

Georreferenciamento de Farmácias para Retirada de Medicamentos via Agente Conversacional

Miguel Brondani¹, Narla Cossentino Fontoura², Daiane Rodrigues de Loreto², Daniel Welfer¹, Liziane Maahs Flores¹

¹ Universidade Federal de Santa Maria (UFSM),
Av Roraima nº1000, Santa Maria, RS, Brasil

² Prefeitura Municipal de Santa Maria
Santa Maria, RS, Brasil

{brondani.miguel, welfer}@gmail.com, liziane.m.flores@ufsm.br

{daiadeloreto, narlacfontoura}@gmail.com

Abstract. *The lack of a centralized way to distribute information about medication withdrawal hinders access to the healthcare system, especially for users with low digital literacy. This problem results in increased queues at basic health units. Reports from Sistema Único de Saúde (SUS) employees converge on the need for a simple and easily accessible way to provide location data for medication withdrawal. This article presents a simple, low-cost, and centralized solution for the georeferencing of municipal and state pharmacies, a list of medications available at each location and the required documentation, through the simulation of a conversational agent.*

Resumo. *A falta de uma forma centralizada de distribuição de informações sobre retirada de medicamentos dificulta o acesso ao sistema de saúde, principalmente para usuários com baixa alfabetização digital. Tal problema acarreta no aumento de filas em unidades básicas de saúde. Relatos de funcionários do Sistema Único de Saúde (SUS) convergem na necessidade de uma forma simples e de fácil acesso sobre dados de localização de retirada de medicamentos. Este artigo apresenta uma solução simples e centralizada de baixo custo para georreferenciamento de farmácias municipais e estaduais, a relação dos medicamentos que podem ser retirados em cada uma delas e a documentação necessária para tal, através da simulação de um agente conversacional.*

1. Introdução

Há desafios na distribuição de informações sobre disponibilidade de medicamentos em farmácias públicas do Sistema Único de Saúde (SUS). Pacientes frequentemente enfrentam longas filas em Unidades Básicas de Saúde (UBS) para obter informações básicas sobre onde retirar seus medicamentos.

A demanda pelo desenvolvimento da aplicação surgiu de relatos de profissionais de saúde vinculados ao Programa de Educação pelo Trabalho para a Saúde: Informação e Saúde Digital (PET-Saúde/I&SD), que atuam diretamente nas unidades de saúde municipais. Segundo esses profissionais, uma parcela recorrente dos atendimentos nas Unidades Básicas de Saúde é destinada a esclarecer dúvidas sobre onde retirar medicamentos. Tal

demanda poderia ser atendida por um canal digital centralizado, liberando tempo dos servidores para atividades assistenciais

Este artigo apresenta o desenvolvimento de uma aplicação web *low-cost* que integra georreferenciamento de farmácias, agente conversacional simplificado e visualização de mapas interativos. Uma solução centralizada para reduzir o tempo de busca dos cidadãos e a carga de trabalho administrativo dos servidores da saúde.

O trabalho faz parte do PET-Saúde/I&SD, realizado em parceria com instituições municipais e estaduais de saúde.

2. Métodos

2.1. Arquitetura do Sistema

A aplicação foi desenvolvida utilizando Django com Django REST Framework no backend. A arquitetura MVT (Model-View-Template) do Django facilitou a rápida implementação da lógica de negócio e endpoints RESTful. O frontend foi construído em HTML5/CSS3 e JavaScript, sem dependências de frameworks adicionais.

A hospedagem da aplicação foi realizada na plataforma PythonAnywhere [PythonAnywhere LLP 2012], um serviço de hospedagem gerenciada especializado em aplicações Python. O ambiente oferece suporte nativo ao protocolo Web Server Gateway Interface (WSGI), utilizado pelo Django para comunicação entre o servidor web e a aplicação. A escolha pela plataforma foi motivada pelo seu plano gratuito, que inclui um *worker* WSGI dedicado, domínio público no subdomínio `.pythonanywhere.com`, e 512 MB de espaço em disco suficiente para hospedar a aplicação, os arquivos estáticos e os dados tabulares em CSV.

A ausência de banco de dados relacional foi uma decisão deliberada de arquitetura motivada pelas restrições de armazenamento da plataforma. Os dados de medicamentos são mantidos em um arquivo CSV único lido em memória a cada requisição via *pandas*, o que é viável dado o tamanho do conjunto de dados. Essa abordagem elimina a necessidade de configuração e manutenção de um Sistema Gerenciador de Banco de Dados (SGBD), mas introduz uma limitação de escalabilidade caso o volume de dados cresça significativamente.

O fluxo de uma requisição típica ocorre da seguinte forma: o cliente realiza uma chamada HTTP POST ao endpoint `/chat`, enviando o texto digitado e a intenção identificada pela Máquina de Estados Finita (FSM), descrita na Seção 2.2. O backend interpreta a intenção, executa a busca no CSV e retorna um objeto JSON contendo a resposta textual ou os dados geográficos para renderização da mensagem.

2.2. Agente conversacional

A implementação do agente conversacional foi norteada pela necessidade de balancear a experiência do usuário com as restrições de latência e custo operacional. Inicialmente, o projeto avaliou o uso de modelos de linguagem de grande escala (LLMs), especificamente o modelo Ollama 3.2 via Ollama. Contudo, a execução de LLMs no ambiente de hospedagem foi descartada pois o armazenamento exigido por eles ultrapassa o limite disponível no plano gratuito do Pythonanywhre. Essa abordagem foi descartada por duas razões principais: o armazenamento exigido pelos pesos do modelo ultrapassa o limite

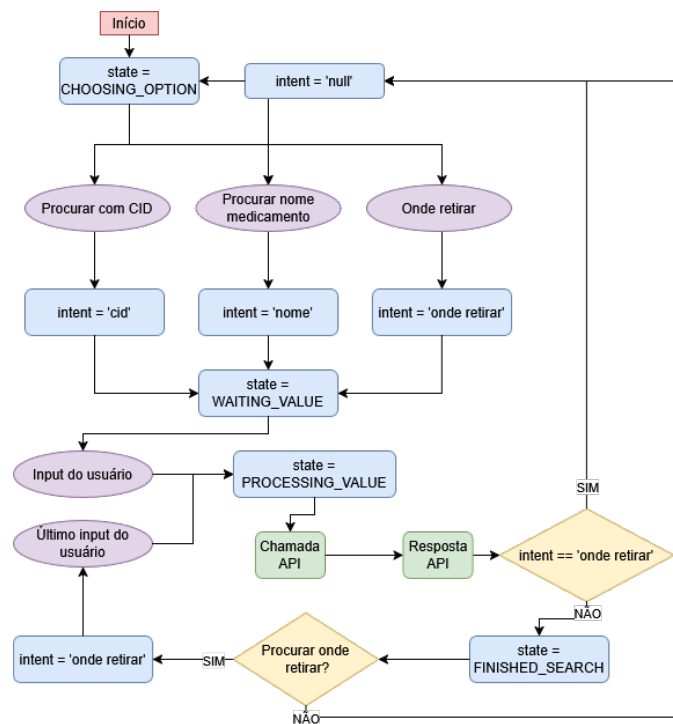


Figura 1. Fluxograma de transições da FSM

disponível no plano gratuito do PythonAnywhere, e a latência de inferência em ambiente de hospedagem compartilhada mostrou-se incompatível com uma experiência de uso aceitável.

Diferentes técnicas podem ser empregadas na construção de agentes conversacionais, incluindo *Artificial Intelligence Markup Language*, correspondência de padrões, redes neurais e LLMs [Abdul-Kader and Woods 2015]. A arquitetura final foi consolidada em uma Máquina de Estados Finitos (FSM) implementada no lado do cliente via JavaScript. Essa abordagem garante execução instantânea da lógica de interface, elimina custos de inferência e reduz a carga sobre o servidor, que passa a ser responsável exclusivamente pela busca e formatação dos dados.

O controle do fluxo de diálogo é regido por duas variáveis globais mantidas no cliente: *state*, que representa o estado atual da FSM, e *intent*, que armazena a intenção de busca e é enviada a cada requisição ao *backend*, determinando qual serviço será acionado.

A FSM opera em quatro estados, ilustrados na Figura 1. Em *CHOOSING_OPTION*, o sistema apresenta Botões de Resposta Rápida e aguarda a seleção do usuário. Selecionada uma opção, *intent* é atribuído e o sistema transita para *WAITING_VALUE*, aguardando a entrada textual do parâmetro de busca. Após o envio, entra em *PROCESSING*, estado de bloqueio que impede requisições simultâneas durante a chamada à API. Conforme a resposta recebida, o sistema retorna a *CHOOSING_OPTION* ou transita para *FINISHED_SEARCH*, onde o usuário pode solicitar a localização do medicamento ou retornar ao menu inicial.

Para lidar com variações ortográficas e erros de digitação nos nomes de medi-

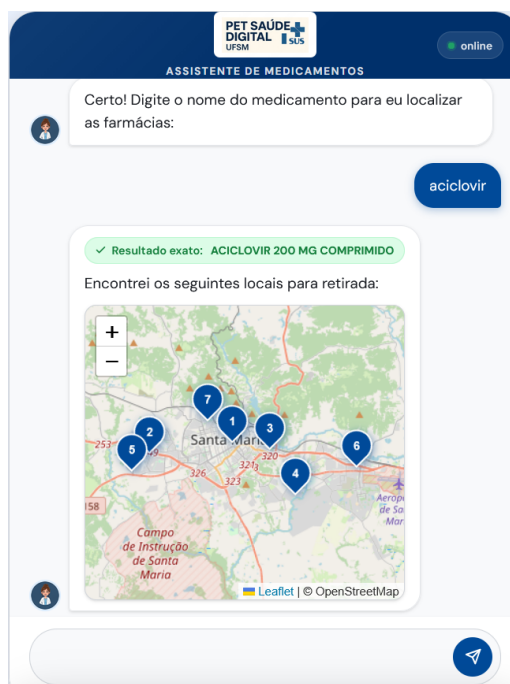


Figura 2. Captura de tela mostrando o mapa, pontos de retirada do medicamento aciclovir e confirmação de busca

camentos e códigos CID, o sistema emprega um algoritmo de busca por similaridade baseado na distância de Levenshtein [Levenshtein 1966], implementado na biblioteca python *TheFuzz* [Choldate and SeatGeek 2021]. A distância de Levenshtein quantifica o número mínimo de operações de edições necessárias para transformar uma cadeia de caracteres em outra. O sistema realiza inicialmente uma busca por correspondência exata. Na ausência de resultados exatos executa uma busca por similaridade com limiar configurável. O usuário é informado sobre o tipo de correspondência retornada diretamente na interface, distinguindo resultados exatos de resultados aproximados.

2.3. Georreferenciamento e Dados

Os dados de medicamentos disponibilizados pelas farmácias distritais, municipal e estadual foram compilados em um único arquivo *CSV*, obtidos pelos integrantes do projeto a partir de documentos disponíveis publicamente. O caráter público dessas informações faz com que não seja necessário qualquer tipo de anonimização, eliminando riscos referentes a ataques de dados sensíveis. O sistema conta atualmente com oito farmácias cadastradas e cerca de 3000 registros de medicamentos.

As coordenadas geográficas de cada farmácia foram obtidas manualmente através do serviço Google Maps e armazenadas em um arquivo *JSON* junto às demais informações institucionais de cada unidade, como endereço, fotografia da fachada e documentação necessária para a retirada ou entrada em processo de retirada de medicamentos. A visualização cartográfica e a interatividade com os marcadores geográficos foram implementadas com a biblioteca Leaflet [Agafonkin 2024], de código aberto. Ao interagir com um marcador no mapa, o usuário tem acesso ao nome, endereço e imagem da farmácia correspondente por meio de um painel deslizante.

É importante destacar que os dados apresentados pelo sistema referem-se aos

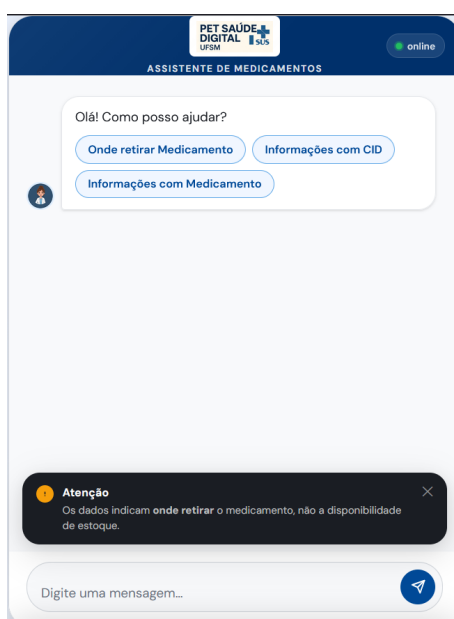


Figura 3. Captura de tela do app mostrando as Respostas Rápidas e o pop-up de aviso sobre disponibilidade dos medicamentos.

medicamentos dispensados em cada farmácia, conforme os documentos públicos de referência, e não à disponibilidade de estoque em tempo real. Essa limitação é comunicada ao usuário diretamente na interface, por meio de um aviso exibido sempre que acessa o site.

2.4. Design

O design da interface foi orientado para atender usuários com baixa familiaridade tecnológica, simulando a experiência visual de aplicativos de mensagens instantânea, formato amplamente difundido na população brasileira. A adoção do modelo conversacional como interface é respaldada por trabalhos anteriores. [Fontainha et al. 2023] demonstraram que chatbots integrados a aplicativos de mensagens atingiram altos índices de compreensão entre usuários do sistema público de saúde, incluindo aqueles com menor grau de escolaridade.

A navegação é guiada por Botões de Resposta Rápida, que apresentam as opções de busca disponíveis sem exigir que o usuário formule uma consulta em linguagem natural. O único input textual livre necessário é o nome do medicamento ou o código da Classificação Internacional de Doenças (CID). O sistema foi projetado com foco em dispositivos móveis, dado que o acesso via smartphone é o meio mais provável de uso pela população-alvo.

3. Testes e Resultados

A disponibilização pública do software está condicionada à aprovação do Comitê de Ética em Pesquisa (CEP), etapa prevista na continuidade do projeto. Até o momento, o sistema foi avaliado internamente por profissionais de saúde membros do PET-Saúde/I&SD envolvidos no desenvolvimento e na coleta de dados, totalizando entre cinco e quinze avaliadores. A avaliação com o público geral não foi realizada nesta etapa, sendo prevista para a etapa seguinte à aprovação do CEP.

Os retornos obtidos nessa fase inicial foram positivos. A interface foi considerada intuitiva, e os Botões de Resposta Rápida foram destacados como facilitadores da navegação, reduzindo a necessidade de entrada textual livre por parte do usuário. O tempo de resposta do servidor se mostrou adequado para o contexto de uso.

O processo de avaliação seguiu um modelo iterativo: à medida que os profissionais de saúde interagem com o sistema e reportavam inconsistências ou ausências, melhorias eram incorporadas de forma contínua. A principal limitação identificada ao longo desse processo refere-se à incompletude da base de dados, com registros de medicamentos ainda pendentes de inserção. Entre as evoluções decorrentes do ciclo de testes, destacam-se: a inclusão de medicamentos e farmácias inicialmente ausentes na base de dados, correção de informações, aprimoramentos na interface e no fluxo de navegação, e a incorporação de novas funcionalidades, como a exibição dos documentos necessários para a retirada de medicamentos e o aviso explícito sobre a não garantia de disponibilidade de estoque em tempo real. Esse ciclo de desenvolvimento orientado pelo uso contribuiu para aproximar o sistema das necessidades reais identificadas pelos profissionais envolvidos.

4. Conclusão

O projeto tem se mostrado promissor. Os testes iniciais, conduzidos com profissionais da área da saúde, demonstraram que o sistema apresenta facilidade de uso e tempo de resposta adequado. Apesar do desenvolvimento ter sido focado para o público geral das farmácias, os relatos dos avaliadores indicam que a ferramenta também apresenta potencial de uso por agentes de saúde. A infraestrutura é financeiramente sustentável, sem geração de custos operacionais recorrentes. A continuação do trabalho contempla o desenvolvimento de um canal que possibilite a profissionais de saúde reportarem inconsistências entre as informações do sistema e a realidade observada, viabilizando a posterior abertura ao público-alvo, mediante aprovação do CEP.

Referências

- Abdul-Kader, S. A. and Woods, J. (2015). Survey on chatbot design techniques in speech conversation systems. *International Journal of Advanced Computer Science and Applications*, 6(7):72–80.
- Agafonkin, V. (2024). Leaflet, a javascript library for interactive maps. Disponível em: <https://github.com/Leaflet/Leaflet>. Licença: BSD-2-Clause.
- Choldate, A. and SeatGeek (2021). thefuzz: Fuzzy string matching in Python. <https://github.com/seatgeek/thefuzz>. Acesso em: fev. 2026.
- Fontainha, T. C., Silva, N. J., and Cappelli, C. (2023). SUSi-Carioca: Chatbot de pós-atendimento de consultas no SUS utilizando práticas de linguagem simples. In *Anais Estendidos do XIX Simpósio Brasileiro de Sistemas de Informação (SBSI)*, pages 38–47, Maceió, AL. SBC.
- Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707.
- PythonAnywhere LLP (2012). PythonAnywhere: Host, run, and code Python in the cloud. <https://www.pythonanywhere.com>. Acesso em: fev. 2026.