



Magia: Aplicativo de Aprendizagem de Espanhol com Inteligência Artificial

Trabalho de Conclusão de Curso

Ian Bernardo Lima dos Santos

Flávia Maristela Santos Nascimento

Orientadora

Instituto Federal da Bahia – IFBA
Curso de Análise e Desenvolvimento de Sistemas
Campus Salvador
Salvador, Bahia, Brasil
Julho 2026

LISTA DE TABELAS

Tabela 1 - Comparativo entre o Magia e soluções existentes

Tabela 2 - Requisitos funcionais

Tabela 3 - Níveis existentes na gamificação do Magia

LISTA DE ILUSTRAÇÕES

Figura 1 - Diagrama de casos de uso

Figura 2 - Diagrama de classes

Figura 3 - Diagrama de Sequência da atividade de Completar

Figura 4 - Visão arquitetural do sistema Magia

Figura 5 - Modelo de banco de dados

Figura 6 - Trigger do sistema de pontuação

Figura 7 - Perfil do usuário

Figura 8 - Chamada à API da OpenAI

Figura 9 - Uso do swagger para testes e documentação da API

Figura 10 - Debug no front-end

Figura 11 - Testes unitários do TrialGuard

Figura 12 - Tela de login

Figura 13 - Tela de cadastro

Figura 14 - Tela Inicial

Figura 15 - Tela de criação de atividades

Figura 16 - Tela de atividades de múltipla escolha

Figura 17 - Tela de atividades de completar

Figura 18 - Telas de flashcard

Figura 19 - Telas de histórico para revisão

Figura 20 - Telas de curiosidade

Figura 21 - Tela de perfil

SUMÁRIO

1. VISÃO GERAL E DECLARAÇÃO DO PROBLEMA.....	6
1.2 PROPOSTA DE SOLUÇÃO DE SOFTWARE.....	8
1.3 TECNOLOGIAS ADOTADAS.....	10
1.3.1 Front-end: Ionic com Angular e Capacitor.....	10
1.3.2 Back-end: NestJs.....	10
1.3.3 Banco de Dados: MariaDB.....	11
1.3.4 Integrações e Ferramentas Adicionais.....	11
1.4 TRABALHOS RELACIONADOS.....	11
1.4.1 O Conceito de MALL e a Evolução da Mobilidade.....	11
1.4.2 Da Fixidez do CALL à Espontaneidade do MALL.....	12
1.4.3 Privacidade e Inclusão Social.....	13
1.4.4 O Cenário Atual e o Magia.....	13
1.4.5 Plataformas Baseadas em Currículos Fixos: Duolingo e Busuu.....	14
1.4.6 Experimentos em IA Generativa: Little Language Lessons (Google Gemini).....	14
1.4.7 Quadro Comparativo de Funcionalidades.....	15
2. REQUISITOS.....	16
2.1 REQUISITOS FUNCIONAIS.....	16
2.2 REQUISITOS NÃO-FUNCIONAIS.....	17
3. DESIGN.....	19
3.1 PROJETO UML.....	19
3.1.1 Diagrama de casos de uso.....	19
3.1.2 Diagrama de classe e de sequência.....	20
3.2 VISÃO ARQUITETURAL.....	21
3.3 MODELO DE BANCO DE DADOS.....	23
3.3.1 Sistema de Pontuação.....	25
3.4 VALIDAÇÃO DE DADOS.....	27
3.5 INTEGRAÇÃO COM A API da OpenAI.....	27
3.5.1 Consumo da API da OpenAI e tratamento de erros.....	28
4. TESTES DE SOFTWARE.....	30
5. IMPLANTAÇÃO.....	33
6. MANUAL DO USUÁRIO.....	34
6.1 ACESSO AO APLICATIVO.....	34
6.2 CADASTRO DE USUÁRIO.....	35
6.3 TELA INICIAL.....	36
6.4 TELA DE GERAÇÃO DE ATIVIDADES.....	37
6.5 TELA DE MÚLTIPLA ESCOLHA.....	38
6.6 TELA DE COMPLETAR.....	39
6.7 TELA DE FLASHCARDS.....	40
6.8 TELA DE HISTÓRICO.....	41
6.9 TELA DE CURIOSIDADES.....	42
6.10 TELA DE PERFIL.....	43
7. DISCUSSÃO, LIMITAÇÕES E TRABALHOS FUTUROS.....	44

7.1 JUSTIFICATIVA DAS ESCOLHAS TECNOLÓGICAS.....	44
7.2 LIMITAÇÕES DO TRABALHO.....	44
7.3 TRABALHOS FUTUROS.....	45
8. CONCLUSÃO.....	47
9. AGRADECIMENTOS.....	48
10. REFERÊNCIAS.....	49

1. VISÃO GERAL E DECLARAÇÃO DO PROBLEMA

A difusão da língua espanhola está sendo deficitária no Brasil. Segundo Pinto (2023), as recentes mudanças na declaração da Lei de Diretrizes e Bases da Educação Nacional criaram um ambiente de profunda fragilização do ensino do idioma. Essa negligência legal, que vem sendo imposta na última década, favorece o apagamento do idioma no país e uma desconexão da identidade latino-americana. Atualmente, no Brasil, não há rigor suficiente do Poder Legislativo para concretizar o ensino do idioma dentro das escolas. E, por isso, o ensino do espanhol para o ensino básico e secundário torna-se somente uma possibilidade: tem caráter optativo e depende de disponibilidade de oferta, locais e horários definidos pelo sistema educacional (BRASIL, 2017). No cenário atual, com o avanço das tecnologias, surgem aplicativos de idiomas que tentam guiar estudantes em seus estudos, fenômeno este impulsionado pela popularização dos smartphones e pela flexibilidade da chamada aprendizagem-móvel (FERREIRA; BULLIO, 2024). Embora essas plataformas digitais tenham ampliado o acesso ao aprendizado, muitas delas ainda utilizam abordagens baseadas em progressões lineares de conteúdo, oferecendo limitada adaptação às necessidades específicas dos usuários. Esse conteúdo mais genérico acaba por se tornar pouco atrativo, o que afasta os falantes em potencial.

A correlação do Brasil com o ensino do idioma espanhol remonta a sua relação com o MERCOSUL. Em 1991, houve a assinatura de um Protocolo de Intenções entre Brasil, Argentina, Paraguai e Uruguai que preconiza a educação como alicerce da integração dos países e afirma que a herança cultural dos povos latino-americanos é comum. Como consequência, declarou-se o interesse de difundir o espanhol e o português nos sistemas educacionais formais, não formais e informais (MERCOSUL, 1991). Esse protocolo culminou, no Brasil, na aprovação da Lei n.º 11.161, de 5 de agosto de 2005, que determinava a oferta obrigatória da língua espanhola no ensino médio (BRASIL, 2005). Além disso, a disciplina deveria ser ofertada no horário regular de aula, o que não exigia do aluno ter que ficar no colégio em seu contraturno. A resolução ainda cita em seu Art.3º, a implantação de Centros de Língua Estrangeira nos sistemas públicos de ensino com programação desenhada para ofertar o espanhol (BRASIL, 2005). Conforme Pinto (2023), essa

lei aumentou a oferta de aulas de espanhol na Educação Básica, expandiu Centros de Línguas Estrangeiras e incentivou a abertura de cursos de licenciatura em Letras-Espanhol nas universidades. Tudo isso foi um avanço considerável para a integração dos países do MERCOSUL que repercutiu, posteriormente, na Lei de Diretrizes e Bases da Educação Nacional (LDB) do Brasil.

Contudo, a partir de 2017 foi possível perceber um retrocesso na LDB com a sanção da Lei n.13.415, popularmente conhecida como a reforma do ensino médio. Esta lei promoveu o inglês como a única língua estrangeira de oferta obrigatória, rebaixando o espanhol à língua optativa (BRASIL, 2017). Em 2024, foi sancionada a lei nº 14.945/2024, com algumas alterações na LDB, mas que manteve o status do espanhol como facultativo. Ela somente instruiu que caso outras línguas estrangeiras fossem ofertadas, que o espanhol tivesse preferência. Na prática, não há grandes alterações, visto que a aprovação de leis em favor do espanhol (e não em favor de sua obrigatoriedade) não assegura seu estabelecimento efetivo (Pinto, 2023).

No contexto de deficiência do ensino formal, numa tentativa de burlar essa limitação, os estudantes fazem o acesso a materiais de estudo por meio de aplicativos de aprendizagem de idiomas. No entanto, a abordagem adotada por essas aplicações traz à baila um problema pertinente e quase perene: rigidez. Plataformas comuns de ensino, como o Duolingo, operam seu funcionamento em regras de negócio que funcionam de maneira linear. Os usuários começam em uma trilha básica de conteúdos e vão avançando conforme completam os módulos disponíveis. E, caso queiram avançar, devem fazer um teste que mostrem que são capazes de responder as perguntas desses conteúdos genéricos. Isso não leva em consideração o caráter dinâmico da vida, que é a mudança e necessidade imediata criada por contexto. Tendo isso em vista, é pouco proveitoso que pessoas com demandas específicas tenham de se sujeitar a cronogramas que não respeitam seu tempo. Por exemplo, um engenheiro com um projeto para apresentar no Chile, seria obrigado a passar por módulos que explanam sobre vocabulário alimentício. Consequentemente, causa desmotivação e abandono, dificultando a adoção da língua de maneira natural.

Portanto, este trabalho de conclusão de curso se debruça na mitigação dos impactos da retirada da obrigatoriedade do espanhol do ensino curricular do Brasil por meio de um software baseado em Inteligência Artificial. O objetivo é oferecer um ensino personalizado, atrativo e ambientado conforme as demandas do usuário. Dessa forma, a proposta busca ampliar o acesso ao aprendizado do espanhol e favorecer a aproximação linguística do Brasil com os demais países hispanofalantes da América Latina.

1.2 PROPOSTA DE SOLUÇÃO DE SOFTWARE

A proposta de solução de software para o problema proposto é o Magia, um aplicativo de aprendizagem de língua espanhola, que se apoia em inteligência artificial generativa para criar experiências de estudo focalizadas. Nesse sentido, o Magia torna-se um sistema que se relaciona com o usuário por meio das demandas provenientes dele, sendo diferente de aplicativos convencionais presentes no mercado, como o Duolingo e o Busuu.

O principal objetivo do software é permitir que o estudante escolha o tema de sua preferência, seja profissional ou recreativo. Após solicitar, ele receberá a atividade estruturada de acordo com a temática e com a estrutura da atividade escolhida. A solução é integrada com a API da OpenAI, pois usa Grandes Modelos de Linguagem (Large Language Models — LLMs) para processar a solicitação feita pelo usuário em linguagem natural.

Todo pedido enviado pelo usuário leva em consideração o nível de dificuldade desejado baseado no CEFR — *Common European Framework of Reference for Languages*. O CEFR é um quadro de referência internacional desenvolvido pelo *Council of Europe*, que estabelece parâmetros comuns para descrever a proficiência linguística de falantes de diferentes idiomas (COUNCIL OF EUROPE, 2020). Seu propósito é oferecer uma base compartilhada para a elaboração de currículos, materiais didáticos e exames de proficiência, permitindo comparar o nível de domínio de um idioma entre diferentes contextos educacionais e países. Para isso, o CEFR avalia quatro competências comunicativas principais:

1. compreensão oral, referente à capacidade de entender a língua falada.

2. compreensão escrita, referente à capacidade de interpretar textos.
3. produção oral, referente à capacidade de se expressar verbalmente.
4. produção escrita, referente à capacidade de redigir textos de forma coerente e gramaticalmente adequada.

Essa classificação divide a complexidade do idioma e facilita o ajuste do vocabulário e da gramática retornados pela LLM. O CEFR classifica seis níveis de proficiência. Cada um dos seis níveis de proficiência do quadro apresenta descrições específicas para essas quatro competências, possibilitando uma avaliação mais objetiva do domínio do idioma pelo aprendiz. Do mais básico para o mais avançado, são eles: A1, A2, B1, B2, C1, C2. O nível A1 é o iniciante e traduz o estudante que consegue lidar com situações extremamente básicas. O A2 ainda é básico, mas o falante consegue resolver questões simples do cotidiano. O B1 é um nível intermediário e já é um marcador de independência. O B2 é um intermediário avançado, sendo possível a comunicação em situações mais complexas, como contextos laborais e acadêmicos. O C1 é um nível avançado e é definido por vocabulário bastante complexo e facilidade com abstrações. Por fim, o C2 é o nível equivalente ao de um nativo altamente instruído.

Munido de temáticas definidas pelo usuário e nível de exigência, o software estrutura a atividade através de quatro modalidades principais de interação:

1. Múltipla Escolha: Focado na identificação de estruturas gramaticais e semânticas dentro do tema escolhido, permitindo que o usuário valide seu conhecimento de forma rápida e intuitiva.
2. Complete: Desafia o usuário a evocar o vocabulário e aplicar a ortografia correta, exigindo a digitação da resposta. Este módulo foca na fixação da escrita e na memorização ativa.
3. Flashcards: Gera cartões de estudo com definições e exemplos contextuais, promovendo a expansão do léxico de forma natural.
4. Imersão Cultural (Curiosidades): Uma seção de leitura complementar, gerada a partir da curiosidade do usuário, que traz fatos e aspectos culturais do mundo hispânico, combatendo o isolamento cultural mencionado no problema e humanizando o aprendizado técnico.

Com essas funcionalidades, o aplicativo é fundamental para mitigar o problema existente. Magia dá autonomia para o indivíduo decidir o que deve ser aprendido e opera como um software que minimiza a lacuna deixada pela retirada do espanhol das escolas.

1.3 TECNOLOGIAS ADOTADAS

Para a implementação do aplicativo Magia, foram selecionadas stacks tecnológicas que estão sendo muito visadas no mercado de desenvolvimento de software. Essa escolha é estratégica, pois, por esses motivos, essas stacks estão ganhando novas versões de maneira recorrente, com correção de bugs e melhorias de segurança.

1.3.1 Front-end: Ionic com Angular e Capacitor

Para o desenvolvimento da interface da aplicação, foi utilizado o Ionic com Angular. O Ionic Framework é um kit de ferramentas de interface do usuário de código aberto que possibilita a criação de aplicações móveis multiplataforma utilizando tecnologias web, tais como HTML, CSS e Javascript (IONIC, 2026). Enquanto o Angular oferece uma arquitetura baseada em componentes, favorecendo a organização e a manutenção da aplicação (ANGULAR, 2026). Em consonância, o Capacitor foi escolhido, pois atua como ponte entre a aplicação web e os recursos nativos do dispositivo, permitindo a geração e execução do aplicativo em plataformas Android (CAPACITOR, 2026). A integração dessas tecnologias contribui para o desenvolvimento de uma aplicação moderna e voltada ao ambiente mobile.

1.3.2 Back-end: NestJs

Para o desenvolvimento da API e a lógica de servidor, foi adotado o NestJs, um framework progressivo para Node.js. O NestJs utiliza TypeScript por padrão, o que facilita o desenvolvimento, uma vez que o Angular também usa essa linguagem. Conforme a documentação do NestJs (2026), a arquitetura modular e inspirada em princípios de engenharia de software, como Injeção de Dependência, torna o back-end altamente testável, escalável, desacoplado e manutenível.

1.3.3 Banco de Dados: MariaDB

Para a persistência dos dados, foi selecionado o MariaDB, um sistema de gerenciamento de banco de dados relacional de código aberto. A plataforma oferece alto desempenho e confiabilidade para o armazenamento de dados. Ademais, a compatibilidade com o ecossistema SQL e a eficiência no tratamento de transações garantem maior integridade e consistência dos dados dos usuários (MariaDB, 2026).

1.3.4 Integrações e Ferramentas Adicionais

- **Inteligência Artificial (API da OpenAI):** A solução utiliza APIs de modelos de linguagem de grande escala (LLMs) para a geração das atividades em formato JSON.
- **TypeScript:** Linguagem base de todo o projeto, escolhida por adicionar tipagem estática ao JavaScript, reduzindo erros em tempo de execução e aumentando a produtividade.
- **Node.js:** Ambiente de execução para o back-end, escolhido pela sua eficiência em operações de I/O assíncronas, ideais para o consumo de APIs.

A combinação dessas tecnologias permite que o aplicativo Magia entregue uma solução tecnicamente atualizada e viável.

1.4 TRABALHOS RELACIONADOS

1.4.1 O Conceito de MALL e a Evolução da Mobilidade

O campo de atuação do projeto Magia fundamenta-se no conceito de *Mobile-Assisted Language Learning* (MALL), uma vertente da Linguística Aplicada que investiga o uso de tecnologias móveis para o ensino de idiomas. Segundo Kukulska-Hulme (2018), o MALL não se define apenas pelo uso do dispositivo em si, mas pela exploração de situações onde a portabilidade e a aprendizagem situada oferecem vantagens específicas que o ensino tradicional em sala de aula ou em laboratórios fixos não consegue suprir.

Desde o início da década de 2000, o MALL evoluiu de experimentos com dispositivos limitados, como *palmtops* e dicionários eletrônicos, para a onipresença dos smartphones modernos. Kukulska-Hulme enfatiza que a força motriz dessa transição é a ubiquidade. No manuscrito original, a autora cita dados da GSMA (2018) que previam que o número de assinantes móveis atingiria 5,9 bilhões em 2025, o equivalente a 71% da população mundial. Essa previsão foi superada pela realidade. Segundo dados da União Internacional das Telecomunicações de 2023, 78% da população acima de 9 anos já possui acesso a aparelhos telefônicos, consolidando o smartphone como uma ferramenta de comunicação e como portal de acesso ao conhecimento global.

1.4.2 Da Fixidez do CALL à Espontaneidade do MALL

Para compreender a relevância do Magia, é preciso analisar a transição do *Computer-Assisted Language Learning* (CALL) para o MALL. O CALL, modelo predominante até o final dos anos 90, baseia-se no uso de computadores fixos. Embora tenha revolucionado o ensino com softwares interativos, o CALL é limitado pelo caráter fixo que possui, pois o aprendizado ocorre em um local específico e com tempo pré-determinado.

Em contraste, Kukulska-Hulme e Shield (2008) argumentam que o MALL difere do CALL por permitir novas formas de aprendizagem que enfatizam a continuidade e a espontaneidade. Enquanto o CALL é frequentemente formal e isolado, o MALL permite que o aprendizado aconteça durante a vida cotidiana. A autora introduz o conceito de *Mobile-Assisted Language Use* (MALU), sugerindo que os aprendizes hoje têm a oportunidade de aprender a língua através do uso diário dos dispositivos para fins sociais ou acadêmicos, tornando a distinção entre estudo e tempo de lazer cada vez mais difícil de definir.

Essa mudança de paradigma transforma qualquer ambiente (um café barulhento, uma fila de banco, a recepção de um hospital ou o trajeto de ônibus) em uma potencial sala de aula. Mas essa transição impõe desafios: o aprendizado em contextos móveis compete com distrações e ambientes estimulantes. Porém, essa mesma característica enriquece a experiência, pois permite que o aluno faça fotos, vídeos e gravações de áudio para registrar o uso real da língua e levá-lo para a

discussão em sala de aula posterior, criando uma ponte entre o aprendizado formal e informal.

1.4.3 Privacidade e Inclusão Social

Um dos pontos mais relevantes destacados no manuscrito de Kukulska-Hulme, e que corrobora a proposta do Magia, é a questão da privacidade. A aprendizagem de uma nova língua é um processo que pode gerar vulnerabilidade, especialmente nas fases iniciais de prática de pronúncia e fala. Computadores compartilhados em ambientes familiares ou laboratórios públicos nem sempre oferecem o ambiente seguro necessário para o erro. Dispositivos móveis, por serem pessoais e portáteis, permitem que o aluno possa praticar a fala ou a escuta em locais onde se sinta confortável, sem medo de julgamento.

O MALL também se manifesta como um poderoso mecanismo de inclusão. Kukulska-Hulme aponta que o acesso a smartphones estende as oportunidades de aprendizagem para grupos que historicamente tiveram acesso limitado a materiais educacionais, como migrantes, refugiados e profissionais com rotinas de viagem intensas. Para esses grupos, a mobilidade não é um luxo, mas uma necessidade de sobrevivência e integração social.

1.4.4 O Cenário Atual e o Magia

O mercado de tecnologia educacional, atento a essas bases teóricas, consolidou gigantes como o Busuu e o Duolingo. A evolução prossegue com a integração de Inteligência Artificial generativa, como o *Little Language Lessons* da Google (2025), que busca oferecer diálogos cada vez mais naturais e situados.

O projeto Magia insere-se justamente nesta trajetória evolutiva. Ao alinhar a necessidade de práticas situadas, o respeito à privacidade do aprendiz e a utilização de ferramentas que diminuem a barreira entre o formal e o informal, o Magia busca não apenas ser mais um aplicativo, mas uma solução que compreende a mobilidade como o eixo central da aprendizagem de idiomas. A análise a seguir detalha como as escolhas pedagógicas do Magia se comparam e se diferenciam das soluções consolidadas no mercado.

1.4.5 Plataformas Baseadas em Currículos Fixos: Duolingo e Busuu

As plataformas Duolingo e Busuu são dois dos maiores e mais consolidados aplicativos no mercado de ensino de idiomas. Ambos fazem uso de elementos de gamificação para envolver o usuário nas atividades, de sistemas de repetição espaçada para revisão de atividades e de trilhas estruturadas de aprendizagem para promover o acesso ao vocabulário.

Porém, esse modelo tem uma limitação significativa no que tange a personalização dos conteúdos. Tanto no Duolingo, quanto no Busuu, a lógica adotada é a de projetar um modelo para todos os usuários, guiando o estudante por tópicos básicos e padronizados.

Nesse contexto, surge o principal diferencial do Magia: substituição do currículo fixo por um sistema de geração de conteúdo sob demanda. Enquanto o Busuu se destaca pela presença de vídeos com falantes nativos e o Duolingo pela abordagem gamificada, os dois aplicativos atuam fortemente por conta de bancos de dados previamente estruturados. Então, o Magia surge para romper essa lógica, permitindo que o aprendizado siga demandas reais e contextualizadas.

1.4.6 Experimentos em IA Generativa: Little Language Lessons (Google Gemini)

O projeto Little Language Lessons, lançado recentemente pelo Google Gemini, permeia o campo da aprendizagem de idiomas por meio da Inteligência Artificial generativa. É uma ferramenta que possibilita a escolha de temas específicos, geração dinâmica de vocabulário e geração de palavras a partir de fotos. São pontos muito positivos quando trata-se de aprendizagem por contexto. Todavia, há uma limitação que reside no aspecto educacional, uma vez que o sistema tem sua mecânica baseada em uma experiência mais passiva. O Little Language Lessons, por mais que seja personalizável, foca bastante em leitura, o que retira a prática ativa do usuário e carece de acompanhamento de progresso e de revisão do conteúdo, pois não salva o histórico. Diferente do Little Language Lessons, o Magia transforma o conteúdo gerado pela IA em atividades estruturadas, promovendo uma aprendizagem ativa e contínua. Dessa forma, o usuário não

apenas consome informações, mas também pratica, revisa e consolida o conhecimento adquirido.

1.4.7 Quadro Comparativo de Funcionalidades

Tabela 1 - Comparativo entre o Magia e soluções existentes

Funcionalidade	Duolingo / Busuu	Little Language Lessons	Magia
Currículo	Linear e fixo	Dinâmico (IA)	Autônomo e personalizado
Base de Conteúdo	Estática	Dinâmico (IA)	Dinâmica (IA)
Interatividade	Ativa (exercícios)	Passiva	Ativa (exercícios)
Feedback	Monolíngue	Inexistente	Bilíngue (PT/ES)
Revisão	Sim	Não	Sim
Flashcards	Não	Não	Sim
Contextualização	Baixa	Alta	Alta
Gamificação	Alta	Baixa	Média

Fonte: Elaborado pelo autor (2026).

2. REQUISITOS

Nesta seção são apresentados os requisitos do sistema desenvolvido, os quais estão divididos em requisitos funcionais e não funcionais.

2.1 REQUISITOS FUNCIONAIS

Os requisitos funcionais descrevem as principais funcionalidades e comportamentos esperados da aplicação para atender os usuários.

Tabela 2 - Requisitos funcionais

ID	Ator	História de Usuário
RF01	Sistema	Como sistema, devo enviar um prompt para a API da OpenAI contendo tema, nível e formato JSON.
RF02	Estudante	Como usuário, quero inserir um tema de interesse para que o conteúdo seja gerado sob demanda.
RF03	Estudante	Como usuário, quero definir o nível de proficiência para que a IA ajuste a complexidade.
RF04	Estudante	Como usuário, quero realizar exercícios de múltipla escolha gerados para validar meu conhecimento.
RF05	Estudante	Como usuário, quero digitar respostas em frases com lacunas para praticar a escrita e a sintaxe.
RF06	Estudante	Como usuário, quero receber explicações pedagógicas para cada questão ou flashcard estudado.

RF07	Estudante	Como usuário, quero criar uma conta e realizar login para manter as minhas atividades personalizadas salvas.
RF08	Sistema	Como sistema, devo salvar os temas já estudados no banco de dados MariaDB.
RF09	Estudante	Como usuário, quero acessar flashcards com definições e exemplos para reforçar o vocabulário do tema.
RF10	Estudante	Como usuário, quero poder gerar curiosidades dos países da América Latina.

Fonte: Elaborado pelo autor (2026).

2.2 REQUISITOS NÃO-FUNCIONAIS

Nesta seção, são descritos os requisitos não-funcionais que o aplicativo Magia deve atender para garantir uma operação eficiente e segura.

[RNF01] Portabilidade: O sistema deve ser capaz de operar em dispositivos Android, utilizando a base de código fornecida pelo Ionic Framework.

[RNF02] Desempenho (Tempo de Resposta): Durante o processo de geração das atividades, a interface deve exibir um indicador de carregamento para o usuário.

[RNF03] Usabilidade: A interface do usuário deve ser intuitiva e seguir princípios de experiência do usuário, permitindo que um estudante consiga gerar uma atividade personalizada em no máximo quatro cliques a partir da tela principal.

[RNF04] Segurança de Dados: Todas as informações sensíveis dos usuários, como senhas de acesso, devem ser armazenadas no MariaDB utilizando algoritmos de criptografia (hashing), garantindo a proteção contra acessos não autorizados.

[RNF05] Manutenibilidade: O sistema deve ser desenvolvido utilizando TypeScript e seguindo os padrões de arquitetura modular do NestJs (back-end) e Angular (front-end), facilitando a identificação de erros e a implementação de futuras funcionalidades.

[RNF06] Disponibilidade: A aplicação deve estar disponível para acesso 24 horas por dia, 7 dias por semana, estando seu funcionamento relacionado à disponibilidade da API da OpenAI.

[RNF07] Escalabilidade: O back-end estruturado em NestJs deve ser capaz de suportar o aumento do número de usuários e de requisições simultâneas sem aumento demasiado no tempo de processamento.

[RNF08] Integridade de Formato: O sistema de processamento deve garantir que a comunicação entre o back-end e a IA resulte estritamente em objetos JSON válidos, prevenindo falhas na interface.

[RNF09] Idioma da Interface: A interface do sistema deve estar em português, enquanto os conteúdos gerados para estudo devem ser apresentados em espanhol ou português, conforme dinâmica da aplicação.

3. DESIGN

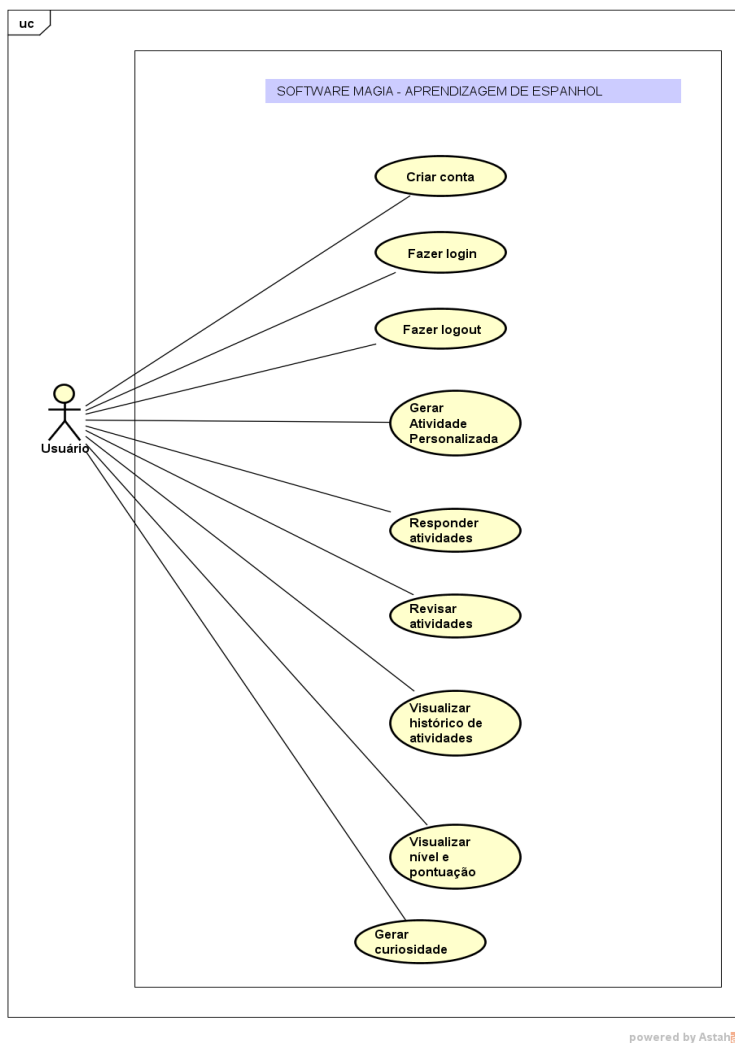
Esta seção apresenta a modelagem estrutural e comportamental do software Magia, utilizando diagramas UML (Unified Modeling Language) para representação do sistema. Os diagramas construídos ajudam na visualização das funcionalidades, das interações entre usuários e sistema e da organização dos componentes internos da aplicação.

3.1 PROJETO UML

Com o objetivo de representar o sistema proposto, foram elaborados diagramas UML por meio do software Astah. Ele foi escolhido por sua interface intuitiva e pelo acesso gratuito para estudantes.

3.1.1 Diagrama de casos de uso

Figura 1 - Diagrama de casos de uso

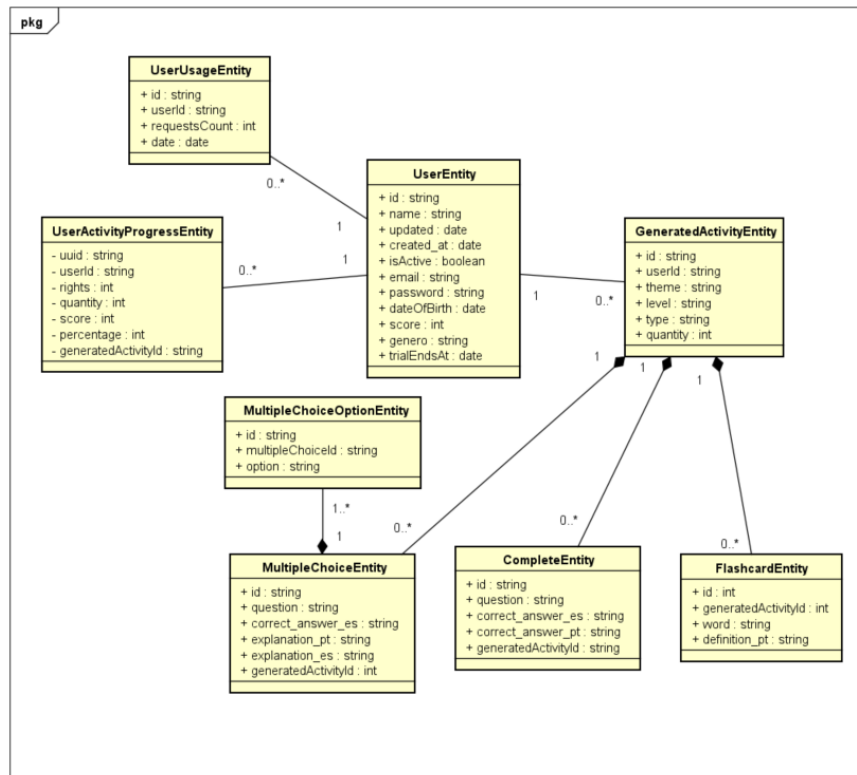


Fonte: Elaborado pelo autor (2026).

3.1.2 Diagrama de classe e de sequência

Para representar a estrutura estática do sistema, foi elaborado o diagrama de classes. As classes apresentadas seguem um modelo anêmico, contendo principalmente atributos responsáveis pela representação dos dados, enquanto as regras de negócio estão concentradas nas camadas de serviço da aplicação, conforme a arquitetura adotada.

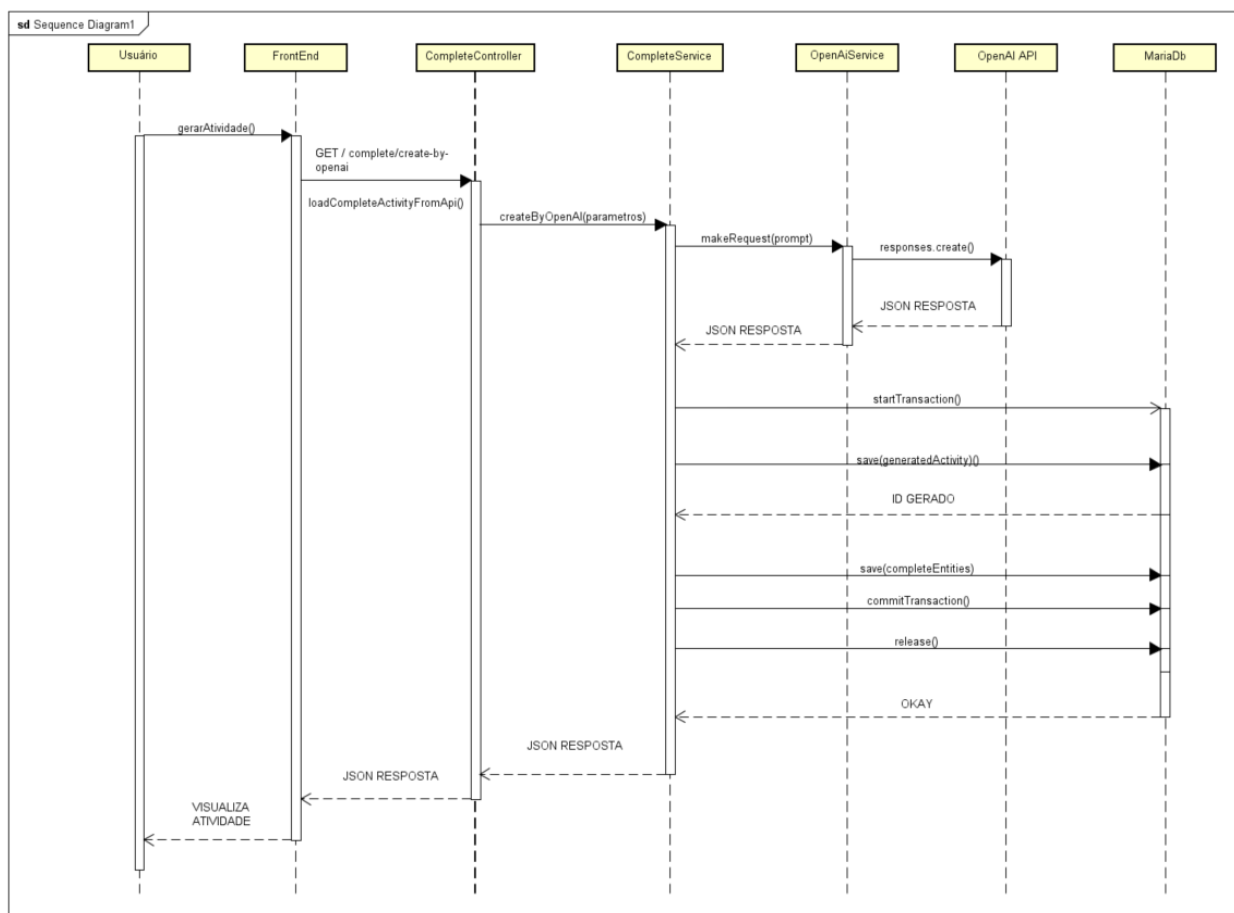
Figura 2 - Diagrama de classes



Fonte: Elaborado pelo autor (2026).

Para esclarecer o fluxo do sistema, foi elaborado um diagrama de sequência para demonstrar o fluxo de interação do usuário ao gerar uma atividade. Diferentemente dos diagramas estruturais, que apresentam a organização dos elementos do sistema, o diagrama de sequência evidencia a ordem das mensagens trocadas e dos métodos invocados entre as diferentes partes do sistema. O fluxo escolhido é o de uma atividade do tipo de completar, permitindo visualizar como a requisição realizada pelo usuário é processada pelo aplicativo móvel, pela API back-end, pela API da OpenAI e pelo banco de dados. Dessa forma, o diagrama contribui para a compreensão do comportamento dinâmico do sistema e da colaboração entre seus elementos durante a execução das operações.

Figura 3 - Diagrama de Sequência da atividade de Completar



Fonte: Elaborado pelo autor (2026).

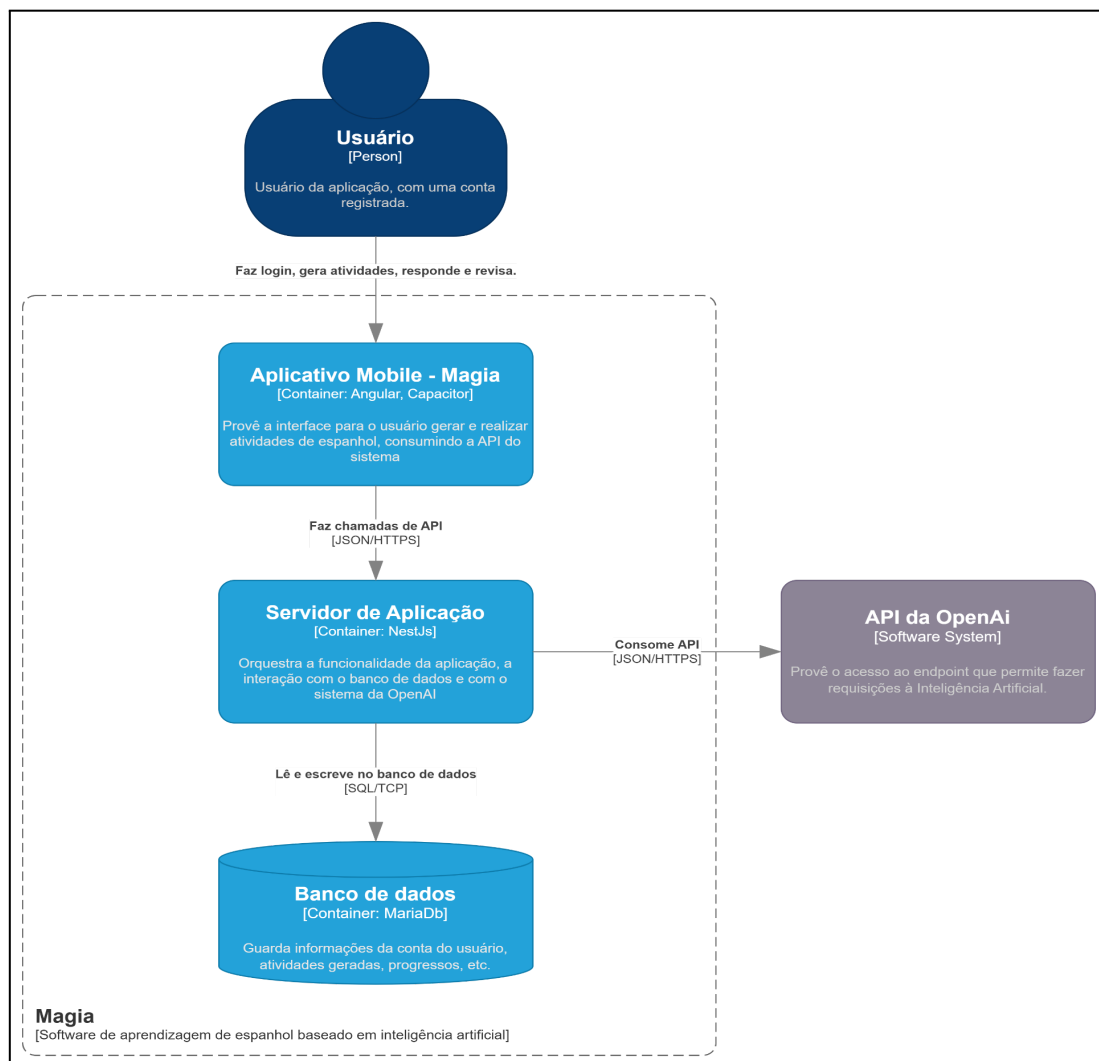
3.2 VISÃO ARQUITETURAL

O sistema Magia – Fale como Mágica foi desenvolvido seguindo o modelo Cliente-Servidor e o estilo arquitetural de Arquitetura em Camadas (Layered Architecture). Esta abordagem promove a separação de responsabilidades entre os componentes do sistema. No modelo cliente-servidor, o aplicativo móvel atua como cliente, sendo responsável pela interação com o usuário e pelo envio de requisições ao servidor. O servidor, por sua vez, processa as requisições recebidas, executa as regras de negócio, realiza operações de persistência de dados e integra-se a API da OpenAI.

A camada de apresentação foi implementada utilizando Angular, sendo posteriormente empacotada como aplicativo Android por meio do Capacitor. A camada de aplicação foi desenvolvida utilizando NestJs. Nessa camada

encontram-se os controladores (controllers), responsáveis pelo recebimento das requisições HTTP, e os serviços (services), responsáveis pela implementação das regras de negócio da aplicação. A persistência dos dados é realizada através do TypeORM, responsável pelo mapeamento objeto-relacional entre as entidades da aplicação e o banco de dados MariaDB. Neste banco, são armazenadas informações de usuários, atividades geradas, progresso de aprendizagem, tentativas realizadas e demais dados necessários ao funcionamento do sistema. Além dos componentes internos, o sistema integra-se à API da OpenAI. As requisições à inteligência artificial são realizadas pelo backend, que recebe as respostas e as armazena no banco de dados. A Figura 4 apresenta a arquitetura geral do sistema.

Figura 4 - Visão arquitetural do sistema Magia



Fonte: Elaborado pelo autor (2026).

3.3 MODELO DE BANCO DE DADOS

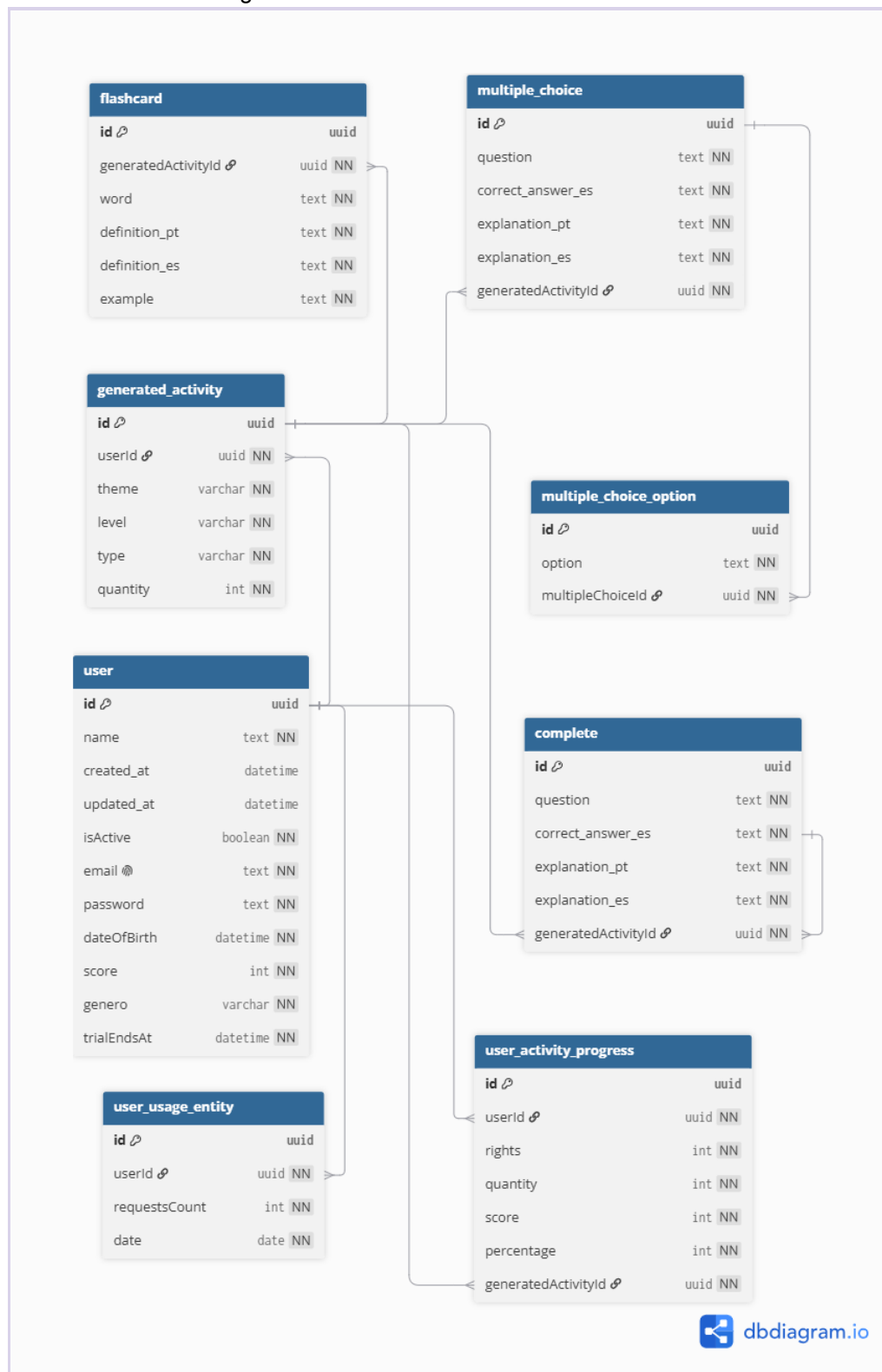
O sistema utiliza um banco de dados relacional para armazenar informações dos usuários, atividades geradas por inteligência artificial, questões, flashcards e progresso de aprendizagem. As principais entidades do sistema são:

- Usuário (user): Armazena os dados cadastrais dos usuários da plataforma, incluindo informações de autenticação, pontuação acumulada e período de teste.
- Atividade Gerada (generated_activity): Representa uma atividade criada pelo sistema a partir de um tema, nível de dificuldade e tipo de exercício selecionados pelo usuário. Cada atividade pode conter diferentes recursos de aprendizagem: questões de múltipla escolha, questões de completar e flashcards.
- Questão de Múltipla Escolha (multiple_choice): Armazena perguntas, respostas corretas e explicações em português e espanhol. Cada questão possui diversas opções de resposta associadas.
- Opção de Resposta (multiple_choice_options): Representa as alternativas disponíveis para uma questão de múltipla escolha.
- Questão de Completar (complete): Armazena exercícios em que o usuário deve preencher lacunas utilizando a resposta correta.
- Flashcard (flashcard): Contém vocabulário em espanhol, definição em português, definição em espanhol e exemplo de uso.
- Progresso da Atividade (user_activity_progress): Permite acompanhar o progresso do usuário em cada atividade realizada, registrando percentual de acertos, quantidade de respostas corretas e pontuação.
- Uso do Usuário: Permite registrar a quantidade de solicitações feitas na API para determinado usuário, de modo a controlá-la.

O principal relacionamento do sistema ocorre entre as entidades Usuário e Atividade Gerada. Cada usuário pode gerar diversas atividades de aprendizagem, enquanto cada atividade pertence a um único usuário. A entidade Atividade Gerada atua como núcleo do sistema, concentrando os conteúdos educacionais produzidos pela inteligência artificial. Além disso, ela também se relaciona com os registros de

tentativas e progresso dos usuários, permitindo o acompanhamento do desempenho durante o processo de aprendizagem.

Figura 5 - Modelo de banco de dados



Fonte: Elaborado pelo autor (2026).

3.3.1 Sistema de Pontuação

Foi implementada uma trigger no banco de dados com o objetivo de atualizar automaticamente a pontuação dos usuários após a conclusão de atividades. A trigger é executada quando um novo registro de progresso é inserido na tabela `user_activity_progress`. Ao registrar um novo resultado, o sistema compara as duas maiores pontuações obtidas para a atividade correspondente. Caso essa diferença seja positiva, ela é adicionada à pontuação total do usuário. Dessa forma, quando o usuário refizer a atividade, só são conferidos pontos quando houver melhoria no desempenho.

Figura 6 - Trigger do sistema de pontuação

```
1 DELIMITER //
2
3 CREATE TRIGGER TG_USER_ACTIVITY_PROGRESS_AFTER_INSERT
4 AFTER INSERT ON user_activity_progress
5 FOR EACH ROW
6 BEGIN
7     DECLARE PrimeiroScore INT DEFAULT 0;
8     DECLARE SegundoScore INT DEFAULT 0;
9     DECLARE AdditionalScore INT DEFAULT 0;
10
11     SELECT U.score
12     INTO PrimeiroScore
13     FROM user_activity_progress U
14     WHERE U.GeneratedActivityId = NEW.GeneratedActivityId
15     ORDER BY U.score DESC LIMIT 1;
16
17     IF NEW.SCORE = PrimeiroScore THEN
18         SELECT U.score
19         INTO SegundoScore
20         FROM user_activity_progress U
21         WHERE U.GeneratedActivityId = NEW.GeneratedActivityId
22         ORDER BY U.score DESC LIMIT 1 OFFSET 1;
23
24         SET AdditionalScore = NEW.SCORE - COALESCE(SegundoScore, 0);
25     END IF;
26
27     IF AdditionalScore > 0 THEN
28         UPDATE user SET score = score + AdditionalScore
29         WHERE id = NEW.userId;
30     END IF;
31
32     END//
33
34 DELIMITER ;
```

Fonte: Elaborado pelo autor (2026).

Na seção de Perfil, o usuário consegue visualizar o acumulado de pontos conquistados, o nível atual e o próximo a ser alcançado. Atualmente, há 9 níveis e cada um deles tem um acréscimo progressivo na quantidade de pontos para ser atingido.

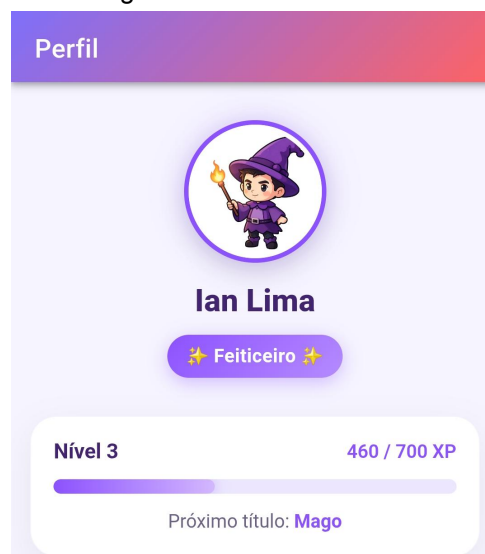
Tabela 3 - Níveis existentes na gamificação do Magia

Quantidade de Pontos	Nível
0	Aprendiz Arcano
100	Iniciado
300	Feiticeiro
700	Mago
1200	Arquimago
2000	Mestre Arcano
2800	Guardião dos Grimórios
4000	Sábio das Runas
6000	Charmed One

Fonte: Elaborado pelo autor (2026).

Na figura abaixo, é possível ver como o sistema apresenta o progresso do usuário, sua pontuação e seu nível correspondente.

Figura 7 - Perfil do usuário



Fonte: Elaborado pelo autor (2026).

3.4 VALIDAÇÃO DE DADOS

Com o objetivo de garantir a consistência e a integridade das informações manipuladas pelo sistema, foram implementados mecanismos de validação tanto no front-end quanto no back-end da aplicação.

No front-end, foi utilizado o Reactive Forms do Angular para a construção dos formulários. Essa abordagem permitiu definir regras de validação diretamente nos componentes da aplicação, verificando, por exemplo, o preenchimento obrigatório de campos e o formato correto de endereços de e-mail antes do envio das informações ao servidor.

Entretanto, as validações realizadas no lado cliente não são suficientes para garantir a segurança da aplicação, uma vez que requisições podem ser enviadas diretamente à API sem passar pela interface gráfica. Por esse motivo, foram implementadas validações adicionais no back-end utilizando DTOs (Data Transfer Objects) e a biblioteca Class Validator do NestJs. Os DTO's foram utilizados para definir a estrutura esperada dos dados recebidos pelas rotas da API, enquanto os validadores verificam automaticamente restrições como obrigatoriedade de campos, formatos válidos e tipos permitidos.

A utilização dessas duas camadas de validação contribui para a confiabilidade do sistema, reduzindo a entrada de dados inconsistentes e aumentando a segurança das operações realizadas pela aplicação.

3.5 INTEGRAÇÃO COM A API da OpenAI

A geração dinâmica de atividades foi implementada por meio da integração com a API da OpenAI, plataforma que disponibiliza modelos de inteligência artificial capazes de interpretar instruções textuais e gerar conteúdos personalizados.

Para realizar a comunicação com o serviço, foi utilizado o SDK oficial da OpenAI para Node.js. A utilização do SDK simplificou o processo de integração, abstraindo detalhes de baixo nível relacionados às requisições HTTP, autenticação e tratamento das respostas retornadas pela plataforma. A comunicação com a

OpenAI ocorre exclusivamente no backend e a chave de acesso da API permanece armazenada no servidor, não sendo exposta aos usuários da aplicação móvel.

A integração foi implementada por meio do serviço `OpenAiService`, responsável por encapsular a lógica de comunicação com a inteligência artificial. Quando o usuário solicita a geração de uma atividade, o sistema constrói dinamicamente um prompt contendo informações como tema, nível de dificuldade e quantidade de exercícios desejada. Em seguida, esse prompt é enviado ao modelo de inteligência artificial da OpenAI. Após o processamento da solicitação, a API retorna uma resposta estruturada em formato JSON contendo os exercícios gerados.

Figura 8 - Chamada à API da OpenAI

```
@Injectable()
export class OpenAiService {
  private readonly openAi: OpenAI;

  constructor() {
    this.openAi = new OpenAI({
      apiKey: process.env.OPENAI_API_KEY,
    });
  }

  async makeRequest(prompt: string): Promise<any> {
    const response = await this.openAi.responses.create({
      model: 'gpt-5.4-mini',
      store: false,
      input: prompt,
    });
    const parsed: any = JSON.parse(response.output_text);
    return parsed;
  }
}
```

Fonte: Elaborado pelo autor (2026).

3.5.1 Consumo da API da OpenAI e tratamento de erros

Como a utilização da API da OpenAI gera custos proporcionais à quantidade de requisições realizadas, foram implementadas estratégias para controlar o consumo desse serviço. Durante o período de teste de 12 dias, cada usuário pode

realizar no máximo três gerações de atividades por dia, limite controlado pelo componente TrialGuard, implementado no back-end.

Além disso, as atividades geradas são armazenadas no banco de dados MariaDB, permitindo que conteúdos já criados sejam recuperados sem a necessidade de novas consultas à inteligência artificial. Dessa forma, reduz-se a quantidade de requisições enviadas à OpenAI, contribuindo para a diminuição dos custos operacionais da aplicação. As medidas adotadas também possibilitam a implementação futura de planos de assinatura com diferentes limites de utilização, adequando o consumo dos recursos da plataforma ao perfil de cada usuário.

No que tange o tratamento de erros, para aumentar a confiabilidade da integração, as respostas retornadas pela OpenAI são validadas utilizando a biblioteca Zod. Essa validação garante que o JSON recebido esteja de acordo com a estrutura esperada pela aplicação antes que os dados sejam processados e armazenados. Dessa forma, reduz-se o risco de erros causados por respostas malformadas ou incompatíveis com os modelos de dados do sistema.

4. TESTES DE SOFTWARE

Para os testes feitos na aplicação, optou-se pela combinação de estratégias de testes manuais e testes automatizados unitários com o objetivo de validar o funcionamento dos recursos implementados, identificar falhas de comportamento e garantir que o software Magia tenha a maior estabilidade possível.

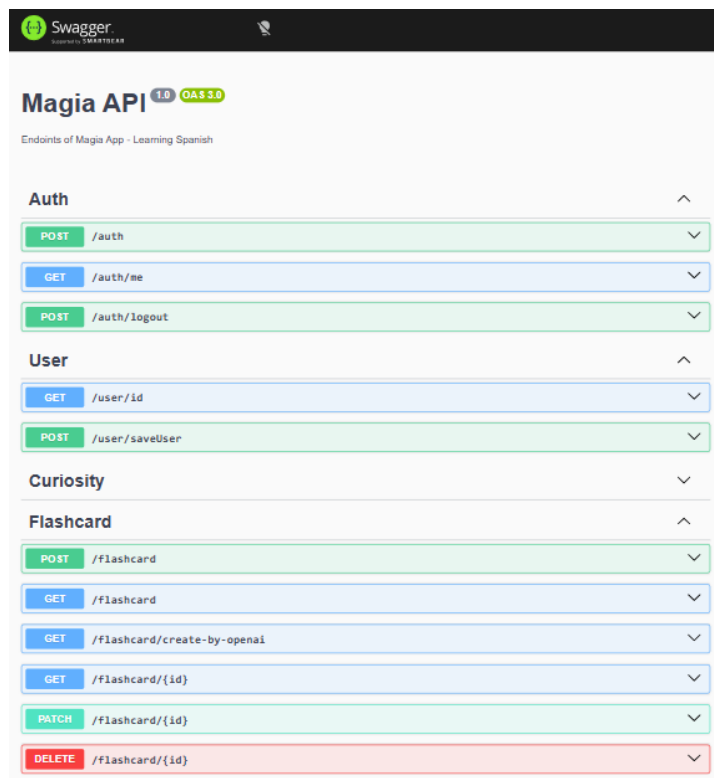
O processo de validação foi feito de maneira contínua, sendo iniciado sempre que uma nova funcionalidade fosse agregada ao sistema e tivesse o status de finalizada. Essa ação foi feita para garantir harmonia entre o back-end, o front-end e o banco de dados. Ademais, essa fase de testes era acompanhada de testes regressivos mais básicos em outras funcionalidades de modo a verificar se as novas alterações tinham impactos negativos nas funcionalidades que já estavam prontas anteriormente.

As áreas mais testadas foram as que são mais críticas para o pleno funcionamento do software, destacando-se o processo de autenticação de usuários, cadastro e login, geração de atividades personalizadas pela API da OpenAI, persistência de dados no MariaDB e a exibição das atividades na interface do usuário.

Para o processo de testes, foram utilizadas ferramentas que aceleram o desenvolvimento e ajudam na documentação, como o Swagger, que foi amplamente utilizado na validação das rotas das controllers e das requisições HTTP. Em consonância, utilizou-se as ferramentas de desenvolvedor presentes no navegador (Chrome), para monitorar as requisições feitas e para inspecionar erros, além de verificar o comportamento da interface. Logs de execução do backend também foram utilizados para identificação e correção de exceções durante a execução do sistema.

A estratégia adotada possibilitou validar o comportamento geral da aplicação, contribuindo para a identificação de inconsistências e para a melhoria da experiência de uso.

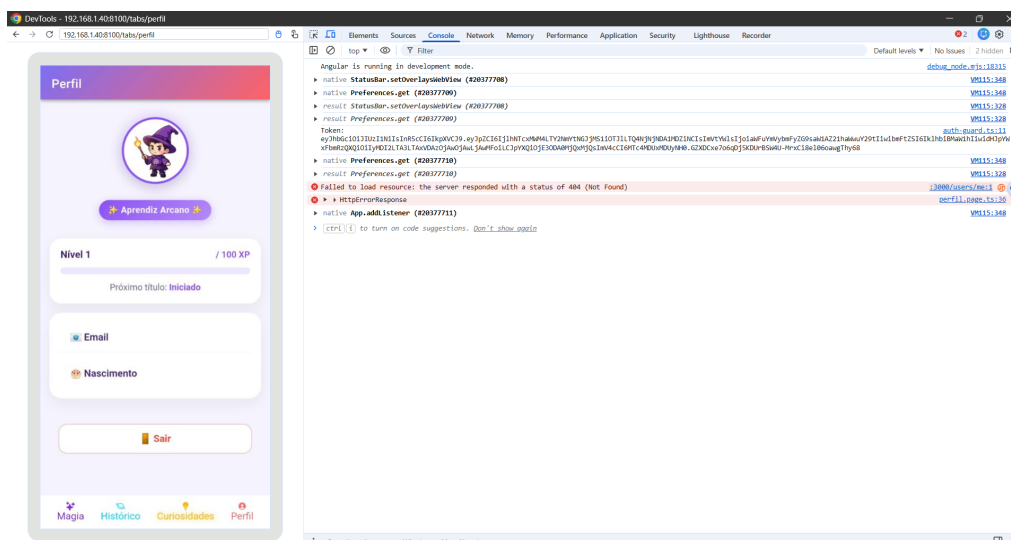
Figura 9 - Uso do swagger para testes e documentação da API



Fonte: Elaborado pelo autor (2026).

Foram feitos também testes no console do front-end, para fazer validação dos retornos das chamadas da API, além de servir para o debug da aplicação.

Figura 10 - Debug no front-end



Fonte: Elaborado pelo autor (2026).

Os testes unitários automatizados foram feitos utilizando Jest com o objetivo de validar o comportamento das principais regras de negócio da aplicação. Esses testes têm como foco garantir o correto funcionamento de componentes críticos do sistema de forma isolada, assegurando que diferentes cenários de execução sejam tratados adequadamente.

No contexto do controle de acesso e uso da plataforma, foram verificados diferentes cenários relacionados ao período de trial do usuário e ao limite diário de utilização, garantindo que o sistema aplique corretamente as restrições definidas.

Figura 11 - Testes unitários do TrialGuard

```
it('deve bloquear quando trial expirou', async () => {
  const context = mockContext({
    user: {
      id: '1',
      trialEndsAt: '2020-01-01',
    },
  });

  await expect(guard.canActivate(context as any)).rejects.toThrow(
    'Período de teste expirado.',
  );
});

it('deve bloquear quando atingir limite diário', async () => {
  const userUsageService = {
    findOne: jest.fn().mockResolvedValue({
      requestsCount: 3,
    }),
    save: jest.fn(),
  };

  const guard = new TrialGuard(userUsageService as any);

  const context = mockContext({
    user: {
      id: '1',
      trialEndsAt: '2099-01-01',
    },
  });

  await expect(guard.canActivate(context as any)).rejects.toThrow(
    'Você atingiu o limite diário de 3 gerações.',
  );
});
```

Fonte: Elaborado pelo autor (2026).

5. IMPLANTAÇÃO

O projeto de implantação consiste em especificar características de hardware e software para que o aplicativo Magia possa ser instalado de maneira eficiente.

- **Requisitos do Servidor**

HARDWARE MÍNIMO	
Componente	Especificação
Processador	2 vCPUs
Memória RAM	4 GB
Armazenamento	80 GB SSD
Conectividade	Acesso permanente à Internet
SOFTWARE	
Node.js	22.22.0
NestJs	11.0.1
OpenAI SDK	6.34.0
MariaDB	11.7.2

- **Requisitos do Dispositivo Cliente**

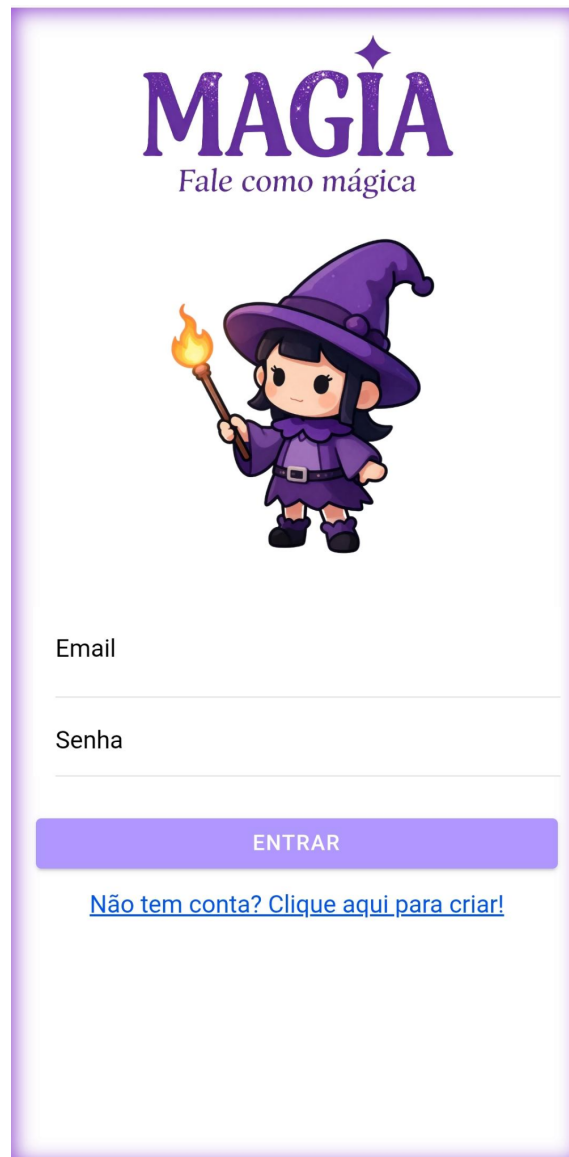
HARDWARE MÍNIMO	
Processador	Quad-Core
Processador	4 GB
Memória RAM	4GB
Armazenamento Livre	100 MB
Conectividade	Wi-Fi ou Dados Móveis
SOFTWARE	
Sistema Operacional	Android 7.0 (API 24) ou superior

6. MANUAL DO USUÁRIO

6.1 ACESSO AO APLICATIVO

Ao acessar a plataforma, o usuário é direcionado para a tela inicial, onde poderá realizar login ou criar uma nova conta.

Figura 12 - Tela de login



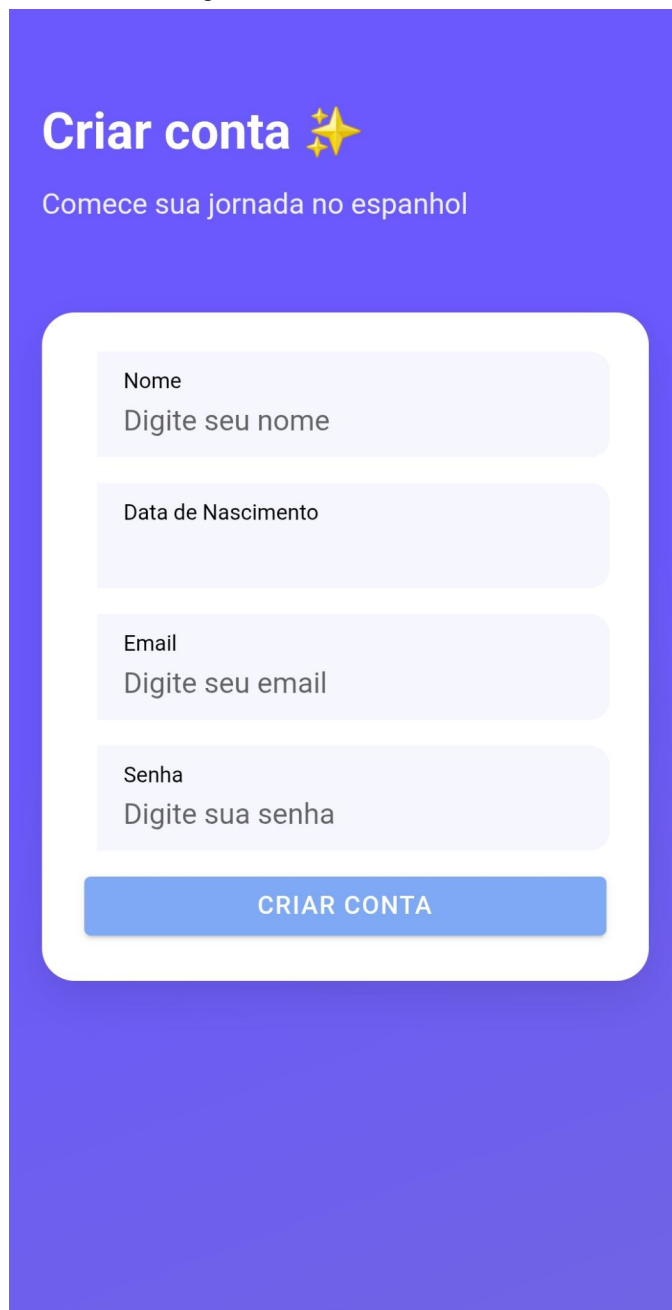
A tela de login do aplicativo 'MAGIA' apresenta o título 'MAGIA' em uma fonte serifada e decorada com estrelas, seguido pelo subtítulo 'Fale como mágica' em uma fonte cursiva. Abaixo, há uma ilustração de uma bruxinha fofa com um chapéu de bruxa roxo, segurando uma varinha mágica com uma chama. O formulário de login contém campos para 'Email' e 'Senha', ambos com linhas de entrada brancas. Abaixo dos campos, há um botão roxo com o texto 'ENTRAR' em branco. Na base da tela, há um link azul com o texto 'Não tem conta? Clique aqui para criar!'.

Fonte: Elaborado pelo autor (2026).

6.2 CADASTRO DE USUÁRIO

Para utilizar os recursos da plataforma, o usuário deve realizar um cadastro informando: nome, data de nascimento, e-mail e senha. Após o preenchimento dos dados, o sistema cria automaticamente o perfil do estudante.

Figura 13 - Tela de cadastro



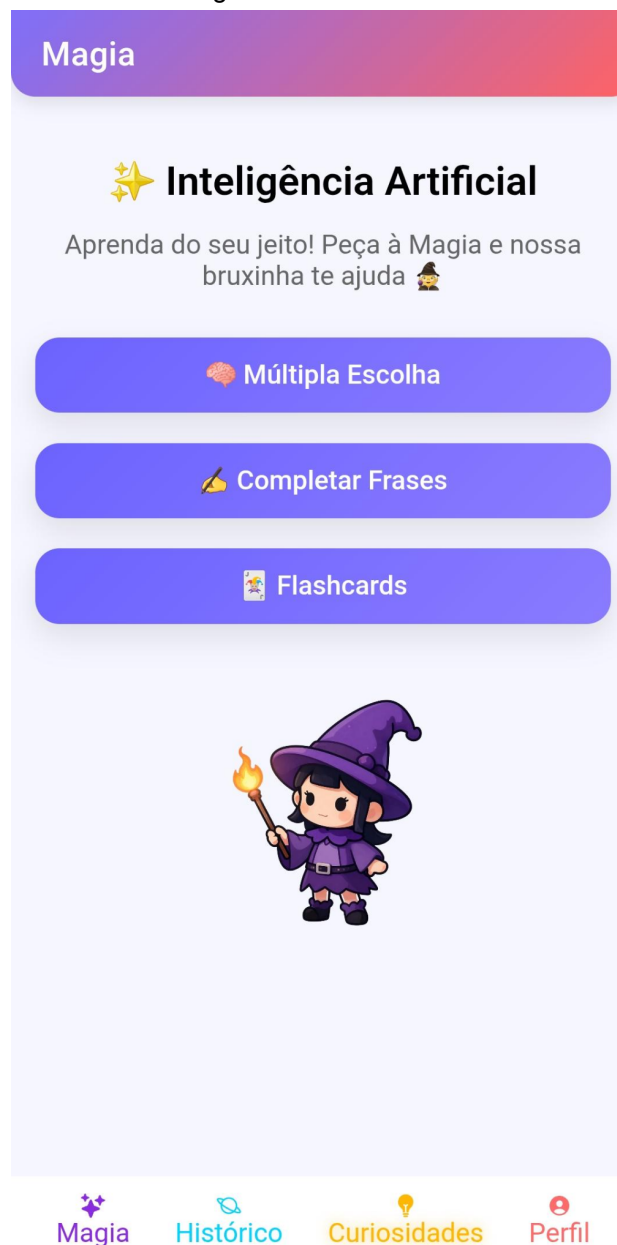
A tela de cadastro possui um fundo azul escuro. No topo, o texto "Criar conta" é exibido em branco, seguido por um ícone de estrela amarela. Abaixo, a frase "Comece sua jornada no espanhol" aparece em uma cor mais clara. O formulário principal é um retângulo branco com cantos arredondados, contendo quatro campos de entrada: "Nome" com o placeholder "Digite seu nome", "Data de Nascimento", "Email" com o placeholder "Digite seu email", e "Senha" com o placeholder "Digite sua senha". No final do formulário, há um botão azul com o texto "CRIAR CONTA" em branco.

Fonte: Elaborado pelo autor (2026).

6.3 TELA INICIAL

Na tela inicial, o usuário possui acesso às principais funcionalidades do sistema: geração de atividades, histórico, curiosidades e gerenciamento do perfil. Tudo isso pode ser acessado pela barra inferior. Na tela principal, o estudante pode escolher qual tipo de atividade ele quer.

Figura 14 - Tela Inicial



Fonte: Elaborado pelo autor (2026).

6.4 TELA DE GERAÇÃO DE ATIVIDADES

Na tela inicial, o usuário pode escolher o nível de complexidade da atividade, o tema e pode também selecionar a quantidade de questões para este tema, sendo mais uma personalização possível.

Figura 15 - Tela de criação de atividades



A interface de criação de atividades, intitulada "Completar Frases", apresenta o subtítulo "Escolha como deseja aprender hoje". O formulário contém três campos de configuração: "Nível" com uma lista suspensa selecionando "B2", "Tema" com um campo de texto contendo "Futuro do pretérito", e "Quantidade" com três botões de seleção (5, 10, 15), onde o botão "5" está ativo. Abaixo do formulário, há um botão de ação "Gerar Atividade" com um ícone de estrela. Na base da tela, há uma ilustração de uma bruxinha segurando uma varinha mágica.

Fonte: Elaborado pelo autor (2026).

6.5 TELA DE MÚLTIPLA ESCOLHA

Na tela de múltipla escolha, após escolhido o tema, o sistema vai mostrar uma pergunta e mostrar 4 opções, sendo somente uma correta. O usuário deve clicar na resposta que acredita estar correta. Há também o feedback sobre a resposta.

Figura 16 - Tela de atividades de múltipla escolha

Multipla Escolha

Pergunta 1 de 5

¿Dónde compras un billete?

En la estación

En la cama

En la sopa

Feitiço lançado com sucesso!
Você acertou a resposta.

Resposta correta: En la estación. A estação vende bilhetes de viagem.

Continuar

Fonte: Elaborado pelo autor (2026).

6.6 TELA DE COMPLETAR

Na tela de completar, após escolhido o tema, o sistema vai mostrar uma incompleta e uma dica entre parênteses. O usuário deve digitar a resposta que acredita estar correta e clicar no botão de conferir para verificar se foi exitoso.

Figura 17 - Tela de atividades de completar

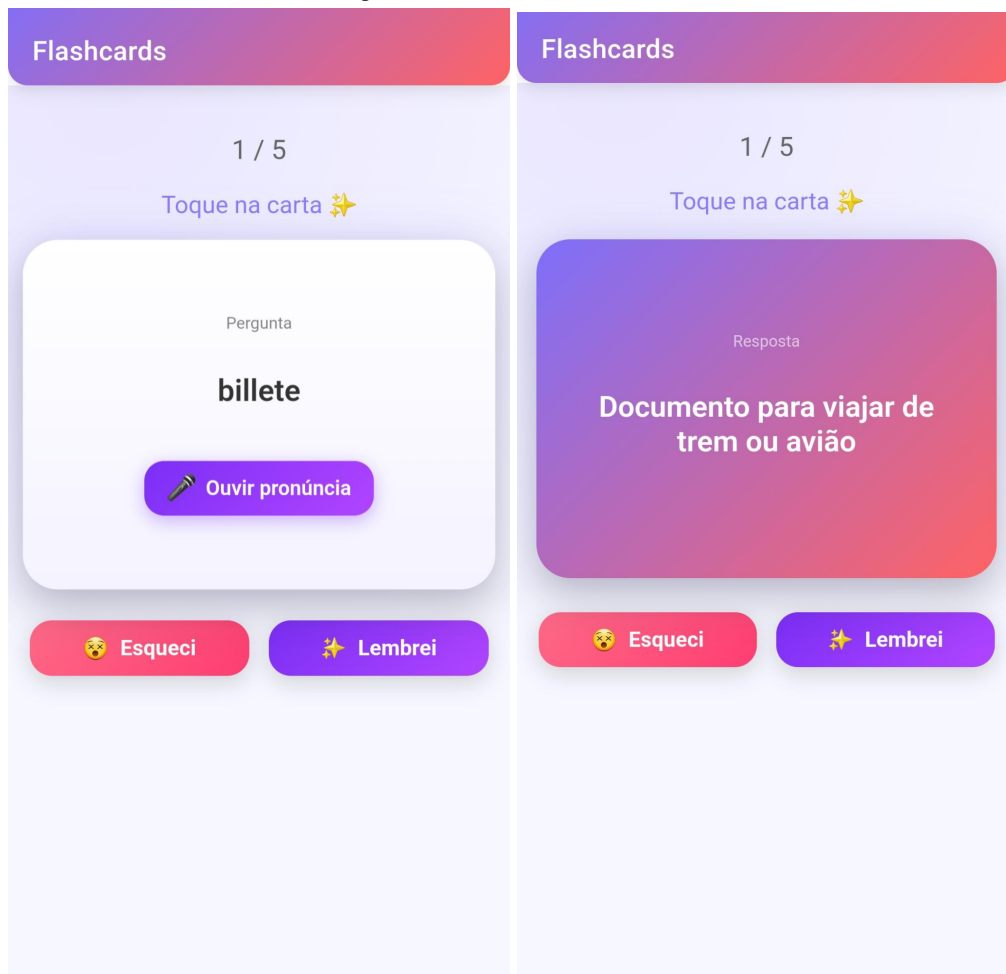


Fonte: Elaborado pelo autor (2026).

6.7 TELA DE FLASHCARDS

Na tela de flashcards, após escolhido o tema, o sistema vai mostrar uma palavra. Em caso de dúvidas sobre a palavra, ele pode clicar na carta. Ela girará e exibirá o significado que estava escondido no verso. Depois, o usuário deve sinalizar se esqueceu ou se lembrou.

Figura 18 - Telas de flashcard



Fonte: Elaborado pelo autor (2026).

6.8 TELA DE HISTÓRICO

Na tela de histórico, os usuários podem ver as pontuações das atividades anteriores e refazê-las.

Figura 19 - Telas de histórico para revisão

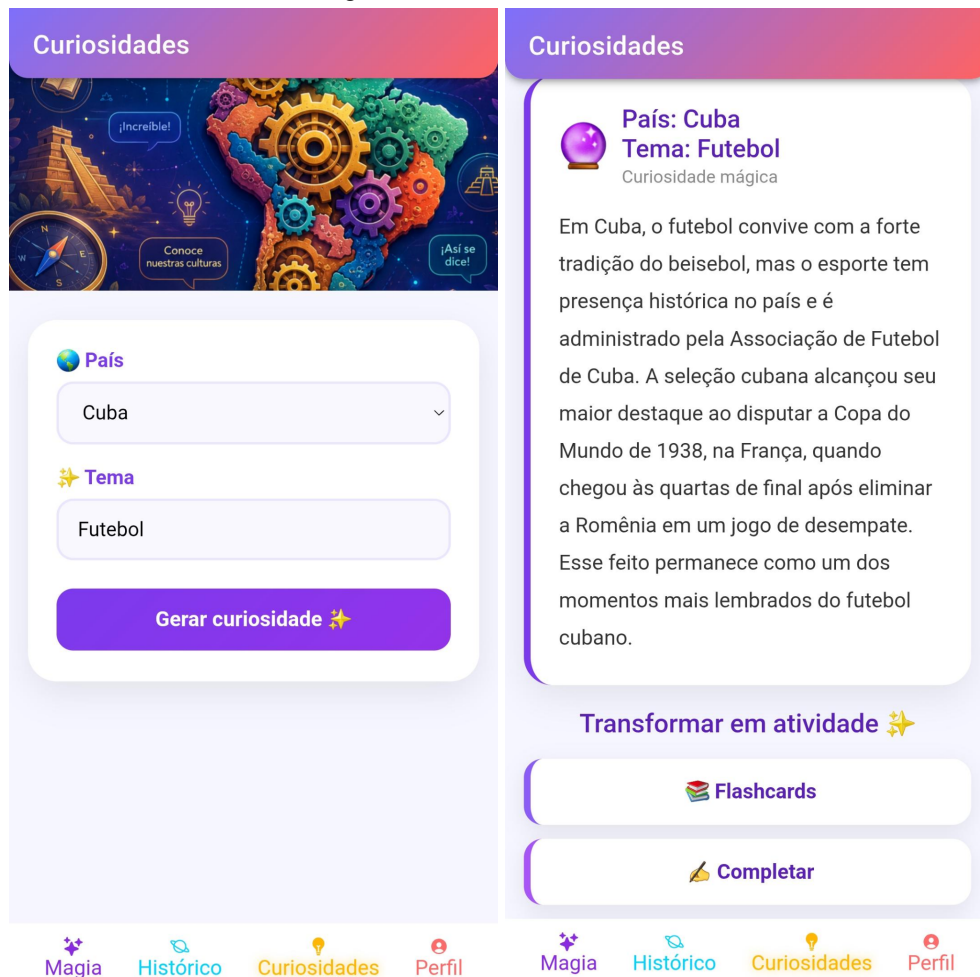


Fonte: Elaborado pelo autor (2026).

6.9 TELA DE CURIOSIDADES

Nessa tela, o usuário pode escolher um dos países da América Latina (Sul e Central) e um tema para receber uma curiosidade customizada baseada nesses parâmetros. A curiosidade é exibida para o usuário e pode virar input para a criação de uma atividade.

Figura 20 - Telas de curiosidade

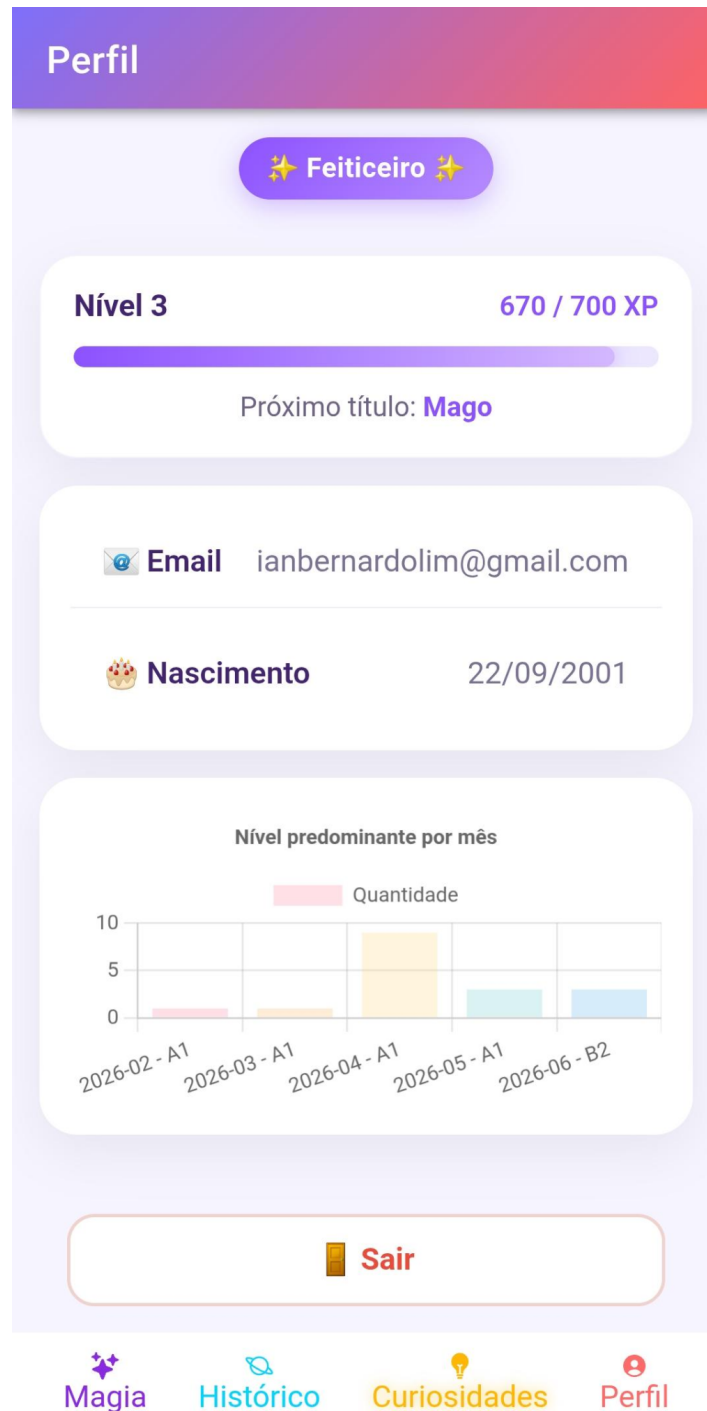


Fonte: Elaborado pelo autor (2026).

6.10 TELA DE PERFIL

Na tela de perfil, os usuários podem ver seus dados, verificar o progresso de nível do CEFR, as atividades geradas pelo dashboard e fazer logout da aplicação.

Figura 21 - Tela de perfil



Fonte: Elaborado pelo autor (2026).

7. DISCUSSÃO, LIMITAÇÕES E TRABALHOS FUTUROS

Esta seção discute justificativas técnicas de projeto, reconhece as limitações identificadas ao longo do desenvolvimento do Magia e indica possibilidades de continuidade da pesquisa.

7.1 JUSTIFICATIVA DAS ESCOLHAS TECNOLÓGICAS

A escolha do NestJs em detrimento de outros frameworks de back-end, como o Spring Boot, justifica-se principalmente pela unificação da linguagem de programação em todo o projeto. Como o front-end foi desenvolvido em Angular, a adoção do NestJs, também baseado em TypeScript, permite o compartilhamento de tipos e uma menor troca de contexto entre as camadas da aplicação, além de já apresentar uma arquitetura modular e baseada em injeção de dependência semelhante à encontrada em frameworks do ecossistema Java. De maneira parecida, a escolha do Ionic em vez do Flutter segue a mesma lógica de coesão tecnológica, viabilizando o reaproveitamento do conhecimento em Angular e TypeScript para a construção da interface móvel, sem a necessidade de adotar uma nova linguagem exclusivamente para o front-end. Já a adoção do MariaDB, conforme discutido na seção 1.3.3, justifica-se pela compatibilidade com o ecossistema SQL, desempenho e natureza de código aberto, características adequadas ao escopo deste projeto.

7.2 LIMITAÇÕES DO TRABALHO

Este trabalho apresenta limitações que devem ser reconhecidas. Em primeiro lugar, não foi realizada uma avaliação empírica com usuários reais, seja por meio de testes de usabilidade estruturados ou de questionários de satisfação. Relacionado a esse ponto, o trabalho também não apresenta métricas quantitativas de uso, como tempo médio de geração das atividades, taxa de acerto ao longo do tempo, retenção de vocabulário ou custo operacional por usuário. Essa coleta de dados dependeria de um período mais longo de uso da aplicação em produção, fora do escopo temporal deste TCC. Dessa forma, embora a proposta se sustente teoricamente na personalização do conteúdo, não há, neste momento, evidências quantitativas que comprovem que o aplicativo efetivamente melhora a aquisição do idioma espanhol em comparação a métodos tradicionais.

No que se refere aos testes de software, a estratégia adotada concentrou-se em testes unitários de regras de negócio críticas (como o TrialGuard), validação manual via Swagger e ferramentas de depuração do navegador. Para trabalhos futuros, seria interessante a realização de testes de carga e integração, pois ajudariam a afirmar que o sistema está pronto para operar em escala.

Quanto ao uso de inteligência artificial generativa, o principal risco identificado é a possibilidade de alucinações, isto é, respostas pedagogicamente incorretas. O sistema mitiga, mas não elimina esse risco. O prompt enviado à OpenAI já delimita explicitamente tema, nível CEFR e estrutura de saída esperada, e a biblioteca Zod garante que o JSON retornado esteja estruturalmente correto antes de ser processado. Entretanto, essa validação atua sobre o formato da resposta, não sobre a exatidão do seu conteúdo pedagógico. Associado a isso, há também uma dependência direta de um único provedor externo de inteligência artificial: interrupções no serviço, alterações de preço ou descontinuação do modelo utilizado impactariam diretamente a principal funcionalidade do aplicativo. Todavia, no escopo de desenvolvimento do projeto, a utilização de uma única API supriu as necessidades, visto que a diversificação de LLM's traria consigo um custo associado à assinatura de algum plano. Foi uma limitação calculada.

7.3 TRABALHOS FUTUROS

Como desdobramento deste trabalho, recomenda-se a realização de estudos futuros voltados à avaliação da experiência do usuário sob uma perspectiva pedagógica, e não apenas de usabilidade e navegação. Entre as questões a serem investigadas, destacam-se a manutenção do interesse do usuário após múltiplas sessões de uso, o real impacto da personalização das atividades sobre a motivação do aprendiz e em que medida o sistema favorece a autonomia do estudante. A investigação dessas questões aproximaria a avaliação de UX das dimensões pedagógicas da aplicação, fortalecendo o caráter interdisciplinar da pesquisa.

Sugere-se ainda aprofundar a investigação sobre como o Magia potencializa, na prática, as vantagens do MALL discutidas na seção 1.4 em relação à privacidade do aprendiz durante a prática de escuta e o suporte a públicos com rotinas de mobilidade intensa, como migrantes e profissionais que viajam com frequência.

Estudos futuros poderiam investigar se a flexibilidade do aplicativo realmente favorece a continuidade e a espontaneidade da aprendizagem propostas pelo conceito de Mobile-Assisted Language Use (MALU), ou se, na prática, o uso do Magia ainda se aproxima de sessões de estudo isoladas e formais. Nesse mesmo sentido, quanto às competências avaliadas pelo CEFR, o Magia contempla parcialmente a compreensão oral por meio do recurso de pronúncia disponível nos flashcards, mas a produção oral ainda não é trabalhada pelo sistema. Sugere-se, como trabalho futuro, a incorporação de reconhecimento de fala, ampliando a cobertura das quatro competências comunicativas do quadro europeu. Já em um âmbito técnico, para um cenário de crescimento da base de usuários, estratégias de evolução arquitetural também podem ser consideradas, como o processamento assíncrono das requisições de geração de atividades por meio de filas.

8. CONCLUSÃO

O desenvolvimento do aplicativo Magia se apresenta como um auxiliador tecnológico ao oferecer uma solução nova para o ensino do idioma espanhol no Brasil, ajudando a mitigar a lacuna deixada pelas recentes mudanças na legislação. O Magia, por meio da integração com a Inteligência Artificial Generativa, aborda a aprendizagem de maneira mais subjetiva, partindo dos objetivos do aprendiz, ao invés de utilizar um currículo padronizado.

Do ponto de vista técnico, a arquitetura do magia, baseada em tecnologias recentes de desenvolvimento de software — tais como Angular e NestJs — é facilmente escalável, manutenível e robusta. A implementação do sistema com atividades interativas e referenciadas nos níveis do CEFR criam um ambiente dinâmico e motivador que respeitam a Zona de Desenvolvimento Proximal do usuário.

Referente à confiabilidade, o Magia foi implementado com validações tanto no front-end, quanto no back-end. O Angular tem uma biblioteca fortíssima chamada de Reactive Forms, facilitando a validação de dados de *input* de formulários. Não obstante, também foram aplicadas validações na API, com proeminência no uso dos DTO's e do Zod. Isso permitiu, respectivamente, duas coisas: proteção dos dados vindos do front-end e resiliência em respostas inconsistentes da Inteligência Artificial. Outrossim, os testes realizado validaram integridade técnica e usabilidade.

Portanto, o projeto entrega uma ferramenta capaz de minimizar o isolamento linguístico do Brasil com os outros países da América Latina. Dessa forma, espera-se que o Magia ajude na integração cultural e no desenvolvimento do idioma espanhol no Brasil. E, como desdobramento natural deste trabalho, recomenda-se a realização de estudos futuros voltados à avaliação da experiência do usuário sob uma perspectiva pedagógica, o caráter estimulante da gamificação ao longo do tempo e a efetividade pedagógica do feedback gerado pela Inteligência Artificial.

9. AGRADECIMENTOS

Agradeço, em primeiro lugar, à minha versão criança, que, com sua imaginação fértil e coração puro, sempre acreditou que o mundo poderia ser maior do que parecia.

À minha mãe, Leila, ao meu pai, Washington, à minha tia Patricia e à minha avó Crispina, deixo minha profunda gratidão por todo amor, apoio e pelos valores de fé e esperança que sempre cultivaram em mim.

Agradeço também às minhas amigas e aos meus amigos, que tornaram a jornada mais leve e me ajudaram a me conhecer melhor ao longo do caminho.

À minha orientadora, Flávia, expresso minha gratidão por acreditar no meu potencial e por mostrar que o trabalho bem feito sempre traz recompensas.

Por fim, agradeço à minha versão atual, por não ter desistido dos próprios sonhos. Algo bom sempre está por vir.

10. REFERÊNCIAS

ANGULAR. *Angular Documentation*. Disponível em: <https://angular.dev/>. Acesso em: 13 maio 2026.

ASTAH. Ferramenta de modelagem UML. Disponível em: [Astah Official Website](#). Acesso em: 5 jun. 2026.

BRASIL. Lei nº 11.161, de 5 de agosto de 2005. Dispõe sobre o ensino da língua espanhola. Brasília, DF: Presidência da República, 2005. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2004-2006/2005/lei/l11161.htm#:~:text=LEI%20N%C2%BA%2011.161%2C%20DE%205%20DE%20AGOSTO%20DE%2005.&text=Disp%C3%B5e%20sobre%20o%20ensino%20da,Art.. Acesso em: 12 maio 2026.

BRASIL. Lei nº 13.415, de 16 de fevereiro de 2017. Altera as Leis nº 9.394, de 20 de dezembro de 1996, e nº 11.494, de 20 de junho de 2007, revoga a Lei nº 11.161, de 5 de agosto de 2005, e institui a Política de Fomento à Implementação de Escolas de Ensino Médio em Tempo Integral. Brasília, DF: Presidência da República, 2017. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2017/lei/l13415.htm#art22. Acesso em: 12 maio 2026.

BRASIL. Lei nº 14.945, de 31 de julho de 2024. Altera a Lei nº 9.394, de 20 de dezembro de 1996 (Lei de Diretrizes e Bases da Educação Nacional), para definir diretrizes para o ensino médio, e as Leis nº 14.818, de 16 de janeiro de 2024, nº 12.711, de 29 de agosto de 2012, nº 11.096, de 13 de janeiro de 2005, e nº 14.640, de 31 de julho de 2023. Brasília, DF: Presidência da República, 2024. Disponível em: <https://www2.camara.leg.br/legin/fed/lei/2024/lei-14945-31-julho-2024-796017-publicacaooriginal-172512-pl.html>. Acesso em: 12 maio 2026.

BUSUU. *Aprenda um idioma em apenas 10 minutos por dia*. [S. l.], 2024. Disponível em: <https://www.busuu.com/pt>. Acesso em: 22 maio 2024.

CAPACITOR. *Capacitor Documentation*. Disponível em: <https://capacitorjs.com/docs>. Acesso em: 13 maio 2026.

COUNCIL OF EUROPE. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment: Companion Volume*. Strasbourg: Council of Europe Publishing, 2020. Disponível em:

<https://www.coe.int/en/web/common-european-framework-reference-languages>.

Acesso em: 13 maio 2026.

DBDIAGRAM. *dbdiagram.io: Database Relationship Diagrams Design Tool*. Disponível em: dbdiagram.io. Acesso em: 5 jun. 2026.

DUOLINGO. Duolingo: a melhor maneira de aprender um idioma. [S. l.], 2024. Disponível em: <https://pt.duolingo.com/>. Acesso em: 13 abril 2026.

GOOGLE. Little Language Lessons: learn vocabulary through context and AI. [S. l.], 2024. Disponível em: <https://labs.google/lll/en>. Acesso em: 22 abril 2024.

IONIC. *Ionic Documentation*. Disponível em: <https://ionicframework.com/docs>. Acesso em: 13 maio 2026

KUKULSKA-HULME, Agnes. Mobile-Assisted Language Learning. In: CHAPELLE, Carol (ed.). **The Concise Encyclopedia of Applied Linguistics**. Disponível em: https://oro.open.ac.uk/57023/1/_userdata_documents5_ak35_Desktop_Accepted%20Manuscript_Concise%20Encyclopedia.pdf.

MariaDB. *MariaDB Enterprise Platform*. Disponível em: <https://MariaDB.com/docs/platform/>. Acesso em: 13 maio 2026.

MERCOSUL. Protocolo de Intenções. Brasília, 13 dez. 1991. In: UNIVERSIDADE CATÓLICA DO SALVADOR. Disponível em: <https://ri.ucsal.br/server/api/core/bitstreams/64aecab5-59e6-4bf4-98c8-168266bdb7a8/content>. Acesso em: 12 maio 2026.

NESTJS. *NestJs Documentation*. Disponível em: <https://docs.NestJs.com/>. Acesso em: 13 maio 2026.