RBMC
Revista Brasileira Militar de Ciências

Revista Brasileira Militar de Ciências, v. 11, n. 25, e198, 2025
ISSN 2447-9071 | DOI: https://doi.org/10.36414/rbmc.v11i25.198

Recebido: 24/06/2025 | Aceito: 27/06/2025 | Publicado: 03/07/2025

Sistema de recomendação em e-commerce com modelos de LLM e computação em nuvem: um relato de experiência

A cloud-based e-commerce recommendation system using LLMs: an experience report

Pablo Felix da Silva[1]
Laís Silva dos Santos[2]
Abner Luiz Pascoal de Oliveira[3]
Ana Clara Fernandes Costa[4]
Jonas Augusto Kunzler[5]

Resumo

Este trabalho apresenta o desenvolvimento da plataforma Diego, um sistema de recomendação inteligente voltado ao e-commerce, que integra modelos de linguagem natural (LLMs), serviços em nuvem e boas práticas de engenharia de software. A motivação do projeto surgiu da necessidade crescente de oferecer recomendações personalizadas em ambientes de compra cada vez mais dinâmicos e competitivos. A metodologia adotada envolveu revisão bibliográfica, desenho arquitetural, implementação e testes. A arquitetura do sistema foi estruturada segundo o padrão MVC, com backend desenvolvido em Python e FastAPI, frontend com ReactJS e autenticação via Firebase. A infraestrutura foi concebida em ambiente multicloud, combinando serviços da AWS (S3 para armazenamento de arquivos) e GCP (Cloud Run para deploy e BigQuery para análise de dados). O sistema interage com agentes de IA generativa da OpenAI e DeepSeek, permitindo recomendações e respostas contextualizadas em linguagem natural. A coleta de dados foi realizada por meio de técnicas de webscraping com Selenium, extraindo informações de produtos em marketplaces como Kabum, Terabyte e Pichau. Os dados foram organizados em camadas de processamento (Raw, Silver e Gold) e armazenados em buckets S3. Os testes envolveram avaliações locais e em produção, verificando autenticação, integração com APIs, persistência e desempenho do sistema. Como contribuição, o trabalho apresenta uma solução escalável, segura e modular para recomendação em e-commerce, com potencial de adaptação para diferentes contextos comerciais. Como trabalhos futuros, pretende-se explorar a integração com modelos open-source, expandir marketplaces suportados, incluir feedback ativo do usuário e aplicar aprendizado por reforço. A plataforma Diego evidencia como o uso integrado de LLMs e computação em nuvem pode aprimorar a experiência do usuário e gerar valor estratégico para o varejo digital.

Palavras Chave: chatbots inteligentes; FastAPI; serviços em nuvem; integração de serviços; firebase; framework CALMS.

[1] Graduando em Gestão da Tecnologia da Informação pela Faculdade de Princípios Militares (FPM).
[2] Graduanda em Gestão da Tecnologia da Informação pela Faculdade de Princípios Militares (FPM).
[3] Graduando em Gestão da Tecnologia da Informação pela Faculdade de Princípios Militares (FPM).
[4] Graduanda em Gestão da Tecnologia da Informação pela Faculdade de Princípios Militares (FPM).
[5] Doutor em Engenharia Elétrica e da Computação, Professor da Faculdade de Princípios Militares (FPM) e pesquisador e técnico em telecomunicações na Universidade Federal de Goiás (UFG).

Silva PF, Santos LS, Oliveira ALP, Costa ACF, Kunzler JA. Sistema de recomendação em e-commerce com modelos de LLM e computação em nuvem: um relato de experiência. Revista Brasileira Militar de Ciências. 2025;11(25):e198.

Abstract

This paper presents the development of Diego, an intelligent recommendation system for e-commerce that integrates large language models (LLMs), cloud services, and modern software engineering practices. The project was motivated by the growing demand for personalized recommendations in increasingly dynamic and competitive shopping environments. The adopted methodology involved literature review, system design, implementation, and testing. The system architecture follows the MVC pattern, with a backend developed using Python and FastAPI, a frontend built with ReactJS, and user authentication handled via Firebase. The infrastructure was deployed in a multicloud environment, combining AWS services (S3 for file storage) and GCP (Cloud Run for deployment and BigQuery for data analytics). The system communicates with generative AI agents from OpenAI and DeepSeek, enabling contextualized product recommendations and natural language responses. Data collection was performed using web scraping techniques with Selenium, targeting e-commerce sites such as Kabum, Terabyte, and Pichau. The extracted data included product names, prices, user ratings, availability, and purchase links, which were organized in a layered data pipeline (Raw, Silver, Gold) and stored in S3 buckets. Testing was conducted both locally and in production, evaluating authentication, API integration, data persistence, and overall system performance. As a contribution, this work presents a scalable, secure, and modular recommendation solution for e-commerce platforms, with potential adaptability to various retail scenarios. As future work, the integration with open-source models should be explored, expand supported marketplaces, include active user feedback, and apply reinforcement learning techniques. The Diego platform demonstrates how the integration of LLMs and cloud computing can enhance user experience and deliver strategic value in the digital commerce landscape.

Keywords: Intelligent chatbots; FastAPI; multicloud services; service integration; firebase; framework CALMS.

INTRODUÇÃO

O avanço das tecnologias de Inteligência Artificial (IA) e o amadurecimento das práticas de engenharia de software vêm transformando profundamente a maneira como sistemas interagem com os usuários. Nesse cenário, os chatbots inteligentes ocupam posição de destaque, permitindo a criação de interfaces conversacionais que combinam personalização, disponibilidade contínua e resposta em tempo real. Este artigo apresenta o desenvolvimento da plataforma Diego, um sistema de recomendação inteligente voltado ao e-commerce, que integra backend robusto com FastAPI, frontend interativo com ReactJS, múltiplos serviços de nuvem e agentes de IA generativa.

Paralelamente a esses avanços, o comércio eletrônico (e-commerce) consolida-se como um dos pilares da economia digital moderna. De acordo com o Sebrae, mais de 61% dos brasileiros preferem comprar online, destacando o impacto crescente dos marketplaces como Mercado Livre, Amazon, Shopee e AliExpress nas decisões de compra dos consumidores[1]. No entanto, à medida que esses ambientes se tornam mais competitivos, cresce também a necessidade de diferenciação por meio de experiências personalizadas — especialmente no processo de recomendação de produtos.

A recomendação inteligente é, portanto, um diferencial estratégico para marketplaces. Agentes de IA capazes de compreender o comportamento, as intenções e o histórico do usuário são capazes de oferecer sugestões altamente relevantes, otimizando a conversão e aumentando a fidelização. Neste contexto, a plataforma Diego adota uma abordagem centrada em modelos de

linguagem natural (LLMs), como o GPT-4o[2], combinando-os a tecnologias de coleta e análise de dados para fornecer respostas humanizadas, insights comerciais e sugestões personalizadas.

A motivação do projeto é explorar o potencial de agentes baseados em LLMs para aumentar a assertividade na recomendação de produtos em ambientes altamente dinâmicos. A arquitetura modular da aplicação permite que o Diego funcione tanto como uma API REST integrada a sistemas de e-commerce, quanto como uma interface web autônoma. Sua infraestrutura abrange serviços como Firebase[3] (para autenticação e persistência segura de usuários), Stripe (para gerenciamento de pagamentos), e BigQuery[4] (para análise de dados em grande escala).

Além disso, a abordagem de desenvolvimento da plataforma está alinhada aos princípios do framework CALMS (Culture, Automation, Lean, Measurement, Sharing), priorizando integração contínua, automação de testes e provisionamento, e cultura colaborativa[5]. O sistema foi concebido em um ambiente multicloud, explorando o ecossistema da Google Cloud Platform (GCP), com suporte a deploys automatizados e escalabilidade sob demanda.

A aplicação de IA generativa ao Diego permite que ele vá além das funções convencionais de chatbot, atuando como um verdadeiro assistente virtual de mercado, capaz de buscar produtos em diferentes marketplaces, comparar preços, oferecer recomendações personalizadas e gerar respostas ricas e contextualizadas. Seu uso de tecnologias modernas como OpenAI[2,6] e DeepSeek AI[7] permite entregar respostas precisas, com linguagem natural e adaptada ao perfil do usuário.

Por fim, destaca-se o papel do Firebase na segurança e privacidade dos dados[3]. Com autenticação baseada em tokens e integração transparente com o frontend React, o Diego garante proteção de endpoints e conformidade com a Lei Geral de Proteção de Dados (LGPD)[8,9], sem comprometer a experiência do usuário. Essa camada de proteção é essencial em sistemas de recomendação, onde a coleta de dados sensíveis exige tratamento ético e tecnicamente sólido.

O restante deste artigo está organizado da seguinte forma: a seção 2 apresenta a fundamentação teórica sobre agentes inteligentes, computação em nuvem e práticas modernas de desenvolvimento. A seção 3 detalha os objetivos do projeto. A seção 4 descreve os métodos e as tecnologias adotadas no desenvolvimento da plataforma. A seção 5 apresenta os resultados obtidos e sua análise crítica, e por fim, a seção 6 discute as conclusões e recomendações para trabalhos futuros.

FUNDAMENTAÇÃO TEÓRICA

Nesta seção são estabelecidos os alicerces conceituais que orientam o desenvolvimento da plataforma Diego, abordando os principais temas relacionados à Inteligência Artificial, arquiteturas modernas de software e computação em nuvem.

Silva PF, Santos LS, Oliveira ALP, Costa ACF, Kunzler JA. Sistema de recomendação em e-commerce com modelos de LLM e computação em nuvem: um relato de experiência. Revista Brasileira Militar de Ciências. 2025;11(25):e198.

Agentes de Inteligência Artificial

Agentes de Inteligência Artificial (IA) são entidades computacionais capazes de perceber seu ambiente, processar essas percepções por meio de mecanismos internos de decisão e realizar ações que influenciam o ambiente em direção a objetivos definidos. Segundo Russell e Norvig (2021), um agente inteligente pode ser formalmente descrito como qualquer sistema que percebe o ambiente por sensores e age sobre ele por atuadores, com o objetivo de maximizar alguma noção de desempenho[10]. A literatura especializada classifica esses agentes em diferentes tipos, de acordo com sua complexidade e arquitetura: agentes reativos, que respondem diretamente a estímulos com regras simples; agentes deliberativos, que mantêm um modelo interno do mundo e tomam decisões com base em planos; e agentes híbridos, que combinam aspectos reativos e deliberativos para maior flexibilidade e robustez[11,12].

A Figura 1 apresenta um diagrama conceitual da arquitetura de um Agente baseado em Modelos de Linguagem de Grande Escala (LLM), utilizado para interação com usuários e execução de tarefas em ambientes digitais. O fluxo parte da entrada do Usuário, que fornece comandos ou perguntas ao agente. O agente LLM é representado como um sistema composto por três módulos principais: Percepção, Planejamento e Ação[10].

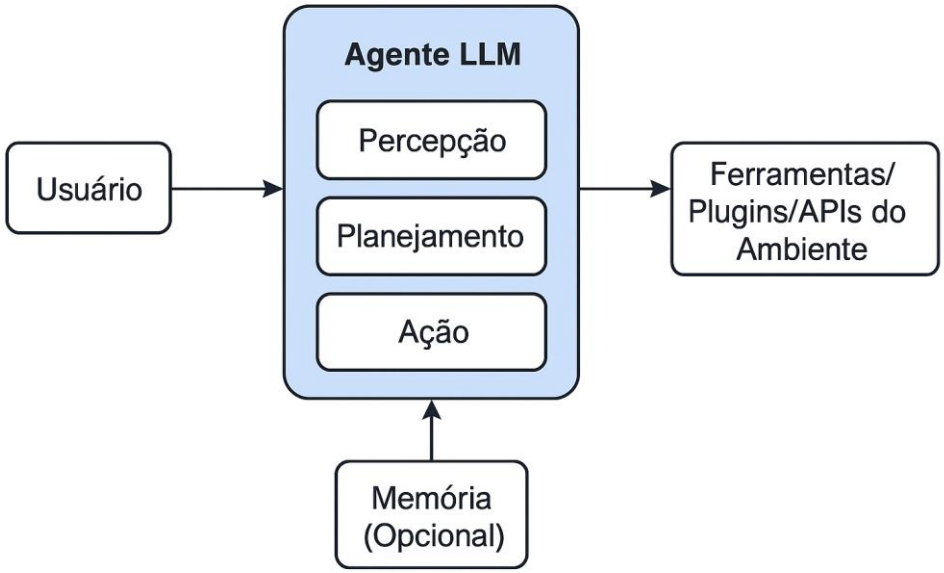
* Percepção: responsável por interpretar a entrada do usuário, convertendo-a em uma representação compreensível pelo sistema (por exemplo, via técnicas de Processamento de Linguagem Natural).

* Planejamento: define a melhor estratégia para atingir o objetivo solicitado, podendo envolver raciocínio, tomada de decisão e seleção de ferramentas apropriadas.

* Ação: executa as tarefas planejadas, podendo acionar recursos externos.

Silva PF, Santos LS, Oliveira ALP, Costa ACF, Kunzler JA. Sistema de recomendação em e-commerce com modelos de LLM e computação em nuvem: um relato de experiência. Revista Brasileira Militar de Ciências. 2025;11(25):e198.

Figura 1. Representação de um agente de inteligência artificial e suas interações.



Além disso, o agente pode contar com um módulo de Memória (Opcional), que permite manter contexto de interações passadas ou registrar informações relevantes para interações futuras. O agente interage com ferramentas externas, plugins ou APIs do ambiente, sendo capaz de realizar buscas, acessar bancos de dados, consultar sistemas de recomendação, entre outros.
No contexto de aplicações voltadas à análise textual e recomendação, destaca-se o uso de agentes baseados em Processamento de Linguagem Natural (PLN)[12]. Esses agentes são projetados para interpretar e gerar linguagem humana, utilizando técnicas que incluem análise de sentimentos, classificação de intenções, extração de entidades nomeadas (*Named Entity Recognition – NER*) e resposta a perguntas (*Question Answering – QA*). Tais abordagens permitem que os sistemas compreendam melhor os objetivos dos usuários e ofereçam recomendações personalizadas com base no conteúdo textual das interações[11,12].

Diversos frameworks modernos têm viabilizado o desenvolvimento desses agentes com alto desempenho. O spaCy é uma biblioteca robusta e eficiente voltada ao PLN, com suporte nativo a tarefas como *tokenization*, *lemmatization* e reconhecimento de entidades. O ecossistema Hugging Face Transformers, por sua vez, revolucionou o acesso a modelos de linguagem profunda (DL), fornecendo implementações pré-treinadas de arquiteturas como BERT, T5 e GPT[13] que podem ser facilmente integradas a aplicações reais[14].

Portanto, o uso de agentes de IA com suporte a PLN representa uma das abordagens mais eficazes para criar sistemas adaptativos e personalizados, principalmente no domínio de e-commerce e interfaces conversacionais, onde a interpretação precisa de linguagem natural é uma exigência crítica para a eficácia do sistema.

Silva PF, Santos LS, Oliveira ALP, Costa ACF, Kunzler JA. Sistema de recomendação em e-commerce com modelos de LLM e computação em nuvem: um relato de experiência. Revista Brasileira Militar de Ciências. 2025;11(25):e198.

Computação em Nuvem

A computação em nuvem (*cloud computing*) representa uma das transformações mais impactantes na área de tecnologia da informação, permitindo o fornecimento escalável, sob demanda e baseado em consumo de recursos computacionais. De acordo com o National Institute of Standards and Technology (NIST), esse paradigma é estruturado em três modelos de serviço principais: *Infrastructure as a Service* (IaaS), *Platform as a Service* (PaaS) e *Software as a Service* (SaaS)[15]. Tais modelos possibilitam desde o provisionamento de servidores virtuais e armazenamento até o desenvolvimento e entrega contínua de aplicações com mínima intervenção manual na infraestrutura.

A computação em nuvem permite que organizações desenvolvam sistemas com alta disponibilidade, tolerância a falhas e baixo tempo de provisionamento, contribuindo diretamente para a agilidade no desenvolvimento de soluções inovadoras. Dentre os principais provedores de serviços em nuvem, destacam-se a Amazon Web Services (AWS)**,** Google Cloud Platform (GCP) **e** Microsoft Azure, que oferecem amplo ecossistema de ferramentas para desenvolvimento, armazenamento, segurança e análise de dados.

A Amazon Web Services (AWS) consolidou-se como uma das líderes globais em serviços de nuvem pública, fornecendo infraestrutura altamente escalável e segura para empresas de todos os portes. O *Amazon Simple Storage Service* (Amazon S3), por exemplo, é amplamente utilizado para armazenamento de objetos, oferecendo altíssima durabilidade e integração nativa com SDKs para diversas linguagens, como Python, Java e Node.js[16]. Em contextos de sistemas baseados em IA e processamento de dados não estruturados, serviços como o S3 se tornam essenciais para armazenar e acessar imagens, áudios e logs de interação em tempo real.

Por sua vez, a GCP tem se destacado tanto por sua abordagem voltada à ciência de dados quanto por seu suporte à automação de pipelines e microsserviços. Com ferramentas como o *Cloud Run*, que permite o deploy de contêineres sem necessidade de gerenciar servidores, e o *BigQuery*, uma plataforma de análise de dados em larga escala, a GCP oferece um ecossistema completo para o desenvolvimento ágil de aplicações baseadas em contêineres, análise preditiva e integração com modelos de aprendizado de máquina[17]. Além disso, a GCP disponibiliza iniciativas de incentivo acadêmico, como créditos para estudantes e pesquisadores, facilitando sua adoção em projetos experimentais ou educacionais.

Complementando esse cenário, o Firebase, também pertencente ao ecossistema Google, desponta como uma solução *backend-as-a-service* (BaaS) moderna, especialmente útil em aplicações web e móveis. Com suporte à autenticação baseada em tokens JWT, banco de dados NoSQL (Firestore), hospedagem e notificações em tempo real, o Firebase permite a construção rápida de sistemas seguros, sem a necessidade de configuração manual de servidores ou bancos

de dados tradicionais[3]. Sua integração com bibliotecas frontend e seu foco na segurança e escalabilidade o tornam uma opção valiosa para projetos que requerem agilidade no desenvolvimento e conformidade com legislações como a Lei Geral de Proteção de Dados (LGPD)[9].

Arquitetura e Desenvolvimento de Sistemas

A construção de sistemas modernos baseia-se na adoção de arquiteturas bem definidas, com destaque para o padrão MVC (Model-View-Controller), que separa a lógica de negócios (Model), a interface do usuário (View) e o controle das interações (Controller). Essa abordagem facilita a manutenção do código, promove reuso e aumenta a escalabilidade da aplicação[18,19].

Esses são princípios caríssimos de engenharia de software que visam modularidade, escalabilidade, manutenibilidade e separação de responsabilidades. O padrão MVC é amplamente adotado em aplicações web e sistemas interativos, pois organiza o software nessas três camadas principais, separando as responsabilidades pela lógica de negócios e acesso a dados da interface de apresentação ao usuário e da intermediação entre entrada do usuário e manipulação de dados[19].

No backend, diversas linguagens e frameworks se destacam por sua produtividade e capacidade de integração com serviços externos. O Python, por exemplo, tem sido amplamente utilizado em soluções orientadas à Inteligência Artificial e serviços web. Em particular, o FastAPI emergiu como um framework moderno e eficiente para construção de APIs RESTful, oferecendo suporte nativo à programação assíncrona via async/await, tipagem estática com Pydantic, validação automática de dados e documentação interativa com base no padrão OpenAPI/Swagger. Essas características tornam o FastAPI uma escolha atraente para aplicações que exigem alta performance, segurança e clareza de especificação da interface[20].

No frontend, frameworks e bibliotecas como ReactJS, Vue.js e Angular são amplamente utilizados para construção de interfaces de usuário modernas. O ReactJS, em particular, possibilita a criação de componentes declarativos e reativos, promovendo maior controle sobre o estado da aplicação e renderizações eficientes. A utilização de Single Page Applications (SPAs) e integração com APIs externas via REST ou GraphQL facilita o desenvolvimento de experiências ricas e responsivas para o usuário final[20].

Quanto à camada de persistência de dados, o modelo NoSQL tem ganhado destaque, especialmente em cenários de aplicações web em tempo real, sistemas distribuídos e dados sem estrutura rígida. Soluções como Firebase Firestore, MongoDB e Amazon DynamoDB permitem armazenamento flexível, escalabilidade horizontal e replicação automática, além de integrações nativas com serviços de autenticação e análise. O Firestore, por exemplo, é um banco de dados orientado a documentos, com sincronização em tempo real, estrutura sem esquema fixo (schema-

Silva PF, Santos LS, Oliveira ALP, Costa ACF, Kunzler JA. Sistema de recomendação em e-commerce com modelos de LLM e computação em nuvem: um relato de experiência. Revista Brasileira Militar de Ciências. 2025;11(25):e198.

less) e integração com outras soluções do ecossistema Firebase, sendo bastante utilizado em aplicações móveis e web modernas[3].

OBJETIVOS

O objetivo geral deste trabalho é desenvolver uma plataforma inteligente, denominada Diego, capaz de recomendar produtos em ambientes de e-commerce utilizando agentes baseados em Inteligência Artificial. A proposta visa integrar tecnologias modernas de desenvolvimento de software com soluções de computação em nuvem, promovendo escalabilidade, segurança e uma experiência de usuário eficiente e personalizada. A plataforma Diego busca aliar a robustez de um backend desenvolvido com FastAPI, a flexibilidade e interatividade de um frontend construído com ReactJS, e a integração com serviços multicloud como Firebase, GCP e AWS, de modo a entregar um sistema funcional, seguro e de fácil expansão. Para alcançar esse propósito, foram definidos os seguintes objetivos específicos:

* Integrar o backend FastAPI com o frontend ReactJS, estabelecendo uma arquitetura coerente baseada em APIs REST, que permita comunicação eficiente entre as camadas da aplicação e mantenha a modularidade do sistema;

* Empregar modelos de linguagem natural (NLP) para compreender as mensagens dos usuários e realizar recomendações assertivas, utilizando técnicas de IA generativa como os modelos GPT, acoplados a um pipeline de entrada textual e extração de intenção;

* Implementar um sistema de autenticação e pagamentos seguro e eficaz, utilizando o Firebase para gerenciamento de usuários e tokens JWT, e o Stripe para integração transparente com métodos de pagamento, viabilizando diferentes planos de uso da plataforma;

* Viabilizar a análise de dados em tempo real através da integração com o BigQuery, permitindo a coleta, o processamento e a visualização de dados relevantes sobre o comportamento dos usuários e a performance das recomendações.

MÉTODOS

Para o desenvolvimento deste trabalho de pesquisa, adotou-se uma abordagem prática e iterativa, combinando o uso de ferramentas de desenvolvimento web, bibliotecas de inteligência

Silva PF, Santos LS, Oliveira ALP, Costa ACF, Kunzler JA. Sistema de recomendação em e-commerce com modelos de LLM e computação em nuvem: um relato de experiência. Revista Brasileira Militar de Ciências. 2025;11(25):e198.

artificial e serviços de computação em nuvem. A estrutura metodológica envolveu a revisão bibliográfica, o design da arquitetura do sistema, a implementação técnica e a realização de testes.

Revisão Bibliográfica

A primeira etapa consistiu na revisão de trabalhos sobre inteligência artificial generativa, análise de dados em marketplaces e segurança digital, com foco em tecnologias aplicáveis a sistemas de chatbot inteligente. Foram definidos os requisitos funcionais, como geração de insights, recomendação de preços e integração com marketplaces, e os não funcionais, como uma interface web interativa e responsiva, escalabilidade e segurança[2]. Com base nesta revisão, o projeto foi subdividido em três áreas principais:

* **Projeto de Backend:** Foram selecionadas tecnologias modernas como Python e o framework FastAPI para a construção da API responsável por integrar a IA generativa, gerenciar requisições do usuário e conectar com os serviços da AWS. A infraestrutura de armazenamento de dados foi baseada em AWS S3[19].

* **Projeto de Frontend:** A interface do usuário foi desenvolvida utilizando o framework React, oferecendo uma experiência fluida, moderna e responsiva. A camada de apresentação permite ao usuário interagir com o sistema via mensagens ou áudios, que são processados pela IA[20].

* **Gerenciamento de Usuários:** O gerenciamento de usuários e a autenticação foram implementados utilizando a plataforma Firebase, responsável pelo controle de acesso ao sistema[3].

Desenvolvimento e Implementação do Sistema

A implementação do sistema Diego envolveu a integração de seus principais componentes: backend em Python com FastAPI, IAs generativas da OpenAI[2] e Deepseek[3], autenticação via Firebase e interface construída com React. O sistema foi projetado para receber dados de produtos e usuários, realizar análise com auxílio de inteligência artificial e apresentar insights em tempo real. Esta abordagem está em consonância com a construção de sistemas modernos, que se baseiam na adoção de arquiteturas bem definidas. A escolha de uma arquitetura garante aspectos importantes como a modularidade, manutenibilidade e escalabilidade das aplicações.

O padrão de arquitetura de software escolhido para o Diego foi o MVC. Tal abordagem é especialmente útil em aplicações de médio e grande porte, pois favorece a reutilização de componentes, a testabilidade e a organização do código, conforme visto na seção 2c.

Silva PF, Santos LS, Oliveira ALP, Costa ACF, Kunzler JA. Sistema de recomendação em e-commerce com modelos de LLM e computação em nuvem: um relato de experiência. Revista Brasileira Militar de Ciências. 2025;11(25):e198.

No backend da aplicação proposta, optou-se pela linguagem Python, uma das mais utilizadas no desenvolvimento de soluções com Inteligência Artificial[21], combinada ao framework FastAPI. Este framework tem se destacado por seu desempenho elevado, suporte nativo a programação assíncrona com async/await, tipagem estática com Pydantic, validação automática de dados e geração de documentação interativa baseada no padrão OpenAPI (Swagger). Esses recursos foram fundamentais para a construção de uma API RESTful robusta e bem documentada, com suporte a autenticação baseada em tokens JWT, integração com provedores externos (Firebase, OpenAI/Deepseek, Stripe) e endpoints seguros para manipulação de dados sensíveis.

Na camada de frontend, foi utilizada a biblioteca ReactJS, desenvolvida pela Meta (Facebook). O React permite a criação de interfaces declarativas e reativas, com renderização eficiente baseada em mudanças de estado, o que é particularmente útil em aplicações interativas como chatbots e sistemas de recomendação. Componentes visuais como os formulários de login, área de chat, cards de recomendação e menus de navegação foram estruturados de forma modular, seguindo boas práticas de desenvolvimento frontend moderno.

Para armazenamento e sincronização de dados, adotou-se o modelo NoSQL, com uso do Firebase Firestore como banco de dados principal. Esse sistema gerenciado de banco de dados oferece alta disponibilidade, replicação automática e sincronização em tempo real, sendo ideal para aplicações baseadas em eventos e atualizações contínuas do estado da interface. Sua integração nativa com o Firebase Authentication e suporte a regras de segurança baseadas em contexto permitiram uma implementação eficiente de controle de acesso e proteção de dados.

O sistema foi concebido para operar em ambiente multicloud, explorando os pontos fortes de diferentes plataformas. A Amazon Web Services (AWS) foi utilizada principalmente para armazenamento de grandes listas de objetos JavaScript (JSON), ativos digitais como imagens de produtos, arquivos de voz, logs de interação e documentos temporários, por meio do serviço Amazon S3. O S3 oferece alta durabilidade, políticas de versionamento, integração com bibliotecas Python (boto3) e permite o controle refinado de acesso por políticas IAM, promovendo segurança e performance na manipulação de dados.

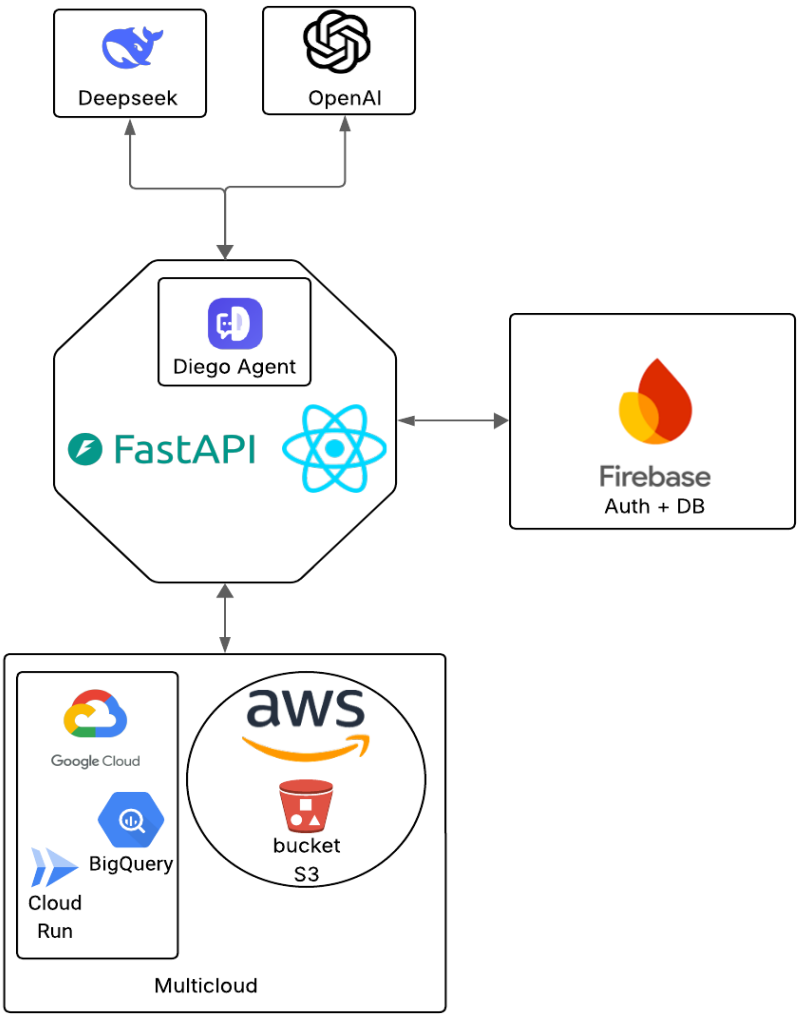
Complementarmente, a Google Cloud Platform (GCP) serviu como base para o deploy do backend (via Cloud Run) e para as análises de dados realizadas com BigQuery. O BigQuery é um data warehouse altamente escalável e otimizado para consultas SQL em grandes volumes de dados. Sua integração com APIs REST e conectores Python permitiu consultar automaticamente indicadores de preços, frequência de buscas e padrões de navegação. Isso tornou possível oferecer recomendações mais precisas com base em dados históricos extraídos de marketplaces.

A escolha por uma estratégia multicloud teve como motivação principal o acesso a ferramentas especializadas de cada provedor, o aumento da resiliência da aplicação e a otimização de custos operacionais, aproveitando créditos educacionais e planos gratuitos das plataformas.

Silva PF, Santos LS, Oliveira ALP, Costa ACF, Kunzler JA. Sistema de recomendação em e-commerce com modelos de LLM e computação em nuvem: um relato de experiência. Revista Brasileira Militar de Ciências. 2025;11(25):e198.

A Figura 2 apresenta um diagrama com os principais componentes do sistema. A arquitetura geral da plataforma Diego possui integração em ambiente multicloud e os principais componentes utilizados na solução se comunicam através do núcleo da aplicação.

Figura 2. Arquitetura da Plataforma Diego Integrando IA Generativa e Serviços Multicloud



.

No centro da arquitetura está o núcleo da aplicação, composto pelo FastAPI (responsável pela lógica de backend) e o ReactJS (camada de frontend), que juntos formam a base da comunicação entre usuários e os serviços de IA. O agente Diego se conecta com modelos generativos de linguagem providos pelas APIs da OpenAI e DeepSeek, recebendo requisições de linguagem natural dos usuários e retornando respostas processadas.

A camada de autenticação e banco de dados é provida pelo Firebase, que centraliza o gerenciamento seguro de usuários e a persistência dos dados com o Firestore (NoSQL). Esta estrutura garante sincronização em tempo real, autenticação via tokens JWT.

Silva PF, Santos LS, Oliveira ALP, Costa ACF, Kunzler JA. Sistema de recomendação em e-commerce com modelos de LLM e computação em nuvem: um relato de experiência. Revista Brasileira Militar de Ciências. 2025;11(25):e198.

Na parte inferior da figura, a arquitetura multicloud é evidenciada:

* GCP: com serviços como Cloud Run (orquestração de containers do backend) e BigQuery (análise de dados em larga escala);

* AWS: com o uso do Amazon S3, destinado ao armazenamento de arquivos como objetos JavaScript, imagens, áudios e registros de interação.

Essa composição permite uma aplicação altamente escalável, modular, com recursos de análise inteligente e integração segura, refletindo os principais pontos metodológicos discutidos até aqui.

Realização de Testes

A etapa de testes é fundamental na validação do desenvolvimento de software, ela assegura a integridade funcional dos componentes implementados e a coerência das integrações entre serviços. Os testes foram conduzidos em duas fases principais: testes locais e testes em ambiente de produção na nuvem.

Durante os testes locais, a aplicação foi executada em ambiente de desenvolvimento com suporte a recarregamento automático (reload) utilizando o uvicorn em conjunto com o framework FastAPI. Essa abordagem permitiu testar rapidamente os endpoints REST criados, validar o funcionamento da autenticação via Firebase, simular chamadas às APIs externas (como OpenAI e Stripe), além de verificar a comunicação entre o backend e o frontend ReactJS por meio do CORS. A base de dados Firestore foi configurada em modo de teste, possibilitando a criação e leitura de documentos sem comprometer dados reais.

Em seguida, foram conduzidos testes de integração em ambiente multicloud, com o backend implantado via Google Cloud Run, o frontend via Firebase Hosting, e o armazenamento de arquivos em AWS S3. Esse ambiente mais próximo do real permitiu avaliar aspectos como latência de requisições, resposta dos agentes inteligentes ao receberem mensagens do usuário e o correto funcionamento dos fluxos de pagamento com Stripe em sandbox. Os logs de execução foram monitorados diretamente pelo console do Cloud Run e complementados por ferramentas de logging da GCP. Além disso, foram validadas as seguintes funcionalidades críticas:

* Autenticação e autorização de usuários, com verificação de tokens JWT e redirecionamento condicional baseado no status do plano;

Silva PF, Santos LS, Oliveira ALP, Costa ACF, Kunzler JA. Sistema de recomendação em e-commerce com modelos de LLM e computação em nuvem: um relato de experiência. Revista Brasileira Militar de Ciências. 2025;11(25):e198.

* Interação com os agentes de IA, observando se a resposta era gerada conforme o prompt de entrada e parâmetros esperados;

* Envio e armazenamento de arquivos para o bucket S3 e o posterior acesso por meio de requisições do núcleo do sistema;

* Persistência e recuperação de dados no Firestore e no BigQuery, garantindo que os dados fossem corretamente estruturados e disponíveis para análise

RESULTADOS E DICUSSÃO

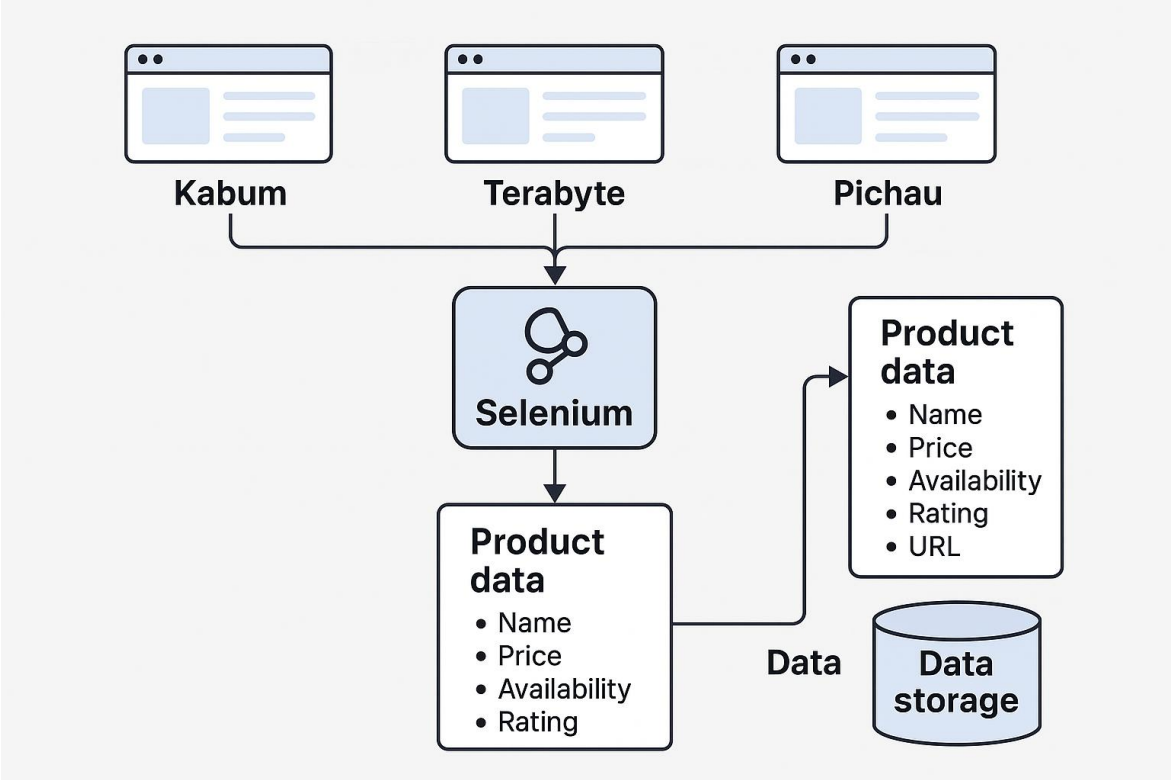
O sistema Diego executa a coleta de dados de marketplaces, tratamento de informações utilizando técnicas de inteligência artificial, aplicação de regras de recomendação e implantação em infraestrutura de nuvem escalável.

Extração de Dados dos Marketplaces

A etapa de obtenção de dados foi conduzida por meio da técnica de webscraping, essencial para alimentar os modelos de recomendação com informações atualizadas sobre produtos de e-commerce. Para isso, utilizou-se a biblioteca Selenium, amplamente reconhecida por sua capacidade de interagir dinamicamente com elementos de páginas web. A coleta foi direcionada para sites de empresas como Kabum, Terabyte e Pichau, respeitando os termos de uso das plataformas e empregando práticas de raspagem responsável, como o uso de delays e user-agents rotativos. Os dados extraídos incluíram nome do produto, preço, disponibilidade, avaliações de usuários e links diretos para compra. Após a coleta, os dados foram armazenados em buckets S3 da AWS e organizados em camadas de processamento no modelo Raw → Silver → Gold, conforme ilustrado na Figura 3. Esse processo garantiu a preparação dos dados com integridade, qualidade e estrutura adequadas para posterior análise com modelos baseados em IA.

Silva PF, Santos LS, Oliveira ALP, Costa ACF, Kunzler JA. Sistema de recomendação em e-commerce com modelos de LLM e computação em nuvem: um relato de experiência. Revista Brasileira Militar de Ciências. 2025;11(25):e198.

Figura 3. Pipeline de coleta e tratamento de dados via webscraping com armazenamento em S3



Tratamento de Dados com Técnicas de NLP

Após a extração, os dados textuais foram processados com técnicas de Processamento de Linguagem Natural. A limpeza dos textos envolveu a remoção de informações irrelevantes, a normalização dos dados e a preparação dos mesmos para a análise de comportamento de consumo. Essa etapa foi essencial para identificar padrões nos dados dos produtos e gerar insights que pudessem orientar as recomendações. Além disso, o tratamento dos dados visou melhorar a interpretação e a resposta humanizada aos usuários.

Modelo de Recomendação

O modelo de recomendação foi baseado em regras predefinidas, considerando fatores como o menor preço, alta avaliação e popularidade dos produtos. Embora o sistema não tenha utilizado modelos avançados de aprendizado de máquina supervisionado ou não supervisionado, ele aplicou regras de negócios robustas que geraram recomendações eficazes para o usuário. A inteligência do sistema focou na análise de dados disponíveis e na geração de respostas contextualizadas.

Silva PF, Santos LS, Oliveira ALP, Costa ACF, Kunzler JA. Sistema de recomendação em e-commerce com modelos de LLM e computação em nuvem: um relato de experiência. Revista Brasileira Militar de Ciências. 2025;11(25):e198.

Estimativa de Custos Mensais

O quadro 1 a seguir apresenta os custos mensais estimados da infraestrutura, apresenta-se uma estimativa dos custos mensais envolvidos na manutenção da plataforma Diego em ambiente de múltiplos provedores em nuvem. Os valores foram obtidos com base em simulações práticas e no uso de ferramentas oficiais de cálculo disponibilizadas por cada provedor de serviços.

Quadro 1: Análise de custo dos componentes do Sistema Diego.

Componente	Serviço/Plano Utilizado	Custo Estimado (USD/mês)	Observações
Armazenamento de Arquivos	AWS S3 (com versionamento e SSE)	~$2,00	Custo estimado com base em ~15GB e poucas requisições.
Hospedagem Frontend	Firebase deploy	~$3,00	Site React hospedado com cache e HTTPS.
Data Warehouse	Bigquery	~$20,00	Para análise de dados estruturados em camada Gold.
Backend API	FastAPI rodando em Cloud Run	~$8,00	Custo baseado em uso leve e sem escalonamento automático.
Autenticação de Usuário		~$0,00 a $5,00	Gratuito no plano básico; custo pequeno se exceder requisições.
IA Generativa	OpenAI GPT-4 API	~$10,00 a 15,00	Com base em interações de uso controlado (~250 mil tokens/mês).
Controle de Versão	GitHub (Free Plan)	$0,00	Inclui Actions e repositórios públicos/privados.
CI/CD	GitHub Actions	~$0,00 a $2,00	Custo se ultrapassar limite gratuito de 2.000 minutos/mês.
Infraestrutura como Código	Terraform + AWS CLI + GCP CLI	$0,00	Ferramentas gratuitas em ambiente local.

As principais plataformas de computação em nuvem, como Amazon Web Services (AWS), Google Cloud Platform (GCP) e Firebase, oferecem calculadoras de preços online que permitem estimar com precisão os custos operacionais de cada serviço, considerando fatores como volume de requisições, armazenamento, tempo de execução de funções, transferências de dados e uso de APIs externas.

A AWS Pricing Calculator, por exemplo, fornece simulações detalhadas para serviços, permitindo o ajuste fino conforme o perfil de uso da aplicação. Já a Google Cloud Pricing Calculator possibilita estimar custos com BigQuery, Cloud Run, Firestore, entre outros serviços da GCP.

Silva PF, Santos LS, Oliveira ALP, Costa ACF, Kunzler JA. Sistema de recomendação em e-commerce com modelos de LLM e computação em nuvem: um relato de experiência. Revista Brasileira Militar de Ciências. 2025;11(25):e198.

No caso do Diego, os maiores custos foram associados ao armazenamento e à manipulação de dados em buckets do Amazon S3, à infraestrutura para deploy da aplicação no GCP e à execução de requisições para modelos de linguagem natural da OpenAI, cujos preços variam conforme o modelo utilizado e a quantidade de tokens processados. Por outro lado, ferramentas como GitHub Actions e Firebase ofereceram um plano gratuito generoso.

Essas ferramentas de estimativa são essenciais tanto na fase de planejamento quanto na fase de operação contínua, permitindo ajustar os serviços contratados de acordo com o orçamento disponível, a demanda do sistema e as metas de escalabilidade. Além disso, viabilizam o uso racional de créditos educacionais ou institucionais, frequentemente oferecidos por essas plataformas a pesquisadores e desenvolvedores. Os valores totais podem ser vistos no quadro 2.

Quadro 2. Custo total para deploy do Sistema Diego.

Categoria	Custo MínimoEstimado (USD/mês)	Custo Médio Estimado (USD/mês)
Infraestrutura AWS + GCP	~$23,00	~$43,00
Firebase	~$0,00	~$5,00
OpenAI (IA Generativa)	~$10,00	~$15,00
GitHub Actions/ CI/DI	~$0,00	~$2,00
Domínio Personalizado	~$1,00	~$1,00
Total Estimado	**~$34,00/mês**	**~$66,00/mês**

CONCLUSÃO

O desenvolvimento da plataforma Diego demonstrou a viabilidade e os benefícios da integração de agentes de inteligência artificial generativa com tecnologias modernas de desenvolvimento web e computação em nuvem em um ambiente de e-commerce. A aplicação se destaca pela adoção de uma arquitetura modular, baseada em boas práticas como o padrão MVC, além do uso estratégico de serviços multicloud para garantir escalabilidade, desempenho e segurança.

O uso de LLMs, como o GPT-4o e o DeepSeek, aliado à coleta automatizada de dados via webscraping e ao armazenamento em buckets estruturados na AWS, permitiu a criação de um sistema de recomendação robusto, capaz de interpretar linguagem natural e oferecer sugestões personalizadas aos usuários. O frontend responsivo desenvolvido com ReactJS, integrado ao backend em FastAPI, assegura uma experiência fluida e segura para o usuário final.

Os testes realizados em ambiente local e na nuvem demonstraram estabilidade, desempenho satisfatório e coerência entre os diferentes componentes do sistema. Como trabalhos futuros, propõe-se a expansão da cobertura de marketplaces analisados, a introdução de modelos de personalização baseados em comportamento de navegação e o aprimoramento do agente com mecanismos de feedback explícito do usuário.

REFERÊNCIAS

1. Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (Sebrae). 61% dos brasileiros preferem e-commerce ao varejo físico. Portal Sebrae, 2023. Disponível em: https://sebrae.com.br/sites/PortalSebrae/61-dos-brasileiros-preferem-e-commerce-ao-varejo-fisico,d6a8dd3257c67810VgnVCM1000001b00320aRCRD. Acesso em: 24 jun. 2025.

2. Achiam J. et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.

3. Google Developers. Firebase Documentation. Disponível em: https://firebase.google.com/docs. Acesso em: 24 jun. 2025.

4. Google Cloud Platform. Cloud Run and BigQuery documentation. Disponível em: https://cloud.google.com/docs. Acesso em: 24 jun. 2025.

5. Balaraman P, Chandrasekar S. E-commerce trends and future analytics tools. Indian Journal of Science and Technology, 2016;9(32):1-9.

6. OpenAI. GPT-4 Technical Report. OpenAI, 2023. Disponível em: https://openai.com/research/gpt-4. Acesso em: 24 jun. 2025.

7. DeepSeek. Documentation. Disponível em: https://api-docs.deepseek.com/. Acesso em: 24 jun. 2025.

8. Souza CE. Atualização do backend do sistema Open Social Care: migrando da arquitetura Serverless para uma API em Laravel. 2024. Trabalho de Conclusão de Curso (Graduação em Engenharia de Software) – Universidade Tecnológica Federal do Paraná, Curitiba, 2024.

9. Losso EB, Freitas COA. A adequação do ChatGPT às exigências da Lei Geral de Proteção de Dados Pessoais. Lex Humana, 2024;16(4):22-47.

10. Russel S, Norvig P. Artificial intelligence: a modern approach. 4. ed. Boston: Pearson; 2021.

11. Wooldridge M. An Introduction to multiagent systems. 2. ed. Chichester: Wiley; 2009.

12. Jurafsky D, Martin JH. Speech and language processing. 3. ed. Draft. Stanford University, 2023. Disponível em: https://web.stanford.edu/~jurafsky/slp3/. Acesso em: 24 jun. 2025.

13. Wolf T. et al. Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). ACL, 2020.

14. Vaswani A. et al. Attention is all you need. In: Advances in Neural Information Processing Systems 30 (NeurIPS 2017). Curran Associates Inc., 2017. p. 5998–6008.

15. Mell P, Grance T. The NIST definition of cloud computing. Gaithersburg: National Institute of Standards and Technology, 2011. (NIST Special Publication 800-145).

16. Amazon Web Services. Amazon S3 documentation. Disponível em: https://docs.aws.amazon.com/s3. Acesso em: 24 jun. 2025.

17. Google Cloud. Documentação oficial: Cloud Run e BigQuery. Disponível em: https://cloud.google.com/docs. Acesso em: 24 jun. 2025.

18. Buschmann F. et al. Pattern-oriented software architecture: a system of patterns. Chichester: Wiley; 1996.

19. Sommerville I. Engenharia de Software. 10. ed. São Paulo: Pearson; 2019.

20. Pressman RS, Maxim BR. Engenharia de software: uma abordagem profissional. 8. ed. Porto Alegre: AMGH; 2016.

21. Ferreira OS, Guimarães LS. Python para Desenvolvedores. 3. ed. São Paulo: Novatec; 2020.

Contato para correspondência:
Jonas Augusto Kunzler

E-mail:
jakunzler.fpm@gmail.com

Conflito de interesse: Não

Financiamento: Recursos próprios