

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

VICTOR HUGO KREOZER DE CARVALHO

**AUTOMATIZAÇÃO DO ATENDIMENTO AO CLIENTE: SELEÇÃO,
CONFIGURAÇÃO E IMPLANTAÇÃO DE UM CHATBOT EM UMA OPERADORA
DE PLANO DE SAÚDE**

TOLEDO

2024

VICTOR HUGO KREOZER DE CARVALHO

**AUTOMATIZAÇÃO DO ATENDIMENTO AO CLIENTE: SELEÇÃO,
CONFIGURAÇÃO E IMPLANTAÇÃO DE UM CHATBOT EM UMA OPERADORA
DE PLANO DE SAÚDE.**

Trabalho de conclusão de curso de graduação
apresentada como requisito para obtenção do título de
Tecnólogo em Sistemas para Internet da Universidade
Tecnológica Federal do Paraná (UTFPR).
Orientador(a): Rosane Fátima Passarini.

Toledo

2024



[4.0 Internacional](https://creativecommons.org/licenses/by-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

VICTOR HUGO KREOZER DE CARVALHO

**AUTOMATIZAÇÃO DO ATENDIMENTO AO CLIENTE: SELEÇÃO,
CONFIGURAÇÃO E IMPLANTAÇÃO DE UM CHATBOT EM UMA OPERADORA
DE PLANO DE SAÚDE.**

Trabalho de conclusão de curso de graduação
apresentada como requisito para obtenção do título de
Tecnólogo em Sistemas para Internet da Universidade
Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 06 de setembro de 2024

Prof. Dra. Rosane Fátima Passarini
Universidade Tecnológica Federal do Paraná

Prof. Dr. Roberto Milton Scheffel
Universidade Tecnológica Federal do Paraná

Prof. Ma. Suzan Kelly Borges Piovesan
Universidade Tecnológica Federal do Paraná

Toledo

2024

Dedico este trabalho à minha mãe, por ter me criado da melhor maneira possível e para a minha orientadora que me auxiliou imensamente na finalização deste trabalho.

AGRADECIMENTOS

Gostaria de expressar minha profunda gratidão a todos que me apoiaram ao longo desta jornada, especialmente aos meus familiares, colegas, professores e ao meu local de trabalho, que possibilitaram a realização deste projeto. Agradeço, em particular, à minha orientadora, Prof. Rosane Fátima Passarini, pela sabedoria com que me guiou ao longo deste percurso, bem como pela paciência e pelo apoio inestimável que tornaram possível a conclusão deste trabalho.

O perigo de verdade não é que computadores passem a pensar como humanos, mas sim que humanos passem a pensar como computadores.
(HARRIS; SYDNEY).

RESUMO

Este trabalho de conclusão de curso aborda a implantação de um software de chatbot em uma empresa do setor de saúde na cidade de Toledo. O objetivo é descrever as etapas e os processos de seleção, configuração e implantação de um chatbot para automatizar o processo de atendimento ao cliente da empresa. O estudo começa apresentando a motivação para implantação da tecnologia, destacando as dificuldades enfrentadas pelos diferentes departamentos da empresa. Em seguida, detalha o levantamento de requisitos necessários para a pesquisa e seleção de uma solução que atendesse às necessidades específicas da empresa, explicando os critérios que influenciaram a decisão de contratação do software. O trabalho também inclui uma descrição das funcionalidades do chatbot e do seu funcionamento, além de uma análise do processo de criação e desenvolvimento da estrutura dentro da plataforma escolhida. Aspectos cruciais do desenvolvimento são discutidos, como a definição de regras de horário de importantes, tais como a criação de menus personalizados para diferentes tipos de clientes e as adaptações necessárias para atender aos requisitos do projeto. Por fim, o estudo apresenta os resultados obtidos após a implementação do chatbot, incluindo gráficos de acesso à plataforma e níveis de satisfação dos clientes com o atendimento prestado. O trabalho conclui destacando os benefícios alcançados com a implantação da ferramenta, bem como as limitações encontradas durante o processo.

Palavras-chave: Chatbot; Atendimento ao Cliente; Blip; Saúde.

ABSTRACT

This work discusses the implementation of chatbot software in a healthcare company in the city of Toledo. The objective is to describe the stages and processes of selecting, configuring, and implementing the project. The study begins by presenting the motivation for adopting the tool, highlighting the challenges faced by the various departments of the company. It then details the requirement gathering necessary for researching and selecting a solution that meets the specific needs of the company, explaining the criteria that influenced the decision to hire the software. The thesis also includes a description of the chatbot's features and how it operates, along with an analysis of the process of creating and developing the structure within the chosen platform. Crucial aspects of development are discussed, such as defining service hour rules, creating customized menus for different types of clients, and making necessary adaptations to meet the project's requirements. Finally, the study presents the results obtained after implementing the chatbot, including graphs of platform access and customer satisfaction levels with the provided service. The thesis concludes by highlighting the benefits achieved with the tool's implementation, as well as the limitations encountered during the process.

Keywords: Chatbot; Customer Service; Blip; Health.

LISTA DE ILUSTRAÇÕES

Figura 1 - Aplicações que fazem uso do NLP	20
Figura 2 - Componentes dos chatbots	22
Figura 3 - Como funciona um chatbot	24
Figura 4 - Abrangência da Unimed Costa Oeste	26
Quadro 1 - Comparação de funcionalidades e diferenciais das plataformas de chatbot	30
Figura 5 - Interface do Blip Desk	33
Figura 6 - Interface do Blip Portal	34
Figura 7 - Interface do Bot Router	38
Figura 8 - Canais de conversa	38
Figura 9 - Menu serviços	39
Figura 10 - Fluxograma inicial	41
Figura 11 - Estrutura final do projeto	42
Figura 12 - Exemplo de criação de um chatbot	44
Figura 13 - Blocos obrigatórios nos chatbots	45
Figura 14 - Diferenciação de cores dos sub-bots	46
Figura 15 - Hierarquia dos bots	47
Figura 16 - Diagrama de Sequência das Condições de saída	49
Figura 17 - Interação inicial com o fluxo	50
Figura 18 - Estrutura do (MAIN) BOT	51
Figura 19 - Estrutura (MENU) EMPRESA	53
Figura 20 - Estrutura (MENU) CLIENTE DE OUTRA UNIMED	54
Figura 21 - Estrutura (MENU) NAO E CLIENTE	55
Figura 22 - Fluxo é cliente Unimed	56
Figura 23 - Dados salvos do cliente	58
Figura 24 - Estrutura (MENU) BENEFICIÁRIO	59
Figura 25 - Estrutura do menu do bot (DEP) FINANCEIRO	61
Figura 26 - Estrutura de validação para o atendimento humano	62
Figura 27 - Configuração do encerramento automático de tickets	63
Figura 28 - Tag do departamento	65
Figura 29 - Variável vStatusTransference	66
Figura 30 - Condição de redirecionamento do (ACT) TRANSFER MANAGER ..	67
Figura 31 - Tags utilizadas pelos atendentes via Blip Desk	68
Figura 32 - Média de acessos no período de julho à dezembro de 2022	72
Figura 33 - Média de acessos no período de fevereiro à julho de 2024	72
Figura 34 - Percentual da nota de atendimento do departamento de Atendimento	73
Figura 35 - Percentual da nota de atendimento do departamento de Relacionamento com o Cliente	74

LISTA DE TABELAS

Tabela 1 - Comparação de custos das plataformas de chatbot	32
---	-----------

LISTA DE ABREVIATURAS E SIGLAS

ACT	Ação
API	Interface de programação de aplicativos
CAEPF	Cadastro de Atividade Econômica de Pessoa Física
CNPJ	Cadastro Nacional de Pessoa Jurídica
CPF	Cadastro de Pessoa Física
DEP	Departamento
DV	Dígito Verificador
HTTP	Protocolo de Transferência de Hipertexto
AI	Inteligência Artificial
MKT	Marketing
ML	Aprendizado de Máquina
NLP	Processamento de Linguagem Natural
PF	Pessoa Física
PJ	Pessoa Jurídica
PR	Paraná
SDK	Kit de Desenvolvimento de Software
TF	Transferência
UTC	Tempo Universal Coordenado
UTFPR	Universidade Tecnológica Federal do Paraná

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Objetivos	15
1.1.1	Objetivo Geral.....	15
1.1.2	Objetivos Específicos	15
1.2	Estrutura do Trabalho	15
2	METODOLOGIA DA PESQUISA	16
2.1	Tipo de Pesquisa	16
2.2	Abordagem da Pesquisa	16
2.3	Procedimento Metodológico	16
3	REFERENCIAL TEÓRICO	19
3.1	Atendimento ao Cliente	19
3.2	Processamento de Linguagem Natural	20
3.3	ChatBot	21
3.3.1	Componentes de um Chatbot.....	21
3.3.2	Tipos de Chatbots	22
4	PROCESSO DE IMPLANTAÇÃO DE UM CHATBOT EM UMA OPERADORA DE PLANO DE SAÚDE	25
4.1	Caracterização da Empresa	25
4.2	Levantamento de Dados	27
4.3	Seleção da Plataforma de ChatBot	28
4.4	Configuração e Personalização do Chatbot	33
4.4.1	Estrutura Básica da Plataforma	33
4.4.2	Configurações Executadas.....	37
4.5	Implementação e Integração do Chatbot	39
4.5.1	Personalização da funcionalidade de Transferência de Atendimentos ..	64
4.5.2	Desafios na Implementação do Chatbot.....	68
4.6	Implantação do Chatbot	69
4.7	Análise do Período Pós-Implantação do Chatbot	71
5	CONCLUSÃO	76
5.1	Trabalhos Futuros	76
	REFERÊNCIAS	77
	APÊNDICE A – SCRIPT PARA DEFINIR OS DIAS E HORÁRIO DE TRABALHO	80

APÊNDICE B – SCRIPT PARA VERIFICAR OS DIAS E HORÁRIO DE TRABALHO	81
---	-----------

1 INTRODUÇÃO

Tecnologias como inteligência artificial (AI) (RUSSEL, NORVIG, 2016) e aprendizado de máquina (ML) (BISHOP, 2009), estão sendo incorporadas para otimizar processos, melhorar a eficiência e fornecer um atendimento mais personalizado aos clientes de empresas e organizações dos mais variados segmentos, incluindo o setor da saúde.

Com a crescente demanda por agilidade, eficiência e personalização, as operadoras de planos de saúde, a exemplo de outros segmentos, têm buscado soluções tecnológicas para aprimorar a experiência dos seus beneficiários. Nesse contexto, os chatbots surgem como uma ferramenta promissora para melhorar a comunicação com os beneficiários, reduzir a sobrecarga de atendimento humano e otimizar processos internos.

Os chatbots utilizam processamento de linguagem natural (NLP) para interagir com os usuários de maneira satisfatória, gerando respostas rápidas e precisas. Eles podem ajudar a resolver dúvidas frequentes, agendar consultas, fornecer informações sobre planos e coberturas e, até mesmo orientar os usuários em processos complexos.

A implantação de um chatbot além de modernizar o atendimento ao beneficiário, também representa um avanço estratégico para as operadoras de planos de saúde, que buscam adaptar-se às novas exigências do mercado e, ao mesmo tempo, garantir a qualidade e a eficiência dos seus serviços.

Dentro deste contexto, este trabalho tem como objetivo principal apresentar o processo de automatização do serviço de atendimento ao cliente de uma operadora de plano de saúde da cidade de Toledo no estado do Paraná. Ele aborda, especificamente, o processo de seleção, configuração, implantação e integração de um chatbot no ambiente operacional da empresa. Além disso, são relatados os resultados obtidos após a implantação da solução, bem como os desafios encontrados e as respectivas soluções aplicadas do ponto de vista tecnológico.

1.1 Objetivos

1.1.1 Objetivo Geral

Uma operadora de plano de saúde da cidade de Toledo no estado do Paraná tomou a decisão de implantar um chatbot para automatizar o serviço de atendimento aos seus beneficiários. Dentro deste contexto, este trabalho tem como objetivo principal relatar o processo de seleção, configuração, implantação e integração do *chatbot* no ambiente operacional da empresa, pontuando os desafios enfrentados e as soluções adotadas do ponto de vista tecnológico.

1.1.2 Objetivos Específicos

- Apresentar as necessidades específicas da operadora e dos beneficiários atendidos pelo *chatbot*.
- Relatar o processo de avaliação e seleção da plataforma de *chatbot* considerada mais adequada para as necessidades da operadora, considerando critérios como custo, funcionalidades e facilidade de integração.
- Demonstrar o processo de configuração e personalização do *chatbot*, incluindo a definição dos fluxos de conversas e a integração com sistemas existentes.
- Demonstrar como foi realizada a implantação do *chatbot* e a execução de testes que garantiram sua funcionalidade e eficácia.

Explicar como foi o processo de avaliação de desempenho do *chatbot* após a implantação, medindo indicadores de sucesso e coletando feedback dos usuários.

1.2 Estrutura do Trabalho

Este trabalho está estruturado em cinco capítulos. O Capítulo 2 apresenta a metodologia de pesquisa utilizada. O Capítulo 3 cita o referencial teórico, abordando conceitos e tecnologias relacionadas a chatbots. O Capítulo 4 detalha o processo de seleção, configuração, implantação e integração do chatbot no ambiente operacional da empresa, bem como os resultados da implantação, os desafios enfrentados e as soluções adotadas. O Capítulo 5 apresenta as conclusões, contribuições do trabalho e sugestões para trabalhos futuros.

2 METODOLOGIA DA PESQUISA

Este capítulo apresenta a metodologia de pesquisa utilizada para a realização deste trabalho. De acordo com (MARCONI; LAKATOS, 2012), a metodologia da pesquisa descreve o processo utilizado para conduzir um estudo, desde a coleta de dados até a análise e interpretação dos resultados.

2.1 Tipo de Pesquisa

O tipo de pesquisa deste trabalho é pesquisa descritiva com características de estudo de caso com participação ativa do pesquisador.

O objetivo deste trabalho é relatar e documentar o processo de seleção, configuração, implantação e integração de um chatbot, utilizando uma plataforma comercial de inteligência artificial, no ambiente operacional de uma operadora de plano de saúde, visando automatizar seu processo de atendimento ao cliente.

Este trabalho foi desenvolvido baseado na experiência direta do pesquisador, uma vez que ele participou de todas as etapas do processo.

2.2 Abordagem da Pesquisa

A abordagem adotada neste trabalho é qualitativa, pois envolve a análise detalhada dos processos de escolha, configuração e integração do chatbot na empresa. Essa abordagem permitiu explorar as características dos processos adotados, documentando os desafios e as soluções implementadas ao longo do projeto.

2.3 Procedimento Metodológico

Este tópico apresenta como este trabalho foi realizado. São descritas descreve as etapas e os métodos aplicados.

Etapa 1: Levantamento de Dados

A coleta de dados envolveu diferentes fontes, alinhadas com o papel do pesquisador como participante ativo:

- **Participação Observacional:** O pesquisador esteve diretamente envolvido em todas as etapas do projeto, participando de reuniões, decisões estratégicas e atividades práticas, como a configuração do chatbot e sua integração aos sistemas da empresa. Essa imersão permitiu um entendimento profundo e detalhado de cada fase do processo.
- **Análise de Documentos Internos:** Documentos técnicos, relatórios de projeto, atas de reuniões e outros materiais internos da empresa foram analisados para complementar a observação participativa e fornecer uma visão mais ampla das decisões e estratégias adotadas.
- **Entrevistas Informais e Discussões:** Conversas e discussões informais com outros membros da equipe envolvida no projeto foram utilizadas para captar percepções e feedbacks sobre o processo, enriquecendo a análise com diferentes pontos de vista.

Etapa 2: Seleção da Plataforma de Chatbot

O processo de seleção da plataforma de chatbot incluiu uma avaliação comparativa entre diferentes soluções pagas, levando em consideração critérios como custo, funcionalidades, escalabilidade, suporte técnico e integração com os sistemas já existentes na empresa. A seleção foi documentada com base em critérios predefinidos, focando na escolha da plataforma que melhor atendesse às necessidades operacionais da empresa.

Etapa 3: Configuração do Chatbot

A configuração do chatbot envolveu o ajuste das funcionalidades da plataforma escolhida para atender aos fluxos de atendimento da empresa. Esse processo foi acompanhado passo a passo, desde a criação dos scripts de conversa até a definição das integrações com sistemas internos.

Etapa 4: Implantação e Integração

A implantação foi realizada de forma gradual, com testes iniciais em ambiente controlado para validar o comportamento do chatbot em cenários reais. A integração com os sistemas existentes foi conduzida pela equipe de TI da empresa, garantindo que o chatbot operasse de forma sincronizada com as demais ferramentas de

atendimento. Esse processo foi descrito detalhadamente, destacando as soluções técnicas adotadas para superar os desafios encontrados.

Etapa 5: Análise dos Dados

Os dados coletados foram analisados qualitativamente, destacando os pontos críticos do processo e as lições aprendidas. A análise incluiu a avaliação dos resultados obtidos após a implantação do chatbot, com foco na eficiência operacional e na satisfação dos usuários finais

Etapa 6: Limitações da Pesquisa

Este estudo está limitado ao contexto específico da empresa em que foi realizado, o que pode não representar a realidade de outras organizações que adotam tecnologias de chatbots. Além disso, a análise foi restrita aos dados disponíveis e às percepções dos envolvidos, podendo haver variações em diferentes cenários de implementação.

3 REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico, abordando conceitos e tecnologias relacionadas a *chatbots* e suas aplicações.

3.1 Atendimento ao Cliente

O atendimento ao cliente é um conjunto de processos e práticas que visam garantir a satisfação deles, durante e após a compra de um produto ou serviço. Ele é a base para construir relacionamentos duradouros e fidelizar os clientes, processo este que é decisivo para o sucesso de um negócio (ZENDESK, 2023)

KOTLER; KARTAJAYA e SETIAWAN (2017) destacam que proporcionar uma experiência de atendimento ao cliente consistente e de alta qualidade tem um impacto direto nos resultados da organização. Em contrapartida, negligenciar essa área pode acarretar perdas significativas para o negócio.

LAS CASAS (2019) argumenta que um atendimento de excelência gera impressões positivas nos clientes, fazendo com que se sintam acolhidos e percebam a empresa como uma referência altamente profissional em seu segmento, o que impacta positivamente nos resultados da organização.

Sendo assim, um bom atendimento é fator decisivo na experiência do cliente com qualquer empresa, isso significa que quando uma pessoa procura um atendimento, é dever da empresa estar preparada para responder da melhor maneira possível às necessidades do cliente e, para isso, é necessário ter um canal de atendimento eficiente e eficaz.

Entretanto, esta não é a realidade de muitas empresas, segundo uma pesquisa feita pela Opinium, é possível observar que 3 em cada 4 brasileiros pararam de fazer compras em algumas empresas por problemas de comunicação com o cliente, sendo cerca de 76% dos participantes e, 39% dos entrevistados deixaram de consumir determinadas marcas por considerarem o tempo de atendimento muito longo (MOURA, 2021).

O uso da tecnologia pode potencializar a comunicação com o usuário final. Atualmente, a adoção de soluções baseadas em Inteligência Artificial, como chatbots

e assistentes virtuais, automatizam tarefas repetitivas, oferecem respostas personalizadas e permitem a análise de dados em tempo real, proporcionando uma experiência mais ágil e satisfatória aos clientes.

3.2 Processamento de Linguagem Natural

O Processamento de Linguagem Natural (NLP) é uma área da Inteligência Artificial que tem como objetivo possibilitar a comunicação natural entre humanos e máquinas. Através de técnicas de aprendizado de máquina e linguística, o NLP busca desenvolver sistemas capazes de entender e responder a comandos e perguntas em linguagem humana, como se estivéssemos conversando com outra pessoa (SANTOS, 2021)

Como pode ser observado na figura 1, o Processamento de Linguagem Natural, é a tecnologia por trás de aplicações como plataformas de busca, assistentes virtuais e chatbots. Ele permite que essas aplicações entendam o que falamos ou escrevemos e que respondam de forma relevante.

Figura 1 - Aplicações que fazem uso do NLP



Fonte: (BLIP, 2022)

A NLP desempenha um papel crucial em um chatbot, permitindo que ele compreenda e interaja com os usuários de maneira mais natural e eficiente. Algumas das funções em um chatbot são:

- **Compreensão da Linguagem:** NLP permite que o chatbot entenda o texto ou a fala do usuário, identificando a intenção e o contexto por trás das palavras. Isso inclui a capacidade de lidar com diferentes formas de expressão e variações linguísticas.
- **Reconhecimento de Entidades:** NLP ajuda a identificar e extrair informações específicas da entrada do usuário, como nomes, datas, locais e outros dados importantes que o chatbot precisa para fornecer uma resposta precisa.
- **Gerenciamento de Contexto:** NLP permite que o chatbot mantenha o contexto da conversa, compreendendo a sequência de interações e mantendo a coerência ao longo do diálogo.
- **Correção de Erros e Ambiguidades:** NLP pode ajudar o chatbot a lidar com erros de digitação, ambiguidade ou falta de clareza na entrada do usuário, ajustando a interpretação para fornecer respostas mais precisas.

3.3 ChatBot

Para MESHARAM et al. (2021), os chatbots são aplicativos de software que simulam conversas ou interações com os usuários, geralmente por meio de texto, mas também podem utilizar a voz. Essa simulação ocorre em tempo real, permitindo que usuários interajam com dispositivos digitais como se estivessem conversando com uma pessoa real. O que proporciona que os usuários tenham uma experiência de toque humanizado em suas conversas com esses robôs digitais. Em outras palavras, chatbots são como assistentes virtuais que podem responder a perguntas, realizar tarefas e fornecer informações de forma automatizada. Essa tecnologia é uma combinação de AI e NLP.

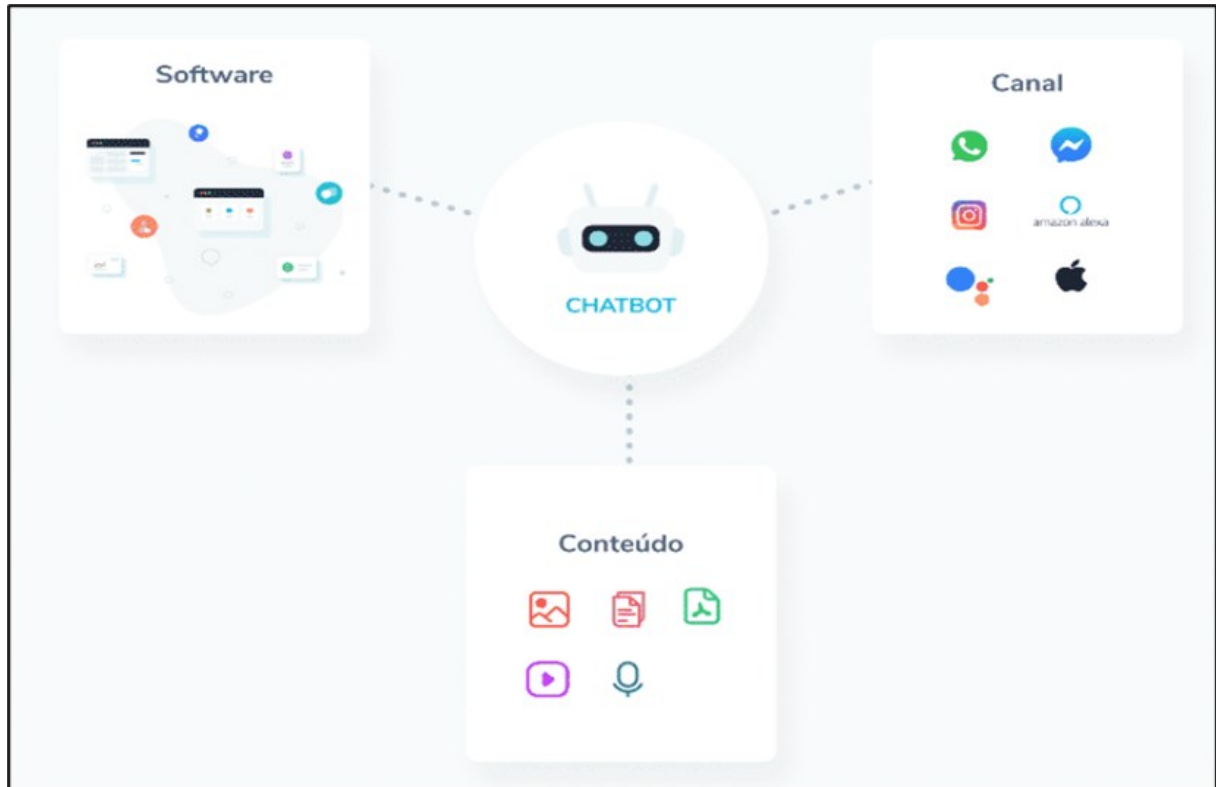
Atualmente os chatbots podem ser integrados a diversas plataformas, como WhatsApp, Instagram, Facebook Messenger, Telegram, sites em geral, sistemas de call center e lojas virtuais.

3.3.1 Componentes de um Chatbot

Segundo (BLIP, 2021), um chatbot é um programa automatizado projetado para interagir com clientes e contatos de empresas, eventos ou organizações. Sendo que,

todo chatbot é formado por três elementos básicos, representados graficamente pela figura 2

Figura 2 - Componentes de um chatbot



Fonte: (BLIP, 2021)

Os elementos apresentados na Figura 2 se inter-relacionam da seguinte maneira: o canal representa a plataforma onde ocorre a interação, como WhatsApp, Facebook Messenger, Telegram, entre outros. O conteúdo inclui os componentes usados na conversa, como textos, emojis e arquivos de mídia. O software, por sua vez, é o programa que determina as regras e controla a dinâmica da interação do chatbot.

3.3.2 Tipos de Chatbots

De acordo com MESHARAM et al. (2021), existem dois tipos básicos de chatbots: os baseados em regras e os baseados em aprendizado de máquina (Machine Learning, ML).

Os chatbots baseados em regras são os mais simples, utilizando um conjunto pré-definido de regras para responder às consultas dos usuários. Essas regras variam

conforme o tipo de aplicação para a qual o chatbot é utilizado. Embora sejam mais fáceis e econômicos de construir, sua principal limitação é a incapacidade de aprender e se adaptar a novas perguntas fora do conjunto de regras estabelecido.

Por outro lado, os chatbots baseados em ML são mais avançados. Eles são treinados para reconhecer palavras-chave e padrões específicos, o que lhes permite fornecer respostas mais precisas. Esses bots aprendem com suas interações anteriores e ajustam seu desempenho com pouca intervenção humana, melhorando continuamente sua capacidade de resposta.

Como pode ser observado, com o tempo, os chatbots evoluíram de simples bots baseados em regras para sistemas mais sofisticados que utilizam Inteligência Artificial (AI), Processamento de Linguagem Natural (NLP) e Aprendizado de Máquina (ML). Essa evolução tem ampliado as possibilidades de uso, permitindo que os chatbots se adaptem de maneira mais eficiente às necessidades específicas dos usuários e se integrem de forma mais eficaz em diversas aplicações.

A Figura 3 ilustra graficamente o funcionamento de um chatbot, destacando as diferenças entre os tipos existentes. Os chatbots baseados em regras realizam tarefas mais objetivas, operando a partir de mensagens pré-estabelecidas. Por outro lado, os chatbots que utilizam Inteligência Artificial (AI) interpretam as mensagens com base nas entradas dos usuários, proporcionando uma interação mais natural e humana. No entanto, esses chatbots requerem extensos treinamentos em processamento de linguagem para serem efetivamente operacionais.

Figura 3 – Como funciona um chatbot



Fonte: (BLIP, 2021)

Neste trabalho, optou-se pelo uso de um chatbot baseado em regras devido à sua simplicidade na configuração e manutenção. Além disso, essa escolha foi influenciada por questões de custo, uma vez que a integração de um chatbot baseado em Inteligência Artificial (AI) geralmente requer a contratação de uma empresa terceirizada e a realização de extensos treinamentos para que a AI compreenda a linguagem específica da empresa.

4 PROCESSO DE IMPLANTAÇÃO DE UM CHATBOT EM UMA OPERADORA DE PLANO DE SAÚDE

Este capítulo apresenta o estudo de caso detalhado sobre a implantação de um chatbot em uma operadora de saúde da cidade de Toledo no Paraná, para a automatização do processo de atendimento ao cliente da empresa. A análise abrange todo o processo, desde a seleção da plataforma até a integração com os sistemas internos da empresa.

Os resultados obtidos demonstram as melhorias alcançadas no atendimento ao cliente, como a redução do tempo de espera e o aumento da satisfação dos usuários. No entanto, o estudo também identifica algumas limitações da tecnologia, como a dificuldade em lidar com consultas complexas e a necessidade de aprimoramento na capacidade de compreensão da linguagem natural.

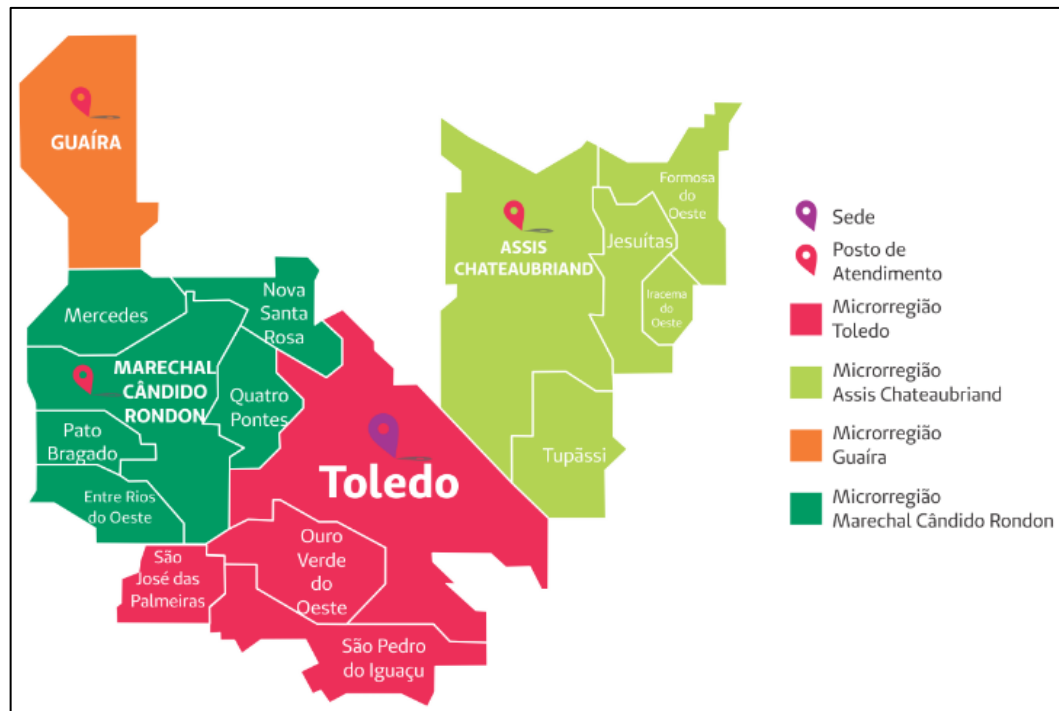
4.1 Caracterização da Empresa

A Unimed Costa Oeste (UNIMED, 2024) é uma cooperativa de planos de saúde atuante na região oeste do Paraná, comprometida em promover a saúde e o bem-estar de seus clientes. Com foco na melhoria contínua da qualidade de vida da comunidade, a cooperativa busca expandir suas ações e oferecer atendimento de excelência, com base nos princípios cooperativistas de trabalho em equipe e valorização humana.

Fundada em 14 de novembro de 1985, a Unimed Costa Oeste surgiu a partir da união de 30 médicos de Toledo, que, reunidos em assembleia, decidiram criar uma sociedade cooperativa para integrar os serviços médicos na região. Inicialmente chamada de Unimed de Toledo Cooperativa de Trabalho Médico Ltda, a cooperativa passou a se chamar Unimed Costa Oeste, ampliando sua atuação para 16 municípios do oeste paranaense.

A abrangência atual da Unimed Costa Oeste pode ser observada na Figura 4.

Figura 4 - Abrangência da Unimed Costa Oeste



Fonte: Portal da UNIMED COSTA OESTE (2024)

A cooperativa conta com uma infraestrutura abrangente, incluindo hospitais, clínicas e centros de atendimento, que oferecem serviços de consultas médicas, exames laboratoriais e de imagem, atendimentos de urgência e emergência, além de programas de saúde preventiva. Atualmente, a Unimed Costa Oeste é composta por centenas de médicos cooperados e atende milhares de beneficiários, proporcionando uma rede de atendimento integrada e qualificada. Sua missão, visão e valores são:

Missão: Promover saúde e qualidade de vida, oferecendo atendimento médico acessível e de excelência, pautado em princípios cooperativistas.

Visão: Ser reconhecida como referência em serviços de saúde na região oeste do Paraná, destacando-se pela inovação, qualidade e compromisso com o cliente.

Valores: Ética, transparência, cooperação, responsabilidade social e comprometimento com a saúde e bem-estar dos clientes.

A Unimed Costa Oeste é reconhecida por suas ações voltadas à responsabilidade social e à sustentabilidade, investindo em programas que visam a prevenção de doenças, o cuidado com o meio ambiente e o apoio a iniciativas comunitárias. A cooperativa se destaca pela busca constante de melhorias em seus

processos e pelo compromisso em manter um atendimento humanizado e centrado nas necessidades dos pacientes.

Com uma área de abrangência que engloba 16 municípios, a Unimed Costa Oeste continua a evoluir, fortalecendo sua posição como uma das principais operadoras de planos de saúde da região e reafirmando seu compromisso com a saúde e a qualidade de vida da população local.

4.2 Levantamento de Dados

O projeto de implantação de uma plataforma de chatbot foi iniciado com o intuito de criar um canal centralizado de comunicação na cooperativa, com o objetivo de aprimorar o atendimento aos clientes da Unimed Costa Oeste. Antes da implantação, o contato com os clientes era realizado principalmente por telefone e, em alguns casos, por WhatsApp, mas cada departamento tinha um número diferente. Isso gerava dificuldades para os clientes que precisavam falar com diferentes setores, resultando em barreiras na comunicação.

Além disso, os departamentos com contato direto com os beneficiários relataram um volume elevado de ligações, o que impactava outros processos e o atendimento presencial. Outro problema identificado foi a perda de backups nos celulares corporativos, já que cada departamento usava um número distinto no WhatsApp, e a falha no aparelho resultava na perda total do histórico de contatos.

Para atender a essas necessidades, foi realizado um levantamento de requisitos com os principais departamentos. Os requisitos funcionais identificados incluem:

- **Backup das conversas:** Garantir a preservação dos históricos de conversas com clientes.
- **Integração com múltiplos canais:** Conectar-se a canais como WhatsApp, Instagram, Facebook Messenger, entre outros.
- **Integração com APIs externas:** Validar dados dos clientes, como número da carteirinha e CPF.
- **Transferência de atendimentos:** Permitir a troca de atendimentos entre atendentes dentro da mesma plataforma, sem a necessidade de usar diferentes

números de telefone.

- **Gerenciamento de atendimentos:** Fornecer métricas para monitorar o tempo médio de atendimento, quantidade de atendimentos diários por atendente, e outros indicadores.
- **Acompanhamento em tempo real:** Monitorar os atendimentos à medida que ocorrem.
- **Aplicativo móvel para vendedores:** Disponibilizar uma aplicação para facilitar o trabalho dos vendedores.

Já os requisitos não funcionais identificados foram:

- **Escalabilidade:** Integrar funções de maneira simples.
- **Melhoria da experiência dos usuários:** Aperfeiçoar o atendimento para Beneficiários, Cooperados e Prestadores.
- **Tempo de Resposta:** O chatbot deve responder às consultas dos usuários em questão de segundos para garantir uma interação fluída.
- **Autenticação e Autorização:** O chatbot deve garantir que apenas usuários autenticados possam acessar informações sensíveis ou realizar ações críticas.
- **Intuitividade:** A interface do chatbot deve ser fácil de usar e intuitiva para garantir que os usuários possam interagir com o bot sem dificuldades.
- **Facilidade de Atualização:** O sistema deve permitir a atualização fácil e rápida dos fluxos de conversa e das bases de conhecimento do chatbot sem causar interrupções significativas no serviço.
- **Alta Disponibilidade:** O chatbot deve estar disponível 99,9% do tempo, para garantir operação contínua mesmo em caso de falhas.

4.3 Seleção da Plataforma de ChatBot

Com a conclusão do levantamento de requisitos para entender as principais funcionalidades necessárias para atender às expectativas da cooperativa, iniciou-se

uma pesquisa com fornecedores para a aquisição do sistema que melhor se adequasse às necessidades da empresa.

De acordo com a política da cooperativa, foi necessário obter cotações de pelo menos três empresas diferentes, resultando na seleção de três empresas: Huggy, Syngoo Talk e Blip. Essas soluções foram escolhidas após consultar outras Unimeds no Paraná que já haviam implantado ferramentas de chatbot, e o feedback positivo levou à continuidade das negociações com essas opções.

A Huggy (HUGGY, 2024), fundada em 2015, surgiu para atender à demanda por soluções eficazes na gestão de conversas via chat online. Suas principais características incluem:

- Integração com WhatsApp, Telegram, Messenger (Facebook), E-mail, SMS e Huggy Chat (para portais);
- Automação de conversas;
- Impulsioneamento de vendas;
- Organização e centralização de atendimentos;
- Atendimento automático e inteligente;
- Criação de bots e integração via API;
- Gerenciamento de tarefas e workflows de atendimento;
- Aplicação de pesquisas;
- Gerenciamento de contatos;

A Syngoo Talk (SYNGOO TALK, 2024), fundada em 2020, é especializada em desenvolver soluções práticas que melhoram o desempenho das empresas. Suas principais características são:

- Integração com WhatsApp, Telegram, Messenger e Instagram (Facebook), além de Chat Online (para portais).
- Automação de conversas.
- Organização e centralização de atendimentos.
- Atendimento automático ou inteligente.
- Criação de bots e integração via API.
- Prospecção de clientes em massa.
- Aplicação de pesquisas.

- Gerenciamento de contatos.
- Integração com Dialogflow (IA Google).
- Disponibilidade de aplicativo mobile.

A Blip (BLIP, 2024), fundada no início dos anos 2000, começou sua trajetória vendendo ringtones monofônicos e polifônicos. Em 2014, a empresa expandiu para o setor de chatbots, oferecendo uma série de características e funcionalidades, incluindo:

- Integração com canais como WhatsApp, Telegram, Messenger (Facebook), chat online (para portais), Google Assistant, AdLingo e RCS Google.
- Automatização de conversas e centralização de atendimentos.
- Atendimento automático.
- Criação de bots e integração via API.
- Prospecção de clientes em massa e aplicação de pesquisas.
- Gerenciamento de contatos.
- Integração com ferramentas de Inteligência Artificial (IA) como Watson, Microsoft LUIS, Azure e Dialogflow (Google).

O quadro 1 detalha como as soluções avaliadas atendem aos requisitos necessários para o projeto.

Quadro 1 - Comparação de funcionalidades e diferenciais das plataformas de chatbot

Funcionalidades/Diferenciais	Take Blip	Syngoo Talk	Huggy
Plataforma para criação de fluxos conversacionais	Sim	Sim	Sim
Integração com WhatsApp, Telegram, Facebook e outros	Sim	Sim	Sim
Monitoramento de Atendimento/Backup/Relatórios	Sim	Sim	Sim
Acompanhamento dos Atendimentos em tempo real	Sim	Sim	Sim
Integração Aberta com API's	Sim	Sim	Não
Transferência de Atendimento para outro departamento	Sim	Sim	Não
Aplicativo Mobile	Em desenvolvimento	Sim	Não
Parceiro Oficial com Facebook	Sim	Não	Não

Fonte: Autoria própria

Ao comparar os valores fornecidos por cada fornecedor, foram observados as seguintes diferenças:

1. Taxa de Implementação:

- **Blip e Huggy:** Ambas cobram uma taxa de implementação.
- **Syngoo Talk:** Isenta da taxa de implementação.

2. Cobrança por Adicionais:

- **Huggy:** Cobra por logins de atendentes, o que pode ser oneroso para a Unimed, dado o grande número de funcionários, resultando em custos mais altos em comparação com as outras opções.
- **Blip e Syngoo Talk:** Não cobram por logins adicionais.

3. Valores de Mensagens Ativas:

- **Blip e Syngoo Talk:** Os valores são semelhantes, refletindo o custo cobrado pela Meta, proprietária do WhatsApp.
- **Huggy:** Realiza o repasse dos valores em dólar, o que pode resultar em variações dependendo da cotação.

4. Quantidade de Mensagens Franqueadas:

- **Blip:** O total de mensagens franqueadas no mês são 66 mil ou realizar o contato com 2.200 clientes diferentes.
- **Syngoo Talk:** A Syngoo ofertou 45 mil mensagens franqueadas por mês.
- **Huggy:** A quantidade de mensagens franqueadas na mensalidade são 15 mil e 15 atendentes simultâneos utilizando a plataforma.

5. Cobrança por Mensagens Excedentes:

- **Blip:** Cobra adicionalmente R\$ 0,03 por mensagem ou R\$ 0,90 por usuário caso passar o total de mensagens franqueadas ou o número de usuários atendidos.
- **Syngoo Talk:** Cobra taxa adicional de R\$ 0,045 por mensagem excedente do total franqueado.
- **Huggy:** Cobra adicional de R\$ 0,03 por mensagem e R\$ 189,90 por usuário adicional.

A tabela 1 ilustra as diferenças de valores para facilitar a comparação entre os fornecedores.

Tabela 1 - Comparação de custos das plataformas de chatbot

Valores	Take Blip	Syngoo Talk	Huggy
Taxa de Implantação	R\$ 3.960,00	R\$ 0,00	R\$ 2.500,00
Mensalidade	R\$ 1.980,00	R\$ 1.990,00	R\$ 4.147,60
Adicional por Mensagem Excedente	R\$ 0,03 ou R\$ 0,90	R\$ 0,05	R\$ 0,03
Valor por Mensagem Ativa (HSM)	R\$ 0,28	R\$ 0,38	R\$ 0,11
Valor por Usuário Adicional	R\$ 0,00	R\$ 0,00	R\$ 189,90
Valor Total	R\$ 5.940,00	R\$ 1.990,00	R\$ 6.647,60

Fonte: Autoria própria

Após a análise das plataformas, ficou claro que a Huggy não atendia aos critérios mínimos necessários para o projeto, principalmente devido à falta de funcionalidades essenciais, como integração com APIs e a capacidade de transferir interações para outros departamentos. Além disso, o modelo de cobrança baseado em usuários ativos representava um custo elevado, especialmente para uma cooperativa com mais de 200 colaboradores, tornando inviável a ampliação para diversos departamentos.

As negociações continuaram com a Blip e a Syngoo Talk, ambas oferecendo as funcionalidades requisitadas e apresentando valores semelhantes. A escolha recaiu sobre a Blip, em parte devido à sua parceria oficial com o Facebook, o que eliminou custos adicionais com o envio de mensagens via WhatsApp, pois, a Meta exige uma empresa parceira para mediar a troca de grandes volumes de mensagens e para evitar o bloqueio de números é preciso ter uma mediadora que possua autorização para transacionar estes dados e a Blip já atendia a esse requisito, evitando a necessidade de contratar um fornecedor adicional.

Outros fatores que influenciaram a decisão foram a isenção da Blip da taxa inicial de R\$ 3.960,00 para implementação e a disponibilidade de quase todas as funcionalidades necessárias, com a aplicação móvel em desenvolvimento. A Blip também foi escolhida devido à sua sólida experiência de mais de 20 anos no mercado e às recomendações positivas de outras Unimed, como BH e Porto Alegre, além da Unimed Cascavel.

Com a contratação da plataforma, foi iniciado o processo de desenvolvimento do projeto, seguindo uma série de etapas bem definidas para garantir uma implementação eficaz. As etapas realizadas são descritas nas próximas seções deste documento.

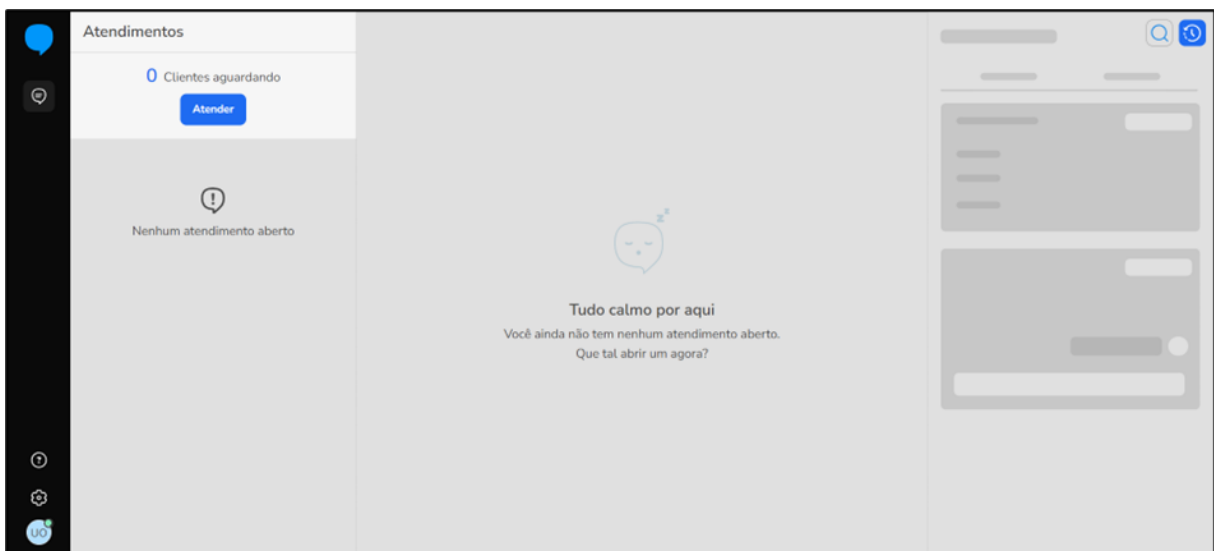
4.4 Configuração e Personalização do Chatbot

4.4.1 Estrutura Básica da Plataforma

Blip (BLIP, 2024) é uma plataforma que opera por meio de uma interface web, o que elimina a necessidade de instalação de software em máquinas corporativas e proporciona maior praticidade. A criação de logins é de responsabilidade dos próprios usuários, pois a equipe de TI não tem permissão para criar logins, ao contrário de alguns outros softwares. Os usuários podem criar suas contas vinculando-as a uma conta do Google ou fornecendo um e-mail e senha, conforme preferirem.

A plataforma Blip oferece dois canais de entrada: Blip Desk e Blip Portal. O Blip Desk é a interface utilizada pelos atendentes para interagir com clientes que entram pelo fluxo conversacional ou para contatá-los diretamente. Como demonstrado na figura 5, esta interface é simplificada para garantir que qualquer atendente possa realizar o atendimento de forma rápida e eficiente, contando apenas as funcionalidades essenciais que utilizarão no atendimento ao cliente.

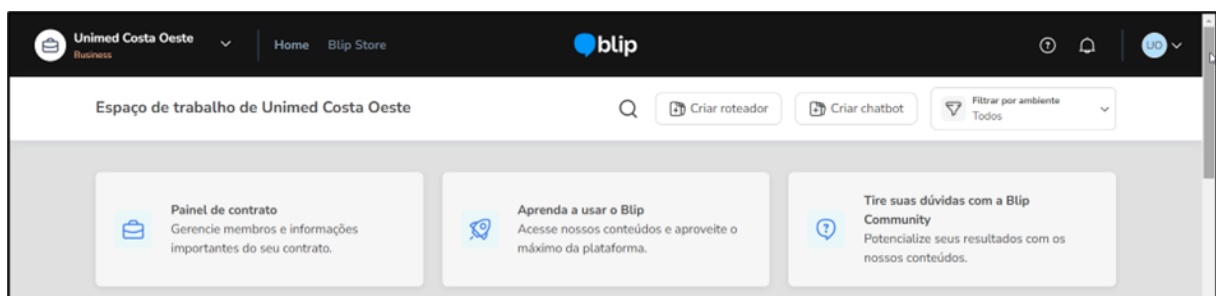
Figura 5 - Interface do Blip Desk



Fonte: Imagem criada pelo autor.

O Blip Portal é a interface onde toda a criação e configuração do fluxo conversacional é realizada. Nele, são configurados os bots e integrados os canais de entrada, além disso, o portal permite gerenciar a equipe de atendimento, concedendo permissões apenas para pessoas autorizadas e será a plataforma onde os supervisores utilizam para gerenciar e monitorar os atendimentos realizados pelos atendentes. Na figura 6 é possível visualizar a interface desta tela, contando com botões e criação de bots, conteúdos para aprimorar o conhecimento da plataforma e realizar o gerenciamento do contrato.

Figura 6 - Interface do Blip Portal



Fonte: Imagem criada pelo autor.

Com a Blip, é possível criar fluxos conversacionais de três maneiras diferentes, visando oferecer suporte completo para a criação, gestão e evolução de contatos inteligentes (Blip, 2024). As opções são:

- **Builder:** Ferramenta de construção visual que permite criar fluxos conversacionais sem necessidade de digitação de código.
- **SDKs:** Permite criar o fluxo utilizando um dos SDKs da plataforma disponíveis nas linguagens C#, JavaScript e Java.
- **API HTTP:** Nesse modelo, o Blip realiza requisições a um endpoint¹ definido pelo desenvolvedor sempre que novas mensagens ou notificações estiverem disponíveis. Da mesma forma, para enviar mensagens ao cliente, é necessário fazer uma requisição HTTP² à API do Blip. Esse modelo é vantajoso para quem

¹ Em tecnologia, um endpoint é um ponto de comunicação final em uma rede ou um serviço web onde dados são enviados ou recebidos. Em APIs (Interfaces de Programação de Aplicações), um endpoint é uma URL específica que permite a interação com um serviço ou recurso, definindo onde as requisições podem ser feitas e as respostas obtidas.

² **HTTP: Protocolo de Transferência de Hipertexto (Hypertext Transfer Protocol)** é um protocolo de comunicação utilizado na web para a troca de informações entre clientes e servidores. Define como as mensagens são formatadas e transmitidas, e como os servidores e navegadores devem responder a diferentes comandos.

deseja construir o bot em linguagens de programação além de C#, JavaScript e Java.

Para o projeto do chatbot da empresa foi optado pelo formato Builder, neste formato é utilizado o chatbot por regras, pois ele permite uma construção visual do fluxo sem a necessidade de codificação, o que facilita a manutenção futura e desta maneira torna a estrutura mais intuitiva para os demais técnicos compreenderem o que foi criado, o que facilita manutenções futuras.

Para um melhor entendimento, a seguir estão descritos alguns conceitos técnicos e nomenclaturas utilizadas pela plataforma, bem como o comportamento da estrutura do Builder:

- **Bot Router:** É o bot responsável por gerenciar os sub-bots. Este bot é o ponto de contato visível para o cliente, devendo ser publicado e testado nos canais. O Bot Router não possui regras ou conteúdo próprio, apenas referências a todos os sub-bots. Cada Bot Router deve ter pelo menos um sub-bot associado.
- **Sub-Bot Principal:** Quando um usuário interage com um Bot Router pela primeira vez, ele é direcionado ao sub-bot principal da hierarquia. Este que é responsável por determinar qual sub-bot atenderá o cliente inicialmente. Se o Bot Router tiver apenas um sub-bot, este será automaticamente o sub-bot principal.
- **Sub-Bot:** Qualquer bot que faça parte da hierarquia de um Bot Router.
- **Serviço:** Cada sub-bot é considerado um serviço dentro do Bot Router. Esta denominação é importante para garantir que sub-bots com o mesmo nome possam ser utilizados dentro do Bot Router. É necessário atribuir um nome (nome do Serviço) a cada sub-bot, que será usado para referenciar o sub-bot durante a troca entre bots.
- **Expiração do Redirecionamento:** É possível definir um tempo de inatividade para cada sub-bot. Esse tempo de expiração estabelece um limite, em segundos, para o período em que o sub-bot deve permanecer ativo na conversa após a última interação do cliente. Isso é útil, por exemplo, para sub-bots que só devem atender clientes em horários específicos.

No padrão Builder, é utilizado um Bot Router para gerenciar os sub-bots e este bot é responsável por criar uma hierarquia de bots que trabalham em conjunto durante a interação com o usuário. Inicialmente, é criado um bot principal, onde as interações com o cliente começam. A partir desse bot principal, é possível redirecionar para os sub-bots, que podem ser criados sem limite.

Ao configurar um novo subbot no Router, é possível definir um tempo de expiração. Essa configuração garante que, se um cliente permanecer por muito tempo em um fluxo específico ou na fila de espera, ele será redirecionado de volta ao fluxo inicial.

Dentro dos sub-bots, a plataforma oferece uma variedade de funcionalidades para a criação e configuração do fluxo, semelhante à construção de um diagrama UML. É possível criar blocos e inserir o conteúdo das mensagens que serão exibidas ao usuário final. Além disso, é possível adicionar ações tanto na entrada quanto na saída dos blocos. As ações disponíveis são:

- **Requisição HTTP:** Permite realizar requisições HTTP para qualquer API pública.
- **Registro de eventos:** Registra eventos importantes ocorridos dentro do bot para análise futura na área de relatórios.
- **Definir contato:** Modifica ou adiciona atributos do usuário utilizando texto fixo ou variáveis do fluxo.
- **Redirecionar a um serviço:** Permite trocar o usuário entre diferentes sub-bots, enviando-o a outro fluxo conforme a hierarquia da plataforma.
- **Gerenciar lista de distribuição:** Administra os usuários que estão interagindo com o bot a partir de uma lista.
- **Executar script:** Permite escrever códigos no formato java script para criar ações lógicas, tratar variáveis e valores, e construir conteúdos dinâmicos, entre outras opções.
- **Definir variável:** Atribui valores a variáveis de contexto no fluxo para uso durante a interação.
- **Consultar assistente de conteúdo:** Envia um valor para análise pelo assistente de conteúdo.
- **Processar comando:** Envia um comando para uma extensão do Blip.

Com essas funcionalidades, é possível configurar uma variedade de opções para melhorar a experiência do usuário e integrar sistemas internos ou de terceiros à sua estrutura. Cada bloco no fluxo é dividido em ações de entrada e saída, permitindo criar ações distintas para quando o cliente entra e sai de um bloco. Por exemplo, ao entrar no bloco, você pode implementar um script para validar o canal de acesso do usuário. Em seguida, tem a possibilidade de solicitar informações adicionais se necessário. Ao sair do bloco, pode-se configurar uma ação para redirecionar o usuário para diferentes sub-bots, dependendo da plataforma utilizada.

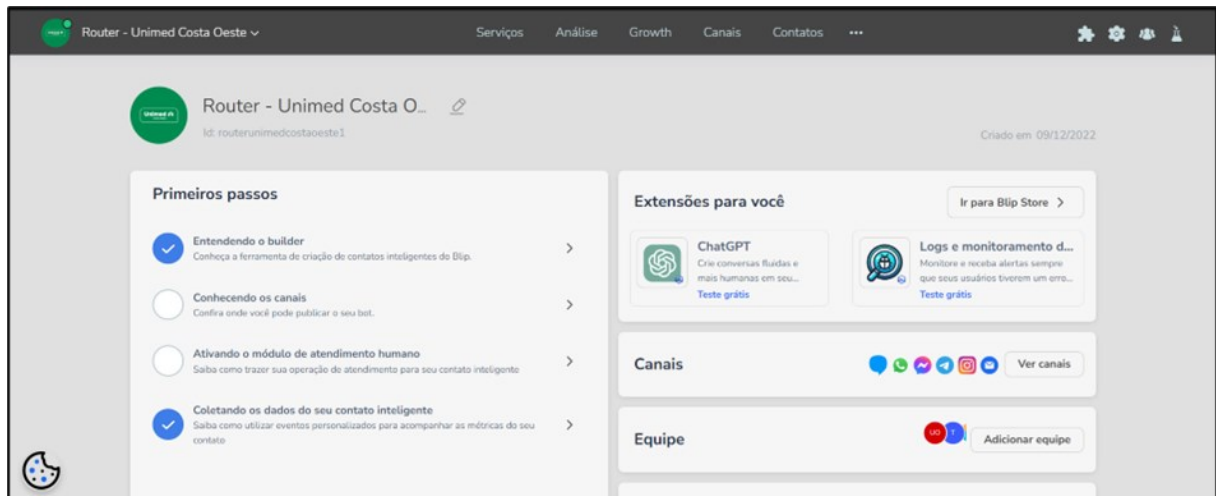
Além dessas ações, cada bloco possui condições de saída, que permitem redirecionar o cliente para outro bloco ou sub-bot com base na resposta fornecida. Também pode-se utilizar variáveis geradas por ações, como uma requisição HTTP, para validar se a pessoa é cliente da empresa. Se nenhuma das condições especificadas for atendida, é possível definir uma saída padrão, geralmente usada para direcionar o cliente a um bloco de erro padrão, alertando-o de que a informação fornecida não está conforme o esperado.

4.4.2 Configurações Executadas

As primeiras configurações no sistema foram ser feitas na aba Router. Nesta seção, foram conectados os canais de comunicação ao fluxo conversacional e a definição de qual será o bot principal. Além disso, foi necessário criar os serviços para tornar os subbots visíveis para os demais bots da organização.

No Router, também é possível adicionar extensões para auxiliar na configuração ou aprimorar funcionalidades que não são nativas do sistema. No entanto, muitas dessas extensões são pagas, pois são desenvolvidas por empresas parceiras da Blip e comercializadas na plataforma. Na figura 7, está demonstrando a interface do Router, onde possui menus como serviços, análise onde pode criar diversos relatórios, acessar os contatos dos usuários, entre outros.

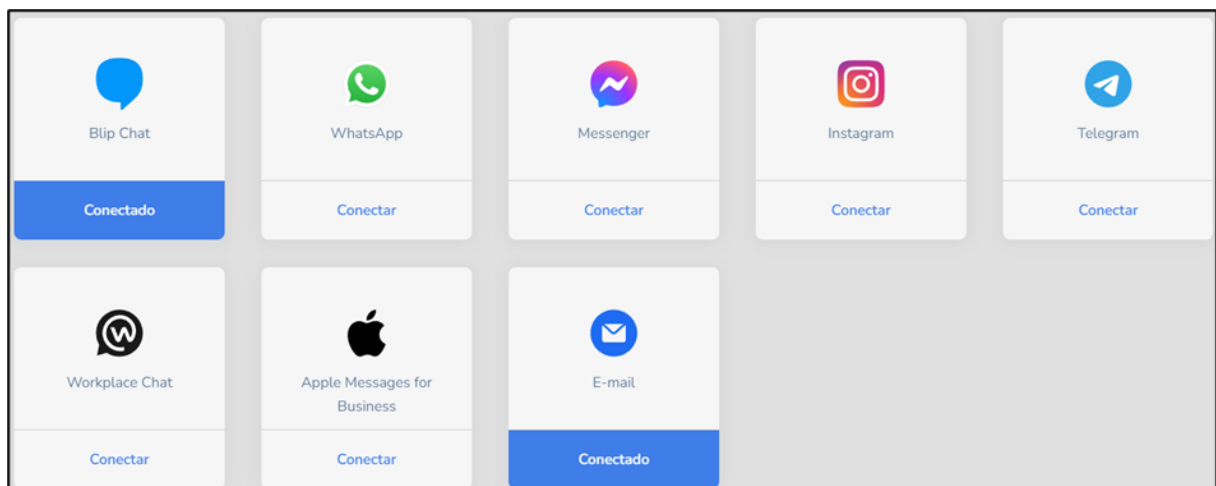
Figura 7 - Interface do Bot Router



Fonte: Imagem criada pelo autor.

A primeira configuração a ser realizada nesta tela é na aba "Canais", onde serão definidos os canais de comunicação disponíveis para interação com o cliente. A plataforma oferece os seguintes canais para configuração, conforme ilustrado na figura 8:

Figura 8 - Canais de conversa



Fonte: Imagem criada pelo autor.

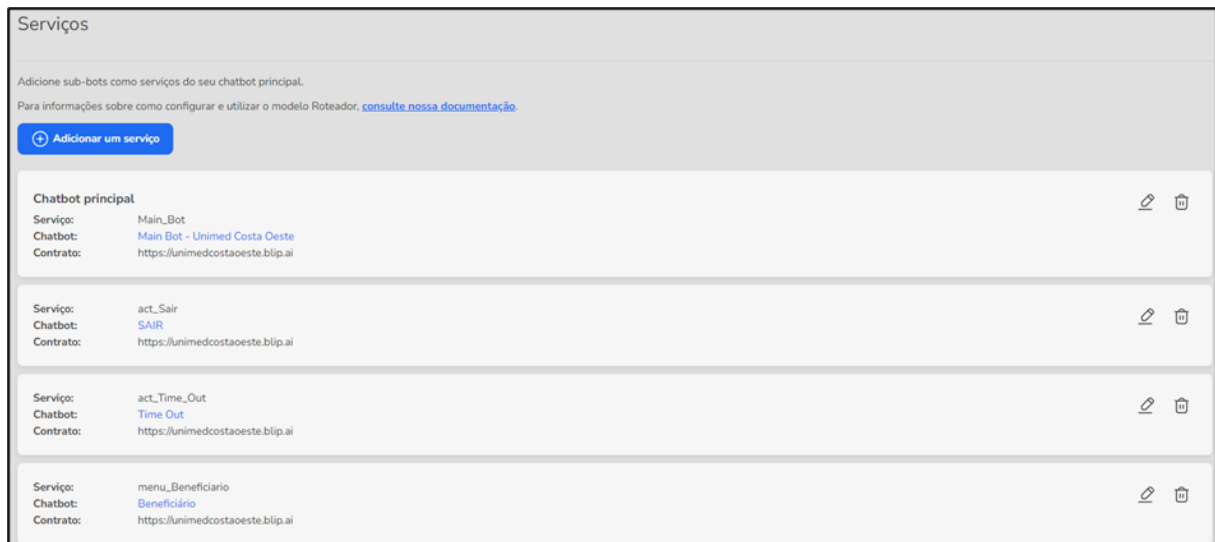
Como pode-se ver na imagem acima, a plataforma oferece a conexão com diversos canais diferentes, alguns não muito conhecidos como o Blip Chat. Este tem como propósito para adicionar em sites web e normalmente ficam localizados no canto inferior direito. O Workplace Chat é uma ferramenta que é utilizada para empresas e tem como objetivo fazer a interação entre colegas de trabalho. Além disso, alguns canais como WhatsApp, Instagram e Apple Messages possuem um aviso de Upgrade

pois estão disponíveis apenas na versão paga da plataforma, e como foi contratado a ferramenta, não aparece este aviso. Os demais canais podem ser utilizados na versão gratuita.

Após análise interna, decidiu-se vincular os canais WhatsApp, Messenger, Instagram e Telegram, com o WhatsApp sendo o canal de entrada principal devido à expectativa de um maior volume de clientes. No entanto, para garantir que todos os clientes sejam atendidos, também serão vinculados os outros canais. Nestes canais secundários, o atendimento será feito exclusivamente por chatbot, e os clientes serão informados de que, para uma conversa interativa com um atendente humano, devem contatar o número de WhatsApp.

Na aba "Serviços", serão adicionados os bots conforme o cadastro e o fluxo criado. Para que um cliente possa interagir efetivamente dentro do fluxo do bot, o sistema precisa reconhecer todos os sub-bots para permitir o direcionamento adequado. Na figura 9 é possível observar alguns serviços cadastrados.

Figura 9 - Menu serviços



Fonte: Imagem criada pelo autor.

4.5 Implementação e Integração do Chatbot

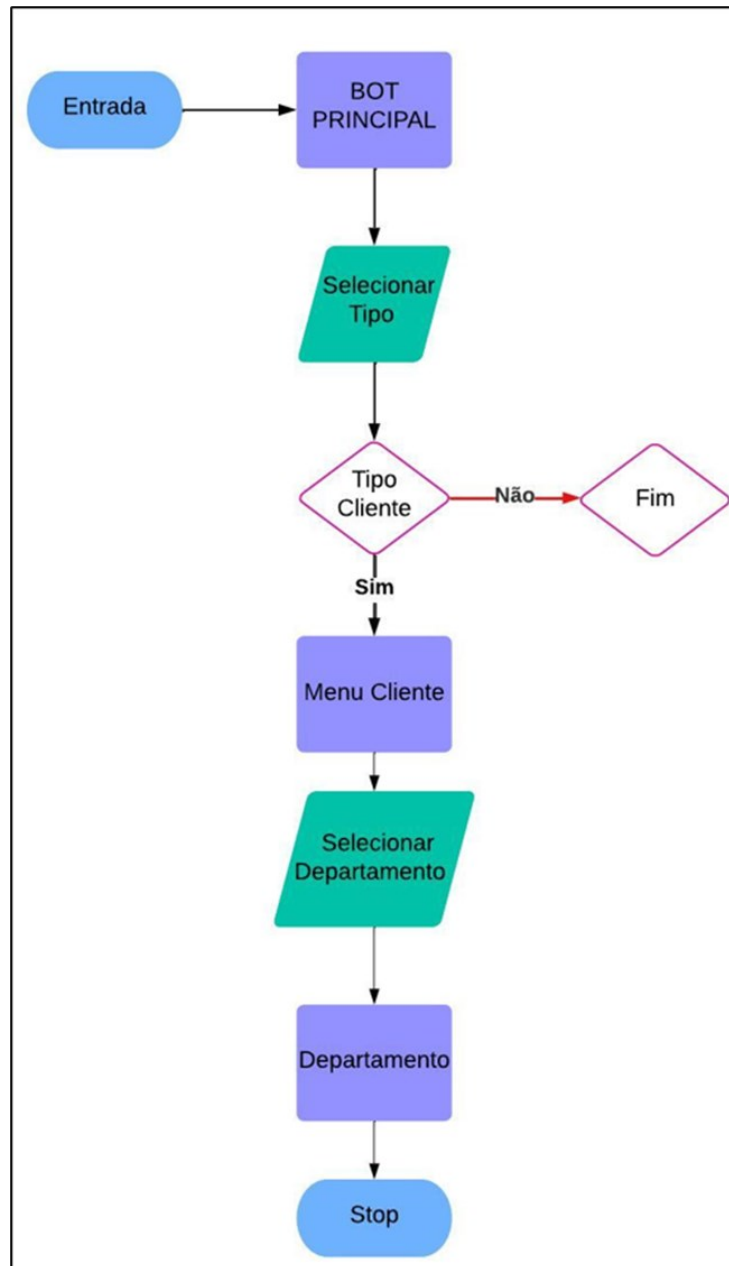
A implementação do chatbot envolveu, essencialmente, a criação e definição dos fluxos de conversação. Para a criação do fluxo de conversação, foi realizada uma pesquisa com algumas Unimed's coirmãs que utilizam chat conversacional. Aquelas que foram contatadas, informaram que toda a estrutura estava consolidada em um

único bot. Nesse modelo, o fluxo conversacional e o redirecionamento para os atendentes eram gerenciados no mesmo bot. No entanto, esse método pode resultar em um fluxo extenso e difícil de manter, e os supervisores de diferentes departamentos teriam acesso a todos os dados, não apenas aos do seu próprio departamento.

Foi decidido, portanto, adotar uma estrutura que considerasse a escalabilidade futura e a separação por departamentos, permitindo que cada supervisor acesse apenas os dados relevantes ao seu departamento e facilitando a manutenção do fluxo.

Na figura 10 a seguir está uma representação visual do fluxograma inicial que foi criado para realizar a criação da estrutura dentro da plataforma Blip.

Figura 10 - Fluxograma inicial



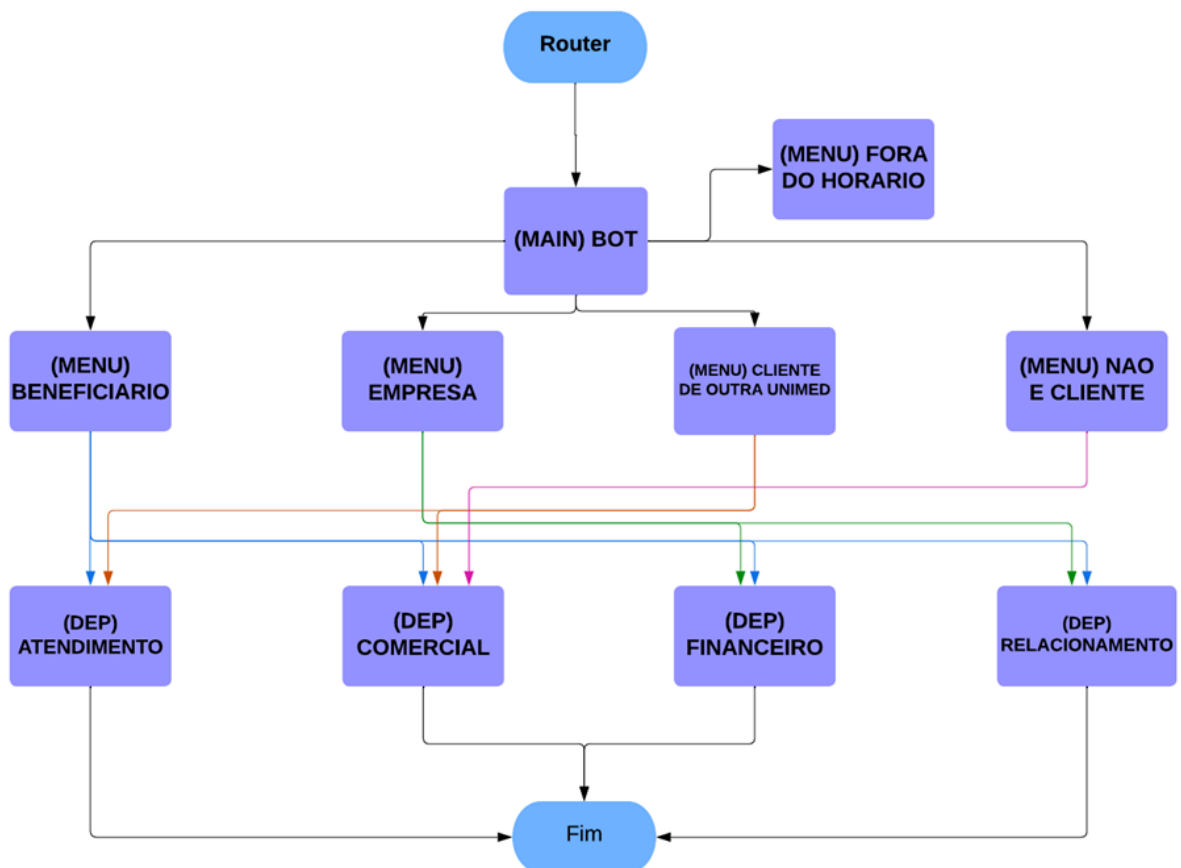
Fonte: Imagem criada pelo autor.

A estrutura inicial incluirá um bot principal, que será a porta de entrada para o cliente, onde será determinado o tipo de cliente para o redirecionar para o menu correto. Nesse menu, o cliente poderá selecionar o departamento com o qual deseja conversar. Cada departamento será representado por um sub-bot no fluxo, e os clientes terão opções para serem direcionados a um atendente apropriado. Além disso, cada menu da estrutura inclui uma opção para retornar ao menu anterior, permitindo que o cliente navegue facilmente de volta caso acesse o bot por engano ou encontre opções que não correspondam à sua dúvida.

Como o novo processo teria um impacto significativo nos departamentos, foi decidido implementá-lo inicialmente em apenas alguns deles. Isso permitirá construir a estrutura gradualmente e obter feedback dos usuários para ajustar os fluxos conforme necessário. Os departamentos escolhidos foram Atendimento, Relacionamento com o Cliente, Comercial e Financeiro, pois são os que têm maior contato direto com os clientes da Unimed. Dessa forma, caso seja necessário transferir um cliente para outro atendente dentro da plataforma, há uma maior probabilidade de que o atendimento possa ser realizado sem que o cliente precise sair do canal.

Assim, a estrutura dos sub-bots criada na ferramenta é ilustrada na Figura 11. O bot principal, denominado (MAIN) BOT, serve como ponto de partida para acessar os sub-bots dos menus e, subsequentemente, os dos departamentos.

Figura 11 - Estrutura final do projeto



Fonte: Imagem criada pelo autor.

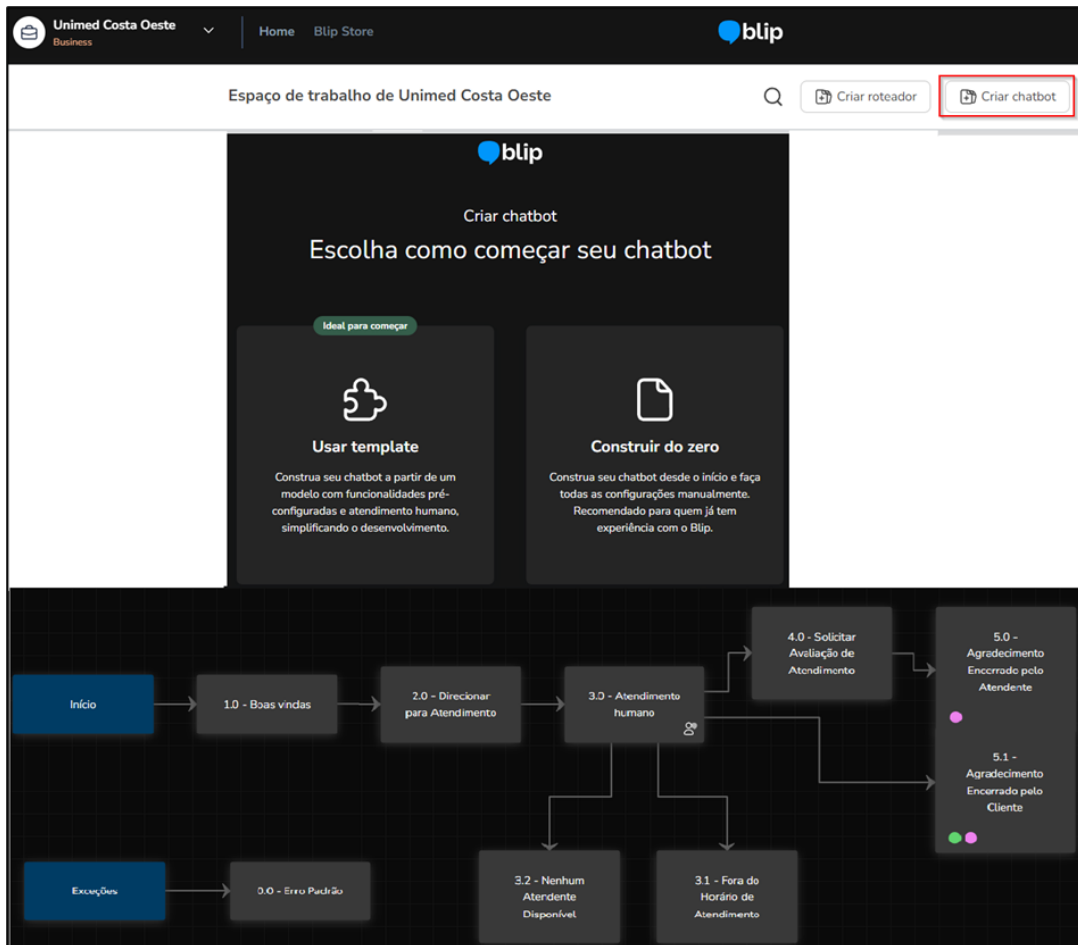
Após a criação do esboço inicial dos fluxos, foi realizada uma reunião com cada departamento, incluindo o supervisor da área e alguns colaboradores-chave, para entender o fluxo habitual de interação com os clientes. Esse encontro proporcionou insights sobre as dúvidas mais frequentes, os tipos de clientes que entram em contato e as responsabilidades de cada departamento. Identificou-se, por exemplo, que há quatro tipos principais de clientes que entram em contato:

1. Clientes físicos que possuem plano com a Unimed Costa Oeste.
2. Clientes jurídicos que possuem plano com a Unimed Costa Oeste.
3. Clientes físicos que possuem plano, mas não com a Unimed Costa Oeste.
4. Clientes que ainda não possuem plano.

Cada tipo de cliente é atendido por um departamento específico. Por exemplo, clientes que não possuem plano com a cooperativa devem ser direcionados ao departamento Comercial para que a equipe possa prospectar uma possível venda; por isso, esses clientes não devem ter acesso às opções dos demais departamentos. Já os clientes que possuem plano com outra Unimed devem se comunicar apenas com o departamento de Atendimento, que é responsável por questões relacionadas à liberação de guias. Por outro lado, clientes que têm contrato com a Unimed Costa Oeste podem entrar em contato com os departamentos de Atendimento, Relacionamento e Financeiro, para esclarecer dúvidas sobre liberação de guias, discutir questões de pagamento, entre outros assuntos. Empresas contratantes (PJ) devem ter acesso apenas aos departamentos de Relacionamento com o Cliente e Financeiro. Este fluxo de entrada do cliente pode ser visto na Figura 11, onde está definido o que cada tipo de cliente tem acesso.

Com base nesses dados, foi dado início a criação do fluxo organizacional na plataforma Blip, começando pelo bot principal. Para criar um bot, é preciso clicar em "Criar Bot" e, em seguida, escolher entre construir o bot do zero ou selecionar um template pré-pronto. O template pré-pronto fornece um fluxo genérico de conversa com o cliente, desde a entrada até o atendimento humano e a avaliação do atendimento. Este processo pode ser visto na figura 12.

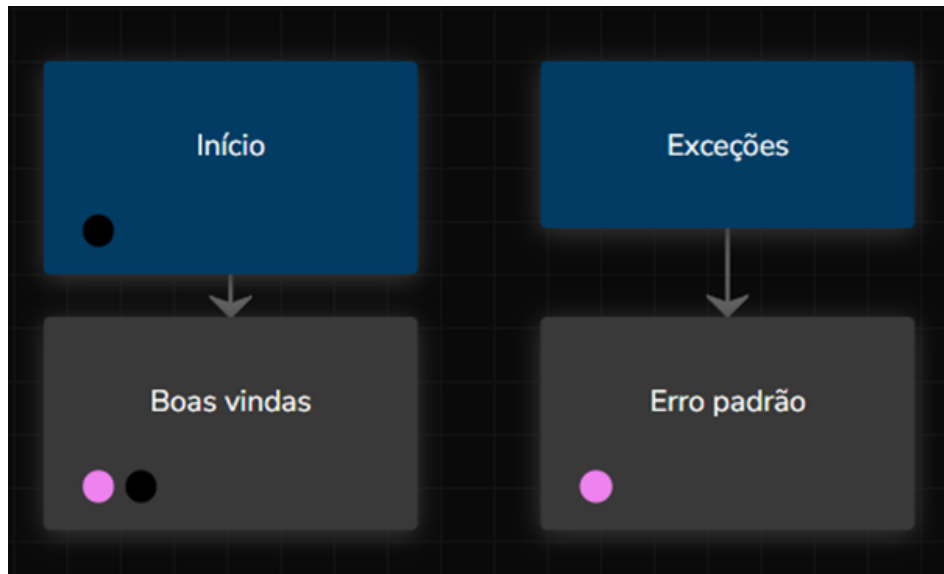
Figura 12 - Exemplo de criação de um chatbot



Fonte: Imagem criada pelo autor.

No Bot Principal, foi optado por criar o fluxo a partir do zero. Nesta configuração, o bot contém apenas o bloco de início e de boas-vindas, além dos blocos de exceções e de erro padrão. O bloco de início é obrigatório em todos os bots criados, pois é nele que ocorre a primeira interação, que posteriormente redireciona para outro bloco onde estão configuradas as interações. O bloco de exceções também está presente em todos os bots e tem a função de redirecionar para uma mensagem de erro caso o cliente interaja com o fluxo e forneça uma informação que o sistema não reconheça. No entanto, o uso deste bloco não é obrigatório, pois é possível criar blocos específicos para tratar eventuais erros ou entradas inesperadas. Na figura 13 podemos demonstrar visualmente estes blocos na plataforma.

Figura 13 - Blocos obrigatórios nos chatbots

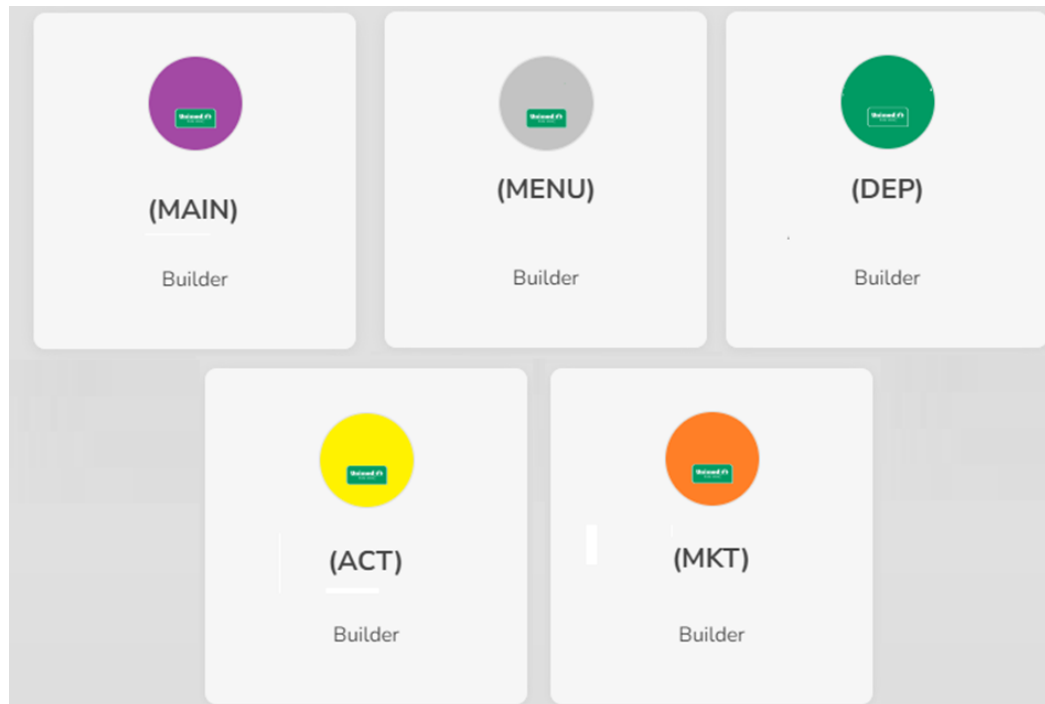


Fonte: Imagem criada pelo autor.

Por motivos de melhor identificação dos bots foram criados tags na frente, sendo elas:

- **(MAIN)**: Identificar qual é o bot principal do Fluxo.
- **(MENU)**: Serão os menus da estrutura, que servirão para direcionar a outros sub-bots.
- **(DEP)**: Estes são os bots dos departamentos onde será realizado o atendimento humano.
- **(ACT)**: É um acrônimo para action, onde serão os bots que possuíam alguma ação específica a ser realizada, seu objetivo é apenas executar uma tarefa.
- **(MKT)**: Os bots de marketings serão os fluxos de entrada para o uso de mensagens ativas específicas realizadas.

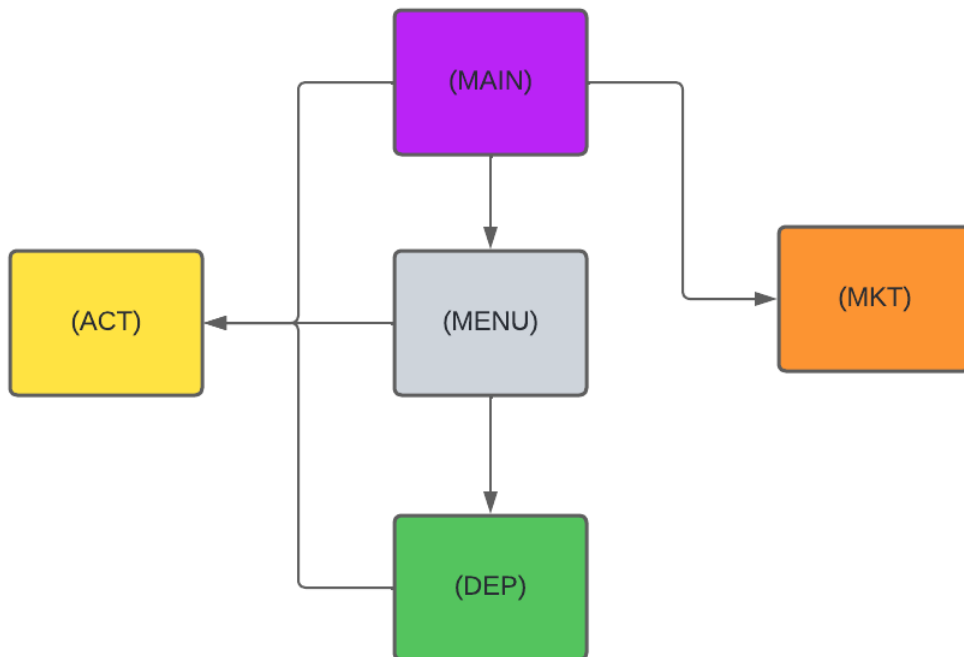
Além da tag, cada sub-bot terá uma cor destinada, para visualmente deixar a estrutura mais organizada, sendo a cor roxa definida para a tag **MAIN**, o cinza para a tag **MENU**, o verde para as tags **DEP**, o amarelo para o **ACT** e a cor laranja para sub-bots que possuem a tag **MKT**, conforme demonstrado na figura 14:

Figura 14 - Diferenciação de cores dos sub-bots

Fonte: Imagem criada pelo autor.

A Figura 15 ilustra a hierarquia dos bots, começando pelo bot principal (MAIN). A partir deste bot, é possível acessar o bot de menus (MENU) e, subsequentemente, o bot de departamentos (DEP). O bot de ações (ACT) pode ser acionado a partir de qualquer um dos bots anteriores, enquanto o bot de marketing (MKT) só pode ser acessado através de uma transferência do bot principal (MAIN).

Figura 15 - Hierarquia dos bots



Fonte: Imagem criada pelo autor.

As variáveis e scripts criados dentro do fluxo seguirão um padrão no qual as palavras serão mantidas em inglês, sempre que possível. Essa escolha visa diferenciar elementos mais técnicos, voltados à programação do fluxo conversacional.

Com a estrutura básica do fluxo principal criada, foram desenvolvidos os bots do fluxo de conversação, começando pelo bot principal (MAIN BOT). Como este será o ponto de entrada para os clientes da empresa, é essencial implementar algumas validações para garantir a melhor experiência do usuário. A principal validação a ser considerada é o horário de funcionamento do fluxo conversacional, para que os clientes possam entrar em contato com a empresa apenas durante o horário de atendimento e não fiquem esperando na fila por horas sem resposta.

A Unimed realiza seus atendimentos em horário comercial, de segunda a sexta-feira, das 07:30h às 18:00h. Por isso, o sistema foi configurado para que os atendimentos com as recepcionistas sejam possíveis apenas dentro desse intervalo. No **(MAIN BOT)**, após o cliente enviar uma mensagem, ele passará por um bloco chamado **(ACT) Verificar Horário**, onde o horário de atendimento está programado. Caso o cliente entre em contato fora desse horário, será redirecionado para um sub-

bot chamado **(MENU) Fora do Horário**, que apresentará um menu para interagir e esclarecer dúvidas que não exijam atendimento humano.

Para criar a regra de horário, dentro do bloco **(ACT) Verificar Horário**, na aba **Ações**, serão criados dois scripts. O primeiro arquivo definirá os dias e horários de funcionamento, enquanto o segundo validará se o horário de entrada no bloco está dentro do horário de funcionamento. Conforme apêndice A é possível visualizar o script criado na linguagem java script, onde tem uma função com nome run, na blip, todas os scripts criados, são obrigatórios ter uma função com este nome para entender que deve executar esta ação e assim, ser possível obter o valor retorno. Após definir uma variável chamada **workSchedule**, é definido o horário de funcionamento e ao final do script a variável é retornada utilizando o método stringify para converter os valores em JSON, desta forma, conseguimos utilizar estes dados durante o fluxo ou em outros scripts. Os retornos desta variável ficaram armazenados em uma variável chamada **objWorkSchedule**.

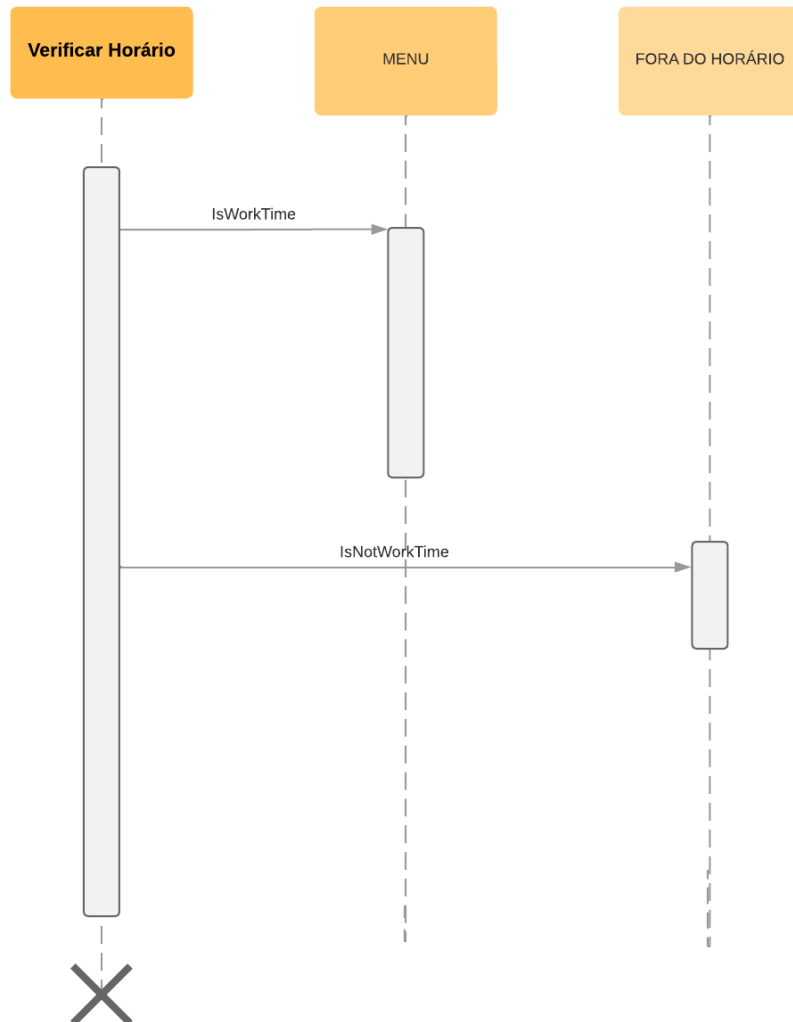
Em seguida, é criado um novo script para validar se o horário da passagem pelo fluxo corresponde com os valores do horário de trabalho e nele é necessário passar duas variáveis para ser possível realizar a validação, uma delas é a **objWorkSchedule** e a outra é a informação da zona da região onde a empresa está localizada, que neste caso o valor a ser utilizado é -3, pois o script utiliza o **UTC** (Tempo Universal Coordenado) para comparar e validar o horário. **UTC** é o padrão de tempo global que serve como base para a sincronização dos relógios ao redor do mundo, sem variações sazonais, sendo essencial para garantir precisão na comparação de horários.

Além da função run, no script é utilizado de demais funções como pegar o valor de UTC do momento, verificar se o dia do momento é um dia de trabalho, do contrário o processo já é encerrado. Também realiza validações como pegar o período de atendimento e comparar com o json definido no script anterior, para garantir que o cliente consiga prosseguir apenas se todos os dados estiverem verdadeiros.

Como pode ser visualizado na apêndice B, o resultado do script é armazenado na variável **objsWorkTime**, que será utilizada para criar uma regra na aba **Condições de Saída**. Se o valor obtido for verdadeiro, o fluxo é direcionado para o

menu principal, caso o valor seja falso, o fluxo é redirecionado para o menu de atendimento fora do horário, que será criado, conforme demonstrado na figura 16.

Figura 16 – Diagrama de Sequência das Condições de saída



Fonte: Imagem criada pelo autor.

Após estabelecer essa regra, reconheceu-se a necessidade de validar se o dia não seria feriado e se não haveria atendimento pela Cooperativa. Para realizar essa verificação, implementou-se um bloco anterior com essa condição, seguindo os mesmos princípios. Foi desenvolvido um script para incluir os dias que são feriados não facultativos para a empresa. Caso seja identificado como feriado, o fluxo é encaminhado para o bot fora do horário; caso contrário, segue o fluxo normalmente.

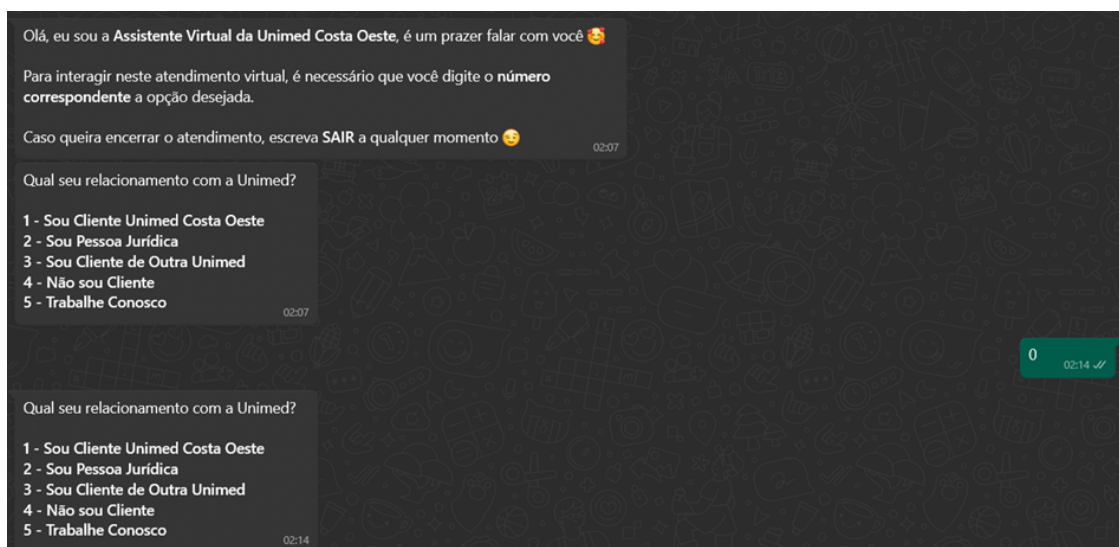
Esses scripts foram desenvolvidos com o apoio de uma área disponibilizada pela plataforma chamada Blip Community (2024). A Blip Community contém diversos

posts da comunidade, compartilhando ideias e dicas para aprimorar e auxiliar o desenvolvimento de fluxos conversacionais. Isso é essencial, pois os desenvolvedores de bots precisam estar sempre atualizados com as melhores práticas para garantir a eficiência no atendimento ao cliente.

Com as regras acima estabelecidas, foi iniciada a criação do menu inicial que os clientes utilizarão para selecionar o tipo de relacionamento que possuem com a Unimed Costa Oeste. Todos os textos apresentados durante as interações foram inseridos após aprovação do departamento de Comunicação, garantindo a manutenção do padrão da marca Unimed e o cumprimento das diretrizes de comunicação com os clientes. Este menu contém 5 opções, que direcionam os clientes a subbots específicos. A seleção de opções é realizada digitando o número correspondente, conforme ilustrado na imagem abaixo. Caso a opção desejada não seja atendida, o menu é apresentado novamente. Além disso, se o cliente desejar encerrar o atendimento, pode digitar a palavra **SAIR** a qualquer momento para finalizar a interação. Há também um tempo de inatividade de 30 minutos definido para cada etapa que requer interação do usuário; se nenhuma ação for realizada dentro desse período, o atendimento será encerrado automaticamente.

Na figura 17 está um exemplo de como o fluxo será apresentado ao cliente ao encaminhar uma mensagem pelo Whatsapp.

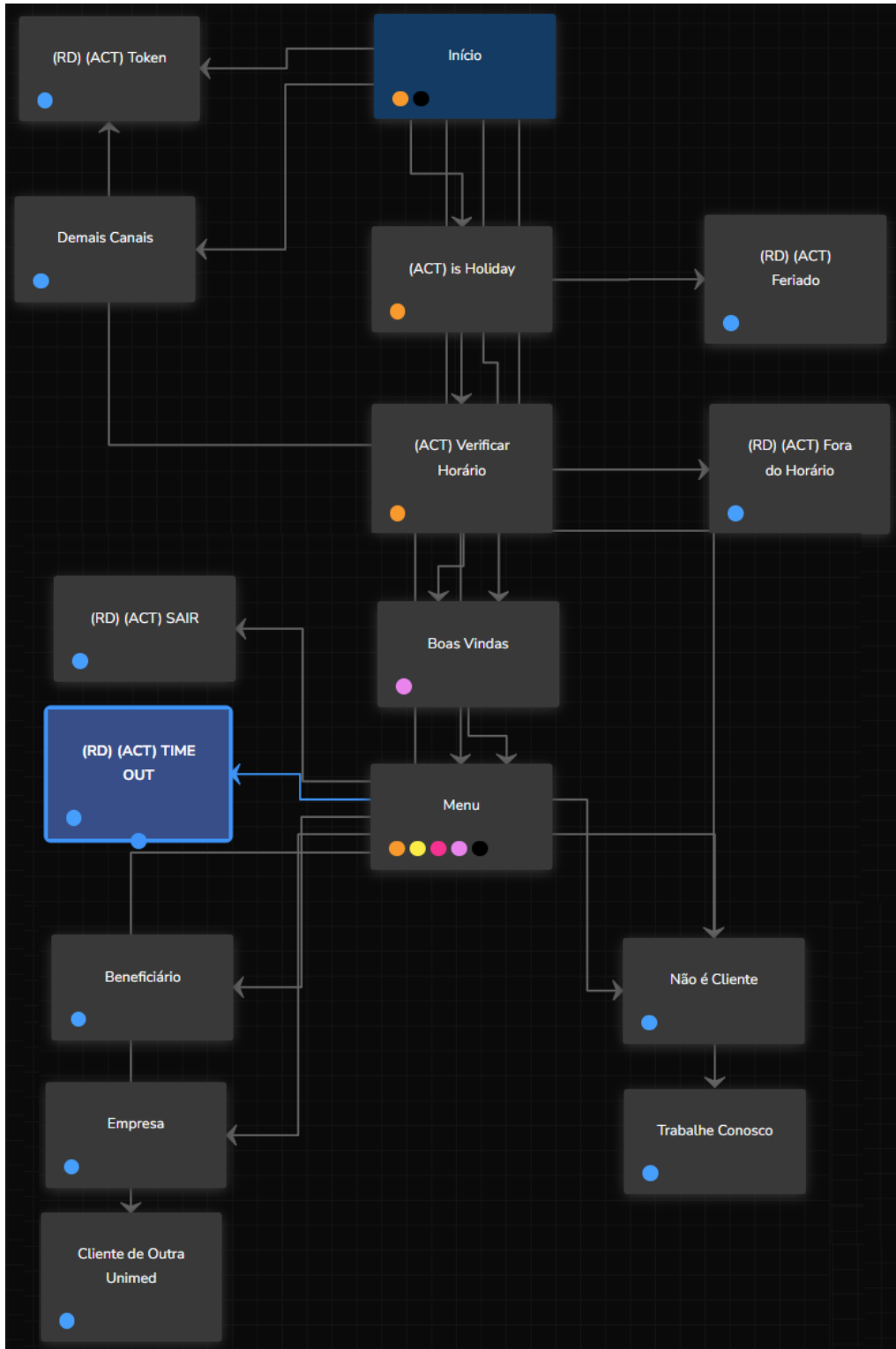
Figura 17 - Interação inicial com o fluxo



Fonte: Imagem criada pelo autor.

A estrutura **(MAIN) BOT** final pode ser visualizada na figura 18, nele consta todos os blocos criados, desde a entrada do usuário, até a chegada ao menu de interação, que após seleção da opção desejada é transferido para um subbot.

Figura 18 - Estrutura do (MAIN) BOT

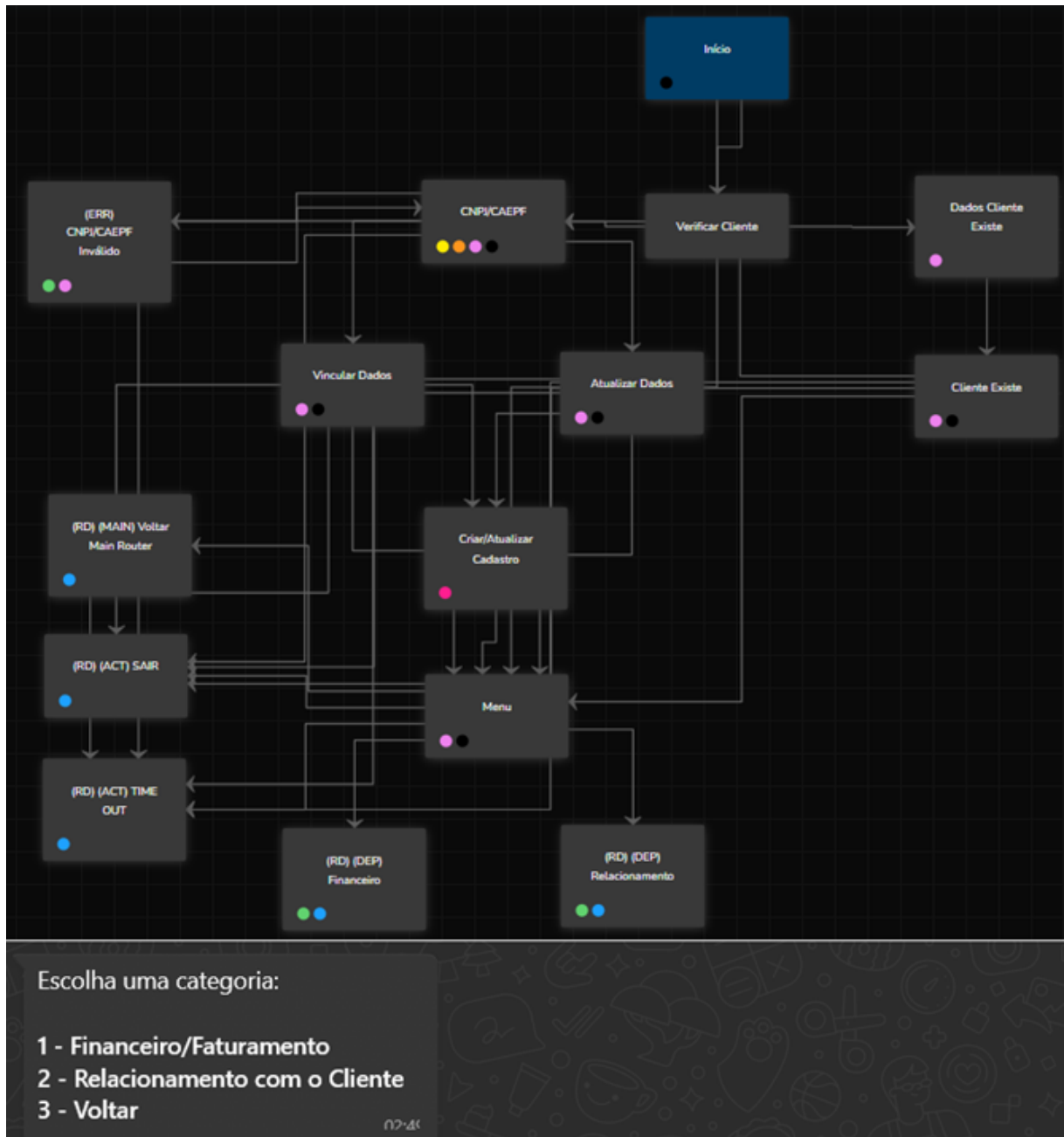


Fonte: Imagem criada pelo autor.

Com a configuração do bot principal concluída, iniciou-se a configuração dos sub-bots do menu principal. A configuração seguiu um padrão semelhante ao do bot principal, com a exceção do bot de beneficiários, que apresenta maior complexidade devido à integração com uma API. Os demais subbots são configurados com menus de interação que permitem aos usuários selecionar o departamento com o qual desejam falar.

No fluxo destinado às contratantes jurídicas, o cliente é solicitado a inserir seu CNPJ/CAEPF. No entanto, como atualmente não há uma API disponível para integração com o sistema de gestão da Unimed, não é possível validar se o número inserido está correto, porém, ele é solicitado para auxiliar o atendente não precisar solicitar este dado na interação com o cliente. O seu menu conta com 5 opções, sendo uma delas para voltar ao menu anterior, como demonstrado na figura 19, juntamente com a estruturação do fluxo conversacional.

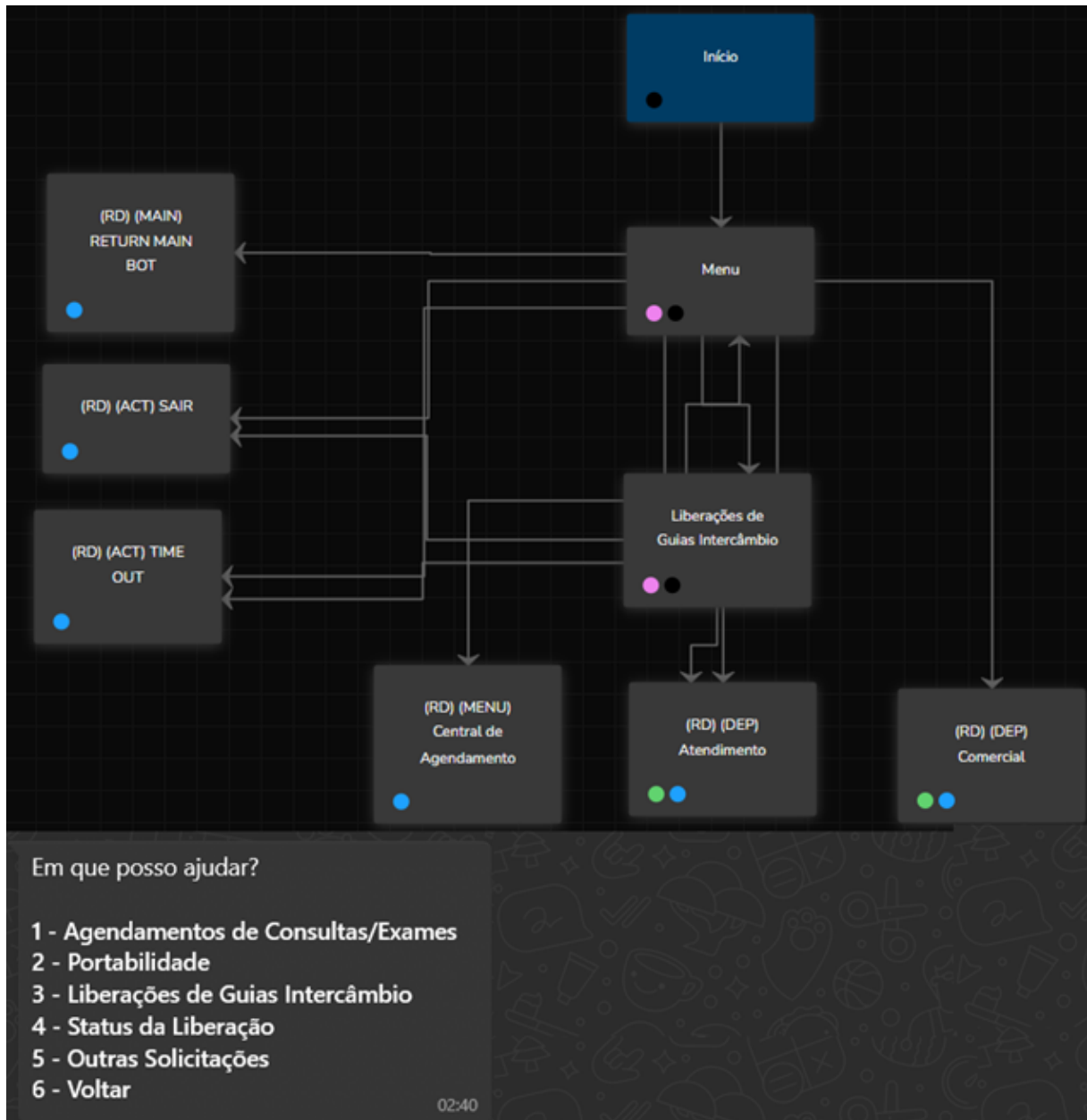
Figura 19 - Estrutura (MENU) EMPRESA



Fonte: Imagem criada pelo autor.

No bot direcionado para clientes de outra Unimed, a sua estrutura ficou simples, possuindo apenas o menu de interação que possui 6 opções, além dos blocos de direcionamento, como é possível visualizar na figura 20.

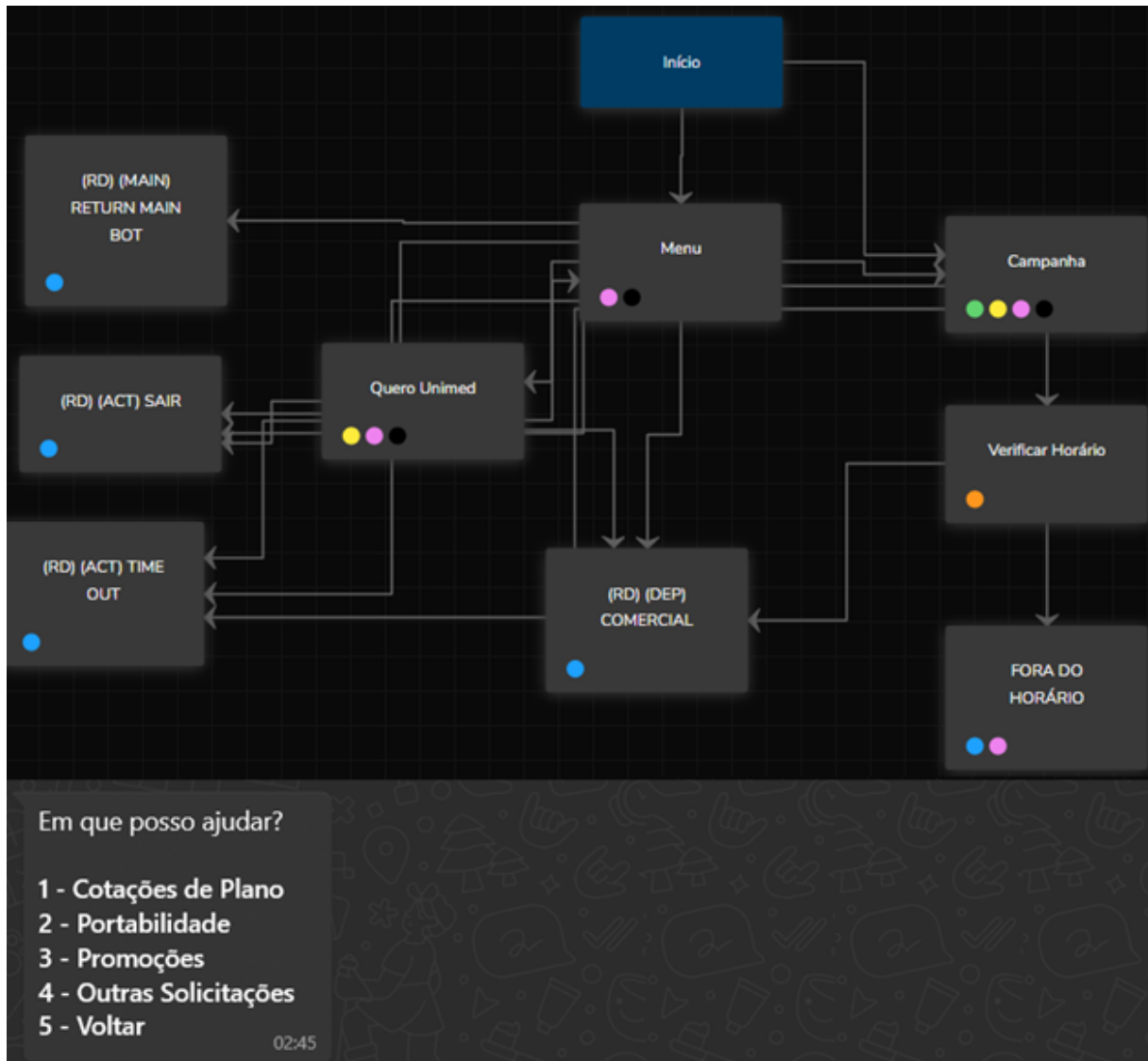
FIGURA 20 - ESTRUTURA (MENU) CLIENTE DE OUTRA UNIMED



Fonte: Imagem criada pelo autor.

A estrutura do fluxo não é cliente seguiu os mesmos padrões do bot anterior, seu menu contém apenas 5 itens e o seu direcionamento ocorrerá apenas para o departamento de comercial. Na figura 21 é possível visualizar a estrutura.

Figura 21 - Estrutura (MENU) NAO E CLIENTE



Fonte: Imagem criada pelo autor.

No fluxo de beneficiários, houve uma complexidade para ser desenvolvido pois foi implementada uma integração via API para validar se a pessoa possui um plano ativo com a Unimed, a fim de assegurar que a pessoa pode usufruir dos serviços da empresa, além de coletar alguns dados para que os atendentes possam utilizar durante o atendimento, para auxiliar e agilizar o atendimento. Nesta validação, se constatarmos que o cliente não possui um plano, ele será direcionado para a área "Não Clientes", onde poderá conversar com um atendente do setor Comercial e tem a possibilidade de gerar uma lead para a equipe de vendas.

O endpoint utilizado da API é fornecido por uma empresa terceira que gerencia o aplicativo Unimed Cliente PR do qual os clientes da Unimed usufruem. Ao entrar no

fluxo, será solicitado para o cliente informar seu CPF, com este dado, foi utilizado a API para assegurar que o plano está ativo e o liberando ao menu deste sub-bot, tendo o fluxo final conforme apresentado na figura 22:

Figura 22 - Fluxo Cliente Unimed

Informe seu CPF (apenas os números): 00:13

00000000055 00:13 ✓

Para confirmarmos a sua identidade, informe a data do seu nascimento:
Ex: 01/12/1990 00:13

00000000055 00:13 ✓

Nome: VICTOR HUGO K CARVALHO
CPF: 00000000055
Número do Cartão Unimed: 011600000
Plano: PLANO I 00:13

Gostaria de vincular estes dados a este contato?

1 - Sim
2 - Não 00:13

1 00:13 ✓

Escolha uma categoria:

1 - Agendamentos de Consultas/Exames
2 - Financeiro/Faturamento
3 - Liberação/Status da Liberação
4 - NAS e Plano Essencial
5 - Relacionamento com o Cliente
6 - Vendas
7 - Voltar ao Início 00:13

Fonte: Imagem criada pelo autor.

Após o preenchimento do CPF pelo usuário, o valor é armazenado na variável **vClientCpf** para ser usado durante o fluxo. Para garantir que apenas números sejam considerados, foi criado um script que remove quaisquer caracteres não numéricos, retornando o valor puro para validação.

Em seguida, um segundo script realiza a validação matemática do CPF. O CPF (Cadastro de Pessoas Físicas) é composto por 9 dígitos base (X1X2X3X4X5X6X7X8X9) e 2 dígitos verificadores (DV1 e DV2), totalizando 11 dígitos no formato XXX.XXX.XXX-DD, onde DD são os dígitos verificadores.

O processo de validação envolve:

Primeiro: Calcular o DV1:

- Multiplique os 9 dígitos base por uma sequência decrescente de 10 a 2, some os resultados, divida por 11 e obtenha o resto.
- Subtraia o resto de 11 para obter o DV1. Se o resultado for 10 ou 11, o DV1 é 0.

Segundo: Calcular o DV2:

- Inclua o DV1 calculado e multiplique os 10 primeiros dígitos por uma sequência decrescente de 11 a 2, some os resultados, divida por 11 e obtenha o resto.
- Subtraia o resto de 11 para obter o DV2. Se o resultado for 10 ou 11, o DV2 é 0.

Se os dígitos verificadores calculados corresponderem aos dois últimos dígitos do CPF, o número é considerado válido.

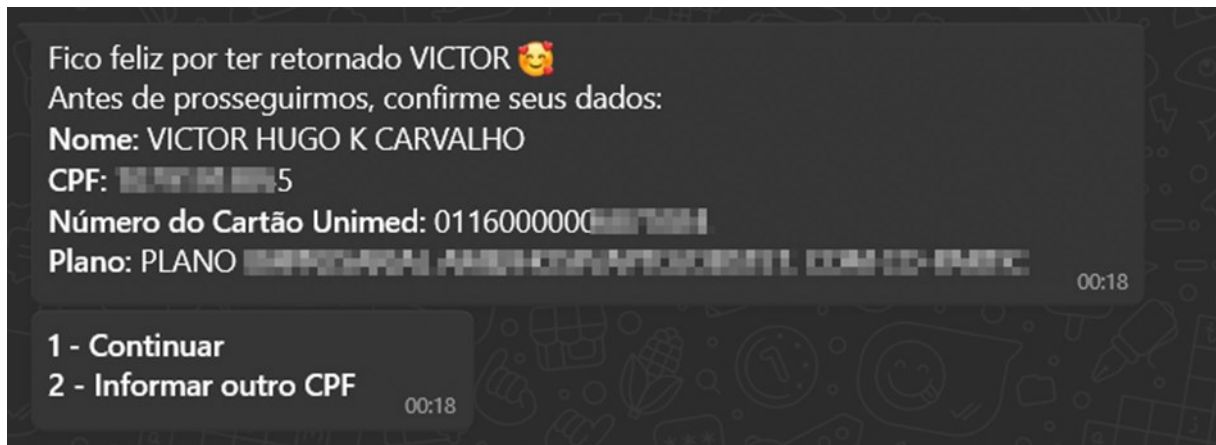
Após confirmar que o CPF está correto, é realizado uma requisição HTTP no endpoint, usando o método GET para verificar se esse CPF está em na base de dados e possui um plano ativo com a cooperativa. Esta API retorna informações como nome completo, nome do plano, data de nascimento, validade do plano e outros detalhes. Se o CPF estiver associado a um plano ativo, é executado um script para identificar quantos planos estão cadastrados, pois um cliente pode ter múltiplos planos, como um particular e outro empresarial. Caso sejam identificados vários planos ativos, o cliente é direcionado a um bloco para selecionar o plano que deseja utilizar durante o atendimento.

Para prevenir tentativas de acesso por terceiros utilizando o CPF dos clientes, após a seleção do cartão Unimed, o usuário deve fornecer sua data de nascimento. Essa medida visa impedir acessos indevidos, mesmo que alguém possua o número do documento. Ao informar a data de nascimento, um script é utilizado para validar o dado inserido com a informação retornada pela API anteriormente. Se as informações coincidirem, o fluxo continua; caso contrário, o cliente é solicitado a inserir a data novamente. Se o erro persistir, o cliente é orientado a fornecer um novo CPF, redirecionar-se para a área de não cliente ou encerrar o atendimento.

Após as validações, para proporcionar uma experiência mais fluida e evitar solicitações repetitivas desses dados, o cliente tem a opção de salvar essas informações na plataforma Blip. Os dados ficam vinculados ao número utilizado para acessar o canal. Se o cliente optar por salvar as informações, nas próximas vezes que acessar o fluxo, o sistema verifica se os dados estão armazenados. Se estiverem, eles são exibidos para confirmação, permitindo que o cliente atualize seus dados, caso necessário, ao inserir o CPF novamente.

Conforme figura 23, é possível visualizar o fluxo da entrada do cliente caso optou por salvar os seus dados na plataforma.

Figura 23 - Dados salvos do cliente



Fonte: Imagem criada pelo autor.

Com os dados confirmados, o cliente é direcionado a um menu onde pode escolher o departamento com o qual deseja conversar. A estrutura deste sub-bot pode ser vista conforme figura 24:

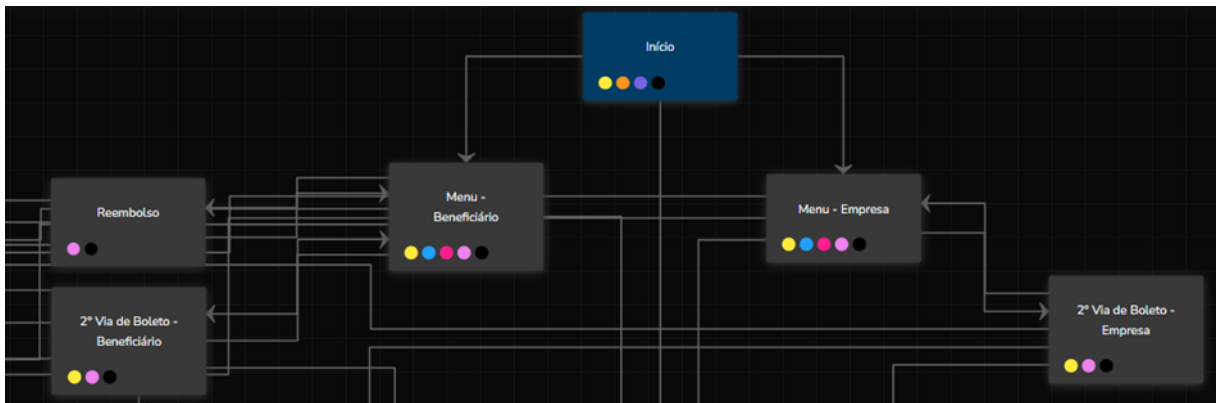
departamento. Dessa forma, os supervisores têm acesso exclusivo aos dados de seu departamento, incluindo monitoramento em tempo real das conversas realizadas pelos atendentes com os clientes, relatórios de tempo de atendimento para fins de métricas, e histórico de conversas realizadas dentro da plataforma. Essa decisão foi tomada não apenas por esses motivos, mas também para estar em conformidade com as políticas de compliance adotadas pela Unimed Costa Oeste, visando mitigar riscos legais, financeiros, operacionais e reputacionais.

Essa abordagem também facilita a criação dos fluxos conversacionais, já que, se todos os departamentos fossem gerenciados dentro de um único bot, seria necessário implementar diversas regras específicas, dado que cada departamento possui seu próprio menu e perguntas relacionadas aos seus processos. Além disso, em caso de manutenção, essa estrutura modular facilita ajustes e reduz as chances de bugs que possam impactar outros processos, pois cada fluxo está isolado em seu próprio bot.

A criação do fluxo para os departamentos foi baseada no feedback obtido nas reuniões realizadas com cada área, permitindo identificar as principais dúvidas e necessidades específicas de cada departamento e implementar essas informações no sistema. Os bots desenvolvidos foram para os departamentos de Atendimento, Comercial, Financeiro e Relacionamento com o Cliente. Em cada fluxo, foram criados menus que pedem ao cliente para selecionar uma das opções apresentadas. Caso nenhuma das opções corresponda à dúvida do cliente, foi incluída uma opção adicional chamada **Outras Solicitações**, destinada a questões mais específicas ou quando o cliente não tem certeza sobre qual opção é a mais adequada.

A estrutura desses bots é basicamente a mesma, diferenciando-se principalmente nos menus. Essas variações ocorrem porque cada departamento lida com processos diferentes e, em alguns casos, atende tanto clientes pessoa física (PF) quanto pessoa jurídica (PJ). Por exemplo, nos departamentos Financeiro e de Relacionamento com o Cliente, foi necessário criar menus separados para cada tipo de cliente, já que as dúvidas de clientes PF e PJ são distintas, como ilustrado na figura 25 do bot do Financeiro.

Figura 25 - Estrutura do menu do bot (DEP) FINANCEIRO



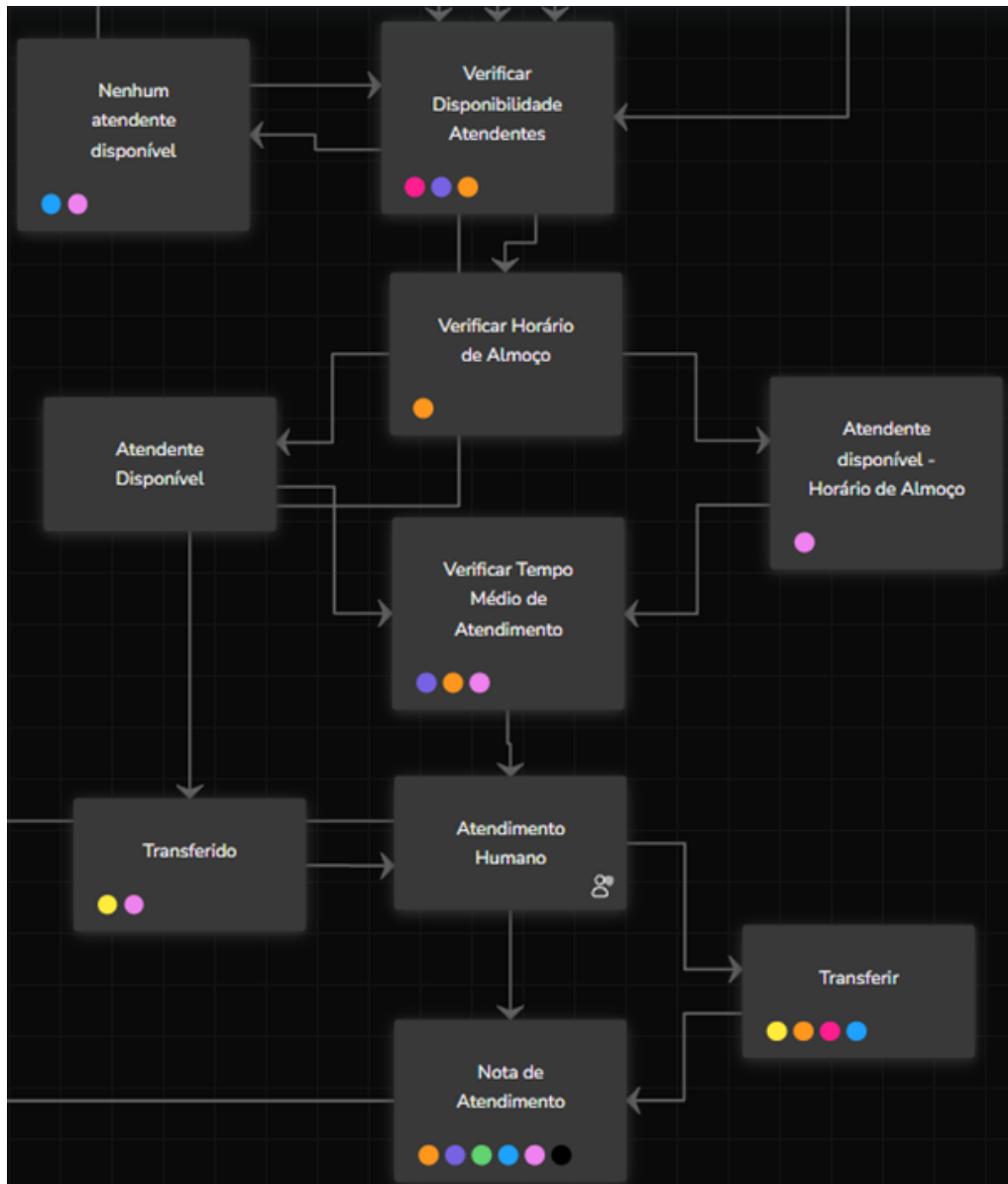
Fonte: Imagem criada pelo autor.

A estrutura desse fluxo é similar aos demais. Após o cliente selecionar a opção desejada, ele será direcionado para conversar com um atendente humano. No entanto, para garantir que o cliente será atendido, é utilizada uma API da Blip para verificar se há atendentes ativos na plataforma naquele momento. Se a resposta dessa requisição HTTP for positiva, uma segunda verificação é feita através de um script JSON para determinar se é horário de almoço, seguindo as mesmas regras de horário de atendimento da empresa configuradas no bot principal. Com base nessa validação, o cliente pode receber uma de duas mensagens: uma indicando que um atendente foi localizado e o atendimento ocorrerá em instantes, ou outra informando que o tempo de espera na fila pode ser um pouco maior devido ao número reduzido de colaboradores durante o horário de almoço.

Em seguida, a API da Blip é utilizada novamente para informar ao cliente o tempo médio de espera na fila. Esse dado é obtido de uma tela de monitoramento dentro do bot, que calcula o tempo com base no total de clientes que ficaram aguardando até serem atendidos.

Se todos os passos realizados estiverem corretos, o cliente será colocado em uma fila de atendimento para conversar com um atendente humano. Após o término do atendimento, o cliente terá duas opções de direcionamento dentro do fluxo: ser redirecionado para outro departamento ou fornecer uma avaliação sobre o atendimento recebido, com o objetivo de medir a qualidade do serviço prestado. O fluxo ficou estruturado conforme figura 26:

Figura 26 - Estrutura de validação para o atendimento humano



Fonte: Imagem criada pelo autor.

Algumas configurações adicionais foram necessárias nos bots para definir regras relacionadas ao atendimento humano, como o tempo limite para que o atendimento seja encerrado automaticamente por inatividade, considerando tanto a inatividade do cliente quanto do atendente. Neste caso, foi configurado que, se o cliente não responder ao atendente dentro de uma hora após o envio da última mensagem, a interação será encerrada. Para reduzir possíveis insatisfações por parte do cliente, um alerta em formato de texto é enviado automaticamente pelo canal de mensagens 15 minutos antes do tempo limite, avisando que, caso não haja interação nos próximos minutos, a conversa será finalizada.

Existe também a opção de encerrar o atendimento se o atendente não responder ao usuário, mas, para proteger a imagem da marca e conforme orientação do departamento de Comunicação da Cooperativa, essa opção não foi ativada. Dessa forma, evita-se que o cliente tenha que passar por todo o fluxo novamente, o que poderia gerar insatisfação e tornar a ferramenta um processo moroso e ineficiente. Na figura 27 é possível visualizar as configurações disponibilizadas pelo sistema.

Figura 27 - Configuração do encerramento automático de tickets

Encerramento automático de tickets
Encerre automaticamente os tickets por inatividade

Regras de encerramento
Defina as regras para encerrar automaticamente os tickets inativos.

Encerrar ticket quando o tempo de inatividade do cliente for maior que:

Tempo de inatividade: 1
Unidade: Horas

O tempo deve ser um valor maior que 0.

Encerrar ticket apenas se o primeiro atendimento já tiver ocorrido

Não encerrar se cliente estiver aguardando resposta do atendente

Remover automaticamente o ticket da tela quando ele for encerrado

Enviar alerta de inatividade para o cliente
Crie uma mensagem para alertar o cliente antes do encerramento automático do ticket

Mensagem
Recado importante: seu atendimento será encerrado automaticamente em 15 minutos, caso não seja identificada nenhuma interação. Para finalizar o atendimento imediatamente digite SAIR.

Quanto tempo antes do encerramento o cliente deve receber a mensagem?

Tempo: 15
Unidade: Minutos

Esse valor não pode ser maior que o tempo de inatividade

Fonte: Imagem criada pelo autor.

Com os bots já criados, foi necessário conectar toda a estrutura dentro do Router. Para isso, foi adicionado os subbots como serviços no Router, permitindo que a transferência de clientes entre os diferentes bots fosse possível. Após conclusão desta configuração, a estrutura final pode ser vista na figura 11, onde o fluxo se inicia no Router, onde estão conectados os canais e amarrado os bots, passando para o bot principal definido, onde após interação do usuário será redirecionado para um dos menus disponíveis e, dos menus poderá ser transferido a um departamento, onde irá selecionar a sua dúvida e entrar na fila para conversar com um atendente humano. Ao fim da interação com o atendente humano, poderá ser transferido para outro departamento ou finalizar o processo tendo a opção de dar uma nota referente ao atendimento recebido.

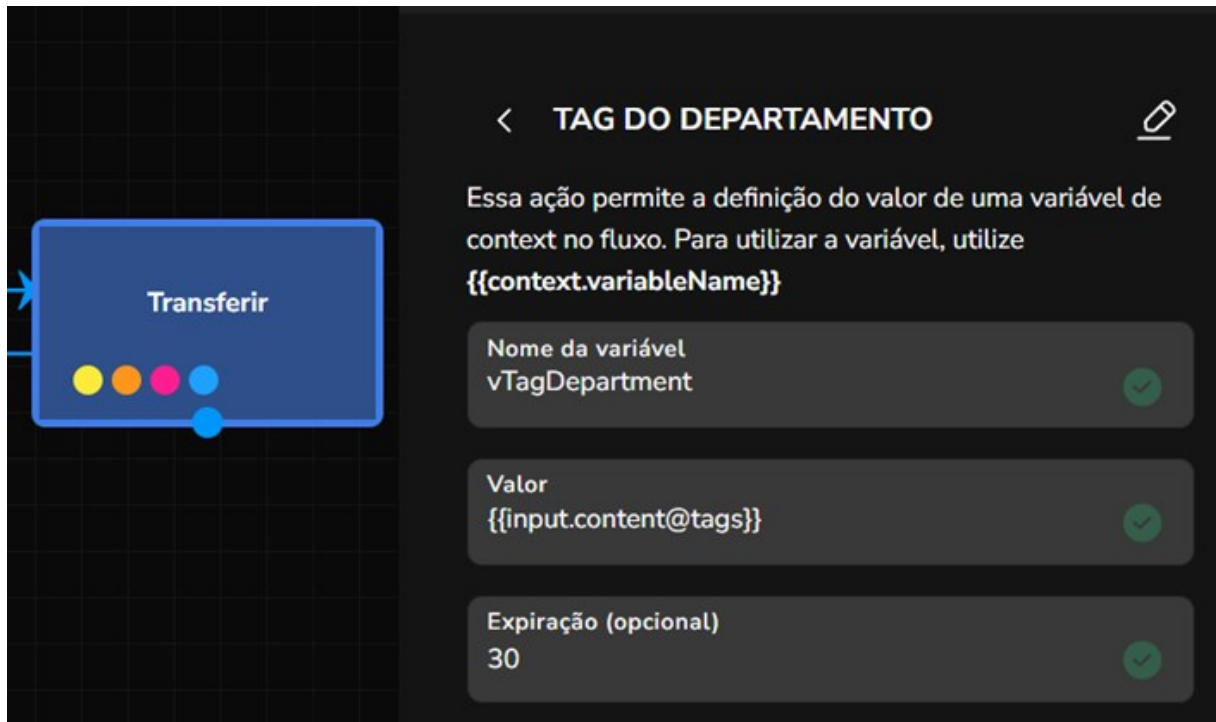
4.5.1 Personalização da funcionalidade de Transferência de Atendimentos

Após a criação dos sub-bots e a realização dos testes iniciais para validar os fluxos da ferramenta, foi identificado que a funcionalidade de transferir clientes entre diferentes subbots não era suportada nativamente pela plataforma, pois a ferramenta permite a transferência de um cliente para outro atendente, mas apenas dentro do mesmo bot. Diante dessa limitação, algumas pesquisas foram iniciadas e houve consultas com o gerente do contrato na Blip para compreender melhor o funcionamento da plataforma. Durante a reunião, a confirmação da restrição que havia sido identificada nos testes foi positiva, no entanto, ofereceu algumas sugestões de como realizar adaptações para que a ferramenta pudesse atender às nossas necessidades sem comprometer a estrutura planejada. Pela regra da plataforma, para realizar uma transferência para outro atendente ou fila, seria necessário centralizar o atendimento humano em um único bot, o que iria contra as políticas de Compliance da Cooperativa, e essa abordagem centralizada comprometeria o acompanhamento e as métricas de atendimento, que seriam contabilizados apenas em um único bot.

Com base nas sugestões recebidas, foi decidido utilizar APIs ou a biblioteca de variáveis da plataforma Blip para identificar quando um atendimento deveria ser transferido. Dessa forma, poderíamos criar regras específicas para direcionar o cliente ao departamento correto. Durante os testes adicionais na plataforma, identificou-se a possibilidade de utilizar Tags, estas que os atendentes selecionam ao finalizar um atendimento na plataforma Blip Desk, possibilitando criar regras que direcionassem o cliente ao departamento apropriado.

Seguindo essa lógica, dentro dos bots de atendimento humano, criou-se um bloco denominado "Transferir". Neste bloco, é feita a validação da Tag associada ao atendimento. Foi implementado uma ação que captura o valor da Tag, armazenado em um JSON gerado automaticamente pelo sistema ao finalizar o atendimento. Esse JSON inclui informações como o nome do atendente, os dados de início e fim do atendimento, e o valor da Tag, que é então armazenado em uma variável chamada **vTagDepartment**, conforme ilustrado na figura 28.

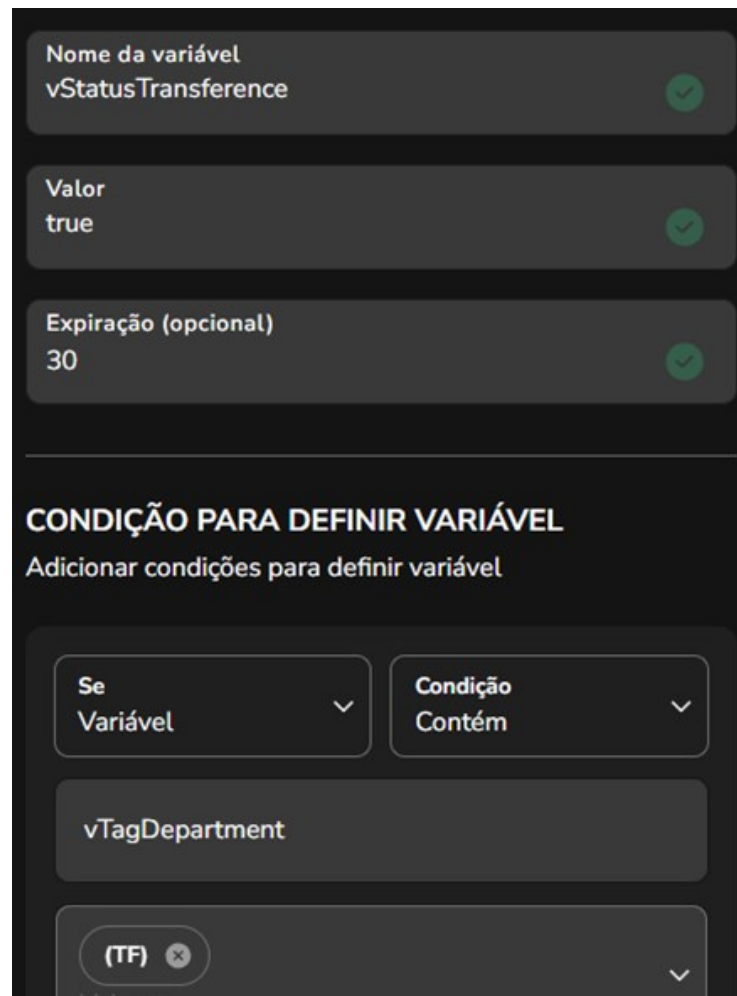
Figura 28 - Tag do departamento



Fonte: Imagem criada pelo autor.

Para determinar se é necessário transferir um usuário, foi utilizado tags do qual os atendentes informam com uma nomenclatura específica, que inclui "(TF)" seguido do nome do departamento. Dessa forma, a regra configurada dentro do subbot do departamento verifica se a tag informada contém "(TF)". Se essa condição for atendida, identificamos que se trata de uma transferência e então o contato é direcionado para outro subbot do fluxo, conforme demonstrado na figura 29. Caso a regra seja verdadeira, foi criada uma variável chamada **vStatusTransference** onde é atribuído a ela o valor **true**. Além disso, é definido um período de expiração de 30 segundos para essa variável, garantindo que seu valor seja resetado antes de passar pelo mesmo bloco novamente, evitando assim loops indesejados.

Figura 29 - Variável vStatusTransference



Nome da variável
vStatusTransference ✓

Valor
true ✓

Expiração (opcional)
30 ✓

CONDIÇÃO PARA DEFINIR VARIÁVEL
Adicionar condições para definir variável

Se Variável ▼ Condição Contém ▼

vTagDepartment

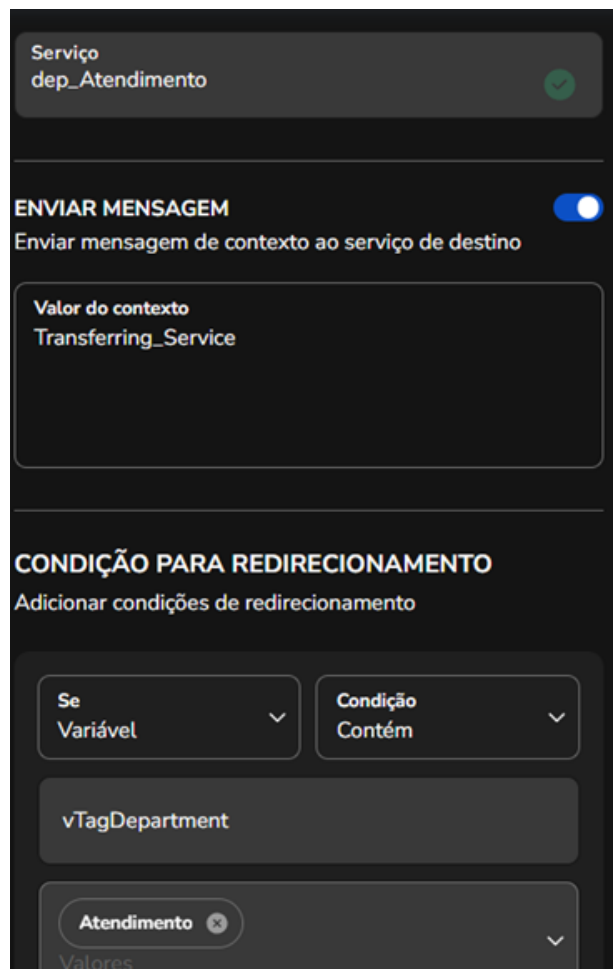
(TF) ✕ ▼

Fonte: Imagem criada pelo autor.

Houve a implementação de algumas versões para chegar nesta versão final para que atendesse as necessidades. Nas primeiras versões era aplicado a validação dentro de cada subbot dos departamentos, individualmente, no entanto, essa abordagem exigia a criação de diversas regras para cada departamento e, caso fosse necessário adicionar um novo departamento, seria necessário acessar cada subbot para incluir novas regras, tornando o processo bastante trabalhoso e demorado. Para resolver esses problemas, foi implementada na última versão um padrão do qual simplificou todo o processo: foi criado um novo sub-bot chamado "(ACT) TRANSFER MANAGER", cuja função é identificar e transferir o usuário para o departamento correto. Com essa solução, não foi mais necessário realizar a validação dentro de cada sub-bot dos departamentos, o que facilitou a manutenção e permitiu maior flexibilidade.

O único propósito do "(ACT) TRANSFER MANAGER" é gerenciar essa regra de transferência. Ele recebe a informação da tag do atendimento e, em um bloco chamado "Transferir", cria uma ação para cada departamento, validando o valor da variável que contém a informação do departamento. Se a condição for atendida, o usuário é redirecionado para o sub-bot correspondente, passando o valor "Transferring_Service". Esse dado é utilizado pelo subbot do departamento de destino para direcionar o cliente diretamente para a fila de atendimento, ignorando o fluxo de seleção de menu. A figura 30 ilustra a condição de redirecionamento: se a tag selecionada contiver a informação "Atendimento", o usuário será redirecionado para o sub-bot "(DEP) ATENDIMENTO".

Figura 30 - Condição de redirecionamento do (ACT) TRANSFER MANAGER



Fonte: Imagem criada pelo autor.

Neste modelo criado, a manutenção necessária é mínima e consiste apenas em acessar os sub-bots dos departamentos para adicionar novas tags caso novos departamentos sejam incorporados à plataforma. Isso ocorre porque o subbot "(ACT)

TRANSFER MANAGER" já possui regras configuradas para todos os departamentos existentes na cooperativa. Na figura 31, é possível visualizar as tags que os atendentes selecionam na tela do Blip Desk, como as tags podem ser utilizadas para diferentes finalidades, aquelas destinadas exclusivamente para transferências são identificadas pelo símbolo "(TF)" no início, um acrônimo para "Transferência".

Figura 31 - Tags utilizadas pelos atendentes via Blip Desk

Fonte: Imagem criada pelo autor.

4.5.2 Desafios na Implementação do Chatbot

Um dos maiores desafios enfrentados durante a execução do projeto foi adaptar a ferramenta para permitir a transferência de atendimentos de um sub-bot para outro. Embora a ferramenta tivesse essa funcionalidade, ela não atendia às necessidades específicas de transferências entre bots que o projeto exigia. Para superar esse obstáculo, foi necessário criar um sub-bot mediador, definir variáveis e utilizar as APIs da Blip para que um atendente do departamento de Atendimento pudesse transferir um cliente para o departamento Financeiro, por exemplo. Esse processo exigiu um grande esforço, com várias tentativas e pesquisas, até que se chegasse a uma solução viável para o projeto final, que manteve a estrutura original planejada, com cada departamento utilizando seu próprio subbot.

Outro desafio técnico enfrentado foi a implementação da funcionalidade de geração de token, que foi integrada à ferramenta após sua inauguração. Essa melhoria foi projetada para facilitar o processo de consulta nos consultórios médicos, especialmente para clientes que não possuíssem o aplicativo de geração de tokens ou que não tivessem o cartão Unimed em mãos.

A implementação envolveu a integração com APIs para validar se o cliente estava ativo e para identificar se possuía mais de um cartão. Nesse caso, o cliente poderia selecionar o cartão desejado para gerar o token. Além disso, foi necessário seguir regras específicas estabelecidas no aplicativo: se o titular possuísse dependentes, apenas os cartões dos dependentes com idade inferior a 18 anos ou superior a 65 anos seriam exibidos. Dependentes com idades fora desses critérios não apareciam na lista, e um dependente poderia visualizar apenas o seu próprio cartão sem acesso aos cartões dos demais membros da família.

Além desse desafio técnico, houve também a dificuldade de gerenciar um projeto dessa magnitude, especialmente por ser a minha primeira experiência lidando com a responsabilidade de coordenar um projeto completo. Isso envolveu etapas iniciais, como o contato com fornecedores, a apresentação do projeto para os coordenadores da empresa e o auxílio na escolha da ferramenta adequada, além de toda a gestão da criação e definição da estrutura e lógica a ser seguida, alinhada aos padrões da marca. Houve também o desafio de conduzir reuniões e treinamentos com as áreas envolvidas e posteriormente para todos os departamentos, explicando o funcionamento da ferramenta, recebendo feedbacks das equipes e implementando melhorias no fluxo que não tinham sido previstas ou que haviam passado despercebidas.

4.6 Implantação do Chatbot

A etapa de treinamento da equipe de atendimento e lançamento do chatbot foram essenciais para garantir uma implementação bem-sucedida e um atendimento automatizado eficaz. Esta etapa envolveu a capacitação da equipe de atendimento, que iria interagir diretamente com os clientes utilizando a ferramenta. A preparação adequada da equipe é fundamental para assegurar que os atendentes compreendam como navegar pelos fluxos conversacionais, interpretar corretamente as mensagens automatizadas e assumir o controle da conversa quando necessário para oferecer um suporte personalizado e eficiente.

Durante o treinamento, os atendentes foram instruídos a operar o sistema Blip de forma completa, o que inclui o uso de suas ferramentas de monitoramento e análise para melhorar continuamente o atendimento ao cliente. Essas ferramentas permitem

que a equipe identifique padrões de comportamento do usuário, ajuste as respostas automatizadas conforme necessário e maximize a eficiência do atendimento.

O lançamento da ferramenta ocorre somente após uma fase de testes para garantir que todos os aspectos operacionais e técnicos estejam funcionando corretamente para que a equipe de atendimento esteja bem preparada para utilizar a ferramenta, garantindo uma experiência positiva e eficaz para os usuários. A equipe de atendimento passou por treinamentos exclusivos, específicos para os departamentos que tiveram fluxos criados dentro da plataforma. Essa abordagem direcionada evita a necessidade de treinamentos gerais e permite uma personalização do conteúdo conforme as necessidades de cada departamento. Durante esses treinamentos individualizados, foi apresentado o fluxo completo que o cliente percorreria, as opções que poderiam ser selecionadas, e como a estrutura dos blocos de diálogo foi desenvolvida com base nas especificações iniciais do projeto.

Além disso, o treinamento inclui a operacionalização prática do atendimento por meio da plataforma Blip Desk. Este componente é vital, pois cobre todas as funcionalidades fornecidas pelo sistema, desde a entrada do cliente na plataforma até o momento em que ele é transferido para um atendente humano. Para garantir a preparação completa da equipe, foram realizados testes práticos simulando cenários reais de atendimento, nos quais os colaboradores puderam praticar desde a navegação na plataforma até a entrada na fila de atendimento para, então, ser atendido por um agente humano.

Após a conclusão dos treinamentos, foi estabelecido que testes adicionais dos fluxos seriam realizados antes do lançamento oficial para os clientes. Esses testes, que duraram uma semana, foram cruciais para identificar e corrigir erros que surgiram durante as interações dos usuários, além de incorporar sugestões de melhorias. Também foram solicitadas alterações em algumas perguntas dentro dos blocos de diálogo de certos departamentos, para garantir uma melhor compreensão por parte dos usuários e adicionar opções que não haviam sido previstas nos primeiros encontros.

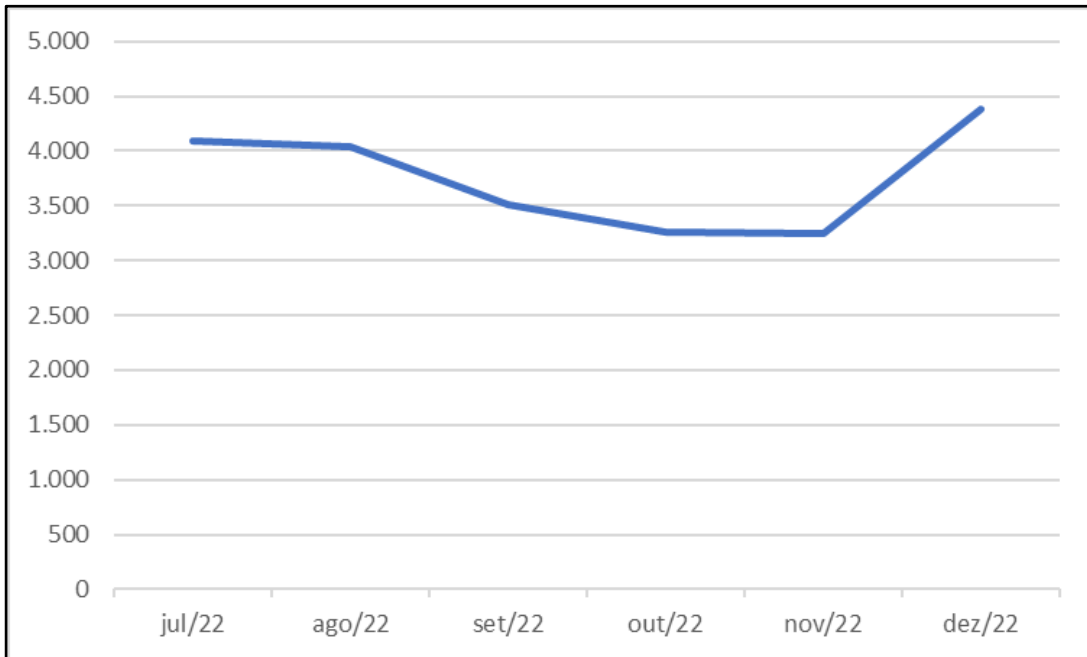
Ao final do período de testes, foi realizada uma reunião com os supervisores dos departamentos envolvidos para avaliar se a ferramenta atendia às expectativas

estabelecidas. Com o feedback verbal positivo obtido nas reuniões, foi definida uma data para o lançamento oficial da plataforma. Antes do lançamento, foi planejada uma campanha de marketing nas redes sociais para informar os clientes e colaboradores sobre a nova possibilidade de atendimento oferecida pela Unimed Costa Oeste. Com todas as etapas de planejamento e validação concluídas, a plataforma foi lançada com sucesso, proporcionando uma nova experiência de atendimento digital para os usuários.

4.7 Análise do Período Pós-Implantação do Chatbot

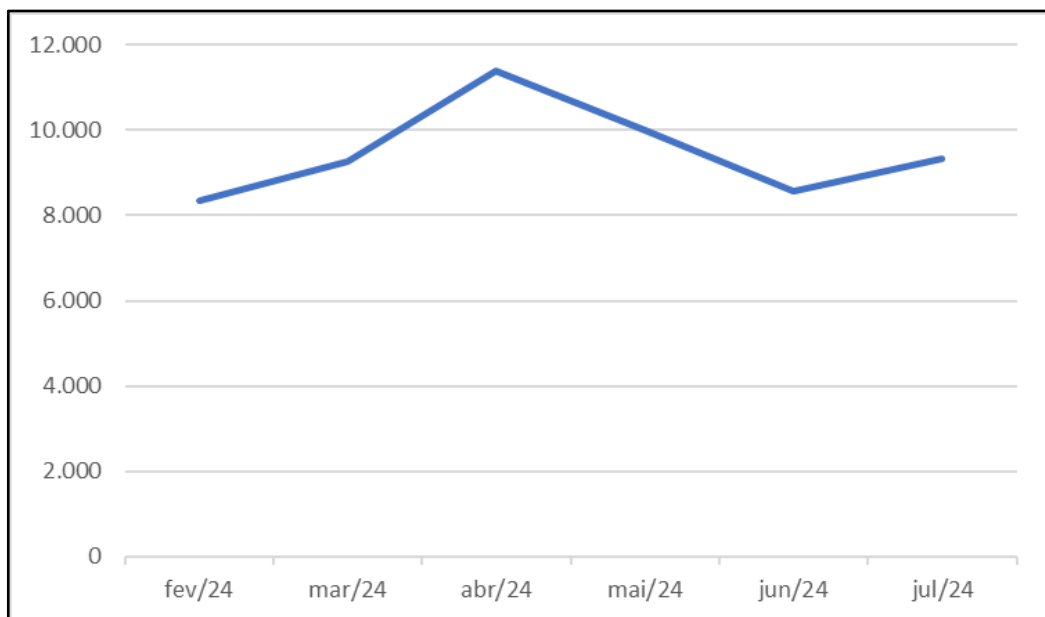
O monitoramento inicial após a implantação foi realizado através de uma análise detalhada do uso geral da plataforma e de cada departamento específico. Foram observados o caminho percorrido pelo cliente e as interações realizadas em cada fluxo. Além disso, foi monitorado o tempo médio de atendimento, com alertas configurados para identificar clientes que estavam aguardando por um período excessivo. Essa abordagem permitiu prestar suporte imediato aos atendentes em caso de dúvidas ou dificuldades, assegurando um atendimento eficiente e contínuo. De maneira geral, a implementação foi bem-sucedida e bem recebida pelos beneficiários, registrando um volume significativo de interações nos primeiros meses.

Devido a um problema na plataforma, não foi possível extrair a quantidade exata de entradas nos meses iniciais. Entretanto, com base nos dados coletados a partir de seis meses após o início do uso da plataforma, observou-se uma média de 3.700 acessos mensais. Na figura 32 pode ser visto um gráfico que demonstra a média de acessos no período de Julho de 2022 até Dezembro de 2022.

Figura 32 - Média de acessos no período de julho à dezembro de 2022

Fonte: Imagem criada pelo autor.

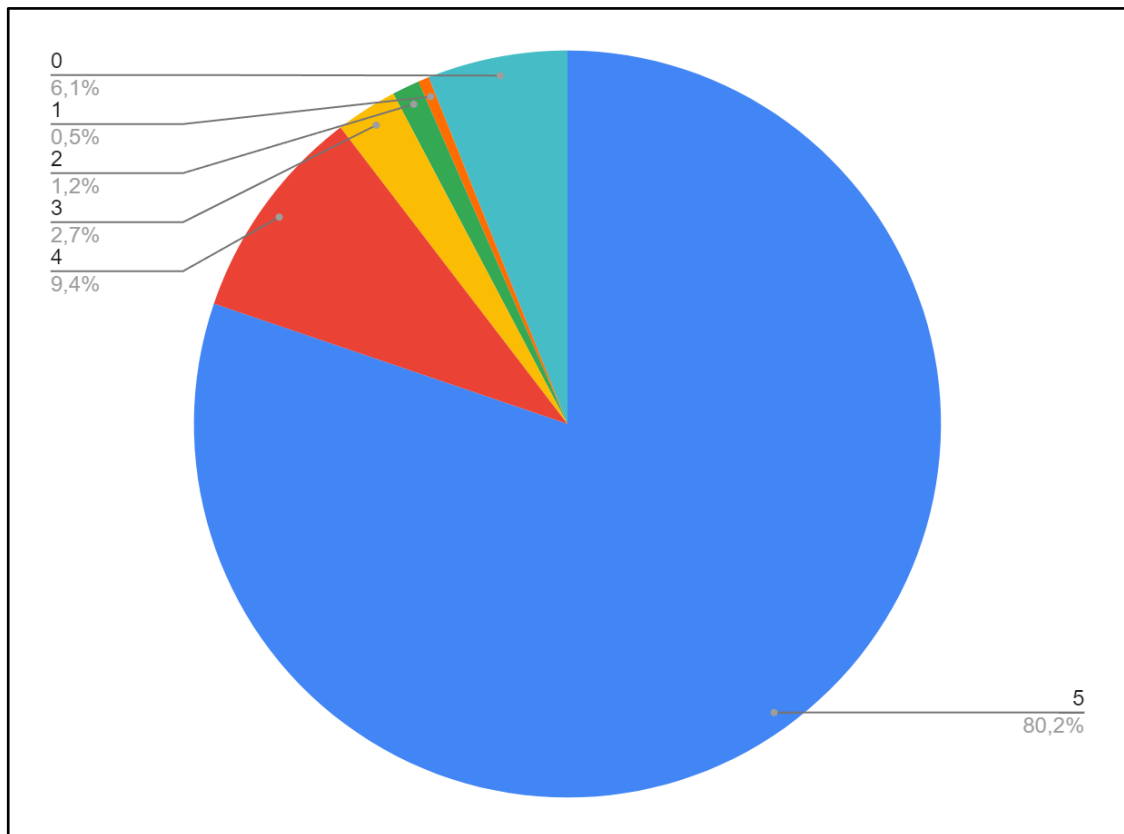
Ao comparar os dados do início da utilização da plataforma conforme gráfico 1 com os dados do período de Fevereiro de 2024 até Julho de 2024, essa média aumentou significativamente para 9.500 acessos mensais. Esses números refletem um crescimento substancial na adoção e utilização da ferramenta, conforme ilustrado na figura 33.

Figura 33 - Média de acessos no período de fevereiro à julho de 2024

Fonte: Imagem criada pelo autor.

Em relação à satisfação com o atendimento, as métricas apresentadas a seguir são dos departamentos de Atendimento e Relacionamento com o Cliente, que são dois setores que mais recebem interações de usuários. No departamento de Atendimento, no período de Fevereiro até Julho de 2024, foram realizados cerca de 13.800 atendimentos diretos com atendentes humanos e 2.952 clientes deixaram uma avaliação numa escala de 0 a 5. Ao analisar as avaliações, verificamos que 89,6% dos clientes que deram uma nota 4 ou 5, indicaram um alto nível de satisfação com o atendimento prestado. Por outro lado, notas abaixo de 2, que sinalizam insatisfação, representam apenas 7,8% das avaliações. Estes valores podem ser vistos na figura 34.

Figura 34 - Percentual da nota de atendimento do departamento de Atendimento



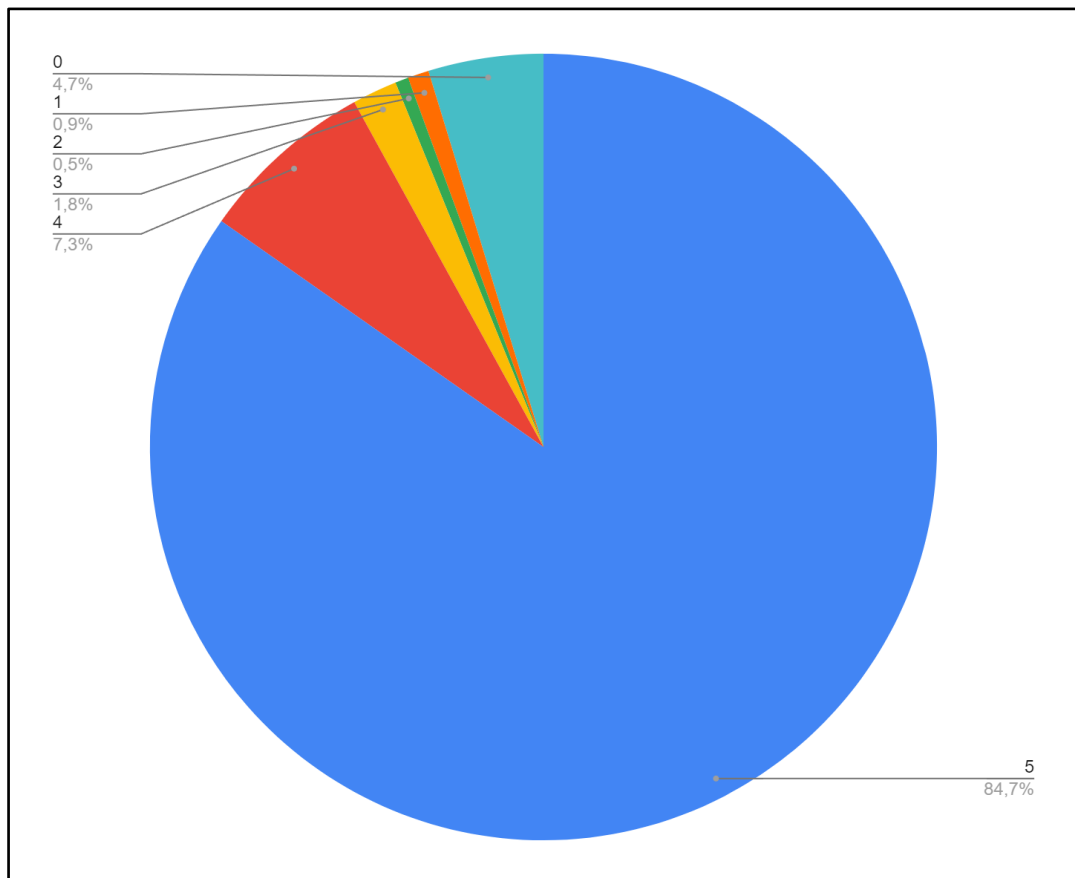
Fonte: Imagem criada pelo autor.

O tempo médio de atendimento registrado durante este período foi de 2 horas e 52 minutos. Esse valor reflete a alta volumetria de conversas e outras atividades que o departamento gerência, e é um aspecto que deve ser constantemente otimizado para reduzir o tempo de atendimento.

Além disso, parte desse tempo médio é atribuída a situações em que o atendimento permanece aberto por vários dias. Isso ocorre, por exemplo, quando o cliente está aguardando uma liberação de guia e para evitar que o cliente tenha que entrar na fila novamente ou para não enfrentar dificuldades ao tentar entrar em contato posteriormente, opta-se por manter o ticket em aberto. Dessa forma, assim que um retorno estiver disponível, podem encaminhar imediatamente ao beneficiário. No entanto, essa prática impacta o tempo médio de atendimento.

No departamento de Relacionamento com o Cliente, no mesmo período de Fevereiro à Julho de 2024, foram registrados em torno de 10 mil atendimentos dentro da plataforma. Desses atendimentos, apenas 1.841 clientes forneceram uma avaliação. Conforme ilustrado na figura 35, as avaliações com notas 4 e 5 representaram 92% do total, já a nota 0 representou um percentual de 4,7%.

Figura 35 - Percentual da nota de atendimento do departamento de Relacionamento com o Cliente



Fonte: Imagem criada pelo autor.

O tempo médio de atendimento registrado foi de 1 hora e 57 minutos. Embora esse tempo seja considerado satisfatório, os departamentos estão constantemente empenhados em aprimorar seus processos para aumentar a eficiência e, conseqüentemente, melhorar a satisfação do cliente.

5 CONCLUSÃO

A implantação do chatbot na Unimed Costa Oeste representou um avanço positivo na automação do atendimento aos clientes da empresa, com impactos positivos na eficiência e na qualidade dos serviços prestados.

Com já citado várias vezes, este trabalho teve como objetivo relatar o processo completo de implantação de um chat, desde o levantamento de dados até a análise dos resultados, passando pela seleção, configuração e integração da ferramenta como ambiente operacional da empresa.

Através do levantamento de dados, foi possível identificar as demandas da empresa, o que orientou a escolha da plataforma mais adequada.

A seleção, configuração e implantação permitiram que o chatbot fosse adaptado às necessidades da empresa, possibilitando uma integração adequada com os sistemas existentes da Unimed Costa Oeste.

A análise dos dados pós-implantação revelou ganhos em termos de agilidade no atendimento e satisfação dos usuários, confirmando que a automação é uma estratégia eficaz para atender à demanda crescente por serviços rápidos e eficientes.

No entanto, o trabalho também destacou algumas limitações, como a necessidade de constante atualização e manutenção da ferramenta para garantir sua eficácia a longo prazo.

A experiência da Unimed Costa Oeste pode servir como um modelo para outras operadoras de planos de saúde e organizações que buscam modernizar seu atendimento ao cliente.

5.1 Trabalhos Futuros

Como sugestão de trabalhos futuros, pode-se indicar:

- Implementar um Chatbot se baseando em Machine Learning e IA.
- Implementar agendamento de consultas pelo Chatbot.
- Implementar integrações com serviços de saúde, como liberação e acompanhamento de guias, resultados de exames, entre outros.

REFERÊNCIAS

RUSSELL, Stuart; NORVIG, Peter. *Inteligência Artificial: Estruturas e Estratégias para Solução Complexa de Problemas*. 3. ed. Rio de Janeiro: Elsevier, 2016.

BISHOP, Christopher M. *Reconhecimento de Padrões e Aprendizado de Máquina*. São Paulo: Editora Campus, 2009.

BLIP. Chatbot: o que é, como funciona, benefícios e cases, 16 set. 2021. Disponível em: <<https://www.blip.ai/blog/chatbots/chatbot/#>>. Acesso em: 25 jun. 2024

BLIP. Tudo sobre NLP: o que é? Quais os desafios? bigblog, 2 ago. 2022. Disponível em: <<https://www.blip.ai/blog/tecnologia/nlp-processamento-linguagem-natural/>>

CARVALHO, A. C. P. DE L. F. DE et al. *Inteligência artificial: uma abordagem de aprendizado de máquina*. 2a. ed. Rio de Janeiro: LTC, 2021.

GUITARRA, P. *Inteligência Artificial*. Escola Brasil, [s.d.]. Disponível em: <<https://brasilecola.uol.com.br/informatica/inteligencia-artificial.htm>>

KOTLER, P.; KARTAJAYA, H.; SETIAWAN, I. *Do tradicional ao digital*. Rio de Janeiro: Sextante, 2017.

LAS CASAS, A. L. *Qualidade total em serviços: conceitos, exercícios, casos práticos*. 7. ed. [s.l.] Editora Atlas SA, 2019.

MARCONI, M. DE A.; LAKATOS, E. M. *Técnicas de pesquisa: planejamento e execução de pesquisa; amostragens e técnicas de pesquisa; elaboração, análise e interpretação de dados*. Em: *Técnicas de pesquisa: planejamento e execução de pesquisa; amostragens e técnicas de pesquisa; elaboração, análise e interpretação de dados*. [s.l.: s.n.]. p. 277–277.

MESHARAM, S. et al. *Conversational AI: chatbots*. 2021 International Conference on Intelligent Technologies (CONIT). Anais...IEEE, 2021.

SANTOS, M. H. DOS. *Inteligência Artificial*. São Paulo: Saraiva, 2021.

ZENDESK. *O que é atendimento ao cliente? Qual a importância desse serviço?* <https://www.zendesk.com.br/>, 18 dez. 2023. Disponível em:

<<https://www.zendesk.com.br/blog/o-que-e-atendimento-ao-cliente/>>. Acesso em: 25 ago. 2024

MOURA, Bernardo. Nova pesquisa global revela que 56% dos consumidores afirmam que o atendimento ao cliente das marcas não corresponde à imagem que retratam. JORNAL JURID, 03/09/2021. Disponível em: <https://www.jornaljurid.com.br/noticias/nova-pesquisa-global-revela-que-56-dos-consumidores-afirmam-que-o-atendimento-ao-cliente-das-marcas-nao-corresponde-a-imagem-que-retratam>. Acesso em: 25 ago. 2024.

UNIMED COSTA OESTE. Unimed Costa Oeste. Disponível em: <<https://www.unimed.coop.br/site/web/costaoeste/unimed-costa-oeste>\>. Acesso em: 25 ago. 2024.

UNIMED COSTA OESTE. Unimed Costa Oeste. Disponível em: <<https://www.unimed.coop.br/site/web/costaoeste/unimed-costa-oeste>>. Acesso em: 25 ago. 2024.

HUGGY. Disponível em: <<https://www.huggy.io/pt-br>>. Acesso em: 25 ago. 2024.

SYNGOO TALK. Disponível em: <<https://www.syngootalk.com/>>. Acesso em: 25 ago. 2024.

BLIP. Disponível em: <<https://digital.blip.ai/growth-blip/>>. Acesso em: 25 ago. 2024.

BLIP. Como construir bots através de SDKs ou API HTTP. Disponível em: <<https://help.blip.ai/hc/pt-br/articles/4474413783959-Como-construir-bots-atrav%C3%A9s-de-SDKs-ou-API-HTTP>>. Acesso em: 25 ago. 2024.

BLIP. Blip Community. Disponível em: <https://community.blip.ai/>.

BLIP. Blip Docs: Introduction. Disponível em: <https://docs.blip.ai/#introduction>.

APÊNDICE A – Script para definir os dias e horário de trabalho

```
function run() {
  let workSchedule = [
    { "num": 1,
      "name": "Monday",
      "portugueseName": "Segunda-feira",
      "workTime": [
        { "start": "07:30",
          "end": "18:00"
        } ]
    },
    { "num": 2,
      "name": "Tuesday",
      "portugueseName": "Terça-feira",
      "workTime": [
        { "start": "07:30",
          "end": "18:00"
        } ]
    },
    { "num": 3,
      "name": "Wednesday",
      "portugueseName": "Quarta-feira",
      "workTime": [
        { "start": "07:30",
          "end": "18:00"
        } ]
    },
    { "num": 4,
      "name": "Thursday",
      "portugueseName": "Quinta-feira",
      "workTime": [
        { "start": "07:30",
          "end": "18:00"
        } ]
    },
    { "num": 5,
      "name": "Friday",
      "portugueseName": "Sexta-feira",
      "workTime": [
        { "start": "07:30",
          "end": "18:00"
        } ]
    },
  ];

  return JSON.stringify(workSchedule);
}
```

Fonte: Autoria própria

APÊNDICE B – Script para verificar os dias e horário de trabalho

```

// Recebe as variáveis como parâmetros
function run(offset, weekSchedule) {
  offset = parseInt(offset);
  weekSchedule = JSON.parse(weekSchedule);
  let today = nowUTC(offset);

  if (isWorkDay(today, weekSchedule)) {
    let todaySchedule = getTodaySchedule(weekSchedule, today);
    let intervalTime = getIntervalTime(todaySchedule, today);
    return checkTime(intervalTime, today);
  }
  return false;
}

function getIntervalTime(todaySchedule, today) {
  let intervalTime = [];
  for (let i = 0; i < todaySchedule.workTime.length; i++) {
    intervalTime.push({
      start: utcDate(todaySchedule.workTime[i].start, today),
      end: utcDate(todaySchedule.workTime[i].end, today)
    });
  }
  return intervalTime;
}

function checkTime(intervalTime, today) {
  for (let i = 0; i < intervalTime.length; i++) {
    if (today - intervalTime[i].start > 0 && intervalTime[i].end
- today > 0)
      return true;
    } return false;
  }

function getTodaySchedule(weekSchedule, today) {
  for (let i = 0; i < weekSchedule.length; i++) {
    if (weekSchedule[i].num == today.getDay())
      return weekSchedule[i]; }
}

//Pega a data UTC atual
function nowUTC(offset) {
  let now = new Date();
  let utc_timestamp = Date.UTC(
    now.getUTCFullYear(),
    now.getUTCMonth(),
    now.getUTCDate(),
    now.getUTCHours(),
    now.getUTCMinutes(),
    now.getUTCSeconds(),
    now.getUTCMilliseconds()
  );
  return new Date(utc_timestamp + offset * 3600 * 1000);
}

```

```
//Pega a data UTC
function utcDate(timeString, today) {
  let hour = getValueByString("hour", timeString);
  let minutes = getValueByString("minutes", timeString);
  let utc_timestamp = Date.UTC(
    today.getUTCFullYear(),
    today.getUTCMonth(),
    today.getUTCDate(),
    hour,
    minutes,
    0,
    0
  );
  return new Date(utc_timestamp);
}

//Pega a hora/minuto por string com padrão HH:mm
function getValueByString(type, timeString) {
  if (type === "hour") {
    return parseInt(timeString.substring(0,
timeString.indexOf(":")));
  } else if (type === "minutes") {
    return parseInt(
timeString.substring(timeString.indexOf(":") + 1,
timeString.length)
    );
  }
  return 0;
}

//Valida se o dia atual é um dia de trabalho
function isWorkDay(today, workDays) {
  for (let i = 0; i < workDays.length; i++) {
    if (workDays[i].num === today.getDay().toString()) return
true;
  }
  return false;
}
```

Fonte: Autoria própria