esses parâmetros permitem avaliar a funcionalidade de identificação de períodos historicamente mais vantajosos para compra, sem pressupor que os padrões simulados correspondam ao comportamento real do mercado.

Em relação aos estoques, os saldos iniciais são gerados aleatoriamente entre aproximadamente 45% e 95% do estoque máximo definido para cada produto. Os níveis de segurança e os limites máximos são estabelecidos conforme o perfil de giro, enquanto os prazos de fornecimento variam entre três e sete dias, de acordo com o fornecedor cadastrado. As entradas de mercadorias são simuladas com base na demanda mensal estimada, na

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

sazonalidade e em probabilidades de compra de 95%, 72% e 38% para os perfis A, B e C, respectivamente.

Como limitação, o processo de geração registra vendas independentemente da disponibilidade efetiva de mercadorias e ajusta para zero os saldos finais negativos. Dessa forma, a base foi utilizada para verificar o funcionamento dos algoritmos e das regras de negócio em ambiente computacional simulado, não permitindo mensurar com fidelidade rupturas reais de estoque ou perdas comerciais.

3. DESENVOLVIMENTO DA PLATAFORMA PREVISIO

O desenvolvimento da plataforma Previsio foi estruturado para reduzir a complexidade de interpretação de dados de estoque e transformar registros operacionais em informações de apoio à decisão. A versão analisada neste trabalho constitui um protótipo funcional, composto por conexão ao MongoDB, processamento em Python, módulos de Curva ABC e previsão de demanda, cálculo de ressuprimento e interface web em Streamlit.

3.1. Arquitetura da aplicação

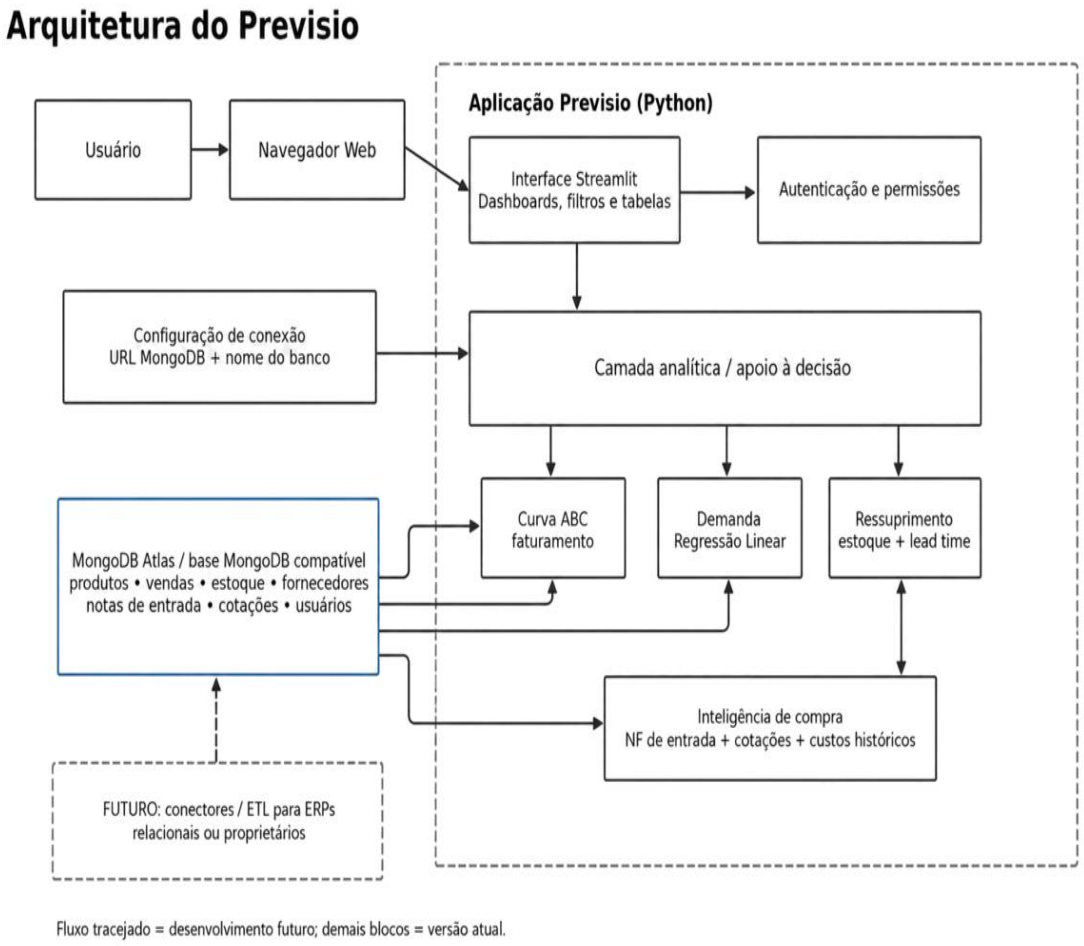
A arquitetura em camadas, apresentada na Figura 2, permite acesso via navegador à interface Streamlit. A aplicação Python realiza autenticação e análises, consultando o MongoDB para executar Curva ABC, previsão de demanda, ressuprimento e inteligência de compra. Um fluxo tracejado indica a futura integração com ERPs relacionais ou proprietários.

3.1.1. Camada de integração e acesso a dados

O módulo etl_pipeline.py concentra a configuração da conexão ao MongoDB, a criação de índices e a verificação do estado da base. A interface permite ao usuário informar a URL de conexão e o nome do banco de dados, possibilitando a consulta dinâmica às coleções com estrutura compatível e a utilização dos registros nos módulos analíticos. Essa configuração permite alternar entre diferentes bases MongoDB sem modificar as rotinas de cálculo. Entretanto, a extração e a transformação automáticas de dados provenientes de bancos relacionais ou de estruturas proprietárias de ERPs ainda não foram implementadas e permanecem como possibilidades de evolução. No protótipo, as senhas são armazenadas como hash SHA-256, sem salt individual. O cadastro utiliza um token fixo e a credencial inicial administrativa também é definida no código. Essas limitações, associadas à ausência de auditoria, exigem reforço da segurança antes da utilização em ambiente real.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 2. Arquitetura lógica do Previsio em camadas



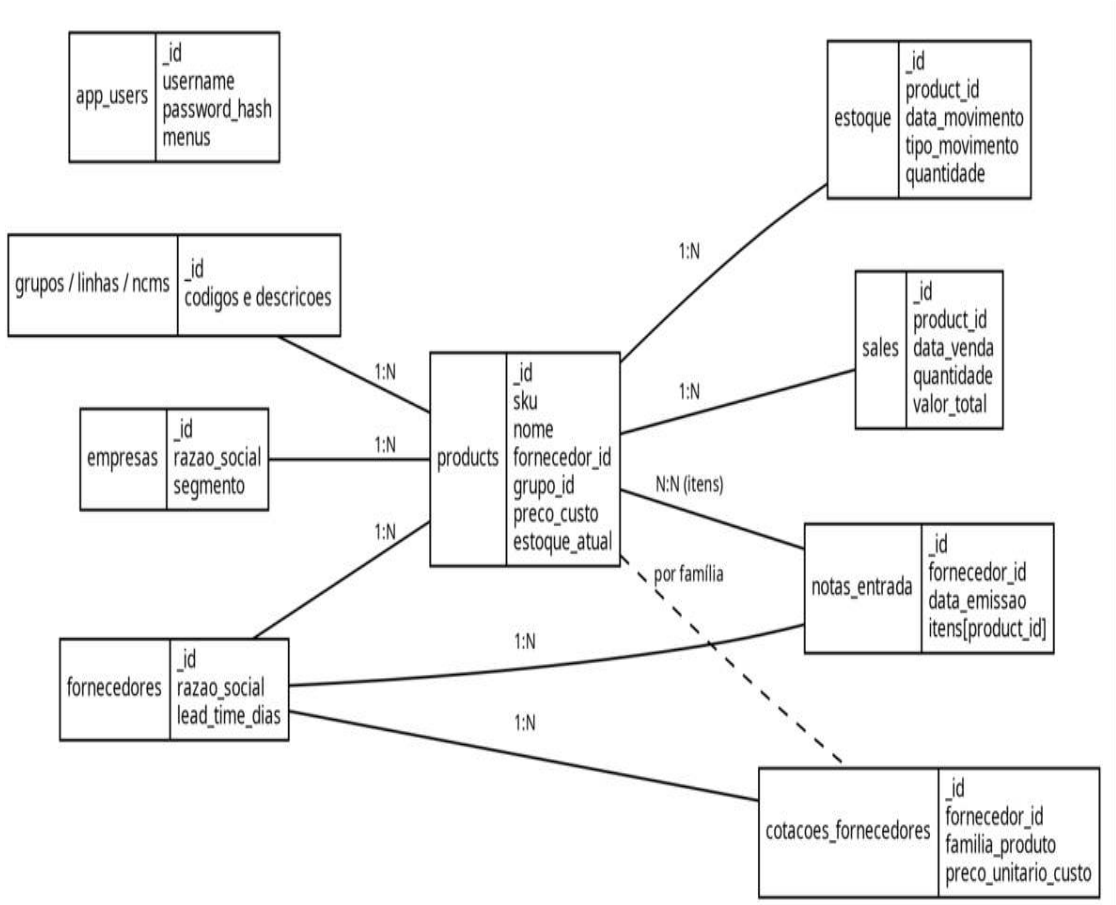
Fonte: elaboração própria (2026).

3.1.2. Modelo lógico de dados (DER simplificado)

A modelagem lógica do protótipo é composta por coleções de documentos do MongoDB relacionadas por identificadores. Entre as estruturas utilizadas estão empresas, produtos, fornecedores, grupos, linhas, NCMs, vendas, movimentos de estoque, notas fiscais de entrada, cotações de fornecedores e usuários da aplicação. Diferentemente do diagrama inicialmente planejado, as previsões não constituem uma entidade persistida obrigatoriamente: elas são calculadas pelo motor analítico a partir dos dados históricos quando solicitadas pela interface.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 3. Modelo lógico simplificado das coleções do protótipo



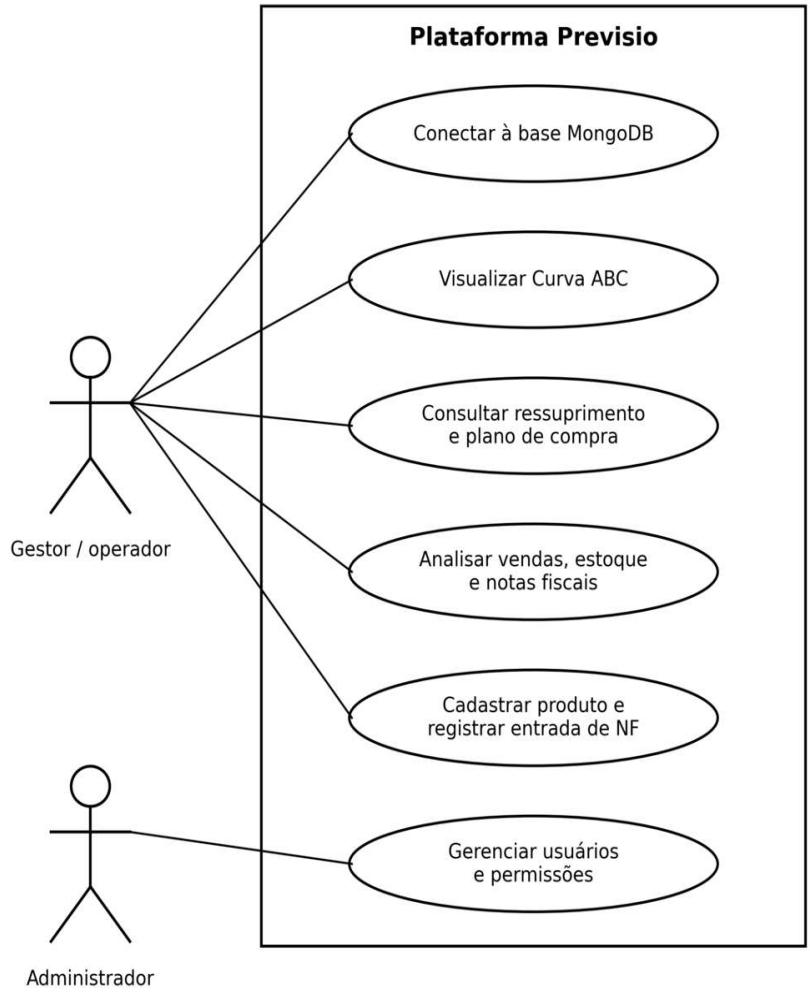
Fonte: elaboração própria (2026).

3.1.3. Diagrama de casos de uso

O diagrama de casos de uso representa as principais interações disponíveis no protótipo. Na Figura 4, o ator Gestor / operador reúne as funções operacionais: conectar a aplicação a uma base MongoDB compatível, visualizar a Curva ABC, consultar o ressuprimento e o plano de compra, analisar vendas, estoque e notas fiscais e registrar produtos ou entradas de mercadorias. O ator Administrador representa o perfil que possui, adicionalmente, a função de gerenciar usuários e permissões. A consulta dinâmica da base MongoDB não deve ser confundida com uma extração universal de ERPs: conectores para bancos relacionais ou estruturas proprietárias continuam como trabalho futuro.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 4. Diagrama de casos de uso da versão atual do Previsio



Fonte: elaboração própria (2026).

3.1.4. Fluxo dos módulos analíticos

Na classificação ABC, o sistema organiza as vendas por produto e soma o faturamento histórico de cada item. Em seguida, ordena os produtos do maior para o menor faturamento, calcula a participação individual no total e o percentual acumulado. A classificação adotada no protótipo atribui classe A aos itens até 80% do faturamento acumulado, classe B aos itens acima

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

de 80% e até 95% e classe C aos demais produtos. O resultado dessa etapa é utilizado nos filtros, indicadores e análises apresentadas ao usuário.

Na etapa de estimativa da demanda, o histórico do produto é filtrado pelo período selecionado e agrupado por dia. Dias sem movimentação são preenchidos com quantidade zero para manter a continuidade da série. A média diária é preservada como referência e também funciona como alternativa quando não existe histórico suficiente para o ajuste estatístico. Quando há dados adequados, a Regressão Linear utiliza a sequência cronológica como variável explicativa e as quantidades vendidas como variável de resposta. As estimativas negativas são limitadas a zero, pois não possuem significado operacional para a demanda.

A demanda futura é acumulada no horizonte e no *lead time*. O ponto de ressuprimento soma o consumo esperado nesse prazo ao estoque de segurança; o saldo é projetado descontando as vendas previstas. O resultado classifica o produto como CRÍTICO, ALERTA ou NORMAL. Notas fiscais e cotações indicam mês e fornecedor de menor custo médio, mas o período de compra mais vantajoso só prevalece se anteceder o risco de ruptura. A quantidade sugerida usa o nível-alvo e o estoque previsto para o recebimento.

3.2. Implementação técnica em Python e Streamlit

A implementação do Previsio foi organizada em módulos Python com responsabilidades separadas. O arquivo app.py coordena a interface, a navegação e a chamada dos módulos analíticos; etl_pipeline.py concentra a conexão e os utilitários de infraestrutura do MongoDB; abc_calculator.py executa a classificação ABC; e forecasting.py reúne a estimativa de demanda, o cálculo de ressuprimento e a inteligência de compra. A separação facilita testes independentes da interface com arquivos JSON e data fixa (10/08/2026). No ambiente conferido, estão instalados Scikit-learn 1.9.0, Streamlit 1.61.1 e PyMongo 4.17.0.

Entre as rotinas implementadas no módulo forecasting.py, destaca-se a lógica responsável pela definição do estado operacional do estoque. Após a previsão da demanda, o sistema calcula o ponto de ressuprimento a partir da demanda esperada durante o *lead time* e do estoque de segurança. Em seguida, projeta o consumo futuro do estoque e identifica quando o produto poderá atingir o ponto de ressuprimento ou sofrer ruptura. Com base nesses resultados, o item é classificado como CRÍTICO, ALERTA ou NORMAL, e o sistema determina se existe necessidade de compra e em qual horizonte ela deve ocorrer. O Código 3 apresenta um trecho dessa implementação.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Código 3. Trecho do módulo forecasting.py responsável pela lógica de ressuprimento do Previsio

```
    ponto_ressuprimento = round(demanda_lead_time + estoque_seguranca, 1)

    # Projeta o estoque futuro abatendo a demanda prevista dia a dia.
    datas_futuras = [hoje + timedelta(days=i) for i in range(1, forecast_days + 1)]
    demanda_acumulada = np.cumsum(previsao_diaria)
    estoque_projetado = np.maximum(0, estoque_atual - demanda_acumulada)

    # Descobre em quantos dias o estoque chega a zero ou ao ponto de ressuprimento.
    dias_ate_ruptura = _primeiro_dia_abaixo(estoque_projetado, 0)
    dias_ate_pr = 0 if estoque_atual <= ponto_ressuprimento else _primeiro_dia_abaixo(
        estoque_projetado,
        ponto_ressuprimento,
    )

    if dias_ate_ruptura is None:
        dias_ate_ruptura = int(estoque_atual / media_diaria) if media_diaria > 0 else 999

    if dias_ate_pr is None and media_diaria > 0 and estoque_atual > ponto_ressuprimento:
        dias_ate_pr = int(np.ceil((estoque_atual - ponto_ressuprimento) / media_diaria))
    elif dias_ate_pr is None:
        dias_ate_pr = 999

    # Define se a compra e imediata, futura ou apenas referencia.
    if estoque_atual <= estoque_seguranca:
        data_operacional = hoje
        status_estoque = "CRITICO"
        necessita_compra = True
        quando_operacional = "IMEDIATO (estoque critico)"
    elif estoque_atual <= ponto_ressuprimento:
        data_operacional = hoje
        status_estoque = "ALERTA"
        necessita_compra = True
        quando_operacional = "IMEDIATO (ponto de ressuprimento)"
    elif dias_ate_pr <= forecast_days:
        data_operacional = hoje + timedelta(days=max(1, dias_ate_pr))
        status_estoque = "ALERTA"
        necessita_compra = True
        quando_operacional = f"em {dias_ate_pr} dias"
    else:
        data_operacional = hoje + timedelta(days=max(1, min(dias_ate_pr, forecast_days)))
        status_estoque = "NORMAL"
        necessita_compra = False
        quando_operacional = f"sem compra prevista em {forecast_days} dias"

    insight = _insight_compra_produto(prod, fornecedor, purchase_intelligence, hoje)
    data_economica = insight["data_compra_economica"]
```

Fonte: elaboração própria (2026).

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Como o Streamlit reexecuta o script quando o usuário interage com filtros e controles, o projeto implementa *cache* temporário para os principais conjuntos de dados carregados do MongoDB. Produtos, fornecedores, grupos, vendas, notas fiscais de entrada e cotações são carregados e reutilizados durante a navegação; quando a origem de dados é alterada ou uma operação exige atualização, o *cache* é limpo para que a aplicação consulte novamente a base.

Na camada de apresentação, a barra lateral controla a conexão com o banco e a navegação entre os módulos liberados para o usuário. Os filtros de período, Curva ABC, fornecedor, produto e horizonte de análise são convertidos em parâmetros utilizados pelas rotinas Python. Após o processamento, o Streamlit organiza os resultados em métricas, tabelas e gráficos, mantendo a tomada de decisão com o usuário e utilizando o sistema como ferramenta de apoio, e não como mecanismo de compra automática.

O controle de acesso também faz parte da implementação. Os usuários da aplicação são mantidos na coleção app_users e, após a autenticação, a sessão registra o perfil e as abas permitidas. A navegação é montada de acordo com essas permissões, enquanto usuários com autorização administrativa podem cadastrar funcionários e ajustar os acessos disponíveis. Dessa forma, os componentes de interface, persistência, análise e controle de acesso atuam de maneira integrada no protótipo.

4. RESULTADOS E DISCUSSÃO

4.1. Validação funcional da Curva ABC

A validação funcional foi executada com a massa sintética gerada pelo próprio projeto para um cenário demonstrativo de *pet shop* varejista. O conjunto compreende aproximadamente 24 meses (730 dias) de histórico, contendo 160 produtos, 56.766 registros de vendas e 125 notas fiscais de entrada. Esses dados foram utilizados para verificar se as rotinas implementadas produzem resultados coerentes com as regras definidas no protótipo, sem a pretensão de representar o desempenho de uma empresa real.

A execução da Curva ABC classificou 38 produtos na classe A, 40 na classe B e 82 na classe C. Embora a classe A corresponda a 23,75% dos SKUs, seus itens concentraram 78,21% do faturamento simulado na validação funcional. As Figuras 5, 6 e 7 apresentam a tela de Visão Geral e Curva ABC, destacando os indicadores de estoque e faturamento, a distribuição dos SKUs por classe ABC, o valor imobilizado por grupo e a tabela geral de estoque.

Para apresentar quantitativamente a distribuição dos produtos e a participação financeira de cada classe, os resultados da classificação ABC foram organizados na Tabela 1. O algoritmo

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

implementado no módulo abc_calculator.py agrupa as vendas por produto, calcula o faturamento individual, ordena os valores de forma decrescente e determina a participação percentual acumulada. A classificação utiliza os limites de até 80% do faturamento acumulado para a classe A, acima de 80% até 95% para a classe B e acima de 95% para a classe C.

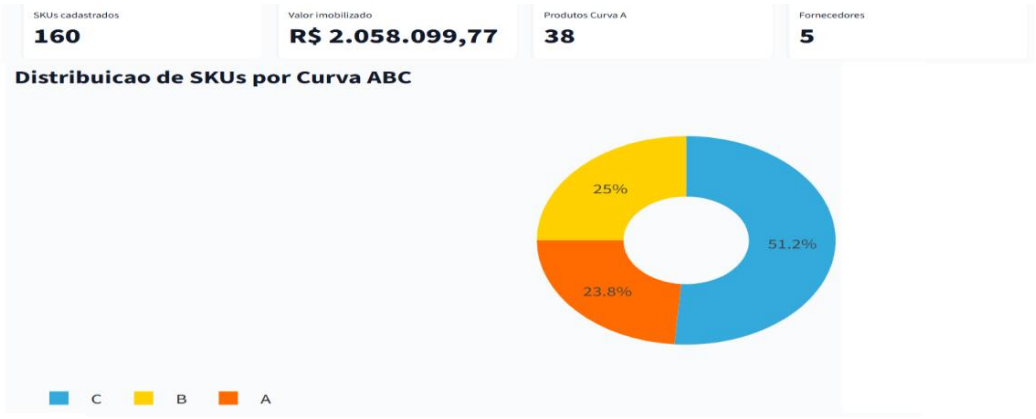
Tabela 1. Distribuição dos produtos e do faturamento por classe ABC

Classe ABC	Quantidade de produtos	Participação nos produtos (%)	Faturamento da classe (R$)	Participação no faturamento (%)
A	38	23,75%	17.912.360,13	78,21%
B	40	25,00%	3.806.504,59	16,62%
C	82	51,25%	1.185.002,95	5,17%
Total:	**160**	**100,00%**	**22.903.867,67**	**100,00%**

Fonte: elaboração própria (2026).

Conforme apresentado na Tabela 1, a classe A reúne 38 produtos (23,75% dos SKUs), responsáveis por R$ 17.912.360,13 (78,21%) do faturamento total simulado de R$ 22.903.867,67. A classe B compreende 40 produtos (25,00%), com faturamento de R$ 3.806.504,59 (16,62%), enquanto a classe C reúne 82 produtos (51,25%), correspondentes a R$ 1.185.002,95 (5,17%). Os resultados evidenciam a concentração do faturamento em uma parcela menor dos produtos, confirmando a priorização financeira obtida pelo algoritmo de classificação ABC.

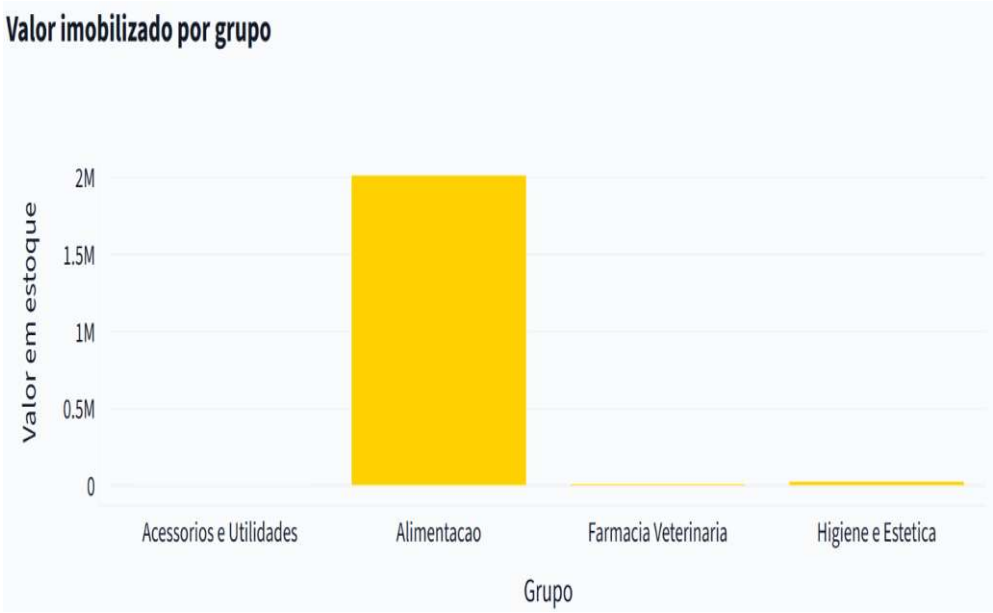
Figura 5. Indicadores e distribuição de SKUs por Curva ABC



Fonte: elaboração própria (2026).

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 6. Valor imobilizado por grupo



Fonte: elaboração própria (2026).

Figura 7. Tabela geral de estoque

Tabela geral de estoque

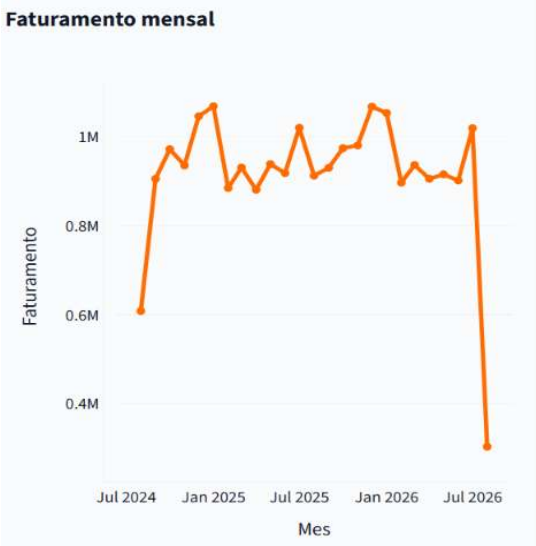
sku	nome	familia_produto	grupo_nome	fornecedor_nome	curva_abc	estoque_atual
PET-1013	Racao Premium Caes Adultos 15kg - Variacao 13	Racao Caes Adultos 15kg	Alimentacao	Royal Canin do Brasil Ltda	A	825

Fonte: elaboração própria (2026).

As Figuras 8, 9 e 10 apresentam, respectivamente, faturamento mensal e indicadores, demanda por classe ABC e gráfico de Pareto. Neste último, a ordenação dos produtos e a linha acumulada demonstram a concentração da receita nos itens mais relevantes, confirmando a classificação implementada como mecanismo de priorização.

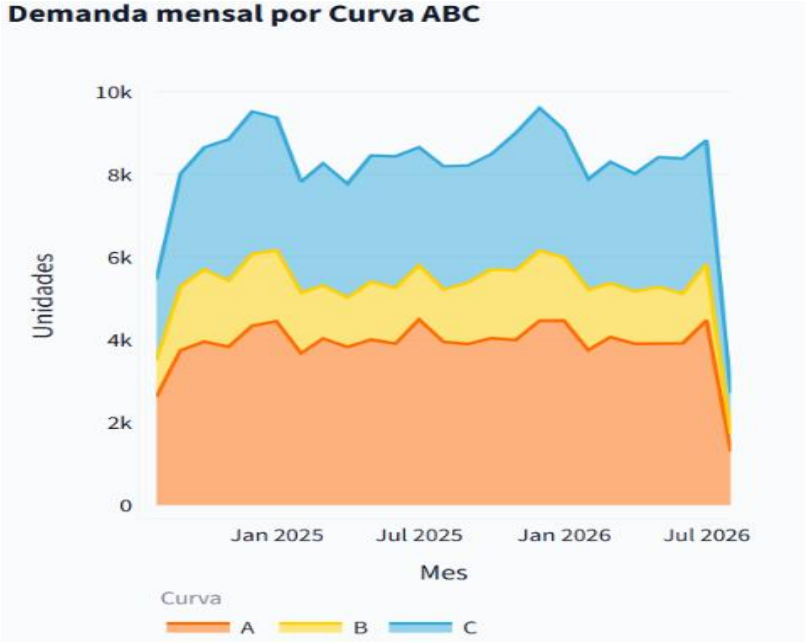
Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 8. Faturamento mensal e indicadores



Fonte: elaboração própria (2026).

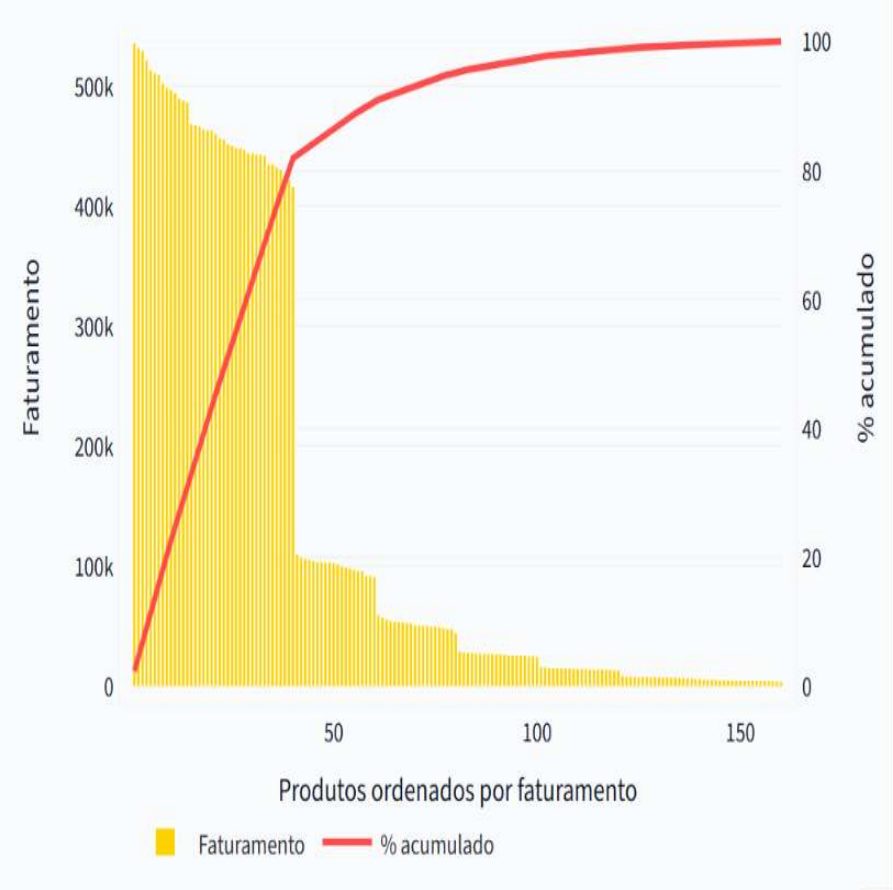
Figura 9. Demanda mensal por Curva ABC



Fonte: elaboração própria (2026).

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 10. Pareto ABC por faturamento



Fonte: elaboração própria (2026).

4.2. Previsão de demanda e ressuprimento

Para a análise de ressuprimento foi utilizado um horizonte de previsão de 180 dias. O sistema primeiro organiza o histórico diário de cada produto e estima a demanda futura; em seguida, acumula essa demanda no *lead time*, adiciona o estoque de segurança para formar o ponto de ressuprimento e projeta o saldo futuro. A recomendação ainda considera a janela de menor custo histórico observada nas notas fiscais e o fornecedor com menor custo médio entre compras e cotações disponíveis. As Figuras 11, 12, 13 e 14 apresentam a visão agregada desse processamento na tela de Ressuprimento Inteligente, incluindo os indicadores, o mapa interativo de compra, o risco de ressuprimento por Curva ABC e a decisão por produto.

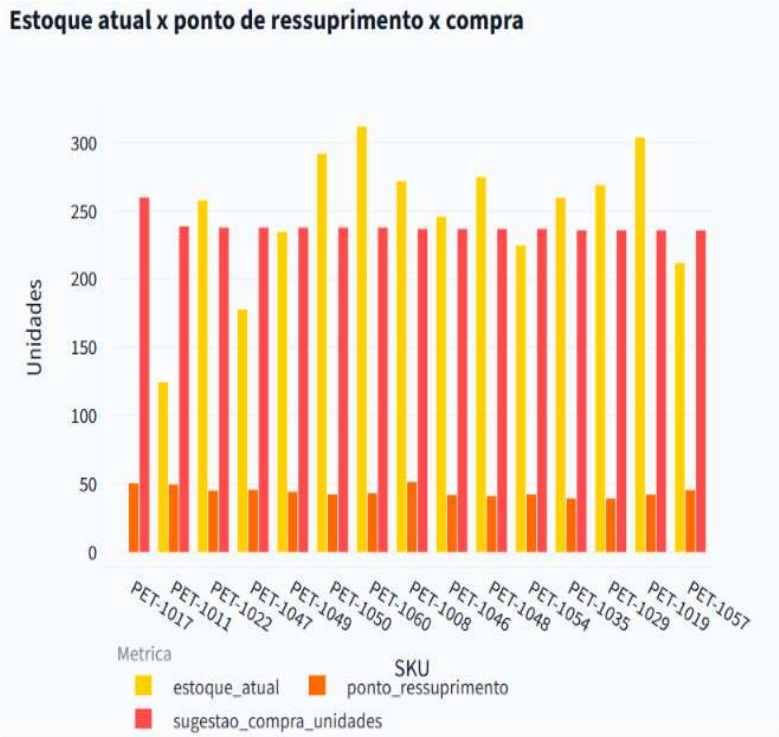
Figura 11. Filtros e indicadores do Ressuprimento Inteligente

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.



Fonte: elaboração própria (2026).

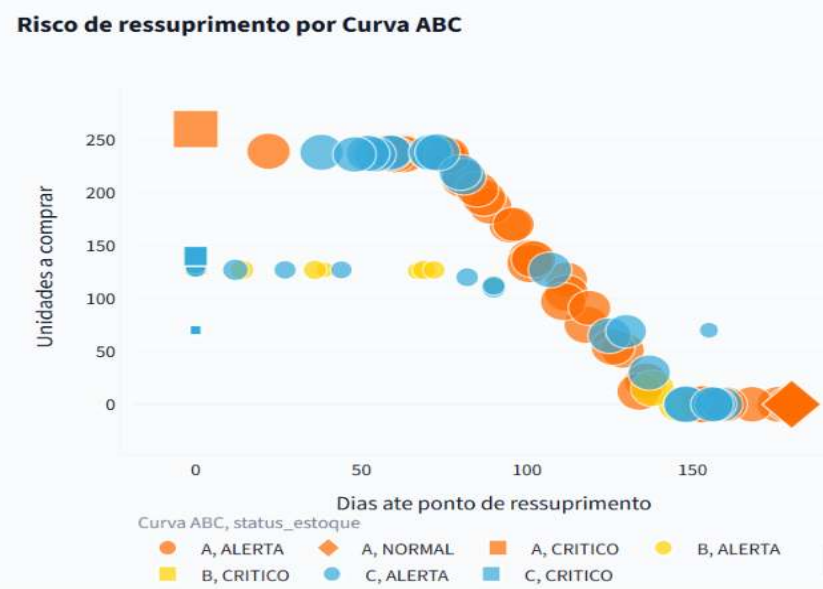
Figura 12. Mapa interativo de compra



Fonte: elaboração própria (2026).

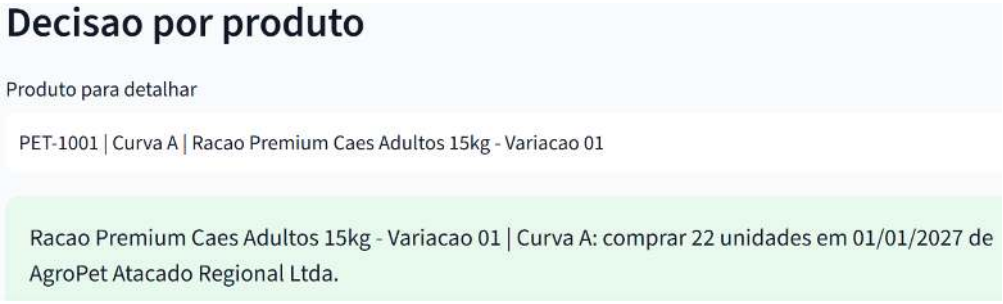
Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 13. Risco de ressuprimento por Curva ABC



Fonte: elaboração própria (2026).

Figura 14. Decisão por produto



Fonte: elaboração própria (2026).

Na execução registrada na interface, considerando um horizonte de previsão de 180 dias, 83 produtos foram classificados como CRÍTICOS e 144 dos 160 produtos apresentaram quantidade sugerida de compra superior a zero. Esses resultados decorrem das regras implementadas no módulo forecasting.py, que considera o estoque atual, o estoque de segurança, o ponto de ressuprimento e a demanda prevista. A classificação CRÍTICO ocorre quando o estoque atual é igual ou inferior ao estoque de segurança, enquanto as demais

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

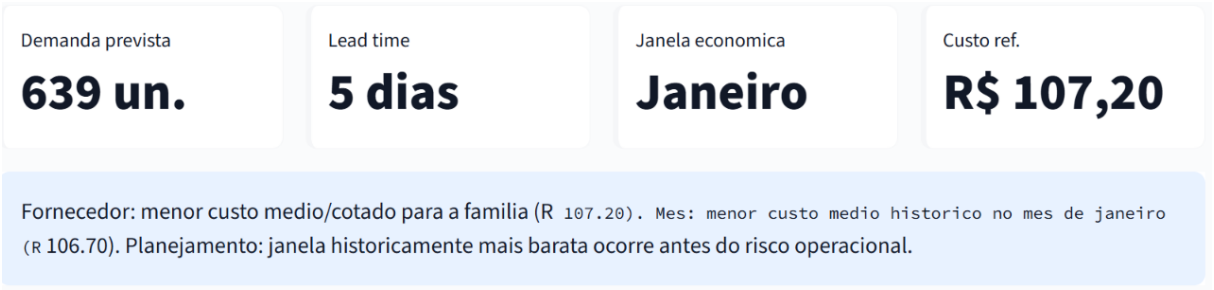
recomendações também contemplam produtos que atingem ou podem atingir o ponto de ressuprimento durante o horizonte analisado. Dessa forma, os 144 produtos não representam necessariamente compras imediatas, mas incluem necessidades de reposição identificadas ao longo dos 180 dias. Os valores refletem o comportamento do algoritmo sobre a base sintética e não devem ser interpretados como necessidades comprovadas de um estabelecimento real. Em ensaio analítico separado, com data de referência fixada em 10/08/2026, foram avaliadas variações do horizonte, do estoque atual e do *lead time* (Tabela 2). As contagens desse ensaio não correspondem à execução da interface apresentada anteriormente. As Figuras 15, 16 e 17 detalham a análise para um produto específico.

Tabela 2. Análise de sensibilidade dos parâmetros de ressuprimento

Horizonte	Condição testada	Críticos	Itens com compra	Unidades sugeridas
30 dias	Parâmetros originais	83	92	10.191
90 dias	Parâmetros originais	83	123	16.633
180 dias	Parâmetros originais	83	157	23.848
180 dias	Estoque atual −50%	87	160	25.277
180 dias	Estoque atual +50%	82	135	18.245
180 dias	Lead time +50%	83	157	23.981

Fonte: elaboração própria (2026).

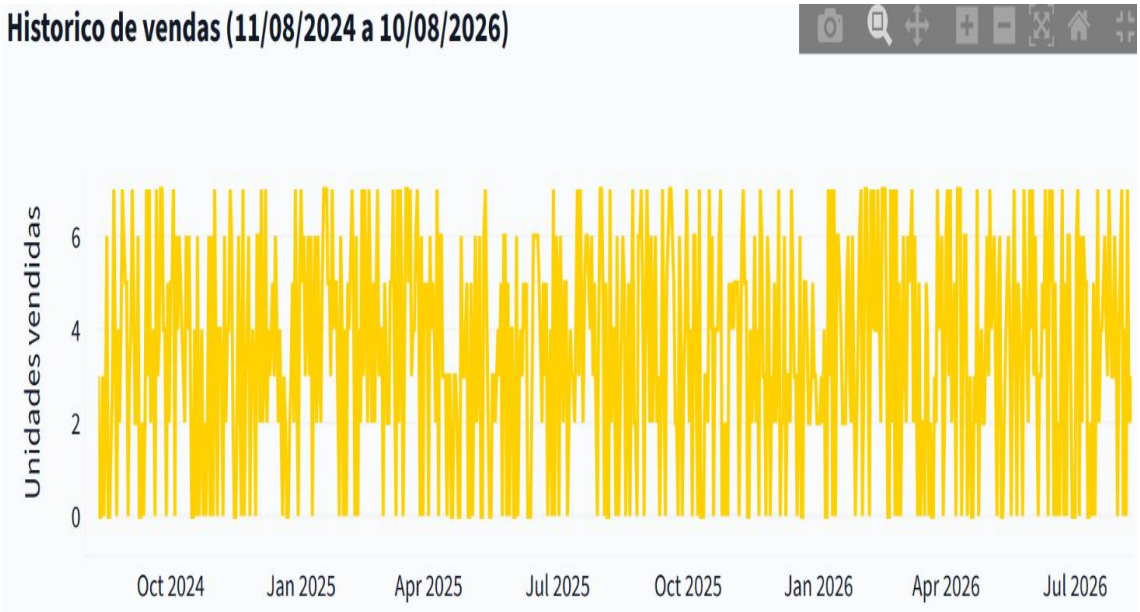
Figura 15. Indicadores da decisão de compra por produto



Fonte: elaboração própria (2026).

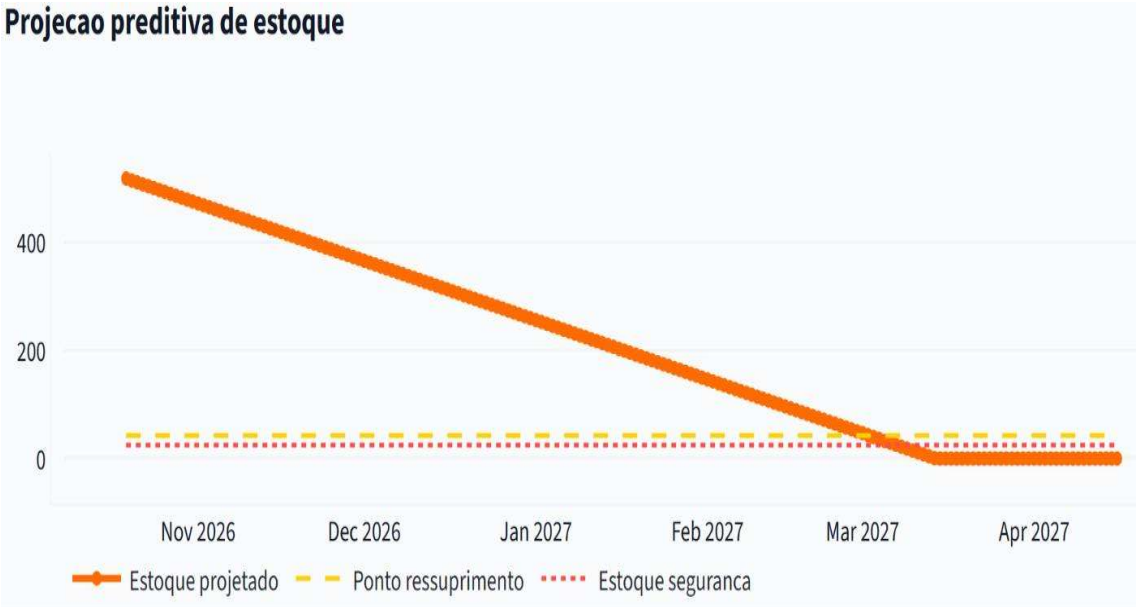
Figura 16. Histórico de vendas do produto

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.



Fonte: elaboração própria (2026).

Figura 17. Projeção preditiva de estoque



Fonte: elaboração própria (2026).

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

O protótipo cruza notas fiscais e cotações para sugerir fornecedor e período de menor custo médio. As notas são agrupadas por produto e mês; notas e cotações também são comparadas por família e fornecedor. Essa análise não altera a regressão da demanda, mas orienta quando e de quem comprar, priorizando evitar rupturas. Como os dados sintéticos têm sazonalidade definida no gerador, os resultados não comprovam economia real.

4.3. Avaliação quantitativa e limitações da validação

Como avaliação complementar, foi realizada uma análise retrospectiva da precisão das previsões de demanda a partir de registros exportados da base sintética do MongoDB Atlas. Foram utilizados os registros históricos anteriores a 12 de junho de 2026 para ajustar os modelos, reservando-se os 60 dias subsequentes, até 10 de agosto de 2026, para teste. A avaliação contemplou 160 produtos e 9.600 observações diárias. A Regressão Linear implementada no Previsio foi comparada com dois métodos de referência, utilizando as métricas Erro Absoluto Médio (MAE), Raiz do Erro Quadrático Médio (RMSE) e Erro Percentual Absoluto Ponderado (WAPE).

Tabela 3. Comparação da precisão dos métodos de previsão

Método	MAE (un.)	RMSE (un.)	WAPE (%)
Regressão Linear	1,323	1,724	75,58
Média histórica	1,332	1,726	76,11
Referência semanal simples	1,648	2,440	94,12

Fonte: elaboração própria (2026).

A Regressão Linear apresentou erros ligeiramente inferiores aos métodos de referência, embora a diferença em relação à média histórica tenha sido pequena. O WAPE de 75,58% também indica limitações relevantes na precisão das previsões diárias. Portanto, os resultados demonstram a possibilidade de avaliação quantitativa do modelo, mas não permitem afirmar superioridade preditiva consistente ou generalizar seu desempenho para estabelecimentos comerciais reais.

Nos testes controlados, o estoque zerado e o saldo igual ao nível de segurança (5 unidades) resultaram em status CRÍTICO, com sugestões de 20 e 15 unidades, respectivamente. O estoque suficiente (50 unidades) retornou NORMAL, sem necessidade de compra. Com estoque de 8 unidades e consumo diário de duas unidades, o resultado foi ALERTA, com sugestão de 20 unidades. Esses resultados corresponderam às regras implementadas. No

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

cenário adicional, o ponto de ressuprimento de 21 unidades, superior ao estoque máximo de 9, produziu nível-alvo de 21 unidades, evidenciando uma limitação operacional a ser corrigida.

5. CONSIDERAÇÕES FINAIS

O desenvolvimento do protótipo de apoio à decisão para gestão de estoques foi concluído no escopo funcional proposto. A validação com dados sintéticos demonstrou a integração da Curva ABC, da estimativa de demanda, das regras de ressuprimento e da interface Streamlit, sem comprovar ganhos operacionais reais.

Em relação ao problema de pesquisa, o protótipo demonstrou, em ambiente simulado, a integração das análises de apoio à decisão sem atuar como ERP completo. Na versão atual, essa coexistência ocorre por meio de uma conexão configurável com uma base MongoDB compatível. Assim, a hipótese é parcialmente confirmada no âmbito funcional: os cálculos e a apresentação simplificada foram automatizados, mas a integração ligar e usar (*plug-and-play*) com diferentes sistemas legados ainda depende do desenvolvimento de conectores e rotinas específicas de ETL.

Também não é possível concluir, nesta fase, que o sistema produz redução comprovada de capital imobilizado, otimização de espaço físico ou ganho financeiro, pois esses efeitos exigem aplicação em ambiente real e acompanhamento longitudinal. Como trabalhos futuros, recomenda-se desenvolver adaptadores para ERPs e bancos relacionais, realizar testes com empresas varejistas, avaliar a usabilidade da interface, ampliar a avaliação preditiva com outras bases e modelos e fortalecer os mecanismos de autenticação e segurança para um eventual uso em produção.

Como possibilidade de desenvolvimento futuro, propõe-se comparar a Regressão Linear atualmente utilizada no Previsio com outros modelos preditivos, como ARIMA, suavização exponencial e algoritmos baseados em árvores, considerando sazonalidade e valores históricos defasados. O Orange Data Mining também poderá ser utilizado em estudos futuros para apoiar análises exploratórias e comparações entre modelos, cuja eficácia deverá ser avaliada por métricas de erro e validação com dados reais.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

REFERÊNCIAS

ASSOCIAÇÃO NACIONAL DE ESTUDOS EM EMPREENDEDORISMO E GESTÃO DE PEQUENAS EMPRESAS (ANEGEPE); SERVIÇO BRASILEIRO DE APOIO ÀS MICRO E PEQUENAS EMPRESAS (SEBRAE). Empreendedorismo no Brasil 2024: relatório executivo GEM. [S. l.]: ANEGEPE: SEBRAE, 2024. 23 p. Disponível em: https://sebraepr.com.br/wp-content/uploads/2025/04/PUB_GEM-Relatorio-Executivo-Global-Entrepreneurship-Monitor-2024.pdf. Acesso em: 6 abr. 2026.

DIAS, Marco Aurélio P. *Administração de materiais: uma abordagem logística.* 5. ed. São Paulo: Atlas, 2010. 528 p.

ERMELINDO, Matheus da Silva. *Previsão de demanda utilizando regressão linear: um estudo de caso no Polo Industrial de Manaus.* 2025. 32 f. Trabalho de Conclusão de Curso (Graduação em Administração) – Faculdade de Estudos Sociais, Universidade Federal do Amazonas, Manaus, 2025. Disponível em: https://riu.ufam.edu.br/bitstream/prefix/9117/3/TCC_MatheusErmelindo.pdf. Acesso em: 5 abr. 2026.

GIL, Antônio Carlos. *Como elaborar projetos de pesquisa.* 7. ed. São Paulo: Atlas, 2022. 175 p.

INÁCIO, Gabriela Magossi. *Reestruturação do sistema de estoque e armazenagem por meio do programa WMS.* 2016. 41 f. Monografia (Especialização em Logística e Canais de Distribuição) – Núcleo de Educação a Distância, Universidade de Araraquara, Araraquara, 2016. Disponível em: https://www.uniara.com.br/download.php?chave=53191. Acesso em: 6 abr. 2026.

MONGODB. *Modelagem de dados no MongoDB.* [S. l.], [s. d.]. Disponível em: https://www.mongodb.com/pt-br/docs/manual/data-modeling/. Acesso em: 23 ago. 2026.

O'BRIEN, James A.; MARAKAS, George M. *Administração de sistemas de informação: uma introdução.* 13. ed. São Paulo: McGraw-Hill, 2007. 537 p.

PYTHON SOFTWARE FOUNDATION. *Applications for Python.* [S. l.], [s. d.]. Disponível em: https://www.python.org/about/apps/. Acesso em: 23 ago. 2026.

RIBEIRO, P. P.; LOPES, C. M.; CORREIA, A. M. Avaliação da gestão de estoque em uma microempresa de autopeças utilizando a curva ABC como ferramenta de apoio. *Revista Cereus*, Gurupi, v. 12, n. 2, p. 130-146, maio/ago. 2020. DOI: https://doi.org/10.18605/2175-7275/cereus.v12n2p130-146. Disponível em: https://ojs.unirg.edu.br/index.php/1/article/view/2791/1641. Acesso em: 5 abr. 2026.

SCIKIT-LEARN DEVELOPERS. *LinearRegression.* Scikit-learn 1.9.0 documentation. [S. l.], [s. d.]. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html. Acesso em: 23 ago. 2026.

STREAMLIT. Streamlit: a faster way to build and share data apps. Versão 1.61.1. [S. l.], 2026. Disponível em: https://streamlit.io/. Acesso em: 23 ago. 2026.

WESTPHAL, Eduardo *et al.* Análise preditiva de vendas em uma indústria de alimentos com uso do Software R: Modelos de Regressão Linear Simples, Suavização Exponencial Dupla e ARIMA.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

In: CONGRESSO BRASILEIRO DE ENGENHARIA DE PRODUÇÃO, 13., 2023, Ponta Grossa. *Anais [...]*. Ponta Grossa: APREPRO, 2023. Disponível em: https://aprepro.org.br/conbrepro/anais/2023/arquivos/10012023_101050_65196f620a314.pdf. Acesso em: 24 maio 2026.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.