v7.n12.2026

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

DESENVOLVIMENTO DO APLICATIVO BULA FÁCIL PARA ACESSO E COMPREENSÃO DE BULAS DE MEDICAMENTOS COM RECURSOS DE INTELIGÊNCIA ARTIFICIAL E ACESSIBILIDADE

DEVELOPMENT OF THE BULA FÁCIL APPLICATION FOR ACCESS TO AND UNDERSTANDING OF MEDICINE LEAFLETS USING ARTIFICIAL INTELLIGENCE AND ACCESSIBILITY RESOURCES

DESARROLLO DE LA APLICACIÓN BULA FÁCIL PARA EL ACCESO Y LA COMPRENSIÓN DE PROSPECTOS DE MEDICAMENTOS CON RECURSOS DE INTELIGENCIA ARTIFICIAL Y ACCESIBILIDAD

Bruno Scutare Branco[1], André Luiz da Silva[2], Fabiana Florian[3]

https://doi.org/10.47820/recima21.v7i12.10087

PUBLICADO: 12/2026

RESUMO

Este trabalho apresenta o desenvolvimento do aplicativo móvel Bula Fácil, criado para facilitar o acesso e a consulta de bulas de medicamentos, com atenção especial às necessidades de pessoas idosas. O sistema utiliza arquitetura cliente-servidor, com React Native e Expo no frontend e Node.js e Express no backend. As bulas oficiais foram obtidas no Bulário Eletrônico da ANVISA, armazenadas no Cloudflare R2 e processadas por script Python para extração das nove seções destinadas ao paciente, posteriormente inseridas no MongoDB Atlas. A aplicação integra Google Gemini para geração de resumos estruturados, síntese de voz, reconhecimento óptico de caracteres, busca por voz, favoritos, lembretes, notificações e acesso ao PDF oficial. O protótipo foi avaliado por testes funcionais manuais, que indicaram integração técnica dos principais componentes. Não foram realizados testes formais de usabilidade ou de compreensão com usuários. As funções de OCR e reconhecimento de voz dependem de validação adicional em compilação nativa, assim como os resumos gerados por inteligência artificial necessitam de avaliação quanto à fidelidade às bulas. O aplicativo mantém o documento oficial como fonte primária de consulta e não substitui orientação profissional.

PALAVRAS-CHAVE: Acessibilidade. Aplicativo móvel. Bulas de medicamentos. Inteligência artificial. OCR. Saúde digital.

ABSTRACT

This study presents the development of Bula Fácil, a mobile application intended to facilitate access to medicine package leaflets, with special attention to older adults. The system uses a client-server architecture with React Native and Expo on the frontend and Node.js and Express on the backend. Official leaflets were obtained from ANVISA, stored in Cloudflare R2, and processed using Python to extract nine patient-oriented sections subsequently stored in MongoDB Atlas. The application integrates Google Gemini for structured summaries, text-to-speech, optical character recognition, voice search, favorites, reminders, notifications, and access to official PDF files. Manual functional tests indicated technical integration of the main components. Formal user testing of usability and comprehension was not conducted. Optical character recognition and voice recognition require further validation in a native development

[1] Universidade de Araraquara (UNIARA), graduando em Sistemas de Informação.
[2] Universidade de Araraquara (UNIARA), doutor.
[3] Universidade de Araraquara (UNIARA), doutora.

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

build, while AI-generated summaries require assessment of their fidelity to official information. The original leaflet remains the primary source of information, and the application does not replace professional medical advice.

KEYWORDS: *Accessibility. Artificial intelligence. Digital health. Medicine leaflets. Mobile application. OCR.*

RESUMEN

Este trabajo presenta el desarrollo de Bula Fácil, una aplicación móvil destinada a facilitar el acceso a los prospectos de medicamentos, con especial atención a las personas mayores. El sistema utiliza una arquitectura cliente-servidor, con React Native y Expo en el frontend y Node.js y Express en el backend. Los prospectos oficiales se obtuvieron del repositorio de ANVISA, se almacenaron en Cloudflare R2 y se procesaron mediante Python para extraer nueve secciones dirigidas al paciente, posteriormente almacenadas en MongoDB Atlas. La aplicación integra Google Gemini para generar resúmenes estructurados, síntesis de voz, reconocimiento óptico de caracteres, búsqueda por voz, favoritos, recordatorios, notificaciones y acceso al PDF oficial. Las pruebas funcionales manuales indicaron la integración técnica de los componentes principales. No se realizaron evaluaciones formales de usabilidad ni de comprensión con usuarios. Las funciones de reconocimiento de imágenes y voz requieren validación adicional mediante una compilación nativa, y los resúmenes generados por inteligencia artificial necesitan evaluación de fidelidad. El prospecto oficial sigue siendo la fuente principal y la aplicación no sustituye la orientación de profesionales sanitarios.

PALABRAS CLAVE: *Accesibilidad. Aplicación móvil. Prospectos de medicamentos. Inteligencia artificial. OCR. Salud digital.*

1. INTRODUÇÃO

O avanço das tecnologias móveis ampliou as possibilidades de disponibilização e acesso a informações relacionadas à saúde por meio de smartphones e aplicações móveis. Esses recursos podem auxiliar na consulta e organização de informações, oferecendo diferentes formas de interação entre o usuário e o conteúdo apresentado. Entretanto, a utilização dessas tecnologias também depende de aspectos relacionados à facilidade de uso, organização da interface e adequação às características dos usuários. No caso das pessoas idosas, fatores relacionados às capacidades perceptivas, motoras, cognitivas e de memória podem influenciar a utilização de aplicações móveis de saúde, tornando importante considerar essas características durante o desenvolvimento das interfaces (LIU *et al.*, 2021). Estudos específicos sobre o desenvolvimento de aplicações móveis para pessoas idosas também destacam aspectos relacionados à navegação, apresentação visual, carga cognitiva e formas de interação como elementos relevantes para a usabilidade desses sistemas (GOMEZ-HERNANDEZ *et al.*, 2023). Entre as fontes de informação relacionadas ao uso de medicamentos encontra-se a bula. Segundo a Agência Nacional de Vigilância Sanitária (ANVISA), bulas e rótulos devem transmitir

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

informações relevantes sobre os medicamentos e contribuir para seu uso adequado, apresentando informações obrigatórias estabelecidas pelas normas sanitárias vigentes (ANVISA, s.d.). A ANVISA também disponibiliza o Bulário Eletrônico, desenvolvido para facilitar o acesso da população e dos profissionais de saúde às bulas de medicamentos (ANVISA, s.d.). Dessa forma, as informações oficiais podem ser consultadas digitalmente, permitindo o acesso a conteúdos relacionados às indicações, modo de uso, contraindicações, advertências, cuidados e possíveis reações dos medicamentos (ANVISA, s.d.). Apesar da disponibilidade digital desses documentos, a extensão das bulas e a necessidade de localizar informações específicas podem tornar sua consulta pouco prática em determinadas situações. Em dispositivos móveis, a apresentação de grandes quantidades de conteúdo também exige atenção à organização e às formas de interação disponibilizadas ao usuário. Para pessoas idosas, características de interface e mecanismos de interação adequados podem contribuir para a utilização de aplicações móveis (GOMEZ-HERNANDEZ et al., 2023; LIU et al., 2021). Nesse contexto, surgiu o Bula Fácil, uma aplicação móvel destinada ao público em geral, mas desenvolvida com atenção especial às necessidades de pessoas idosas. O sistema concentra em uma única interface a busca de medicamentos, a consulta às nove seções da bula destinada ao paciente, o acesso ao documento oficial em formato PDF e recursos de apoio, como síntese de voz, reconhecimento de texto em imagens, pesquisa por voz, lembretes, favoritos e geração de resumos com inteligência artificial.

O objetivo geral deste trabalho é desenvolver uma aplicação móvel capaz de facilitar o acesso e a compreensão das informações presentes em bulas de medicamentos, utilizando recursos de inteligência artificial, acessibilidade e funcionalidades nativas de dispositivos móveis. Como objetivos específicos, pretende-se disponibilizar pesquisas por nome, voz e imagem; organizar as informações das bulas em seções; gerar resumos estruturados a partir do conteúdo oficial; permitir a reprodução das informações em áudio; disponibilizar o acesso à bula integral; implementar recursos de favoritos e lembretes; e oferecer opções de interface, como tema claro ou escuro e ampliação do tamanho da fonte.

A justificativa para o desenvolvimento do projeto está relacionada à necessidade de disponibilizar informações farmacêuticas extensas de maneira organizada e de fácil localização em dispositivos móveis. O foco especial no público idoso está associado à adoção de diferentes formas de interação, evitando que a consulta dependa exclusivamente da leitura de textos extensos. Estudos sobre aplicações móveis destinadas a pessoas idosas indicam a importância de considerar características perceptivas, motoras e cognitivas durante o desenvolvimento das

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

interfaces (LIU *et al.*, 2021). Diretrizes de design identificadas a partir de estudos de usabilidade com pessoas acima de 60 anos também ressaltam aspectos como simplificação da navegação, apresentação visual adequada e características dos elementos interativos (GOMEZ-HERNANDEZ *et al.*, 2023). Esses princípios são considerados no Bula Fácil por meio de recursos como ampliação de fonte, temas claro e escuro, síntese de voz, pesquisa por voz e organização das informações em seções. O aplicativo, entretanto, não se propõe a substituir profissionais de saúde, realizar diagnósticos ou prescrever medicamentos. As sugestões apresentadas a partir de sintomas possuem caráter exclusivamente informativo, e a bula oficial permanece disponível para consulta no aplicativo.

Diante desse contexto, estabelece-se o seguinte problema de pesquisa: como uma aplicação móvel, apoiada por recursos de inteligência artificial e acessibilidade, pode facilitar o acesso e a compreensão das informações presentes em bulas de medicamentos? Parte-se da hipótese de que a organização das informações em seções, a simplificação textual assistida por inteligência artificial, a reprodução do conteúdo por síntese de voz e a disponibilização de diferentes formas de pesquisa podem reduzir o esforço necessário para localizar e consultar as informações presentes nas bulas, tornando o processo mais acessível ao usuário.

Quanto à metodologia, o trabalho caracteriza-se como um desenvolvimento tecnológico de natureza aplicada, com caráter exploratório e descritivo. O protótipo foi desenvolvido utilizando arquitetura cliente-servidor. O frontend foi implementado em JavaScript utilizando React Native e Expo, enquanto o backend foi desenvolvido com Node.js e Express. Os dados referentes aos usuários e medicamentos são armazenados no MongoDB Atlas, os arquivos PDF das bulas são mantidos no serviço de armazenamento Cloudflare R2 e a geração dos resumos estruturados utiliza modelos de inteligência artificial Google Gemini. Para a composição do protótipo, foram selecionados 60 medicamentos de forma não probabilística, buscando proporcionar diversidade de medicamentos, finalidades e áreas de aplicação, sem a pretensão de representar os medicamentos mais utilizados no Brasil. A avaliação apresentada neste trabalho baseia-se em testes funcionais manuais realizados durante o processo de desenvolvimento.

2. REFERENCIAL TEÓRICO

2.1. Bulas de medicamentos e acesso à informação

A bula constitui uma fonte formal de informações relacionadas aos medicamentos. Segundo a Agência Nacional de Vigilância Sanitária (ANVISA), bulas e rótulos devem transmitir

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

informações relevantes sobre os medicamentos e contribuir para seu uso adequado, contendo informações obrigatórias estabelecidas pelas normas sanitárias vigentes (ANVISA, s.d.). A ANVISA também estabelece distinções entre as informações destinadas aos pacientes e aquelas direcionadas aos profissionais de saúde, considerando as características e necessidades de cada público (ANVISA, s.d.).

Como forma de ampliar o acesso a essas informações, a ANVISA disponibiliza o Bulário Eletrônico, desenvolvido para facilitar o acesso da população e dos profissionais de saúde às bulas de medicamentos (ANVISA, s.d.). Por meio desse serviço, é possível consultar os documentos disponibilizados pela Agência e acessar informações oficiais relacionadas aos medicamentos (ANVISA, s.d.). No desenvolvimento do Bula Fácil, o Bulário Eletrônico foi adotado como fonte das bulas utilizadas no protótipo, mantendo no aplicativo a possibilidade de acesso ao documento oficial em formato PDF.

2.2. Acessibilidade digital e público idoso

A acessibilidade digital está relacionada à possibilidade de diferentes usuários perceberem, compreenderem, navegarem e interagirem com conteúdos e sistemas digitais. As diretrizes de acessibilidade para conteúdo digital abordam aspectos relacionados à percepção, operação, compreensão e robustez das interfaces, buscando reduzir barreiras de acesso para pessoas com diferentes necessidades (W3C, 2018).

No desenvolvimento de aplicações móveis destinadas também ao público idoso, devem ser consideradas características que podem influenciar a interação com dispositivos digitais. Liu *et al.* destacam fatores relacionados às capacidades perceptivas, motoras, cognitivas e de memória na utilização de aplicações móveis de saúde por pessoas idosas (LIU *et al.*, 2021). Essas características podem influenciar elementos como apresentação das informações, navegação e formas de interação oferecidas pelo sistema.

Gomez-Hernandez *et al.* analisaram estudos envolvendo testes de usabilidade de aplicações móveis com pessoas acima de 60 anos e identificaram diretrizes relacionadas ao design visual, navegação, carga cognitiva, ajuda e formas de interação (GOMEZ-HERNANDEZ *et al.*, 2023). Entre os aspectos considerados estão a simplificação das interfaces e características dos elementos interativos, evidenciando a importância de adaptar aplicações móveis às particularidades desse público (GOMEZ-HERNANDEZ *et al.*, 2023).

Esses princípios foram considerados no Bula Fácil por meio da adoção de recursos como ampliação do tamanho da fonte, temas claro e escuro, organização visual das informações,

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

síntese de voz e diferentes formas de pesquisa. Esses mecanismos buscam oferecer alternativas de interação com o conteúdo, especialmente para usuários que possam apresentar dificuldades na utilização convencional de aplicações móveis.

2.3. Aplicações móveis e arquitetura cliente-servidor

A organização da arquitetura de um sistema de software permite distribuir responsabilidades entre diferentes componentes. A separação de responsabilidades contribui para aspectos relacionados à organização, manutenção e evolução dos sistemas, sendo um princípio relevante durante o desenvolvimento de software (PRESSMAN, 2016; SOMMERVILLE, 2019).

Em uma arquitetura cliente-servidor, diferentes responsabilidades podem ser distribuídas entre a aplicação utilizada pelo usuário e os serviços responsáveis pelo processamento e fornecimento dos dados. Essa separação permite que a interface cliente realize requisições a serviços responsáveis por operações como autenticação, persistência de informações e integração com outros recursos.

No Bula Fácil, essa estrutura é representada pela separação entre a aplicação móvel e o backend. O frontend concentra a navegação, a apresentação dos dados e a interação com recursos do dispositivo, enquanto uma API REST centraliza operações relacionadas à autenticação, consulta de medicamentos, acesso ao banco de dados e comunicação com serviços externos. Para a implementação dessa API foi utilizado o Express, framework para Node.js que oferece recursos de roteamento e utilização de middlewares em aplicações web (EXPRESS.JS, s.d.).

2.4. Banco de dados orientado a documentos

O MongoDB utiliza um modelo orientado a documentos, no qual os registros são armazenados na forma de documentos e organizados em coleções (MONGODB, s.d.). Essa abordagem possibilita representar informações utilizando estruturas que podem conter diferentes campos, objetos e conjuntos de dados relacionados dentro de um mesmo documento.

Essa característica é adequada à estrutura adotada pelo Bula Fácil. Cada medicamento pode ser representado por um documento contendo informações de identificação, uma referência ao arquivo PDF da bula e um objeto composto pelas nove seções textuais utilizadas pelo aplicativo. De maneira semelhante, os documentos referentes aos usuários podem armazenar informações relacionadas aos favoritos e lembretes.

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

Para a infraestrutura de banco de dados do protótipo foi utilizado o MongoDB Atlas, serviço gerenciado da plataforma MongoDB que permite a implantação e administração de bancos de dados em ambiente de nuvem (MONGODB, s.d.). Dessa forma, os dados utilizados pelo backend podem ser mantidos separadamente da aplicação instalada no dispositivo móvel.

2.5. Inteligência artificial generativa

A Inteligência Artificial (IA) compreende diferentes métodos e técnicas destinados ao desenvolvimento de sistemas capazes de executar tarefas relacionadas a capacidades associadas à inteligência, incluindo processamento de linguagem, aprendizagem e tomada de decisões (RUSSELL; NORVIG, 2021). Dentro desse campo, modelos capazes de processar linguagem natural podem receber informações textuais como entrada e produzir conteúdo de acordo com o contexto e as instruções fornecidas.

O aprendizado profundo contribuiu para avanços em diferentes áreas da inteligência artificial e utiliza modelos compostos por múltiplas camadas de representação para aprender características a partir dos dados (GOODFELLOW; BENGIO; COURVILLE, 2016). Esses avanços constituem parte da evolução tecnológica que possibilitou o desenvolvimento de sistemas modernos capazes de processar e gerar conteúdo textual.

No Bula Fácil, a inteligência artificial generativa é utilizada como recurso auxiliar para produzir resumos estruturados a partir das informações presentes nas bulas. A IA não é utilizada como fonte primária de informação médica. O conteúdo das nove seções da bula é fornecido ao modelo juntamente com instruções que determinam que a resposta seja baseada somente nas informações recebidas e apresentada de maneira curta e estruturada.

O resultado é organizado em blocos relacionados à visão geral do medicamento, funcionamento, frequência de administração, contraindicações, cuidados e possíveis efeitos. A utilização da inteligência artificial possui caráter complementar, uma vez que o usuário continua tendo acesso ao conteúdo das seções e ao documento oficial da bula. Dessa forma, o resumo gerado não substitui a documentação oficial nem a orientação de profissionais de saúde.

2.6. Reconhecimento óptico de caracteres (OCR)

O Reconhecimento Óptico de Caracteres, conhecido pela sigla OCR (Optical Character Recognition), corresponde ao processo de identificação de caracteres textuais presentes em imagens e sua conversão para uma representação digital que possa ser processada por sistemas computacionais.

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

O ML Kit disponibiliza uma API de reconhecimento de texto capaz de identificar e extrair conteúdo textual presente em imagens em aplicações móveis (GOOGLE, s.d.). Segundo a documentação da ferramenta, características como resolução da imagem e foco podem influenciar a precisão do reconhecimento, sendo necessário que os caracteres possuam quantidade suficiente de pixels para que possam ser adequadamente processados (GOOGLE, s.d.).

No Bula Fácil, o OCR é empregado como etapa inicial do recurso de identificação de medicamentos por imagem. Uma fotografia da embalagem pode ser processada para obtenção do texto nela presente. Posteriormente, o texto identificado é normalizado e comparado com os nomes dos medicamentos disponíveis na aplicação, permitindo apresentar os resultados considerados mais prováveis. Assim, o fluxo desse recurso pode ser representado como imagem, reconhecimento do texto, normalização, comparação e apresentação dos medicamentos correspondentes.

2.7. Reconhecimento de voz e text-to-speech (TTS)

O reconhecimento automático de fala e a síntese de voz representam formas distintas de processamento da linguagem falada. Sistemas de reconhecimento de fala processam sinais de áudio com o objetivo de obter uma representação linguística do conteúdo pronunciado, enquanto sistemas de síntese de voz realizam o processo de geração de fala a partir de conteúdo textual (JURAFSKY; MARTIN, 2025).

O Text-to-Speech (TTS), portanto, permite que informações originalmente apresentadas em formato textual sejam reproduzidas por meio de fala sintetizada. No contexto de aplicações móveis, esse mecanismo pode oferecer uma forma alternativa de acesso ao conteúdo, reduzindo a dependência exclusiva da leitura visual. A biblioteca Expo Speech fornece uma interface para utilização de funcionalidades de TTS em aplicações móveis e permite configurar características da reprodução da fala (EXPO, s.d.).

No Bula Fácil, o reconhecimento de voz e o TTS são utilizados com finalidades distintas. O reconhecimento de voz transforma a fala do usuário em texto, que pode ser empregado como termo para pesquisa de medicamentos. O TTS, por sua vez, permite a reprodução em áudio tanto dos conteúdos apresentados nos cards dos resumos gerados por inteligência artificial quanto do texto integral de cada seção da bula. Esses recursos ampliam as formas de interação disponibilizadas pelo aplicativo.

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

2.8. Armazenamento de objetos e acesso temporário

Serviços de armazenamento de objetos possibilitam manter arquivos separadamente das informações estruturadas armazenadas em bancos de dados. No Bula Fácil, essa estratégia é utilizada para os documentos oficiais das bulas em formato PDF, enquanto as informações textuais e os demais registros necessários ao funcionamento do sistema são mantidos no banco de dados.

O Cloudflare R2 é um serviço de armazenamento de objetos que oferece compatibilidade com a API S3, possibilitando realizar operações de armazenamento e recuperação de objetos por meio dessa interface (CLOUDFLARE, s.d.). No projeto, os arquivos PDF das bulas são armazenados em um bucket privado, evitando que os documentos permaneçam diretamente disponíveis por meio de endereços públicos permanentes.

Para permitir o acesso aos arquivos quando solicitado pelo aplicativo, são utilizadas URLs pré-assinadas. Esse mecanismo permite conceder acesso temporário a um determinado objeto e operação sem tornar o bucket publicamente acessível (CLOUDFLARE, s.d.). No Bula Fácil, o backend gera uma URL pré-assinada para o PDF correspondente ao medicamento solicitado, permitindo sua visualização pelo aplicativo durante o período de validade definido.

3. METODOLOGIA E DESENVOLVIMENTO

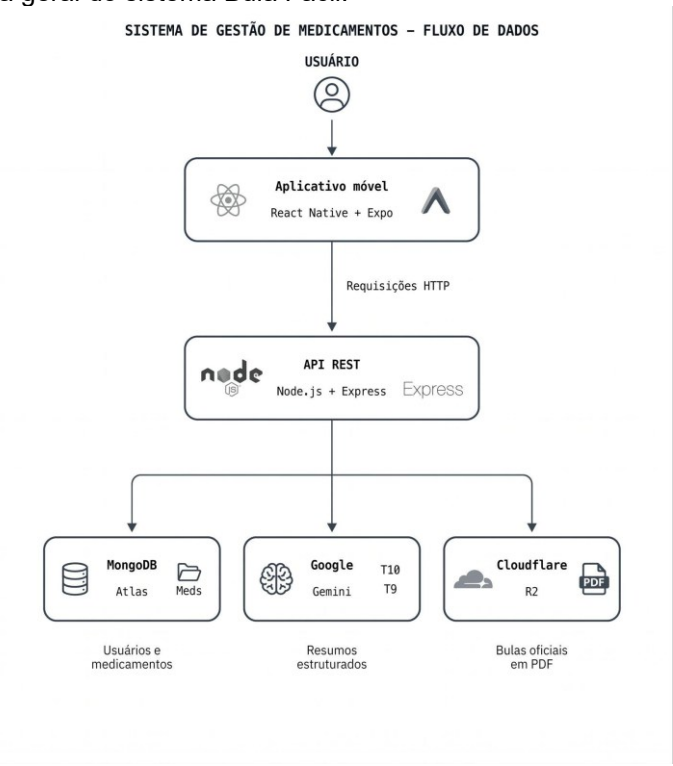
3.1. Visão geral e arquitetura do sistema

O Bula Fácil foi desenvolvido utilizando uma arquitetura cliente-servidor, organizada em duas camadas principais. A camada cliente corresponde à aplicação móvel, implementada em JavaScript com React Native e Expo, sendo responsável pela interface, navegação e interação do usuário com as funcionalidades do sistema. A camada servidor corresponde a uma API REST desenvolvida em JavaScript com Node.js e Express, responsável pelo processamento das requisições, autenticação dos usuários, acesso aos dados e integração com serviços externos.

A API realiza a comunicação com três serviços principais: MongoDB Atlas, utilizado para armazenamento dos dados de usuários e medicamentos; Google Gemini, empregado na geração dos resumos estruturados das bulas; e Cloudflare R2, utilizado para armazenamento dos documentos oficiais em formato PDF. Durante o desenvolvimento apresentado neste trabalho, a API foi executada localmente na porta 3001, enquanto o banco de dados e os demais serviços permaneceram hospedados externamente. A Figura 1 apresenta a arquitetura geral do sistema e o fluxo de comunicação entre seus principais componentes.

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

Figura 1. Arquitetura geral do sistema Bula Fácil.



Fonte: o autor, 2026.

3.2. Frontend mobile

O frontend do Bula Fácil foi desenvolvido em JavaScript utilizando React Native com Expo SDK 57, React 19 e React Native 0.86. A navegação entre as telas foi estruturada com o Expo Router, utilizando o sistema de roteamento baseado em arquivos. O código foi organizado de forma modular, separando telas, componentes reutilizáveis, serviços, contextos, hooks, temas, utilitários e constantes.

A Context API foi utilizada para o gerenciamento de estados compartilhados entre diferentes partes da aplicação, principalmente aqueles relacionados à autenticação do usuário e ao tema da interface. Os React Hooks, por sua vez, foram empregados para controlar estados locais, efeitos e ciclos de vida dos componentes.

A interface foi construída a partir de componentes reutilizáveis, incluindo cards, cabeçalhos, listas, Bottom Sheets, mensagens do tipo toast e uma barra inferior de navegação.

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

Essa organização permite reutilizar elementos visuais em diferentes telas e manter maior consistência na interface da aplicação.

Para o armazenamento local, o aplicativo utiliza AsyncStorage para dados como histórico de pesquisas, preferências de interface e cache dos resumos gerados por inteligência artificial. O SecureStore é utilizado para informações relacionadas à autenticação que demandam armazenamento local protegido.

A tela inicial reúne os principais meios de acesso às funcionalidades do sistema, incluindo pesquisa de medicamentos por texto, pesquisa por voz, identificação por imagem, histórico de pesquisas recentes e acesso às sugestões organizadas por sintomas. A Figura 2 apresenta a tela inicial, que concentra pesquisa, acesso por voz ou câmera, histórico recente e atalhos de sintomas.

Figura 2. Tela inicial do aplicativo Bula Fácil.



Fonte: o autor, 2026.

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

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

3.3. Backend e API REST

O backend do Bula Fácil foi desenvolvido em JavaScript utilizando Node.js e Express, com adoção do padrão ES Modules. Sua estrutura foi organizada em rotas, controllers, models, services e middlewares, distribuindo as responsabilidades relacionadas ao recebimento das requisições, processamento das regras da aplicação, acesso aos dados e integração com serviços externos.

A comunicação entre o aplicativo móvel e o backend ocorre por meio de uma API REST. Os endpoints disponibilizados atendem principalmente às operações relacionadas ao gerenciamento de usuários, consulta de medicamentos e geração dos resumos estruturados das bulas. O frontend realiza as requisições HTTP por meio da API fetch, utilizando a variável de ambiente EXPO_PUBLIC_API_URL para definir o endereço do servidor. Durante o desenvolvimento, essa variável foi configurada com o endereço IP da máquina responsável pela execução local da API, permitindo a comunicação entre o dispositivo móvel e o servidor por meio da rede local.

Para o controle de acesso às funcionalidades protegidas, foi implementado um mecanismo de autenticação baseado em JSON Web Token (JWT). Após a autenticação, o token é utilizado nas requisições protegidas por meio do esquema Bearer. No backend, um middleware verifica a validade do token e identifica o usuário associado à requisição. As senhas dos usuários não são armazenadas diretamente, sendo protegidas por meio de hash utilizando a biblioteca bcrypt.

Também foram utilizados mecanismos adicionais de segurança e configuração. O Helmet é empregado na definição de cabeçalhos HTTP relacionados à segurança, enquanto o CORS controla as origens permitidas nas requisições. O express-rate-limit limita a quantidade de requisições recebidas em determinado intervalo, contribuindo para a proteção da API contra uso excessivo.

3.4. Banco de dados MongoDB Atlas

Os dados utilizados pelo Bula Fácil são armazenados no MongoDB Atlas. O banco de dados “bulafacil” possui duas coleções principais: medicamentos e usuarios. Essa organização permite separar as informações referentes às bulas e aos medicamentos dos dados associados às contas dos usuários.

A coleção “medicamentos” contém 60 documentos selecionados para a composição do protótipo, buscando incluir medicamentos com diferentes finalidades e áreas de aplicação. Essa

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

seleção não possui caráter estatístico e não pretende representar os medicamentos mais utilizados no Brasil. Cada documento armazena informações como nome, identificador, referência para o arquivo PDF e um objeto contendo as nove seções padronizadas da bula destinada ao paciente.

A coleção “usuarios”, por sua vez, armazena os dados relacionados às contas cadastradas no sistema e estruturas associadas às funcionalidades individuais de cada usuário. Entre essas informações estão os medicamentos adicionados aos favoritos e os lembretes cadastrados. Essa organização permite que favoritos e lembretes permaneçam vinculados à respectiva conta e possam ser recuperados após a autenticação do usuário.

O campo pdf_path mantém a referência ao arquivo PDF correspondente armazenado no Cloudflare R2. Dessa forma, o banco de dados não armazena diretamente o arquivo da bula, mas apenas a informação necessária para que o backend localize o documento no serviço de armazenamento de objetos.

3.5. Obtenção, armazenamento e processamento das bulas

As bulas utilizadas no protótipo foram obtidas manualmente no Bulário Eletrônico da ANVISA. A plataforma oficial permite baixar a Bula do Paciente em PDF (ANVISA, s.d.). Após o download, os arquivos foram enviados ao Cloudflare R2, em bucket privado.

Em seguida, cada PDF foi processado por um script em Python. O script utiliza a biblioteca pdfplumber para extrair o texto das páginas e uma expressão regular para localizar as seções numeradas da bula do paciente. As nove primeiras seções encontradas são organizadas em um objeto e gravadas em JSON. Os arquivos JSON resultantes foram inseridos manualmente na coleção “medicamentos” do MongoDB Atlas.

3.6. Acesso protegido ao PDF oficial

O campo pdf_path armazenado no MongoDB funciona como referência ao objeto no R2. Antes da consulta, o backend normaliza o caminho, removendo o prefixo bulas/ quando necessário. Em seguida, utiliza GetObjectCommand do SDK compatível com S3 e gera uma URL assinada temporária. No protótipo, a validade definida é de uma hora. Essa abordagem permite manter o bucket sem acesso público permanente e disponibilizar o documento ao aplicativo apenas por um período limitado, comportamento compatível com o uso de URLs pré-assinadas descrito pela documentação do R2 (CLOUDFLARE, s.d.).

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

3.7. Resumo da bula com Google Gemini

O resumo por inteligência artificial é gerado sob demanda e utiliza cache local. Quando o usuário abre o medicamento, o aplicativo verifica primeiro o AsyncStorage, na estrutura bulafacil_resumos_ia. Se existir um resumo previamente gerado para o medicamento, o conteúdo é reutilizado. Caso contrário, o aplicativo obtém a bula e solicita ao backend a geração do resumo. O resultado retornado é salvo localmente e não é armazenado no MongoDB.

O backend envia ao serviço de IA o objeto completo med.secoes, contendo as seções 1 a 9 quando disponíveis, serializado em JSON. O prompt orienta o modelo a responder em português do Brasil, utilizar exclusivamente as informações recebidas, não inventar dados, produzir textos curtos e objetivos, informar frequência de administração, destacar informações importantes e retornar somente JSON válido. Os campos gerados contemplam visão geral, funcionamento, frequência, contraindicações, cuidados e efeitos possíveis.

O modelo principal configurado é gemini-2.5-flash, com novas tentativas em caso de indisponibilidade e gemini-2.0-flash como alternativa. O SDK utilizado na implementação é @google/genai 1.45.0. A interface identifica explicitamente o conteúdo como resumo gerado por inteligência artificial a partir da bula oficial e informa que dúvidas devem ser encaminhadas a um profissional de saúde.

3.8. Organização das nove seções da bula

Todos os 60 medicamentos cadastrados no protótipo possuem nove seções correspondentes às perguntas padronizadas da bula do paciente. O aplicativo apenas mostra essas seções caso o usuário decida ter uma compreensão mais aprofundada do medicamento, pois inicialmente apenas o resumo por IA da bula é mostrado. A lista de seções permite abrir cada conteúdo separadamente. Essa organização reduz a necessidade de percorrer o PDF integral quando o usuário procura uma informação específica. O botão para visualizar a bula completa permanece disponível ao final da listagem.

3.9. Reconhecimento óptico de caracteres e identificação por imagem

A identificação por imagem utiliza @react-native-ml-kit/text-recognition. O fluxo da interface é direcionado à fotografia da embalagem do medicamento, embora tecnicamente o serviço possa processar outras imagens contendo texto. O usuário pode tirar uma foto ou selecionar uma imagem da galeria. Após o ML Kit extrair o texto, o aplicativo normaliza o conteúdo e o compara com a lista de medicamentos carregada da API.

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

O algoritmo atribui 10 pontos quando o nome completo do medicamento aparece no texto reconhecido. Também compara palavras individuais do nome com pelo menos quatro caracteres e acrescenta 2 pontos por correspondência. Os resultados são ordenados pela pontuação e o aplicativo retorna até dez medicamentos mais prováveis. Essa etapa complementa o OCR: o ML Kit realiza a extração textual, enquanto a lógica de pontuação do Bula Fácil determina os candidatos.

3.10. Busca por voz e síntese de voz

A pesquisa por voz utiliza @react-native-voice/voice. A fala é reconhecida em português do Brasil e convertida em texto, que passa a ser utilizado como termo de busca sobre os medicamentos carregados no aplicativo. O recurso oferece uma alternativa à digitação, especialmente útil em situações nas quais o usuário apresenta dificuldade para utilizar o teclado virtual.

A síntese de voz utiliza expo-speech com idioma pt-BR, velocidade 0,9, tom 1,0 e volume 1,0. O serviço possui operações para iniciar, pausar e interromper a leitura; nas telas atuais, o controle visível é utilizado principalmente para iniciar ou parar. O TTS está disponível nos cards do resumo e no conteúdo completo de cada seção, permitindo que a informação textual seja consumida também em formato de áudio.

3.11. Pesquisa, lista de medicamentos e sintomas

A pesquisa textual pode ser realizada na tela inicial e na lista completa de medicamentos. O protótipo mantém 60 registros e permite filtrar os itens pelo nome. A tela de sintomas apresenta associações fixas no frontend entre sintomas comuns e medicamentos usualmente relacionados a esses contextos. Essas associações foram incluídas apenas como recurso informativo de demonstração e não constituem prescrição, diagnóstico ou recomendação médica. A própria interface orienta o usuário a consultar um profissional de saúde.

3.12. Histórico, favoritos e lembretes

O histórico de últimas pesquisas é armazenado localmente em AsyncStorage. A chave utilizada é associada ao identificador do usuário, evitando o compartilhamento do histórico entre contas diferentes no mesmo dispositivo. Atualmente são mantidos até quatro registros recentes.

Os favoritos são persistidos no documento do usuário, no campo “favoritos” da coleção usuarios. Os lembretes são armazenados no campo “notificacoes”. O backend participa da

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

criação, edição, consulta e exclusão desses registros. O disparo das notificações, entretanto, ocorre localmente no dispositivo por meio de expo-notifications. O aplicativo mantém localmente apenas o marcador dos lembretes já consumidos no dia.

3.13. Configurações e acessibilidade

As preferências de tema claro ou escuro e fonte ampliada são persistidas em AsyncStorage e controladas pela aplicação por meio de contexto de tema. Além dessas preferências, a proposta de acessibilidade é reforçada pelo TTS, pesquisa por voz, organização das seções e interface com componentes de toque destacados. O perfil do usuário e as opções de segurança também são reunidos na tela de configurações.

3.14. Compatibilidade e limitações do ambiente de desenvolvimento

A base do projeto possui suporte estrutural a Android e iOS por utilizar React Native e Expo, e o aplicativo foi testado manualmente também em Android durante o desenvolvimento. A execução apresentada nas capturas utiliza Expo Go. Entretanto, o OCR e o reconhecimento de voz dependem de módulos nativos que exigem Development Build ou compilação própria. Por essa razão, embora a lógica dessas funcionalidades esteja implementada, elas não foram validadas no ambiente Expo Go utilizado na etapa atual. Essa limitação é considerada na análise dos resultados.

4. RESULTADOS E DISCUSSÃO

Os testes funcionais foram realizados manualmente durante o desenvolvimento, com o objetivo de verificar a comunicação entre frontend e backend, persistência de dados, navegação e comportamento das principais funcionalidades. Não foi realizado estudo de usabilidade com participantes, portanto os resultados apresentados se restringem à implementação e ao funcionamento técnico observado no protótipo.

A organização das bulas em nove seções permitiu apresentar conteúdos específicos sem exigir que o usuário percorresse todo o PDF. Paralelamente, a manutenção do documento oficial no aplicativo preserva a possibilidade de conferir o conteúdo original. A integração com o Gemini acrescentou uma camada de resumo estruturado, e o uso de cache local evita nova geração sempre que o mesmo medicamento é aberto no dispositivo.

Os dados persistentes foram distribuídos conforme a natureza de cada informação. Medicamentos, favoritos e lembretes permanecem no MongoDB Atlas, enquanto histórico,

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

preferências e cache dos resumos ficam no dispositivo. Os PDFs são armazenados separadamente no R2 e acessados por URLs temporárias. Essa divisão demonstrou ser adequada ao protótipo por separar dados estruturados, preferências locais e arquivos de maior porte.

Quadro 1. Síntese dos testes funcionais

Funcionalidade	Resultado no protótipo
Cadastro, login e autenticação	Implementado e testado manualmente
Busca textual, seções e PDF oficial	Implementado e testado manualmente
Resumo Gemini, favoritos e lembretes	Implementado e testado manualmente
TTS, temas e fonte ampliada	Implementado e testado manualmente
OCR por imagem e reconhecimento de voz	Implementado; validação depende de Development Build

Fonte: o autor, 2026.

5. CONSIDERAÇÕES FINAIS

O trabalho teve como objetivo desenvolver uma aplicação móvel capaz de facilitar o acesso e a compreensão das informações presentes em bulas de medicamentos por meio de recursos de inteligência artificial e acessibilidade. A versão atual do Bula Fácil substituiu a arquitetura inicial em Flutter por uma solução baseada em React Native e Expo no frontend e Node.js e Express no backend, integrando MongoDB Atlas, Cloudflare R2 e Google Gemini.

Em resposta ao problema de pesquisa, o desenvolvimento demonstra que é tecnicamente possível reunir, em uma única aplicação móvel, diferentes formas de acesso às bulas: pesquisa textual, organização em seções, PDF oficial, resumo assistido por inteligência artificial, síntese de voz, reconhecimento por imagem, busca por voz, favoritos e lembretes. A utilização da bula oficial como fonte de dados e a manutenção do PDF original disponível ao usuário são elementos importantes para distinguir o conteúdo documental do resumo gerado pela IA.

Entre as vantagens observadas estão a centralização das informações, a possibilidade de consultar apenas a seção desejada, o uso de cache para evitar gerações repetidas do resumo, a leitura em áudio e a separação entre banco de dados e armazenamento dos PDFs. Como limitações, destacam-se a seleção de medicamentos restrita ao escopo de protótipo, a ausência de avaliação formal com usuários e a necessidade de Development Build para validar integralmente os módulos nativos de OCR e reconhecimento de voz.

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

REFERÊNCIAS

AGÊNCIA NACIONAL DE VIGILÂNCIA SANITÁRIA (ANVISA). Bulas, rótulos e nome comercial. Brasília, DF: ANVISA. Disponível em: https://www.gov.br/anvisa/pt-br/assuntos/medicamentos/bulas-e-rotulos. Acesso em: 13 set. 2026.

AGÊNCIA NACIONAL DE VIGILÂNCIA SANITÁRIA (ANVISA). Como acessar o Bulário Eletrônico? Brasília, DF: ANVISA. Disponível em: https://www.gov.br/anvisa/pt-br/assuntos/medicamentos/bulas-e-rotulos/como-acessar-o-bulario-eletronico. Acesso em: 13 set. 2026.

AGÊNCIA NACIONAL DE VIGILÂNCIA SANITÁRIA (ANVISA). Perguntas e respostas sobre bulas. Brasília, DF: ANVISA. Disponível em: https://www.gov.br/anvisa/pt-br/assuntos/medicamentos/bulas-e-rotulos/perguntas-e-respostas-sobre-bulas. Acesso em: 13 set. 2026.

CLOUDFLARE. Presigned URLs: Cloudflare R2 Docs. Disponível em: https://developers.cloudflare.com/r2/api/s3/presigned-urls/. Acesso em: 13 set. 2026.

CLOUDFLARE. S3 API compatibility: Cloudflare R2 Docs. Disponível em: https://developers.cloudflare.com/r2/api/s3/api/. Acesso em: 13 set. 2026.

EXPO. Speech: Expo Documentation. Disponível em: https://docs.expo.dev/versions/latest/sdk/speech/. Acesso em: 14 set. 2026.

EXPRESS.JS. Express: Node.js web application framework. Disponível em: https://expressjs.com/. Acesso em: 13 set. 2026.

GOMEZ-HERNANDEZ, M.; FERRE, X.; MORAL, C.; VILLALBA-MORA, E. Design Guidelines of Mobile Apps for Older Adults: Systematic Review and Thematic Analysis. JMIR mHealth and uHealth, v. 11, e43186, 2023. DOI: 10.2196/43186. Disponível em: https://mhealth.jmir.org/2023/1/e43186. Acesso em: 14 set. 2026.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep Learning. Cambridge: MIT Press, 2016.

GOOGLE. Recognize text in images with ML Kit on Android. Disponível em: https://developers.google.com/ml-kit/vision/text-recognition/v2/android. Acesso em: 13 set. 2026.

JURAFSKY, D.; MARTIN, J. H. Speech and Language Processing. 3. ed. draft. 2025. Disponível em: https://web.stanford.edu/~jurafsky/slp3/. Acesso em: 14 set. 2026.

LIU, N.; YIN, J.; TAN, S. S. L.; NGIAM, K. Y.; TEO, H. H. Mobile health applications for older adults: a systematic review of interface and persuasive feature design. Journal of the American Medical Informatics Association, v. 28, n. 11, p. 2483–2501, 2021. DOI: 10.1093/jamia/ocab151. Disponível em: https://pubmed.ncbi.nlm.nih.gov/34472601/. Acesso em: 14 set. 2026.

MONGODB. Databases and Collections. MongoDB Documentation. Disponível em: https://www.mongodb.com/docs/manual/core/databases-and-collections/. Acesso em: 13 set. 2026.

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

MONGODB. MongoDB Atlas Documentation. Disponível em: https://www.mongodb.com/docs/atlas/. Acesso em: 13 set. 2026.

PRESSMAN, R. S. Engenharia de Software: uma abordagem profissional. 8. ed. Porto Alegre: AMGH, 2016.

RUSSELL, S.; NORVIG, P. Inteligência Artificial: uma abordagem moderna. 4. ed. Rio de Janeiro: Elsevier, 2021.

SOMMERVILLE, I. Engenharia de Software. 10. ed. São Paulo: Pearson, 2019.

WORLD WIDE WEB CONSORTIUM (W3C). Web Content Accessibility Guidelines (WCAG) 2.1. 2018. Disponível em: https://www.w3.org/TR/WCAG21/. Acesso em: 13 set. 2026.

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