UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI
Curso de Graduação em Sistemas de Informação
Victor Almeida de Amorim

DESENVOLVIMENTO DE UMA FERRAMENTA CHATBOT PARA ATENDIMENTO PÓS-ALTA DE PACIENTES.

Diamantina
2025

Victor Almeida de Amorim

DESENVOLVIMENTO DE UMA FERRAMENTA CHATBOT PARA ATENDIMENTO PÓS-ALTA DE PACIENTES.

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação da Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM), como parte dos requisitos exigidos para a obtenção título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Alessandro Vivas Andrade

Diamantina
2025

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI

FOLHA DE APROVAÇÃO

Victor Almeida de Amorim

DESENVOLVIMENTO DE UMA FERRAMENTA CHATBOT PARA ATENDIMENTO PÓS-ALTA DE PACIENTES.

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de Informação da Universidade Federal dos Vales do Jequitinhonha e Mucuri, como requisito parcial para conclusão do curso.

Orientador: Prof. Dr. Alessandro Vivas Andrade

Aprovado em 18 de novembro de 2025

BANCA EXAMINADORA

Prof. Dr. Alessandro Vivas Andrade
Faculdade de Ciências Exatas - DECOM - UFVJM

Prof. Patrick de Paula Barros
Faculdade de Ciências Exatas - DECOM - UFVJM

Dr. Henrique Carlos Fonte Bôa Carvalho
STI/UFVJM

Documento assinado eletronicamente por **Alessandro Vivas Andrade**, **Servidor(a)**, em 18/11/2025, às 16:18, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Henrique Carlos Fonte Bôa Carvalho**, **Servidor(a)**, em 18/11/2025, às 17:00, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Patrick de Paula Barroso**, **Servidor(a)**, em 01/12/2025, às 08:15, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://sei.ufvjm.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1945259** e o código CRC **A845BF33**.

Referência: Processo nº 23086.000729/2023-83 SEI nº 1945259

AGRADECIMENTOS

Agradeço primeiramente a Deus, por todo o cuidado a mim concedido, escolhendo as melhores pessoas, situações, oportunidades e condições me guiando pelos melhores caminhos.

Agradeço igualmente à minha família, em especial aos meus pais, Evandro e Andréia, por sempre me proporcionaram apoio incondicional e as melhores bases para enfrentar cada etapa desta jornada. Sem eles, nada disso seria possível.

Agradeço também meus amigos e colegas de curso, especialmente Mariana, Pedro, Ian, Jhonathan Marcos e Igor, por fazerem essa jornada mais descontraída e crível.

Por fim, registro minha sincera gratidão aos técnicos e docentes do curso, cujo trabalho e dedicação foram fundamentais para a realização desta conquista. Um agradecimento especial ao professor Dr. Alessandro Vivas, meu orientador, pelos valiosos ensinamentos, paciência e orientação ao longo de todo o processo.

RESUMO

O acompanhamento de pacientes no período pós-alta hospitalar é um grande desafio para as instituições de saúde, impactando diretamente na continuidade do acompanhamento e na prevenção de readmissões. Diante deste cenário, o presente estudo teve como objetivo o desenvolvimento e a validação de uma ferramenta de chatbot para automatizar e otimizar a comunicação entre hospital e paciente. A metodologia envolveu o projeto de um protótipo funcional, utilizando tecnologias de baixo custo para criar um fluxo conversacional estruturado. Os resultados demonstraram que a ferramenta é tecnicamente viável e de fácil utilização, executando com sucesso as rotinas de acompanhamento propostas. Portanto, o chatbot representa uma solução prática e acessível para qualificar o monitoramento pós-alta, promovendo maior engajamento do paciente e oferecendo um canal eficiente para a coleta de dados de saúde.

Palavras-chave: Chatbot. Pós-alta. Cuidado ao Paciente. Tecnologia em Saúde. Monitoramento de pacientes.

ABSTRACT

Post-hospital discharge patient follow-up is a significant challenge for healthcare institutions, directly impacting the continuity of care and the prevention of readmissions. Given this scenario, the objective of this study was the development and validation of a chatbot tool to automate and optimize communication between the hospital and the patient. The methodology involved designing a functional prototype using low-cost technologies to create a structured conversational flow. The results demonstrated that the tool is technically feasible and user-friendly, successfully executing the proposed follow-up routines. Therefore, the chatbot represents a practical and accessible solution to enhance post-discharge monitoring, promoting greater patient engagement and offering an efficient channel for health data collection.

Keywords: Chatbot. Post-discharge. Patient Care. Healthcare Technology. Patient Monitoring.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

UFVJM	Universidade Federal dos Vales do Jequitinhonha e Mucuri
ISC	Infecções de Sítios Cirúrgicos
LGPD	Lei Geral de Proteção de Dados
AILM	Linguagem de Marcação de Inteligência Artificial
XML	Linguagem de Marcação Extensível
LSA	Análise Semântica Latente
NLP	Processamento de Linguagem Natural
NLU	Compreensão de Linguagem Natural
INEP	Instituto Nacional de Estudo e Pesquisas
API	Interface de Programação de Aplicações
IBM	International Business Machines Corporation
LLM	Modelos de Linguagem Generativos
ANPD	Autoridade Nacional de Proteção de Dados
HTTPS	Protocolo de Transferência de Hipertexto Seguro
URL	Localizador Uniforme de Recursos

SUMÁRIO

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO DO PROBLEMA

As Infecções de Sítios Cirúrgicos (ISC) são aquelas que ocorrem no local em que o procedimento cirúrgico foi realizado, seja na superfície da epiderme até em alguma parte interna do corpo humano. As ISC's estão entre as infecções mais preocupantes presentes nos hospitais, podendo ser diagnosticada em até 30 dias após o procedimento (Batista; Rodrigues, 2012). É uma infecção problemática pois afeta a saúde física e emocional dos pacientes, aumentando os índices de permanência hospitalar e contribuindo para altas nas taxas de mortalidade dos pacientes (Oliveira; Ciosak; D'Lorenzo, 2007).

Segundo Portaria nº 2616 do Brasil. Ministério da Saúde (1998), os hospitais devem possuir um conjunto de normas e diretrizes que cuidam da saúde do paciente, desde sua internação até sua alta. É aconselhado que as normas sigam métodos de busca ativa, sistemática e contínua de infecções nos pacientes. No contexto da ISC, observa-se que grande parte das normas abrangem apenas a etapa de internação, sendo que grande parte das ISC dão-se após a alta (Oliveira; Ciosak; D'Lorenzo, 2007).

Neste contexto, o acompanhamento na etapa pós-alta do atendimento hospitalar mostra-se fundamental, bem como a implementação de métodos baseados em métodos proativos de monitoramento contínuo dos pacientes (Oliveira; Ciosak; D'Lorenzo, 2007). Vale ressaltar que não há um procedimento que resulte em uma taxa de 0% de infecção no sítio cirúrgico, mas a implementação de boas práticas reduz os índices de permanência hospitalar e também os índices de fatalidades. Ainda dentro desta, o pós-alta é muito difícil de ser mapeado, devido a uma grande resistência por parte dos pacientes a seguirem os poucos procedimentos a eles repassados. É neste cenário que a implementação de um chatbot mostra-se interessante, pois, de maneira simples, irá fazer um controle do pós operatório remotamente, sem causar transtornos ao paciente.

Para realizar o acompanhamento do paciente pós-alta, a abordagem proposta neste trabalho é a utilização de um agente de chatbot, por meio do aplicativo gratuito de mensagens instantâneas *WhatsApp*, para realizar o atendimento inicial. O uso do aplicativo é justificável pelo número de dispositivos digitais (smartphones, tablets, computadores pessoais) existentes no Brasil, 480 milhões de dispositivos, 2.2 por cidadão, sendo que só de *smartphones* são 258 milhões de dispositivos, 1.2 por cidadão (Meirelles, 2023). Outros argumentos que corroboram para o uso do aplicativo é sua simplicidade no processo de instalação, bastando apenas efetuar o *download* na loja de aplicativos do *smartphone* e os requisitos para sua execução são acessíveis até por dispositivos antigos. O *WhatsApp* também é conhecido por ser um aplicativo que não demanda uma conexão de rede robusta para seu funcionamento, não utiliza muito do armazenamento do dispositivo e é usado por pessoas de diversas idades e classes sociais.

Um agente chatbot tem como objetivo simular uma interação social da maneira mais humana possível, permitindo que a experiência do paciente durante o atendimento seja eficaz, efetiva e rápida. Com os avanços tecnológicos, a implementação desses agentes tornou-se mais acessível, permitindo até a implementação de conceitos mais complexos, como o processamento de linguagem natural (Oliveira Aquino; Costa Adaniya, 2018).

Essa automatização será feita por meio de uma série de perguntas pré programadas, tendo suas respostas simples - sim ou não - analisadas e repassadas para análise da equipe hospitalar. O agente analisará as respostas recebidas e acionará gatilhos caso identifique sinais de infecção no local onde foi realizado o procedimento, enviando a foto anexada do local para análise humana e recomendando o paciente a dirigir-se à unidade de saúde mais próxima. A automação desse atendimento é um processo fundamental para a construção de melhores procedimentos que buscam a identificação de infecções, pois quanto mais cedo for a infecção for identificada, melhor serão as suas condições de tratamento pela equipe hospitalar.

O presente estudo é estruturado da seguinte forma: Introdução, a Fundamentação Teórica do trabalho, o uso da Lei Geral de Proteção de Dados Pessoais (LGPD) no contexto do agente chatbot, Metodologia usada para o desenvolvimento, Desenvolvimento do chatbot, Resultados e Testes, Discussão e Conclusão.

1.2 OBJETIVOS

O principal objetivo geral do presente trabalho é a construção de uma ferramenta chatbot para auxiliar o monitoramento do paciente na etapa pós cirúrgica.

Os objetivos específicos envolvem a coleta de dados dos pacientes para realizar a consulta, o envio de alertas para os profissionais da saúde caso necessário e facilitar a comunicação do paciente com o responsável pelo chatbot.

2 REFERENCIAL TEÓRICO

2.1 CHATBOTS E AGENTES CONVENCIONAIS

A inteligência artificial tem se tornado cada vez mais corriqueira em nosso dia a dia, principalmente pela figura dos agentes inteligentes. Tais agentes podem exercer diversas funcionalidades, desde tarefas simples e repetitivas a tarefas sofisticadas e completas (Adamopoulou; Moussiades, 2020). Dentro dos agentes, o chatbot é uma classe que está cada vez mais presente em nosso cotidiano. Chatbot é um programa que simula e processa a conversa humanas, permitindo a interação entre pessoas e dispositivos digitais como se comunicam com seres humanos (Oracle, 2025). Podem ser programas rudimentares, abstendo-se a consultas simples, ou tão complexos como assistentes virtuais, Google Assistente e Siri.

Apesar de serem construídos para imitar interações humanas, possuem diversas funcionalidades, como o rastreio de encomendas, resposta à perguntas frequentes, notificações e lembretes automáticos. Grande parte das implementações independem do aplicativo de mensagens, acessando os contatos de quem o implementa sem sair do ambiente do aplicativo e não causa transtornos aos usuários - trocar de aplicativo para usar a ferramenta ou lentidão na comunicação com o chatbot. (Adamopoulou; Moussiades, 2020).

A grande promessa desta tecnologia é o atendimento rápido e eficiente e um suporte específico para questões de usuários (Følstad; Nordheim; Bjørkli, 2018), aumentando a satisfação dos usuários e a produtividade da organização. A confiabilidade depende de fatores sociais, como sua semelhança à interação humana, e também fatores técnicos, como a segurança das informações tratadas nele (Følstad; Nordheim; Bjørkli, 2018).

2.1.1 Conceitos básicos

Prosseguindo com o entendimento sobre o que é um agente chatbot, é necessário definir alguns conceitos. Uma entidade é entendida como parâmetro relevante passado pela entrada, o que deve ser tratado na saída (Adamopoulou; Moussiades, 2020). O contexto de uma entidade é uma ou um conjunto de strings armazenadas que mantém o estado da conversa, permitindo que o agente trate de entradas que o usuário refere de maneira subsequente (Adamopoulou; Moussiades, 2020). Em chatbot de atendimento médico, por exemplo, a Entidade passada pode ser a data e o horário de uma consulta, enquanto seu contexto é mantido uma lista de *string* [data, consulta, marcar], podendo ser retomada a qualquer momento da interação.

A correspondência de padrões é a abordagem escolhida para definir a como o agente tratará a entrada - estímulo - e gerará a saída - resposta. Esta abordagem deixa o agente repetitivo e previsível, diminuindo a humanidade do agente em suas interações (Adamopoulou; Moussiades, 2020). Visando mitigar a falta da humanidade na interação, a Linguagem de Marcação de Inteligência Artificial (AILM) foi desenvolvida para facilitar a modelagem dos diálogos possíveis pelos agentes chatbots. É inspirada na Linguagem de Marcação Extensível (XML), pois usa-se de tags para identificar código e comandos - <comando>Parâmetros</comando>. Cada

tag corresponde a um comando diferente (Marietto et al., 2013). Para melhorar a AILM, surgiu a Análise Semântica Latente (LSA), sendo esta uma técnica para identificação de similaridade entre as palavras, através do uso de um vetor matemático (Thomas, 2016). Para a implementação dessas técnicas, ferramentas como ChatScript e o RiveScript foram criados. O agente efetua uma busca por tópicos em suas regras pré-definidas e gera uma resposta de acordo com a entrada inserida (Adamopoulou; Moussiades, 2020).

Por fim, temos o Processamento de Linguagem Natural (NLP) como a área da inteligência artificial que busca entender os desdobramentos da linguagem natural de forma que a máquina seja capaz de realizar as tarefas a ela solicitadas (Vicente Neto; Espirito Santo; Goldman, 2020) . O conceito de Compreensão de Linguagem Natural (NLU) é o core da NLP, sendo a extração do significado das entradas de linguagem natural por parte do usuário (Vicente Neto; Espirito Santo; Goldman, 2020).

2.1.2 Classificação de agentes chatbot

Os agentes podem ter diversas classificações, de acordo com sua usabilidade e funcionalidades. Acerca do seu domínio de conhecimento, eles podem ser generalistas - não se restringindo a apenas uma área do conhecimento - ou focados em um domínio específico - treinados em uma área específica, podendo retornar respostas erráticas sobre outras áreas (Adamopoulou; Moussiades, 2020). Um exemplo de um chat generalista é o Chat GPT da OpenIA, capaz de fornecer informações e gerar respostas sobre assuntos de diferentes domínios. Por outro lado, um exemplo de modelo focado por domínio específico é o chatbot Lu, da Magazine Luiza, sendo capaz de auxiliar os clientes em todas as etapas no processo de compra e venda de produtos.

Podem ainda ser classificados de acordo com o serviço prestado, sendo eles interpessoais quando oferecerem serviços específicos - agendamento de consultas para uma clínica específica, remarcação de viagens aéreas - ou pessoais quando são assistentes pessoais dos usuários (Adamopoulou; Moussiades, 2020), como a Siri da Apple ou Google Assistente.

Outra classificação dos agentes consideram os objetivos primários. Podem ser informativos - agentes projetados para transmitir informações já pré definidas ao usuário -, de conversação - chats criados para interagir com o usuário como se humano fosse - e baseado em tarefa - criados para executar uma tarefa, auxiliando seu usuário (Kucherbaev; Bozzon; Houben, 2018). O Instituto Nacional de Estudo e Pesquisas (INEP) implementou em sua plataforma diversos chatbots informativos para auxiliar na realização do Exame Nacional do Ensino Médio (ENEM), agilizando os processos de inscrição, consulta de provas anteriores, o local de prova e outras informações. Como modelo de conversação existem os chats da *Character.ai*, disponibilizando diversas figuras personificadas em chatbots, permitindo que os conversem com diversas figuras diferentes. Por fim, como baseado em tarefas tem-se o Google Assistente ou Siri, eles ajudam o usuário com tarefas do dia a dia.

O método de processamento de entrada e geração de saída também são critérios usados para a classificação de agentes, sendo eles baseados em regra - cria sua resposta a partir de

um conjunto de regras pré definidas, escolhendo a que mais se adequada à entrada, tendo seu conhecimento fornecido pelos programadores e organizado por meio de padrões e estruturas lógicas -, baseado em recuperação - utiliza de Interfaces de Programação de Aplicações (API) para identificar a saída mais adequada para a entrada - e o modelo generativo - utilizam algoritmos de *machine learning* e *deep learning* para treiná-los, sendo estes os mais próximos de um humano (Adamopoulou; Moussiades, 2020). Um exemplo de chatbot baseado em regras pode ser um que faça atendimento ao cliente e que responde a perguntas frequentes sobre horários de funcionamento utilizando-se de respostas programadas. Como exemplo de um modelo baseado em recuperação, existe o Watson Assistant, chatbot da International Business Machines Corporation (IBM), que pode ser treinado com um conjunto de perguntas e respostas frequentes, recuperando a resposta mais relevante com base na consulta do usuário. E como exemplo de modelo generativo, tem-se o Chat GPT que baseia-se em modelos de linguagens para gerar respostas em linguagem natural, totalmente adaptadas ao contexto. O modelo proposto no presente trabalho é melhor classificado como um agente baseado em regras - possuindo padrões e respostas pré-definidas para as interações com os pacientes.

2.2 PLATAFORMAS DE COMUNICAÇÃO

Em uma pesquisa recente feita pela empresa We Are Digital, foi identificado no território brasileiro um total de 217 milhões de *smartphones* conectados na rede e um total de 144 milhões de usuários ativos foram identificados no Brasil até o mês de Novembro de 2024.

Marta Cardoso de Andrade, em sua pesquisa, verificou que o *WhatsApp* conta com mais de 2 bilhões de *downloads* por todo o globo, sendo um dos aplicativos de mensagens instantâneas mais utilizado no planeta (Andrade, 2021). Em relação ao número de usuários ativos em 2024, cerca de 97% usam regularmente o aplicativo. Alguns fatores que fazem o *WhatsApp* ser uma aplicação muito usada são o fato de ser gratuito, a facilidade de adquirir, os requisitos baixos para seu uso e o baixo consumo de dados de rede.

Explorando outros aplicativos, o *Instagram* é amplamente utilizado pelos usuários ativos, 92% fazem o seu uso regularmente. O fato de ser uma aplicação integrada ao ecossistema da Meta - *WhatsApp*, *Facebook* e *Instagram*, e também ser gratuita para uso justificam sua alta de uso, mas diferentemente do *WhatsApp*, não possui como foco a troca de mensagens, tornando uma opção menos interessante para o uso neste trabalho.

Por fim, o *Telegram* é outro dos mais utilizados pela população brasileira, sendo usado por 56% dos usuários ativos de internet no país. Diferentemente do *WhatsApp*, para fazer total uso de seus recursos é necessário assinar um plano *premium*, reduzindo funcionalidades - como a velocidade de *download* e *upload* - limitando a experiência do usuário.

3 LGPD E QUESTÕES DE PRIVACIDADE

3.1 INTRODUÇÃO À LEI GERAL DE PROTEÇÃO DE DADOS

O Brasil possui a LEI Nº 13.709, DE 14 DE AGOSTO DE 2018, um conjunto de normas sobre o tratamento de dados pessoais por pessoas naturais ou jurídicas no território brasileiro. Contém mais de 50 artigos que abordam todo o processo de tratamento de dados até a definição de uma autoridade nacional - Autoridade Nacional de Proteção de Dados (ANPD) - responsável por fiscalizar a aplicação das normas prescritas.

Dados pessoais são informações relacionadas ao titular e que podem o identificar, conforme disposto no Art. 5º da Lei Geral de Proteção de Dados (Brasil, 2018, art. 5º). Entre os dados pessoais, existe uma categoria chamada de dados pessoais sensíveis, sendo eles aqueles acerca da origem racial ou étnica, convicção religiosa, orientação política, filiação a sindicato ou a organização de caráter religioso, filosófico ou político, dado referente à saúde ou à vida sexual, dado genético ou biométrico, quando vinculado a uma pessoa natural. Ainda no contexto do Art. 5º, o de titular dos dados pessoais é a pessoa natural a quem os dados referem-se. Já o controlador refere-se a pessoa jurídica de natureza pública ou privada a quem compete as decisões do tratamento dos dados e a figura do operador da-se na pessoa jurídica de natureza pública ou privada responsável pelo tratamento dos dados. Controlador e operador em conjunto são definidos como os agentes de tratamento dos dados. O tratamento refere-se a toda operação realizada com dados pessoais, como coleta, recepção, classificação, entre outras. Por fim, o consentimento é manifestação livre, informada e inequívoca pela qual o titular concorda com o tratamento de seus dados pessoais para uma finalidade determinada.

3.2 PRINCÍPIOS DA LGPD APLICADOS AO CHATBOT

Acerca do tratamento pelo chatbot, o uso é permitido em lei caso atenda os princípios de legitimidade, adequação e necessidade no tratamento, livre acesso e transparência do uso aos titulares, segurança e prevenção em seu uso e por fim, responsabilização por quem os trata, conforme Art. 6º da Lei Geral de Proteção de Dados (Brasil, 2018, art. 6º). Para iniciar o tratamento dos dados, a legislação prevê que todo o processo deve possuir a autorização do titular como previsto nos Art. 7º e 11º da Lei Geral de Proteção de Dados (Brasil, 2018, art. 7º, art. 11º). Contudo, existe a possibilidade de tratar os dados sem o consentimento do titular, sendo quando se trata de proteção à saúde e integridade física do titular. Os dados pessoais a serem tratados são: nome, telefone, data de nascimento e data de alta coletadas do paciente via interação com a enfermeira responsável pelo projeto. A interação com o chatbot armazenará um conjunto com 3 respostas de sim ou não, não contemplando nenhum dado pessoal.

3.3 CONSENTIMENTO DO PACIENTE

O titular dos dados deverá ceder o consentimento de maneira direta com o tratamento de dados que serão feitos pelo chatbot, aceitando os termos e condições de uso do agente conforme previsto no Art. 8º da Lei Geral de Proteção de Dados. Por exemplo, será salvo em sua instância no banco de dados o campo de consentimento preenchido com o *link* para um documento salvo em nuvem, sendo este o termo de consentimento assinado pelo titular dos dados.

Os agentes de tratamento (controlador e operador) devem garantir aos titulares acesso aos seus dados em tratamento contendo finalidade, forma, duração do tratamento e a sua identificação. Devem garantir também a correção de dados, caso necessário, o direito à anonimização e até exclusão dos dados como consta no Art. 18º da Lei Geral de Proteção de Dados (Brasil, 2018, Art. 18º). A qualquer momento, o titular poderá solicitar ao controlador e operador quais dados estão em sua posse, podendo pedir sua correção ou exclusão deles no banco de dados, encerrando assim o tratamento do agente.

3.4 ARMAZENAMENTO E TRATAMENTO E DADOS

O armazenamento e tratamento de dados pessoais, segundo a legislação, deve seguir os seguintes critérios definidos no Art. 6º da Lei Geral de Proteção de Dados:

> ”I - finalidade: realização do tratamento para propósitos legítimos, específicos, explícitos e informados ao titular, sem possibilidade de tratamento posterior de forma incompatível com essas finalidades; II - adequação: compatibilidade do tratamento com as finalidades informadas ao titular, de acordo com o contexto do tratamento; III - necessidade: limitação do tratamento ao mínimo necessário para a realização de suas finalidades, com abrangência dos dados pertinentes, proporcionais e não excessivos em relação às finalidades do tratamento de dados; IV - livre acesso: garantia, aos titulares, de consulta facilitada e gratuita sobre a forma e a duração do tratamento, bem como sobre a integralidade de seus dados pessoais; V - qualidade dos dados: garantia, aos titulares, de exatidão, clareza, relevância e atualização dos dados, de acordo com a necessidade e para o cumprimento da finalidade de seu tratamento; VI - transparência: garantia, aos titulares, de informações claras, precisas e facilmente acessíveis sobre a realização do tratamento e os respectivos agentes de tratamento, observados os segredos comercial e industrial; VII - segurança: utilização de medidas técnicas e administrativas aptas a proteger os dados pessoais de acessos não autorizados e de situações acidentais ou ilícitas de destruição, perda, alteração, comunicação ou difusão; VIII - prevenção: adoção de medidas para prevenir a ocorrência de danos em virtude do tratamento de dados pessoais; IX - não discriminação: impossibilidade de realização do tratamento para fins discriminatórios ilícitos ou abusivos; X - responsabilização e prestação de contas: demonstração, pelo agente, da adoção de medidas eficazes e capazes de comprovar a observância e o cumprimento das normas de proteção de dados pessoais e, inclusive, da eficácia dessas medidas.”(Brasil, 2018, art. 6º)

Fica estabelecido aos agentes de tratamento que para a realização do armazenamento e tratamento de dados pessoais, o controlador deve estabelecer critérios claros de finalidade e necessidade, adequar o tratamento às finalidades estabelecidas, garantir ao titular o acesso aos seus dados de forma objetiva, além da possibilidade de solicitar correções e exclusões de seus

dados pessoais e garantir medidas de segurança, prevenção contra falhas responsabilização e prestação de contas por vazamentos que possam ocorrer.

Além dos critérios anteriormente estabelecidos, o armazenamento dos dados pessoais poderá ocorrer somente caso o controlador siga e cumpra os requisitos definidos no Art. 7º da Lei Geral de Proteção de Dados:

> "I - mediante o fornecimento de consentimento pelo titular; II - para o cumprimento de obrigação legal ou regulatória pelo controlador; III - pela administração pública, para o tratamento e uso compartilhado de dados necessários à execução de políticas públicas previstas em leis e regulamentos ou respaldadas em contratos, convênios ou instrumentos congêneres, observadas as disposições do Capítulo IV desta Lei; IV - para a realização de estudos por órgão de pesquisa, garantida, sempre que possível, a anonimização dos dados pessoais; V - quando necessário para a execução de contrato ou de procedimentos preliminares relacionados a contrato do qual seja parte o titular, a pedido do titular dos dados; VI - para o exercício regular de direitos em processo judicial, administrativo ou arbitral, esse último nos termos da Lei nº 9.307, de 23 de setembro de 1996 (Lei de Arbitragem) ; VII - para a proteção da vida ou da incolumidade física do titular ou de terceiro; VIII - para a tutela da saúde, exclusivamente, em procedimento realizado por profissionais de saúde, serviços de saúde ou autoridade sanitária; (Redação dada pela Lei nº 13.853, de 2019) Vigência IX - quando necessário para atender aos interesses legítimos do controlador ou de terceiro, exceto no caso de prevalecerem direitos e liberdades fundamentais do titular que exijam a proteção dos dados pessoais; ou X - para a proteção do crédito, inclusive quanto ao disposto na legislação pertinente." (Brasil, 2018, art. 7º)

Em resumo, para realizar o tratamento é necessário que o controlador, em sua atividade, atenda um dos critérios estabelecidos na legislação, sendo os mais relevantes para o trabalho o consentimento do titular sobre o uso de seus dados pessoais, proteção de vida e tutela da saúde realizado por profissionais da saúde e serviços de saúde.

3.5 RISCOS E MEDIDAS DE MITIGAÇÃO

Para garantir a segurança dos dados pessoais, mesmo após o término do tratamento, é essencial adotar medidas técnicas e administrativas alinhadas às obrigações legais. Entre as medidas técnicas, destacam-se o uso do protocolo HTTPS, o controle rigoroso de acesso com privilégios mínimos - o usuário do sistema terá acesso aos níveis de leitura e escrita apenas - e a proteção contra acessos não autorizados, possibilitando o controle de quem usa a ferramenta e visando evitar o acesso de terceiros sem autorização. Outra medida para garantir a segurança dos dados no banco de dados é a implementação da biblioteca *django-cryptography*. A biblioteca criptografa os campos assinalados no banco de dados com uma chave segura gerada por métodos presentes na biblioteca e não permite o seu acesso por terceiros que não possuam a chave segura, garantindo a segurança sobre acessos indevidos aos dados pessoais armazenados.

Além disso, políticas de retenção e eliminação segura de dados devem ser estabelecidas, assegurando a conformidade com prazos legais. Os dados pessoais podem ser apagados

com a solicitação do paciente via o próprio chatbot, já as alterações são solicitadas diretamente à pesquisadora responsável pela ferramenta.

No âmbito administrativo, é fundamental documentar políticas de segurança, para os profissionais que vão utilizar, gerar relatórios detalhados para notificação à ANPD em caso de incidentes e conduzir auditorias regulares para identificar e corrigir vulnerabilidades. Em casos de responsabilização do controlador e operador, é exigida uma auditoria e a demonstração de conformidade, exceto em casos comprovados de culpa exclusiva de terceiros ou ausência de violação legal. Essas práticas integram-se ao ambiente *Django*, assegurando conformidade com a LGPD e mitigando riscos de tratamento inadequado.

3.6 RESPONSABILIDADES DO CONTROLADOR E DO OPERADOR

O controlador e operador são responsáveis pelos dados em tratamento, sendo obrigados a reparar o titular em razão de danos causados pelos tratamentos dos dados. Somente não serão responsabilizados em casos em que forem comprovados que os agentes não fizeram tratamentos nos dados que causaram os danos, mesmo realizando o tratamento, não houver violação da legislação ou a culpa do dano for exclusivamente de terceiros. O tratamento de dados é considerado irregular quando não é considerado o modo que o dado é utilizado, o risco do tratamento e as técnicas utilizadas.

Ao manter os dados armazenados e o tratamento dos dados ativos, o operador e controlador devem estar cientes da possibilidade da ANPD exigir a elaboração de um relatório acerca do impacto do tratamento à proteção de dados pessoais, contemplando as suas operações de tratamento. Tal relatório deverá incluir descrição dos tipos de dados coletados, da metodologia de coleta, tratamento e garantia de proteção aos dados pessoais.

Ainda na alçada do controlador, fica a ele atribuída a indicação de um encarregado pelo tratamento, a começar pelas informações de identidade e contato. As atividades deste são: lidar com a comunicação com os titulares, receber comunicados da ANPD e tomar as providências cabíveis e orientação dos funcionários e contratados a respeito das boas práticas e medidas a serem tomadas. Em relação ao operador, ele deve realizar as operações de dados de acordo com as instruções passadas pelo controlador, observando as disposições normativas estabelecidas na legislação.

4 METODOLOGIA

4.1 TECNOLOGIAS UTILIZADAS

A linguagem de programação utilizada para o desenvolvimento será o *Python*, junto do *framework Django* para a construção de um servidor web, que terá como funcionalidades iniciar o contato com o paciente, receber as suas mensagens e excluir os dados, caso seja solicitado. A escolha das tecnologias justifica-se por ser uma linguagem validada no mercado, por sua vasta gama de bibliotecas e *frameworks* que auxiliam no desenvolvimento das aplicações e na vasta documentação e conteúdos sobre a linguagem encontrados na rede.

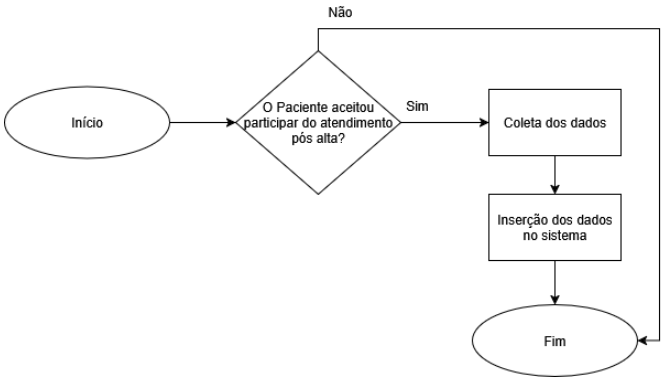
O uso do *WhatsApp* é justificável por, além de ser um dos aplicativos de mensagem instantâneas mais usados no país, possuir uma API que é capaz de integrar chatbots em sua aplicação. Para integrar o *WhatsApp* e o *Django*, a API utilizada será usada a *Evolution API*, uma solução open source gratuita responsável pelo envio de mensagens tanto do *Django* ao *WhatsApp* quanto do *WhatsApp* para o *Django*. Para sua implementação, será necessário a instalação de um objeto, responsável por recuperar as mensagens recebidas e enviar as respostas. Tais funções serão efetuadas através de um *Webhook*.

Para a criação do chatbot, será usada a plataforma n8n, uma plataforma de automação com o uso de Inteligência Artificial, possuindo conectividade com diversas plataformas - *Google Docs*, *Twilio* e *Evolution API*. É possível automatizar tarefas repetitivas usando as ferramentas disponíveis, sendo ideal para a implementação do fluxograma de interação idealizado para o chatbot. Possui suporte tanto para aplicações locais e ambientes de teste quanto para hospedagens na nuvem, em ambiente de produção (n8n.io). Para fins de contextualização do chatbot na conversa, será utilizada uma planilha eletrônica na plataforma *Google Docs* contendo os seguintes campos: numero, message_count, dia, resposta1, resposta2 e resposta3.

4.2 MODELAGEM DO FLUXO DE CONVERSA

Para garantir o correto funcionamento do chatbot, é necessário definir o fluxo de como a ferramenta funcionará. A primeira etapa é a coleta de dados dos pacientes que estão dispostos a participar da pesquisa.

Figura 1 – Fluxograma - Etapa 1.

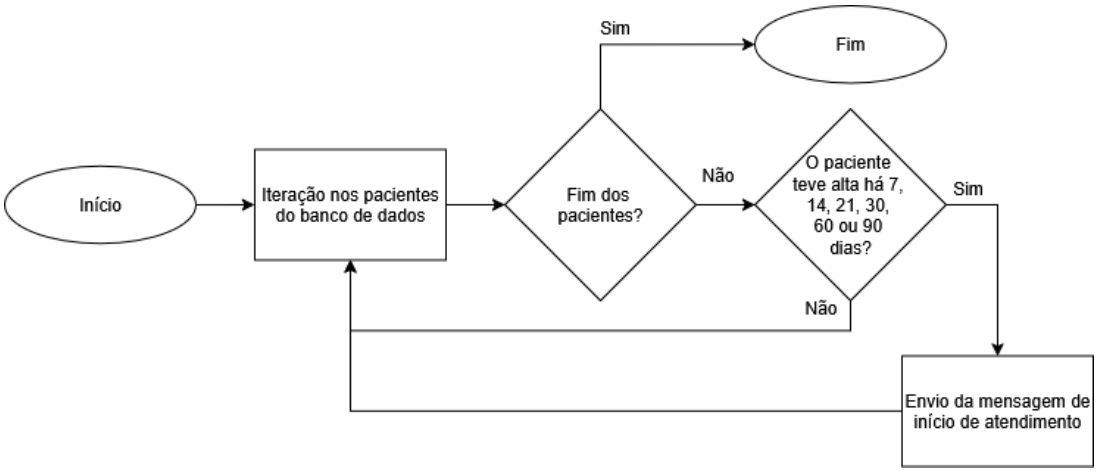


Fonte: Elaborado pelo autor (2025).

Nesta etapa, caso o paciente aceite participar do atendimento pós alta, a pesquisadora colherá a anuência por meio de um termo de consentimento de uso dos dados pessoais e em seguida, o nome, telefone e a data de alta do paciente serão coletados. Então, os dados serão adicionados ao sistema, junto da cópia digitalizada do termo de consentimento no banco de dados do sistema.

A segunda etapa é verificação dos pacientes que estão aptos ao atendimento, filtrando os dados encontrados no servidor *Django*.

Figura 2 – Fluxograma - Etapa 2.



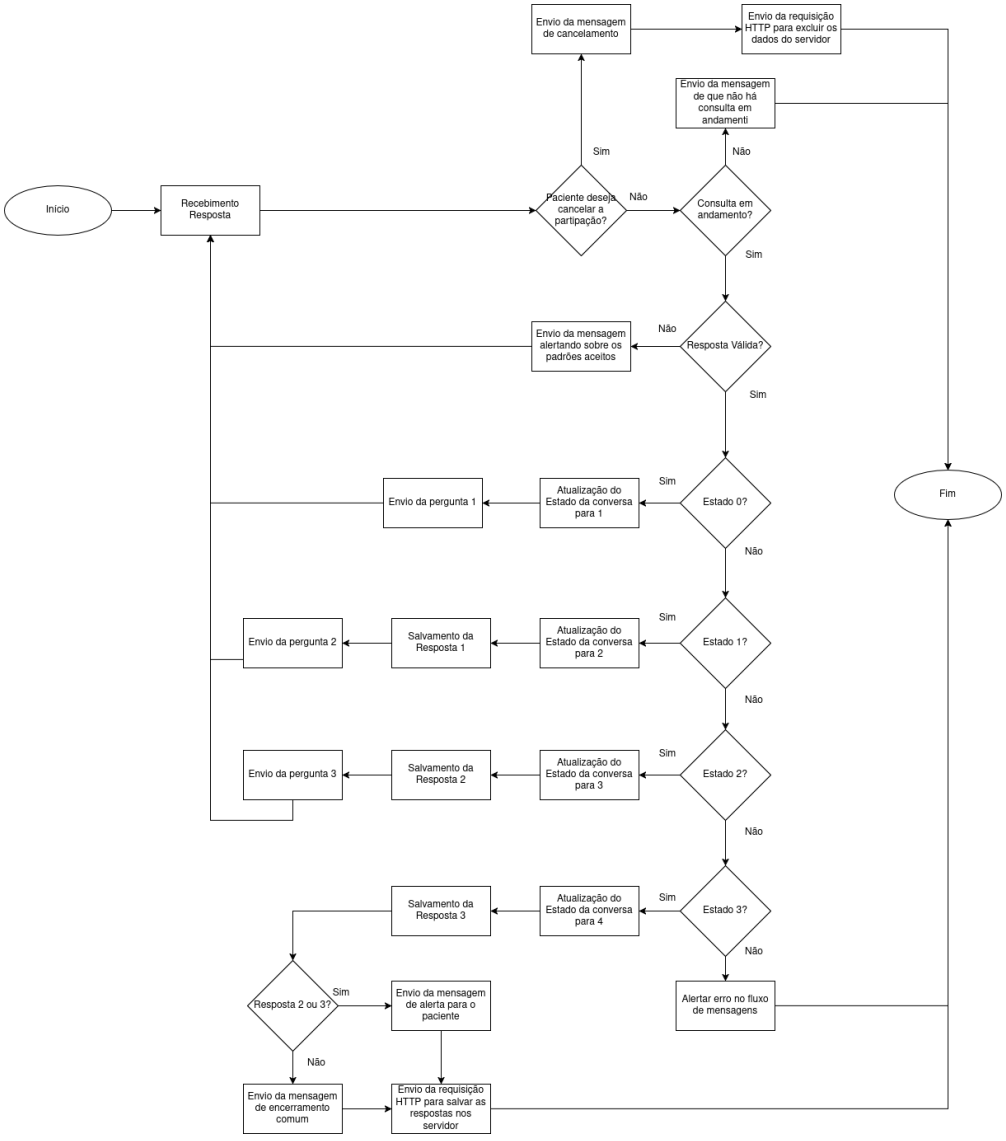
Fonte: Elaborado pelo autor (2025).

O sistema busca em seu banco de dados pelo registro dos pacientes, serão selecionados os pacientes cuja alta ocorreu a 07, 14, 21, 30, 60 e 90 dias. Após essa identificação, o sistema começa a interação com o paciente. A etapa encerra-se quando o sistema varre todos os pacientes.

Para começar o atendimento, o sistema envia uma mensagem de saudação ao paciente. Então, o sistema começa o atendimento, que consiste em realizar três perguntas: “O local da cirurgia está quente ou avermelhado?”, “Você percebeu abertura dos pontos cirúrgicos (deis-

cência) na área operada?" e "Você observou saída de pus ou secreção pela ferida cirúrgica?". O paciente responde com "SIM" ou "NÃO" e após respondê-las, o sistema irá guardar as mensagens no seu banco de dados. Caso o paciente responda "SIM" em 1 ou mais perguntas, o sistema enviará uma mensagem de alerta: "Entraremos em contato para orientações adicionais. Enquanto isso, procure assistência médica no posto de saúde mais próximo.". Caso contrário , retornará a mensagem "Que bom que tudo está indo bem! Continuaremos acompanhando você nos próximos dias. Lembre-se de seguir as orientações pós-operatórias." A pesquisadora receberá o relatório da consulta, constando as respostas positivas e negativas.

Figura 3 – Fluxograma - Etapa 3.



Fonte: Elaborado pelo autor (2025).

Como visto no fluxograma, existem dois padrões de encerramento do atendimento, sendo o primeiro uma mensagem normal caso as duas perguntas cruciais sejam respondidas como "NÃO", sendo ela "Que bom que tudo está indo bem! Continuaremos acompanhando você nos próximos dias. Lembre-se de seguir as orientações pós-operatórias". O outro padrão

se encontra quando uma das respostas às perguntas chave seja “SIM”, o padrão enviado será “Entraremos em contato para orientações adicionais, enquanto isso, procure assistência médica no posto de saúde mais próximo”.

4.3 ESTRATÉGIAS DE COLETA E ANÁLISE DE DADOS

A ferramenta utilizada para a coleta dos dados será o *WhatsApp*, as respostas às perguntas feitas pelo agente serão encaminhadas para o servidor *backend*, onde serão tratadas. As respostas vão gerar uma instância no banco de dados do sistema, contendo informações de nome, número de telefone, data de alta, data da consulta e as respostas pelo paciente informadas. Caso necessário, essa instância será enviada para a pesquisadora auxiliar na continuidade do atendimento ao paciente.

4.4 REGRAS DE DECISÃO PARA IDENTIFICAR AGRAVAMENTOS

Como definido pela pesquisadora em reunião de identificação das regras de funcionamento e levantamento de requisitos, as decisões para identificação de agravamentos no quadro do paciente são as questões do rompimento do ponto ou de surgimento de secreções. Caso a segunda ou a terceira respostas sejam positivas, sendo ela em relação à uma das perguntas críticas citadas anteriormente, o agente irá alertar a pesquisadora sobre o paciente em questão, sugerindo a verificação do sítio cirúrgico em consulta presencial.

5 DESENVOLVIMENTO

5.1 DESENVOLVIMENTO DO CHATBOT

A fim de implementar o agente conversacional, é necessário a instalação da biblioteca NPX, um complemento ao NPM da linguagem de programação *JavaScript*. O n8n está presente na NPX, sendo necessário apenas utilizar o comando npx n8n para utilizar os seus recursos. Dentro do nbn, é necessário criar um ambiente para desenvolver o fluxo de trabalho que o chatbot seguirá.

5.1.1 Ponto de entrada e tratamento inicial

O ponto de partida do fluxo é um nó do tipo *Webhook*. Este nó é responsável por receber as mensagens enviadas do paciente. Ele disponibiliza um URL que aponta para o servidor em que a ferramenta está hospedada, devendo ser indicado na instância da *Evolution API*. Após o recebimento, as informações são tratadas por um nó do tipo Set nomeado **Formatação da resposta do WhatsApp**, ele extrai os dados relevantes do *payload*, como o conteúdo da mensagem e o número do remetente, e os atribui em variáveis locais a fim de facilitar sua utilização em outras etapas do fluxo.

5.1.2 Cancelamento da partipação

Após a formatação dos dados, o fluxo realiza uma verificação acerca da mensagem recebida em nó *If* intitulado de **Cancelar a participação?**, a mensagem extraída do *payload* é comparada com a chave CANCELAR. Caso o nó retorne verdadeiro, o fluxo segue para uma sequência de finalização da participação do paciente na pesquisa. Primeiramente, um nó da *Evolution API* envia uma mensagem de confirmação ao paciente: **Seus dados serão corretamente excluídos**. Em seguida, um nó HTTP *Request* é acionado, enviando uma requisição do tipo *DELETE* ao servidor principal. Esta requisição contém o número do paciente como parâmetro, instruindo o *backend* a localizar e excluir todos os registros associados a ele, tanto na base de dados principal quanto na planilha de contexto. Caso o fluxo retorne falso, indica que o paciente não solicitou o cancelamento, então o fluxo segue para a próxima etapa.

5.1.3 Validação de contexto e entrada

Para a próxima etapa, o fluxo é direcionado a um nó *If*, nomeado de Consulta em Andamento?, caso nenhuma linha for retornada, saída *False*, significa que o paciente não está em um ciclo de perguntas ativo. O fluxo então envia a mensagem **Nenhuma consulta identificada em andamento. Para demais dúvidas, dirija-se à unidade de atendimento à saúde mais próxima** e é encerrado. Se uma consulta for encontrada, saída *True*, o fluxo prossegue para a validação da resposta do paciente.

A validação da resposta é feita via outro nó *If* nomeado **Verificar se as respostas são válidas**, validando a conformidade da mensagem recebida com os padrões esperados SIM e NÃO. Em caso negativo, a mensagem de instrução **Responda com SIM ou NÃO** é enviada, e o fluxo aguarda uma nova interação. Em caso positivo, o fluxo segue para a próxima etapa.

5.1.4 Estados da consulta

O fluxo caminha para o carregamento dos dados da planilha de contexto e os salva em variáveis locais. Então, o próximo nó é um *Switch* que tem o papel de direcionar o atendimento para o estado correspondente na planilha de contexto. O primeiro é o estado 0 e indica o início da consulta. O sistema envia a primeira pergunta: **O local da cirurgia está quente ou avermelhado? Responda com SIM ou NÃO** e atualiza o estado do paciente para 1 na planilha de contexto.

O estado 1 é acionado após o recebimento da primeira resposta. O sistema armazena a Resposta 1, envia a segunda pergunta **Você percebeu abertura dos pontos cirúrgicos (deiscência) na área operada? Responda com SIM ou NÃO** e atualiza o estado para 2.

O estado 2 é acionado após o recebimento da segunda resposta. O sistema armazena a Resposta 2, envia a terceira e última pergunta **Você observou saída de pus ou secreção pela ferida cirúrgica? Responda com SIM ou NÃO** e atualiza o estado para 3.

Por fim, o último estado é o 3, recebe a terceira resposta e inicia a lógica de conclusão do atendimento. Neste estado ocorre também a verificação das respostas com o uso de um nó *If* chamado **Pergunta 2 ou Pergunta 3 iguais à 'SIM'?**, caso o retorno seja *True* o chatbot envia uma mensagem de orientação ao paciente **Por favor, encaminhe-se à unidade de atendimento à saúde mais próxima**. Caso o retorno seja *False*, indica que não foram detectados sinais de alerta imediatos. O sistema envia uma mensagem de encerramento padrão: **Tudo certo! Lembre-se de seguir as recomendações pós-cirúrgicas**. Ao servidor receber os dados da consulta, ele dispara uma notificação de alerta para a pesquisadora responsável caso a Pergunta 2 ou 3 sejam SIM.

5.1.5 Persistência dos dados

Após a análise das respostas, ambos os caminhos convergem para uma etapa final. Um nó HTTP *Request* é executado para enviar todas as respostas coletadas ao servidor principal, que as armazena de forma segura e estruturada no banco de dados. Esta mesma requisição também comanda a exclusão da linha correspondente na planilha de contexto, finalizando o ciclo da consulta.

5.2 DESENVOLVIMENTO DA API

Para promover a comunicação entre a ferramenta desenvolvida e o servidor de suporte, foi escolhida uma aplicação intermediária chamada *Evolution API*. É uma solução de có-

digo aberto que fornece uma interface para realizar o envio e uma porta para receber mensagens diretamente do *WhatsApp*. Esta ferramenta foi escolhida por ser de código aberto, permitindo autonomia ao projeto por não depender de ferramentas pagas, e por fornecer uma grande personalização no serviço oferecido, permitindo muitas possibilidades de customização.

5.2.1 Configuração do ambiente

A implementação da *Evolution API* começa com o download do repositório fornecido no *github* em seu diretório apropriado. A segunda etapa é a configuração do arquivo de variáveis ambiente da aplicação, o arquivo .env. As variáveis a serem configuradas são a *AUTHENTICATION_API_KEY* - definição de um *token* seguro sendo enviado como parâmetro das requisições feitas à API -, *DATABASE_CONNECTION_URI* - parâmetro padrão para direcionar a API para a instância correta onde os dados das operações serão salvos - e as variáveis relacionadas ao Redis - um banco de dados em memória para o gerenciamento de cache e informações dinâmicas - *REDIS_HOST*, *REDIS_PORT*, *REDIS_PASS* e *REDIS_DB*. Tendo feita a parametrização, a API é executada em um *container* Docker em um ambiente isolado e consistente.

5.2.2 Instanciação e sincronização

A próxima etapa é a criação de uma instância dentro da *Evolution API*. A instância é responsável por gerenciar a comunicação de um número sincronizado no *WhatsApp* com o chatbot e servidor. Após sua criação, é necessário sincronizar a conta do *WhatsApp* com um dispositivo móvel via *QR code*, similar ao sincronizar uma conta em um dispositivo web.

5.2.3 Configuração do webhook

A etapa final para a configuração da API é a criação do *Webhook* responsável por enviar as mensagens recebidas ao chatbot. A ferramenta disponibiliza de uma série de ferramentas diferentes para a configuração robusta do *endpoint*, porém, para fins de simplicidade, a solução usará apenas a função *MESSAGE_UPSERT*. O evento é disparado sempre que o número recebe ou envia uma mensagem, a principal vantagem de usá-lo é *payload* robusto e detalhado que é enviado ao chatbot, permitindo a extração de diversas informações para o seu funcionamento.

O último campo a ser preenchido é a URL de destino do envio de mensagens, ele é configurado para o destino do servidor em que o chatbot está hospedado e pronto para o receber e processar as mensagens. Assim, a API está corretamente configurada e operacional.

5.3 DESENVOLVIMENTO DO SERVIDOR DJANGO

A fim de atender os requisitos de segurança para o armazenamento de dados pessoais dos pacientes e suas respostas em cada consulta efetuada e controle dos fluxos de interação com

o chatbot, será necessário a implementação de uma solução em aplicação web. A sua construção será trabalhada no decorrer deste capítulo.

5.3.1 Estruturação do ambiente virutal

O primeiro passo é a criação de uma pasta para conter os arquivos do projeto e iniciar um ambiente virtual utilizando-se da linguagem de programação Python. O ambiente é criado utilizando o comando *python -m venv TCC*. A criação do ambiente virtual encapsula as dependências e bibliotecas utilizadas no projeto, evitando conflitos com outras dependências e aplicações presentes no sistema. Seu ativamento é feito via *./venv/bin/activate*. Após sua ativação, a instalação do Django é feita com o comando *pip install django==4.2.9*. O uso de uma versão anterior do Django é justificada para o uso posterior de uma biblioteca de criptografia do banco de dados da aplicação, que sofre conflitos na versão 5 do *framework*.

5.3.2 Construção do servidor django

O início da estrutura em Django é feita pela execução do comando *django-admin startproject TCC*, criando a estrutura fundamental e padrão para as aplicações compostas pelos arquivos *manage.py* - um utilitário de linhas de comandos permitindo acessar diversas facetas do Django -, o diretório TCC sendo o pacote do projeto, o arquivo *settings.py*- onde ocorre as configurações centrais do servidor, definindo parâmetros como banco de dados, aplicações instaladas e outros - e o *urls.py* - arquivo onde as rotas para os demais aplicativos são definidas -, ambos dentro do pacote TCC.

Antes de iniciar o servidor e verificar o funcionamento do servidor, dentro do arquivo *settings*, deve-se alterar os parâmetros *TIME_ZONE* e *LANGUAGE_CODE* a fim de corrigir a localização do servidor e de seu fuso horário e alterar a linguagem do servidor de inglês para português.

A etapa final da configuração do *settings.py* é a mudança das variáveis referentes ao banco de dados do servidor. Para a utilização do PostgreSQL, será necessário a configuração de 6 variáveis de ambiente, sendo elas: *ENGINE*, *NAME*, *USER*, *PASSWORD*, *HOST* e *PORT*. Dessa maneira, o servidor está pronto para ser executado e prosseguir com seu desenvolvimento.

5.3.3 Desenvolvimento da aplicação chatbot

Dentro do servidor construído, uma aplicação a fim de modularizar as funcionalidades do chatbot deve ser criada a partir do comando *python manage.py startapp chatbot*. Dessa forma, este módulo tem como função encapsular as operações relacionadas à gestão de pacientes e interação via chatbot, sendo essas as operações relacionadas ao cadastro, alteração e exclusão de dados, envio de mensagens e controle do fluxo de consultas.

5.3.4 Construção da base de dados

A próxima etapa do desenvolvimento da aplicação em questão é a construção dos modelos que serão utilizados na criação do seu banco de dados. Serão definidos os modelos de Pacientes, destinado à armazenar as informações cadastrais dos participantes e Respostas, destinado à armazenar as interações de cada consulta.

O modelo Pacientes contém os campos de nome do paciente - este campo é criptografado utilizando a biblioteca *django-cryptography* -, número do *WhatsApp* - o outro campo do modelo que é criptografado utilizando a biblioteca *django-cryptography* -, o termo de consentimento - espaço para armazenar o link do termo digitalizado via drive -, "criado em" e "modificado em". Já o modelo Respostas contém os campos id da conversa do *WhatsApp*, nome do paciente, número do paciente, dias do pós alta, respostas 1,2 e 3 - todos os citados criptografados - e o campo criado em.

Após a criação dos modelos, para efetivar a sua criação, deve-se utilizar os comandos *python manage.py makemigrations* e *python manage.py migrate*, desta maneira, as tabelas são criadas no banco de dados da aplicação.

5.3.5 Implementação da lógica da aplicação chatbot

As implementações lógicas necessárias para o pleno funcionamento da ferramenta foram implementadas no arquivo *views.py*, dentro da aplicação chatbot.

A primeira função implementada foi a função do envio programado de mensagens, que consiste em uma sequência de 5 passos. O primeiro passo é a iteração sobre todos os pacientes cadastrados no banco de dados do modelo Pacientes, o segundo é calcular para cada paciente a diferença entre a data atual e a data em que o paciente recebeu alta. O terceiro passo é verificar se o paciente encontra-se no 7º, 14º, 21º, 30º, 60º ou 90º dia pós-alta. O quarto passo é verificar a existência de uma consulta em aberta na planilha de contexto, caso o paciente esteja em um dos dias pós-alta definidos. O quinto e último passo consiste na criação da linha de consulta - caso não haja uma consulta em andamento - e aciona a função para enviar a requisição à *Evolution API*, que encaminha a mensagem ao paciente, iniciando a consulta.

Em conjunto da função esplanada, foi implementada outra que faz a solicitação para a *Evolution API*. As informações dos pacientes são enviadas como parâmetros e são utilizadas para a construção de uma mensagem personalizada para o envio ao paciente. As informações a serem enviadas são a URL da *Evolution Api*, o *payload* - número e a mensagem - e os *headers* - compostos pela *apikey* de autenticação e o *content-type*.

Outra função implementada é a responsável por receber os dados das consultas realizadas e salvá-los no banco de dados como modelo Respostas. A função recebe uma requisição *POST* e o seu conteúdo é desserializado, os atributos da requisição são guardados como um modelo Respostas no banco de dados. Caso a resposta 1 ou resposta 2 correspondam ao padrão "SIM", uma mensagem é disparada para o responsável pela consulta, solicitando atenção espe-

cial ao paciente que relatou algum problema no pós alta. Por fim, utiliza o número do paciente para filtrar a linha presente na tabela de contexto e a exclui, encerrando o atendimento.

A última função implementada permite que o usuário cancele sua participação no atendimento online e permite a exclusão de seus dados pessoais do banco de dados do servidor. Ela recebe uma requisição *DELETE* com seu corpo sendo composto por um número de celular enviado via *Evolution API* diretamente e filtra todos os registros dos modelos Pacientes e Respostas, incluindo-os em seguida. Também verifica se há consultas em aberto na tabela de contexto e as exclui idem.

Por fim, a última etapa da configuração é adicionar os dois modelos, Pacientes e Respostas, na página de administração do Django. Dessa forma, será possível cadastrar os pacientes e gerenciar ambos os modelos, além de permitir alterações e deleção manual dos pacientes, caso seja solicitado.

6 TESTES E RESULTADOS

6.1 TESTES E RESULTADOS DO CHATBOT

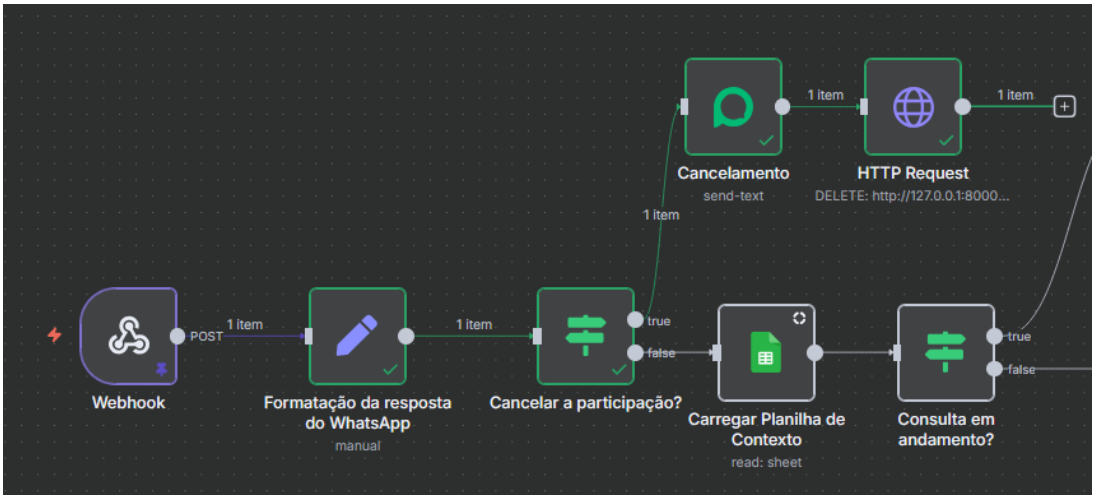
A presente seção é dedicada à execução de testes e apresentação dos resultados sobre a ferramenta. Existem três subsistemas para testar o funcionamento da ferramenta e serão explorados o teste e seu resultado.

6.1.1 Teste de cancelamento da participação

O primeiro teste a ser realizado é o cancelamento da participação do paciente via mensagem direta para o chatbot. Para tal verificação, será efetuado a inserção de dados de teste no chatbot para verificar a funcionalidade. Começando pelo chatbot, será fixado um *payload* fictício com o campo de *conversation* definido como "cancelar". Será verificado se o fluxo é capaz de transformar os caracteres minúsculos em caracteres maiúsculos, se o fluxo é direcionado no *If* para o nó da *Evolution API* e por fim o ativamento da requisição HTTP para solicitar a exclusão dos dados do paciente no banco de dados.

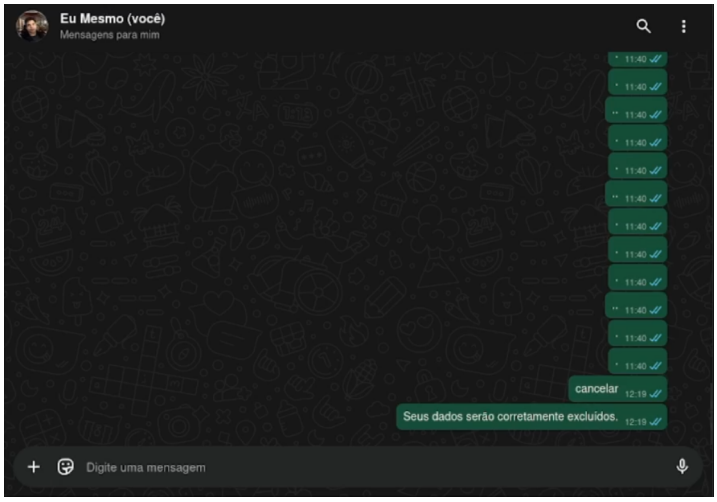
Como resultado, verificou-se que a ferramenta é capaz de alterar os caracteres para maiúsculas, o fluxo foi direcionado para os nós corretamente e a ativação da requisição HTTP foi efetuada como mostra a imagem abaixo.

Figura 4 – Resultado do teste de cancelamento.



Fonte: Elaborado pelo autor (2025).

Figura 5 – Resultado do teste de cancelamento.



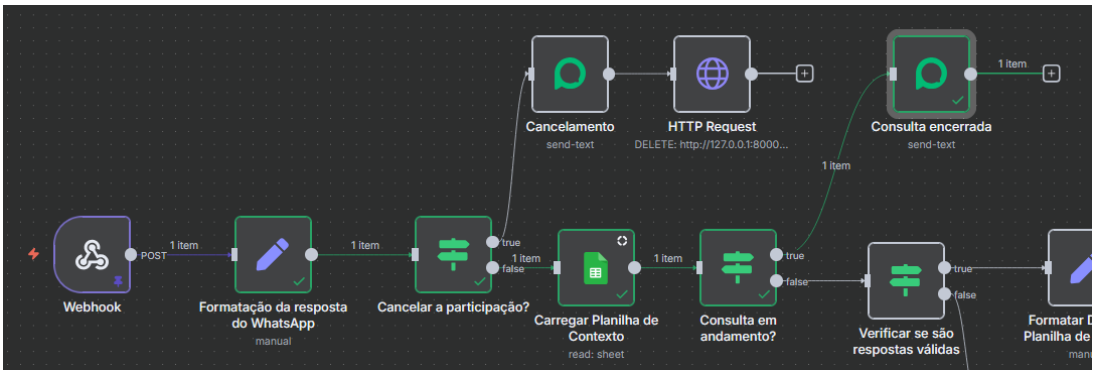
Fonte: Elaborado pelo autor (2025).

6.1.2 Teste de consulta em andamento

O próximo teste verifica o que acontece quando um paciente que não tem consultas abertas. Novamente, será fixado um *payload* fictício com sua *conversation* sendo "oi" e com o número de teste sendo '55999498268". Será analisado se a mensagem será diferenciada de "CANCELAR" e se a planilha de contexto, ao ser consultada, retornará um valor vazio, direcionando o fluxo para o nó da *Evolution API* que disparar a mensagem "Nenhuma consulta identificada em andamento. Para demais dúvidas, dirija-se à unidade de atendimento à saúde mais próxima.".

Como relatado na imagem a seguir, o fluxo fez exatamente o que foi planejado. Ao receber uma mensagem, ela passa pela verificação de cancelamento e a planilha é carregada filtrando pelo número inserido no *payload*. Ao retornar um valor nulo, o fluxo é direcionado para o nó em que envia a mensagem avisando a ausência de consultas em andamento.

Figura 6 – Resultado do teste de consulta em andamento.



Fonte: Elaborado pelo autor (2025).

Figura 7 – Resultado do teste de consulta em andamento.

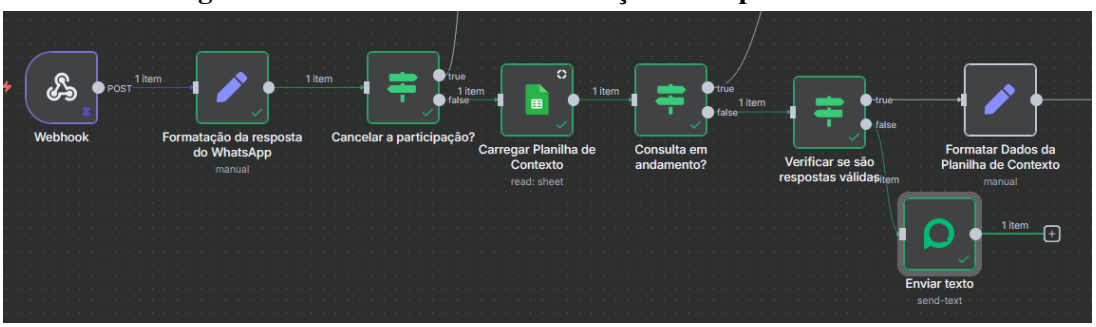
Fonte: Elaborado pelo autor (2025).

6.1.3 Teste de mensagem fora do padrão

O sistema aceita apenas 3 respostas como padrão, CANCELAR, SIM e NÃO. O usuário pode responder sem estar necessariamente em letras maiúsculas, pois o sistema converte os caracteres para o padrão. Serão definidos três *payloads* com as mensagens “nao”, “sim” e “sim, abriu o ponto”, além de uma linha na planilha de contexto com o número igual ao definido no *payload* e o estado 1. O esperado é que, após as verificações de cancelamento e de consulta em aberto, o primeiro e o terceiro *Evolution payloads* sejam rejeitados pelo chatbot e uma mensagem solicitando o envio novamente das mensagens seguindo o padrão. Já o segundo avança para o prosseguimento do fluxo.

Como esperado, ao receber uma mensagem fora do padrão o fluxo é direcionado de maneira correta, a fim de receber a mensagem no seu padrão esperado. As imagens a seguir demonstram o fluxo respectivamente dos três *payloads* ”nao”, “sim” e “sim, abriu o ponto”.

Figura 8 – Resultado do teste validação de respostas - “nao”.



Fonte: Elaborado pelo autor (2025).

Figura 9 – Resultado do teste validação de respostas - “sim”.

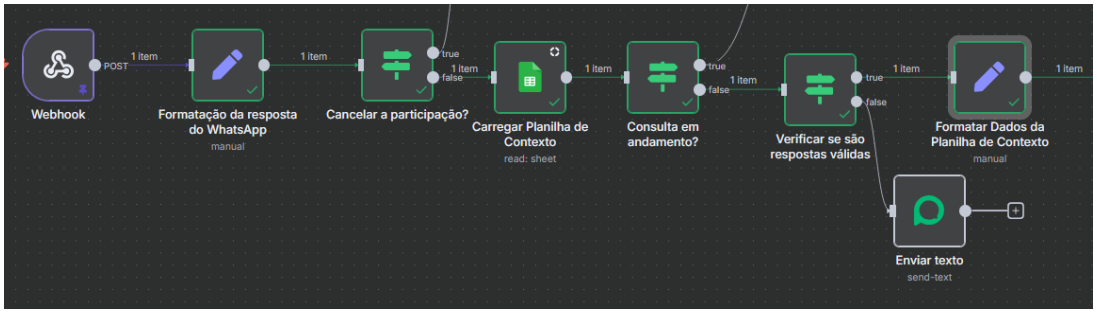
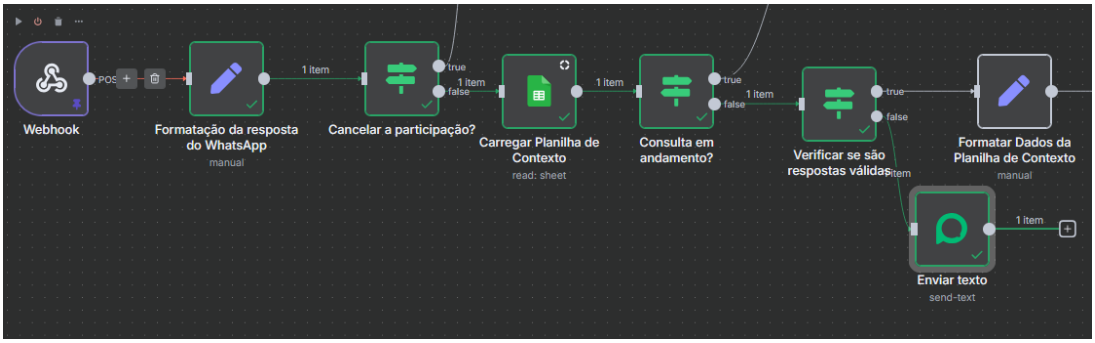


Figura 10 – Resultado do teste validação de respostas - “sim, abriu o ponto”.



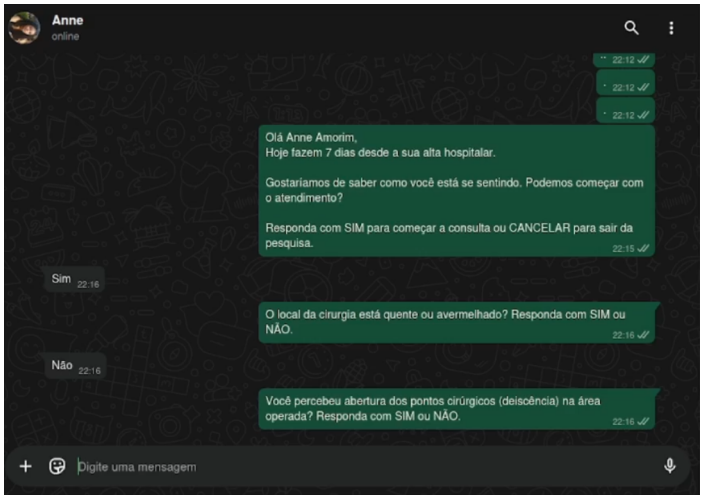
Fonte: Elaborado pelo autor (2025).

6.1.4 Teste de consulta completa

Por fim, será efetuado o teste de três cenários críticos: o primeiro, o paciente responde Não a todas as perguntas; o segundo, responde Sim à segunda pergunta; e, no terceiro, responde Sim à última pergunta.

Como esperado, o primeiro cenário retorna uma mensagem de encerramento normal, enquanto o segundo e terceiro cenários retornam uma mensagem de encaminhamento dos pacientes para a unidade de atendimento mais próxima.

Figura 11 – Cenário 1



Fonte: Elaborado pelo autor (2025).

Figura 12 – Cenário 1

Fonte: Elaborado pelo autor (2025).

Figura 13 – Cenário 2

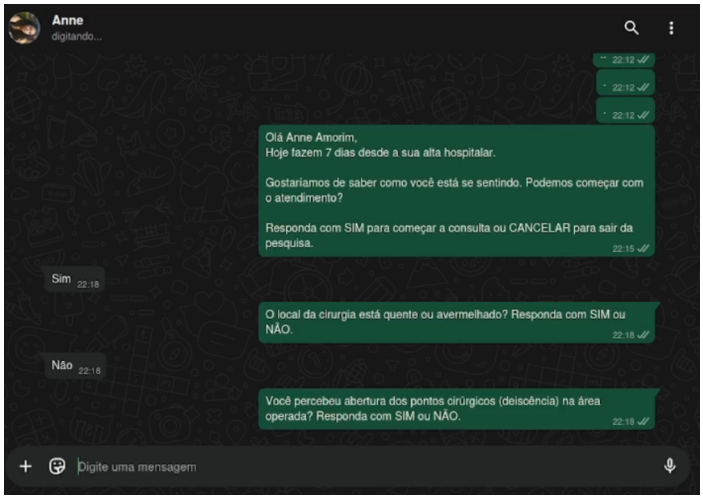


Figura 14 – Cenário 2

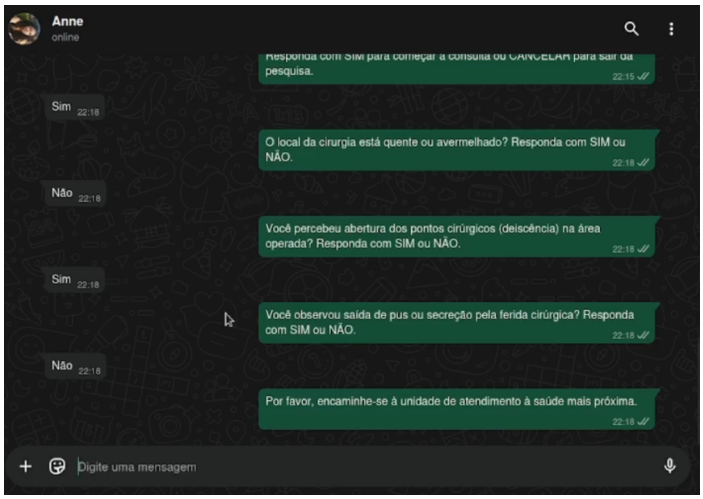
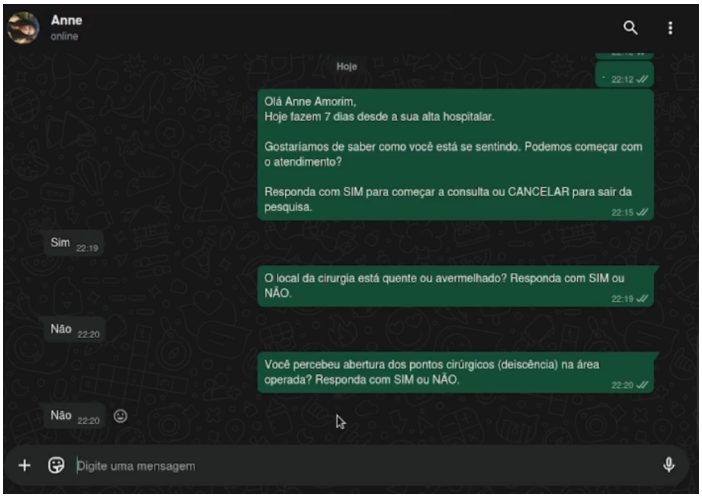
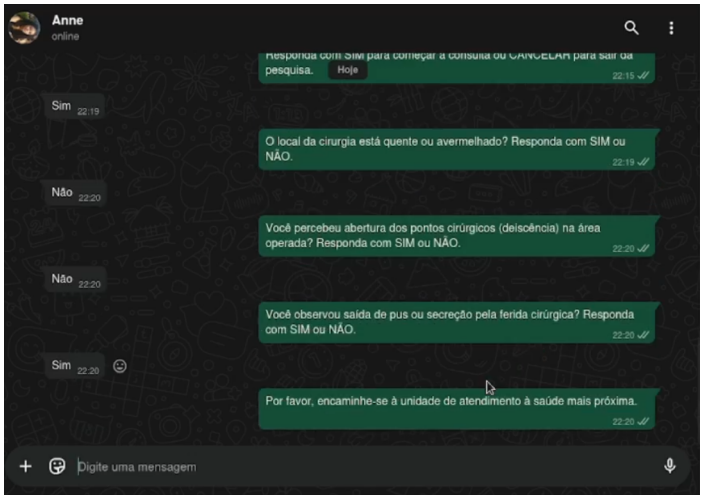


Figura 15 – Cenário 3



Fonte: Elaborado pelo autor (2025).

Figura 16 – Cenário 3



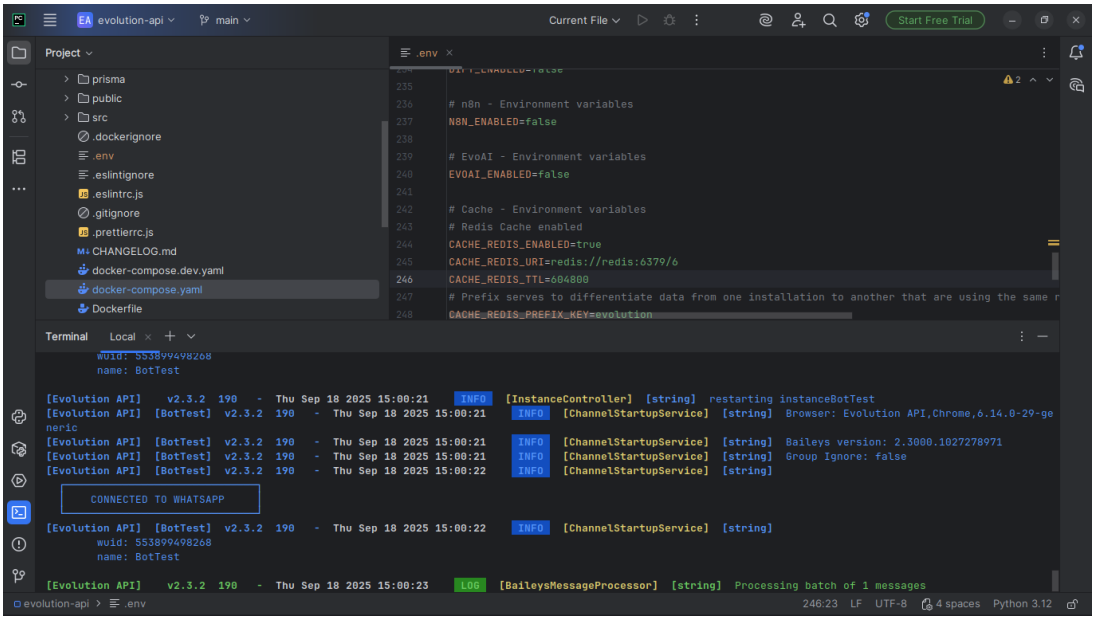
Fonte: Elaborado pelo autor (2025).

6.2 TESTES E RESULTADOS DA API

A fim de testar a *Evolution API*, será feita a conexão, envio de mensagens, recepção de mensagens e analisar seus *logs*.

A primeira etapa é a conexão com o número de *WhatsApp* escolhido. Ao se conectar, será gerado um *log* no *container* onde a aplicação estará em funcionamento.

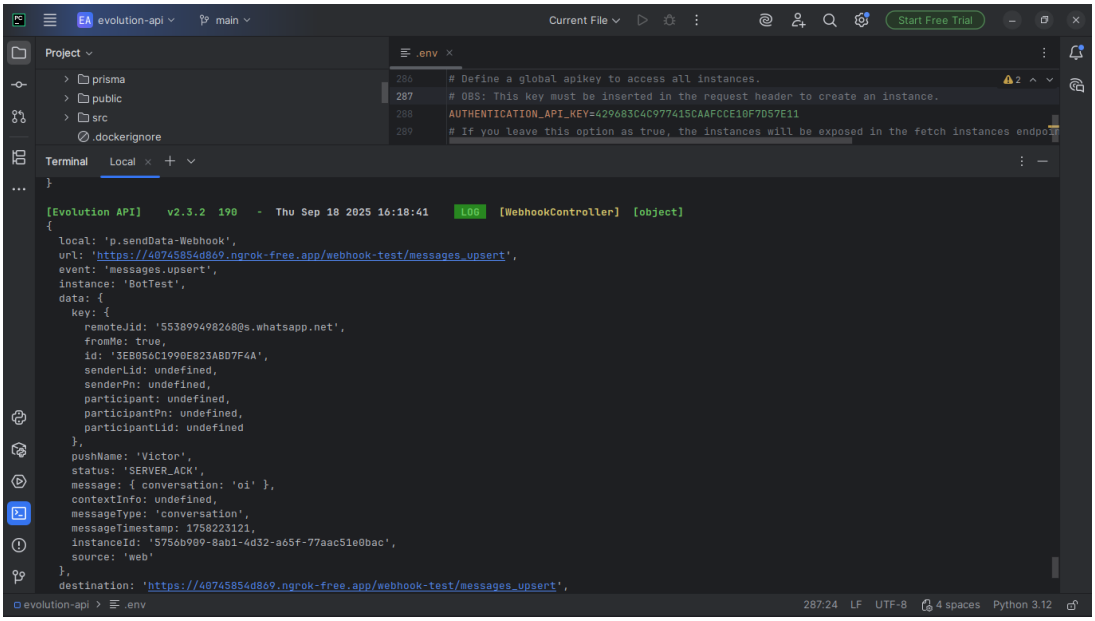
Figura 17 – API no ar e conectada ao WhatsApp.



Fonte: Elaborado pelo autor (2025).

A segunda etapa é testar o envio de mensagens. Ao fazê-lo, um *log* será gerado no *container* com o *payload* e as informações de sucesso ou não.

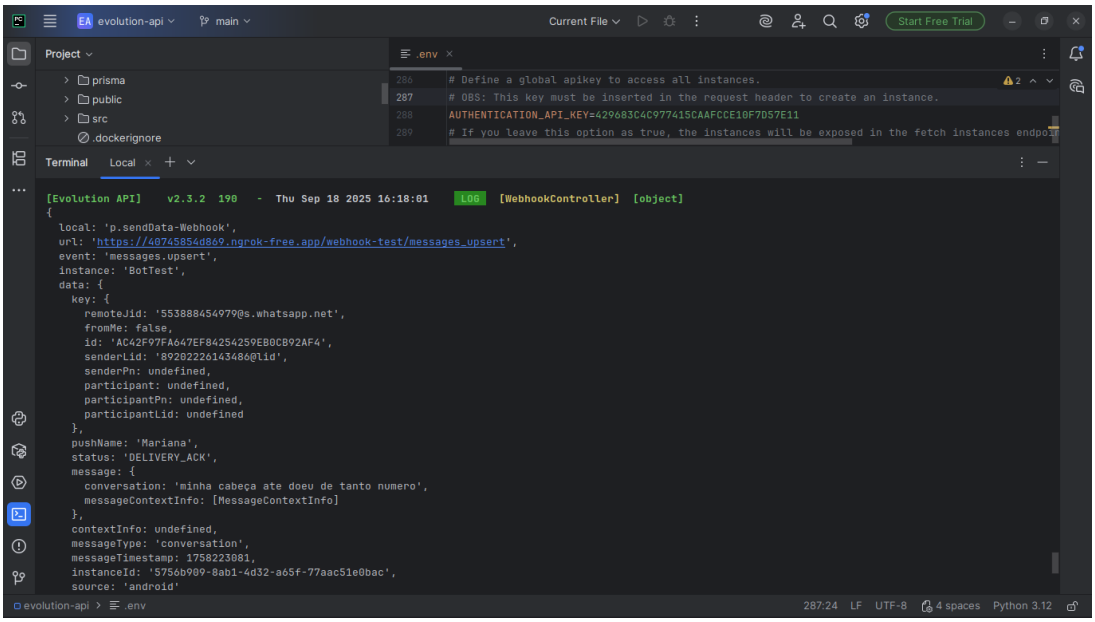
Figura 18 – Envio de mensagem funcionando corretamente.



Fonte: Elaborado pelo autor (2025).

A última etapa é testar a recepção de mensagens. Ao receber uma nova mensagem, um *payload* é gerado no *container* da API.

Figura 19 – Recebimento de mensagem funcionando corretamente.



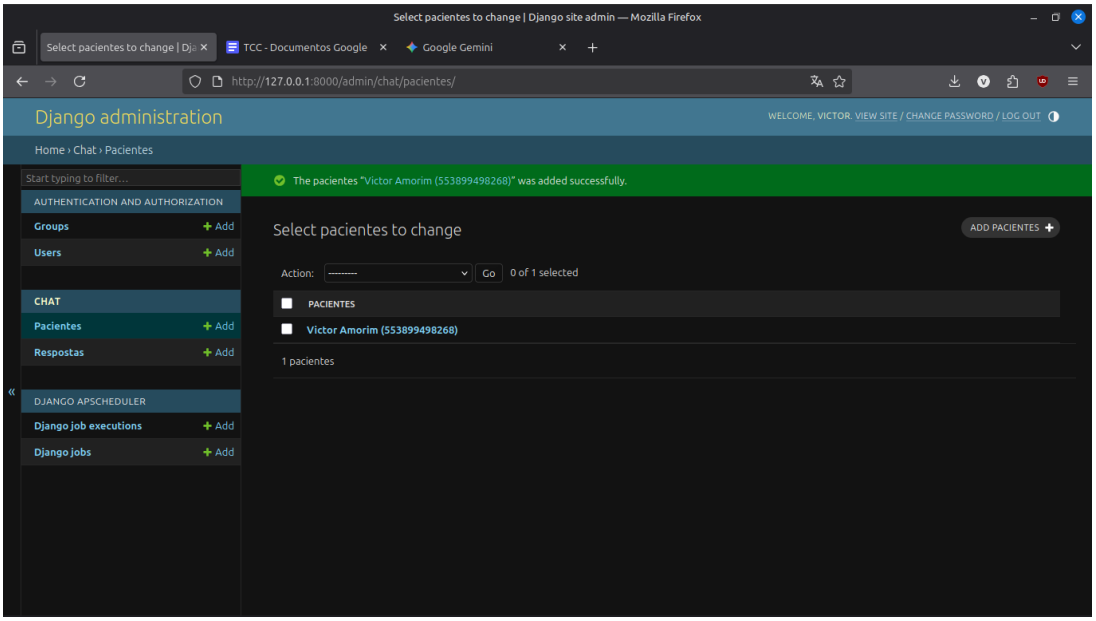
Fonte: Elaborado pelo autor (2025).

6.3 TESTES E RESULTADOS DO SERVIDOR DJANGO

6.3.1 Testes da página de administração

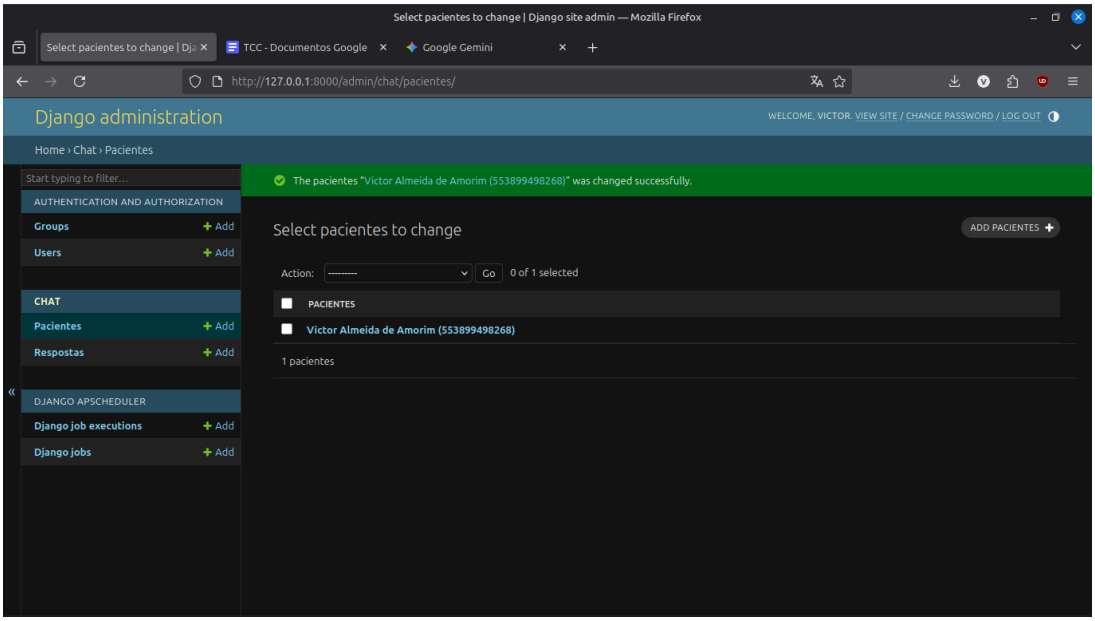
Começando a validação do servidor Django, serão executadas as operações de cadastro (*CREATE*), exclusão (*DELETE*) e atualização (*UPDATE*) de ambos os modelos, Pacientes e Respostas. O êxito das operações será verificado nas respostas retornadas pelo servidor.

Figura 20 – Criação de uma instância do modelo Pacientes.



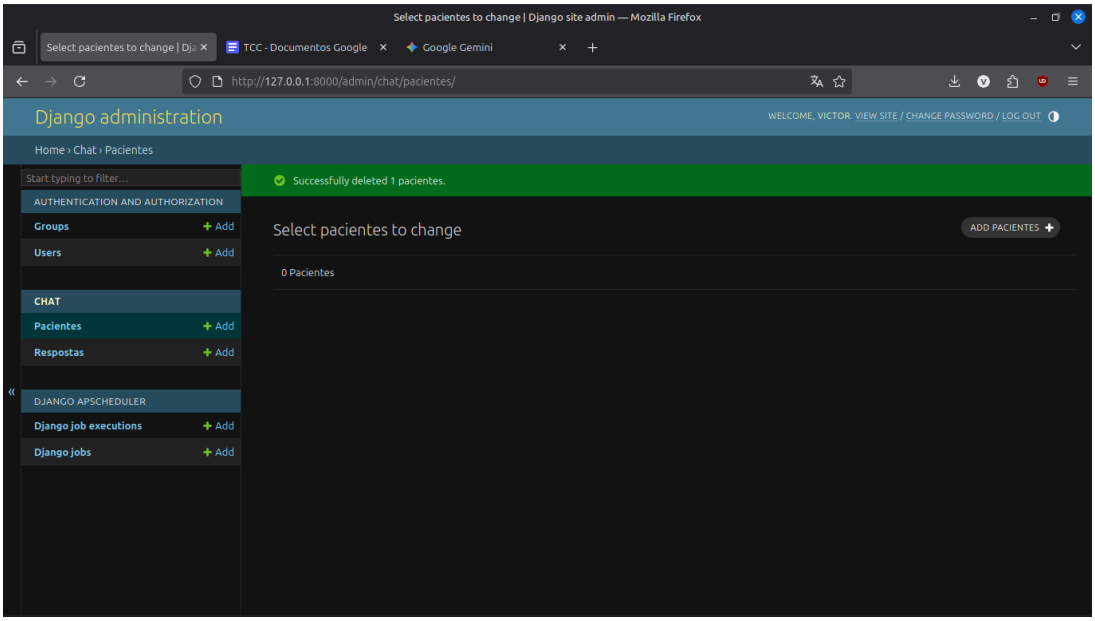
Fonte: Elaborado pelo autor (2025).

Figura 21 – Atualização de uma instância do modelo Pacientes.



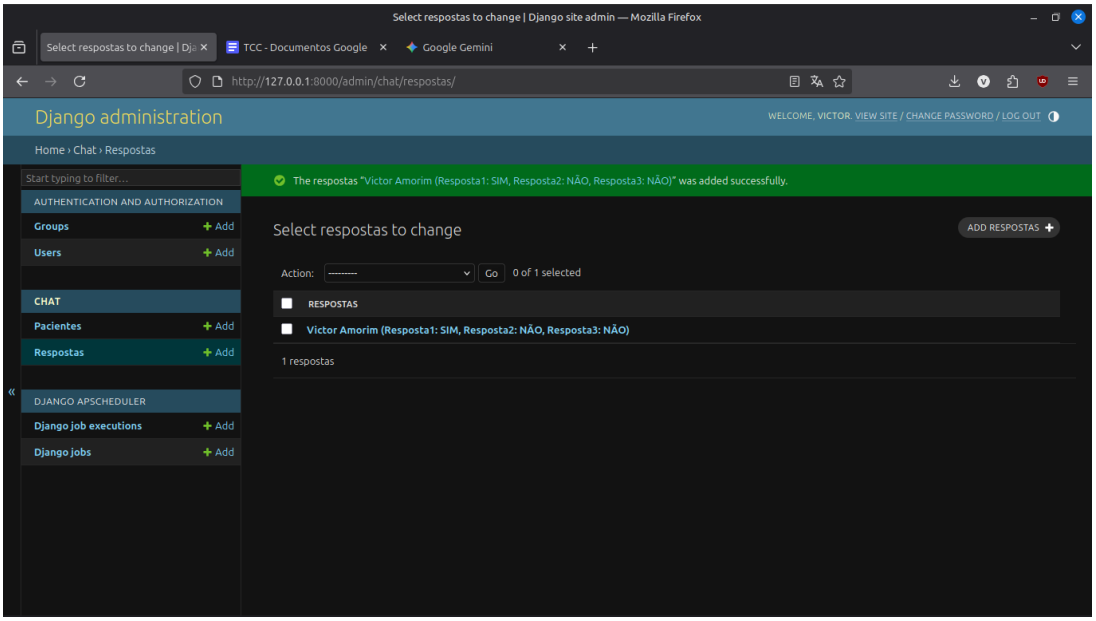
Fonte: Elaborado pelo autor (2025).

Figura 22 – Exclusão de uma instância do modelo Pacientes.



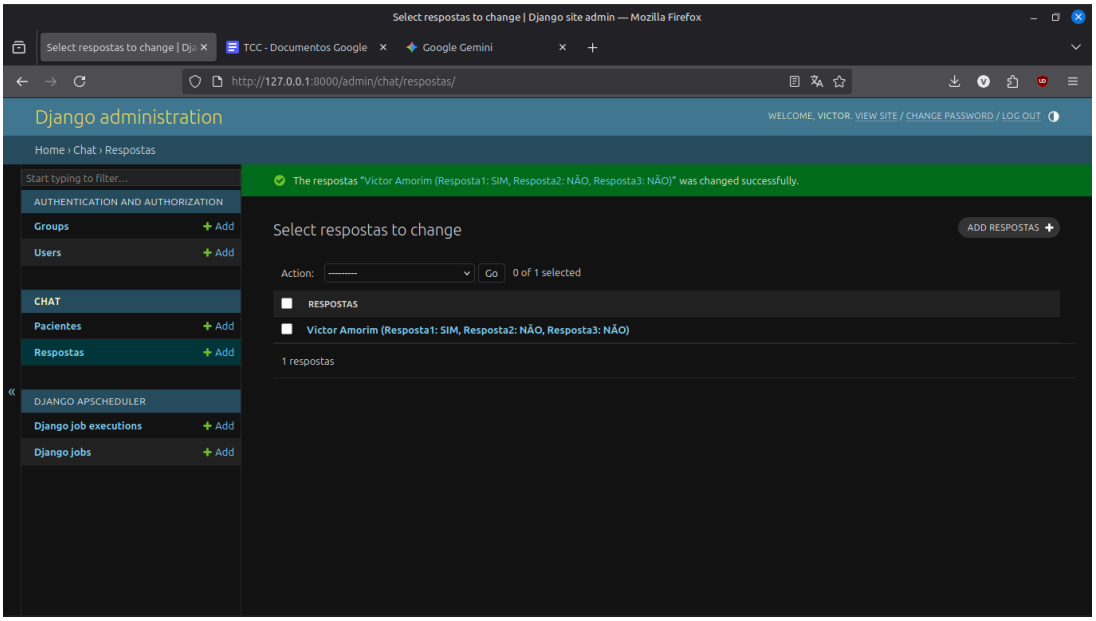
Fonte: Elaborado pelo autor (2025).

Figura 23 – Criação de uma instância do modelo Respostas.



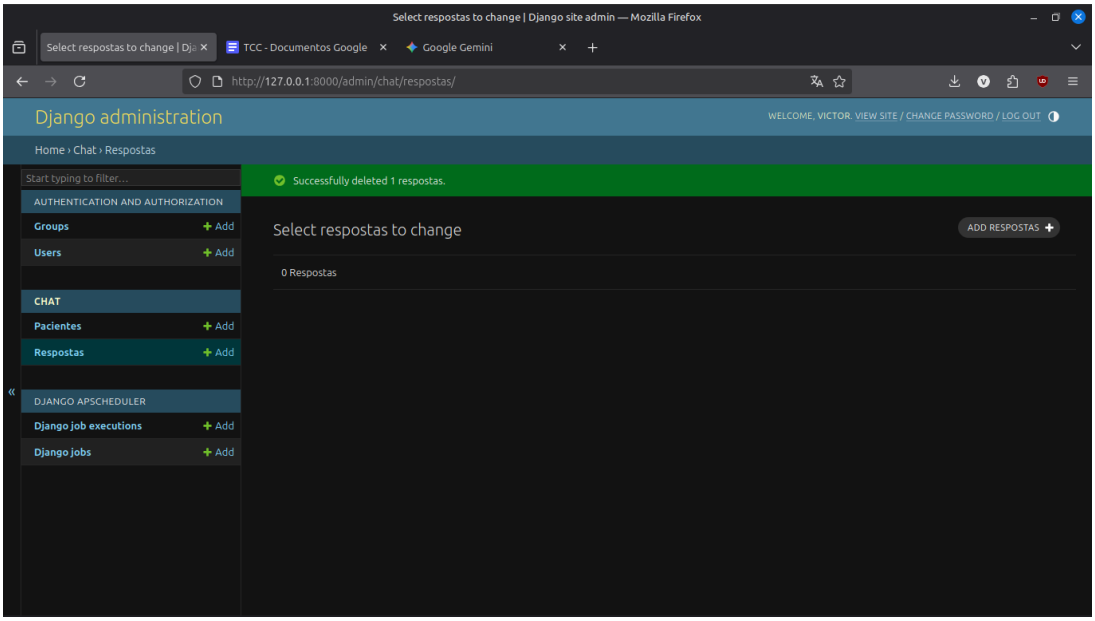
Fonte: Elaborado pelo autor (2025).

Figura 24 – Atualização de uma instância do modelo Respostas.



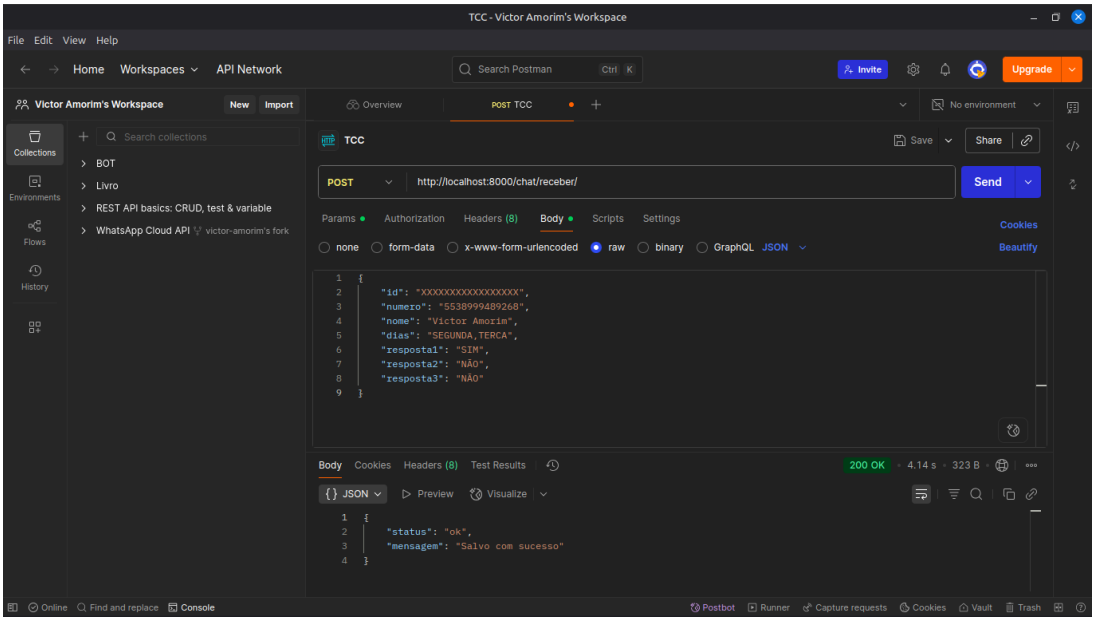
Fonte: Elaborado pelo autor (2025).

Figura 25 – Exclusão de uma instância do modelo Respostas.



Fonte: Elaborado pelo autor (2025).

Figura 26 – Status de sucesso resgatadas pelo log do servidor.



Fonte: Elaborado pelo autor (2025).

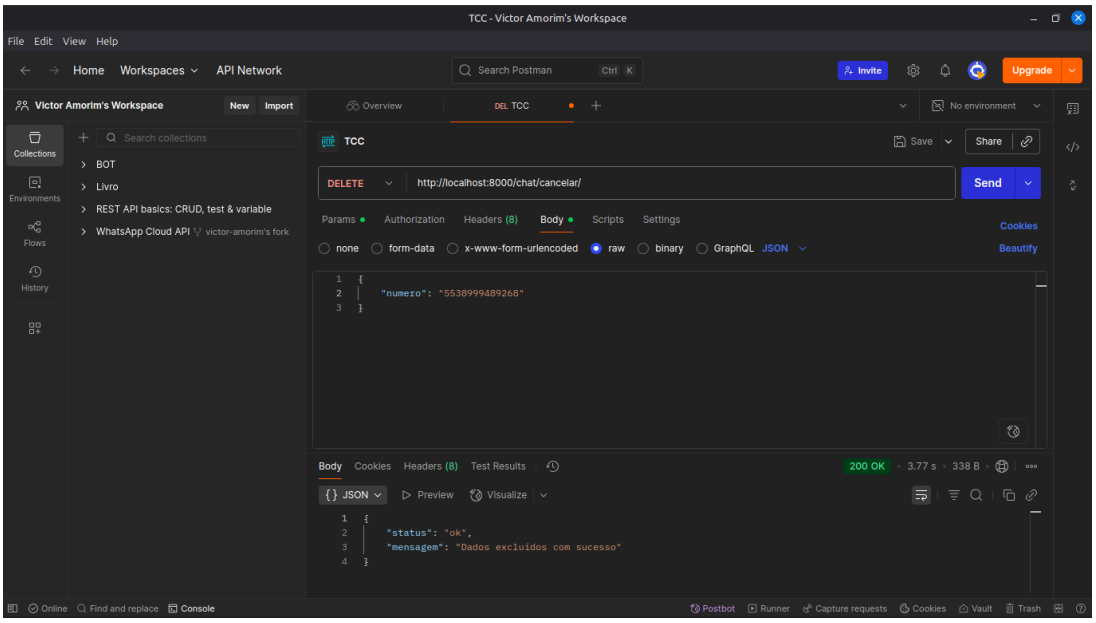
Para além das confirmações visuais presentes no servidor Django, a última imagem relata de forma objetiva o sucesso de todas as operações testadas no servidor.

6.3.2 Testes de respostas às requisições views

A próxima fase dos testes é testar como o servidor lidará com as requisições enviadas pelo chatbot. O primeiro a ser testado é a função de receber as mensagens e salvar no servidor.

Será criada uma requisição *POST*com dados fictícios de id do *WhatsApp*, número, nome do paciente, respostas 1, 2 e 3. É esperado que ao receber a requisição o servidor crie uma nova instância do modelo respostas, retornando o código 200 de sucesso.

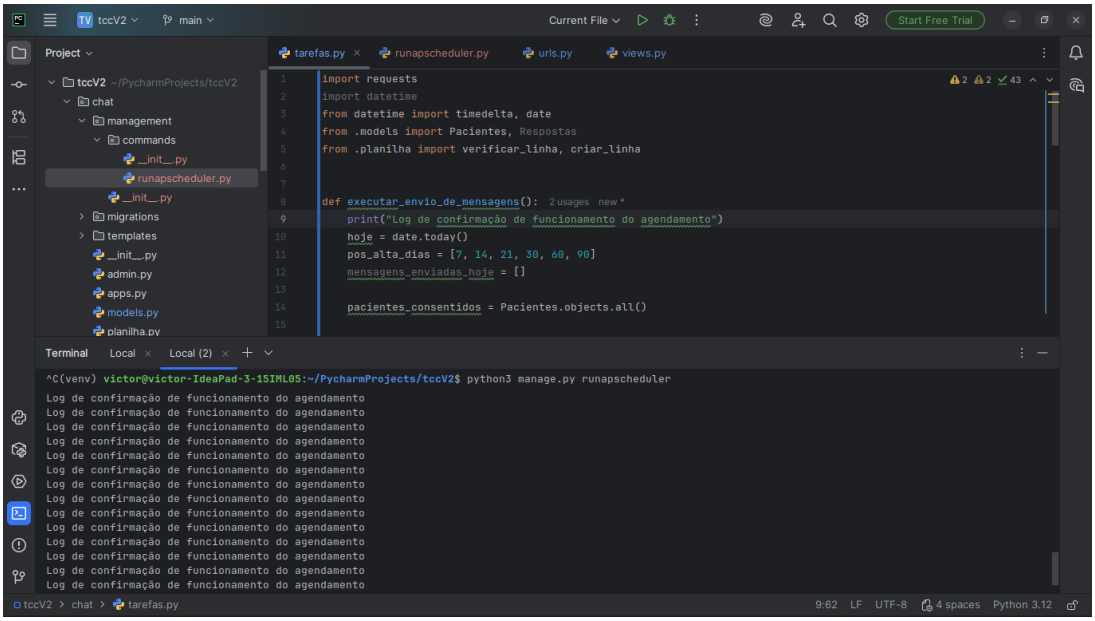
Figura 27 – Requisição realizada via Postman.



Fonte: Elaborado pelo autor (2025).

A outra função a ser validada é deletar os dados do paciente que solicitar o cancelamento da participação na pesquisa. O servidor recebe uma requisição *DELETE* com a informação de número e é esperado que o servidor filtre os dados de ambos os modelos e os delete, excluindo permanentemente os dados pessoais do banco de dados.

Figura 28 – Requisição de cancelamento realizada via Postman.

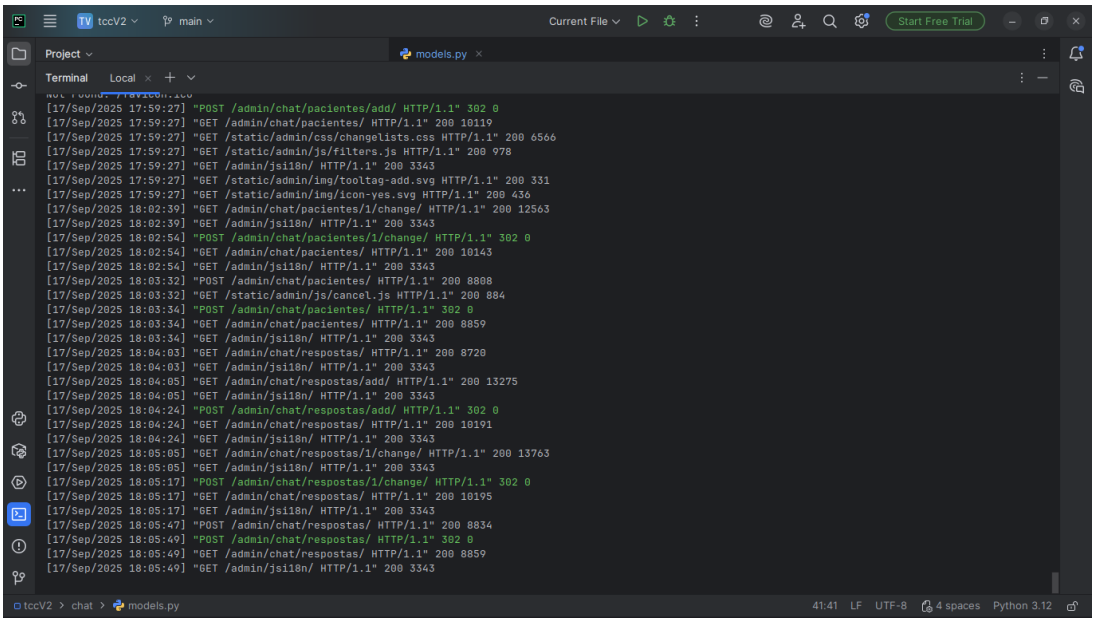


Fonte: Elaborado pelo autor (2025).

6.3.3 Testes das funções de envio de mensagem

Por fim, a função para efetuar o envio das mensagens será testada. Será implementado um *log* para validar o funcionamento da função. Será posto em cada ponto chave da execução do código, na listagem dos pacientes e ao verificar encontrar pacientes para iniciar a consulta.

Figura 29 – Confirmação do agendamento do envio de mensagens.



Fonte: Elaborado pelo autor (2025).

6.4 RESULTADOS

Como verificado nos testes acima, tanto o chatbot (n8n) quanto a API (*Evolution API*) e o servidor (Django) passaram nos testes propostos e estão em pleno funcionamento, garantindo a execução da ferramenta.

7 DISCUSSÃO

Os resultados apresentados validaram a eficácia do protótipo de chatbot como ferramenta de apoio no acompanhamento de pacientes em período pós-alta. A solução demonstrou não apenas ser funcional, mas também de fácil interação para os participantes, cumprindo o objetivo central de oferecer um canal de comunicação ágil e estruturado em uma fase crítica da jornada do paciente. Dessa forma, o trabalho atende à sua hipótese inicial: proporcionar aos profissionais de saúde uma ferramenta que otimiza a continuidade do cuidado e, ao mesmo tempo, facilita o engajamento do paciente devido à baixa barreira de uso da tecnologia. Ademais, para etapas futuras de implementação, destaca-se a necessidade de implementar a aplicação proposta em ambientes hospitalares reais, a fim de validar a solução proposta em um abiente de trabalho.

Em relação à tecnologia utilizada, é notável as limitações inerentes à arquitetura empregada. A utilização da plataforma n8n, sem a integração de Modelos de Linguagem Generativos (LLM), resulta em um fluxo conversacional fixo e dependente de respostas pré-definidas. Esta característica limita o tratamento de entradas do usuário que fogem do padrão esperado e, mesmo que o paciente forneça uma resposta semanticamente correta (afirmativa ou negativa), as variações na escrita podem impedir o prosseguimento do fluxo, representando a principal fragilidade do sistema atual.

Por outro lado, esta simplicidade se traduz em vantagens práticas notáveis: uma implementação mais rápida, custos operacionais reduzidos e, crucialmente, uma maior facilidade de manutenção e compreensão da lógica do sistema por parte da equipe de saúde responsável. Enquanto soluções de mercado mais avançadas podem oferecer uma conversação mais fluida, elas frequentemente demandam um alto investimento e conhecimento técnico especializado, o que poderia inviabilizar sua adoção no contexto hospitalar em questão.

Portanto, considerando que a ferramenta passou com êxito pelos testes de validação propostos, conclui-se que ela está apta para ser implementada em um ambiente de produção. Sua utilização pode gerar implicações operacionais positivas para as unidades hospitalares, como a padronização do acompanhamento pós-alta, a coleta estruturada de dados sobre a recuperação dos pacientes e a possibilidade de identificar precocemente sinais de complicação, contribuindo para a redução de taxas de readmissão hospitalar.

8 CONCLUSÃO

O presente trabalho propôs-se a projetar, desenvolver e validar uma ferramenta tecnológica, em formato de chatbot, para auxiliar no processo de acompanhamento de pacientes após a alta hospitalar. Conforme demonstrado, a solução foi implementada com sucesso e submetida a uma série de testes de validação, nos quais alcançou todos os resultados esperados, comprovando sua funcionalidade e usabilidade.

Conclui-se, portanto, que o protótipo atingiu plenamente os objetivos propostos. A principal contribuição deste estudo reside na entrega de uma solução de baixo custo e alta aplicabilidade, que promove um maior controle sobre a jornada do paciente no período pós-alta e amplia a acessibilidade ao cuidado continuado. A ferramenta demonstrou ser um elo viável entre a equipe de saúde e o paciente, mitigando riscos e fortalecendo o vínculo terapêutico fora do ambiente hospitalar.

Apesar do êxito, reconhece-se como limitação central a baixa capacidade do sistema em processar linguagem natural de forma flexível. Com base nisso, delineiam-se as seguintes sugestões para trabalhos futuros como uma evolução para modelos conversacionais avançados utilizando-se modelos de linguagem generativas e uma otimização dos fluxos de trabalhos da ferramenta.

REFERÊNCIAS

ADAMOPOULOU, Eleni; MOUSSIADES, Lefteris. **An overview of chatbot technology**, p. 373–383, 2020.

ANDRADE, Marta Cardoso de. **WhatsApp é o novo "queridinho" da Comunicação Mercadológica ou é da Comunicação Organizacional?** Dito Efeito-Revista de Comunicação da UTFPR, v. 12, n. 20, p. 1–14, 2021.

BATISTA, Taína Fagundes; RODRIGUES, Maria Cristina Soares. **Vigilância de infecção de sítio cirúrgico pós-alta hospitalar em hospital de ensino do Distrito Federal, Brasil: estudo descritivo retrospectivo no período 2005-2010**. Epidemiologia e Serviços de Saúde, Geral de Desenvolvimento da Epidemiologia em Serviços/Secretaria de …, v. 21, n. 2, p. 253–264, 2012.

BRASIL. **Lei nº 13.709, de 14 de agosto de 2018**. [S.l.: s.n.], 2018. Dispõe sobre a proteção de dados pessoais e altera a Lei nº 12.965, de 23 de abril de 2014 (Marco Civil da Internet). Publicada no Diário Oficial da União de 15.ago.2018. Disponível em: http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm. Acesso em: 24 set. 2025.

BRASIL. MINISTÉRIO DA SAÚDE. **Portaria nº 2.616, de 12 de maio de 1998**. [S.l.: s.n.], 1998. Expede diretrizes e normas para a prevenção e o controle das infecções hospitalares. Publicada no Diário Oficial da União de 13 de maio de 1998. Disponível em: https://bvsms.saude.gov.br/bvs/saudelegis/gm/1998/prt2616_12_05_1998.html. Acesso em: 25 set. 2025.

FØLSTAD, Asbjørn; NORDHEIM, Cecilie Bertinussen; BJØRKLI, Cato Alexander. **What makes users trust a chatbot for customer service? An exploratory interview study**, p. 194–208, 2018.

KUCHERBAEV, Pavel; BOZZON, Alessandro; HOUBEN, Geert-Jan. **Human-aided bots**. IEEE Internet Computing, IEEE, v. 22, n. 6, p. 36–43, 2018.

MARIETTO, Maria das Graças Bruno et al. **Artificial intelligence markup language: a brief tutorial**. arXiv preprint arXiv:1307.3091, 2013.

MEIRELLES, Fernando S. **Pesquisa do uso da TI: tecnologia de informação nas empresas**. São Paulo: Fundação Getúlio Vargas, 2023.

OLIVEIRA, Adriana Cristina; CIOSAK, Suely Itsuko; D'LORENZO, Claudia. **Vigilância pós-alta e o seu impacto na incidência da infecção do sítio cirúrgico**. Revista da Escola de Enfermagem da USP, SciELO Brasil, v. 41, p. 653–679, 2007.

OLIVEIRA AQUINO, Victor Hugo de; COSTA ADANIYA, Mario Henrique Akihiko da. **Desenvolvimento e aplicações de Chatbot**. Revista Terra & Cultura: Cadernos de Ensino e Pesquisa, v. 34, p. 56–68, 2018.

ORACLE. **O que é um chatbot?** Disponível em:
<https://www.oracle.com/br/chatbots/>. Acesso em: 24 set. 2025.

THOMAS, NT. **An e-business chatbot using AIML and LSA**, p. 2740–2742, 2016.

VICENTE NETO, Augusto; ESPIRITO SANTO, Guilherme Feulo do; GOLDMAN, Alfredo. **Compreensão de linguagem natural: uma solução em português brasileiro**. Anais, 2020.


UFVJM