

Projeto de Aplicação Chatbot Web para o gerenciamento de chamados de atendimento: um estudo de caso na UFPB

Diego Felipe Gonçalves do Nascimento



CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, 2024

Diego Felipe Gonçalves do Nascimento

Projeto de Aplicação Chatbot Web para o
gerenciamento de chamados de atendimento: um
estudo de caso na UFPB

Relatório técnico apresentado ao curso Ciência da Computação do Centro de Informática, da Universidade Federal da Paraíba, como requisito para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Raoni Kulesza

Maio de 2024

Catálogo na publicação
Seção de Catalogação e Classificação

N244p Nascimento, Diego Felipe Goncalves do.
Projeto de aplicação chatbot web para o
gerenciamento de chamados de atendimento: um estudo de
caso na UFPB / Diego Felipe Goncalves do Nascimento. -
João Pessoa, 2024.
49 f. : il.

Orientação: Raoni Kulesza.
TCC (Graduação) - UFPB/CI.

1. Aplicação Web. 2. Sistemas web. 3. Chatbot. I.
Kulesza, Raoni. II. Título.

UFPB/CI

CDU 004.8



CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

Trabalho de Conclusão de Curso de Ciência da Computação intitulado ***Projeto de Aplicação Chatbot Web para o gerenciamento de chamados de atendimento: um estudo de caso na UFPB***, de autoria de Diego Felipe Gonçalves do Nascimento, aprovada pela banca examinadora constituída pelos seguintes professores:

Prof. Dr. Raoni Kulesza
Universidade Federal da Paraíba - UFPB

Prof. Dr. Carlos Eduardo Coelho Freire Batista
Universidade Federal da Paraíba - UFPB

Prof. Dr. Marcelo Iury de Sousa Oliveira
Universidade Federal da Paraíba - UFPB

Coordenador(a) do Departamento Departamento de informática
Leandro Carlos De Souza
CI/UFPB

João Pessoa, 22 de maio de 2024

Centro de Informática, Universidade Federal da Paraíba
Rua dos Escoteiros, Mangabeira VII, João Pessoa, Paraíba, Brasil CEP: 58058-600
Fone: +55 (83) 3216 7093 / Fax: +55 (83) 3216 7117

DEDICATÓRIA

Este trabalho é dedicado a todas as pessoas que sempre me perguntam: "Já concluiu o curso?".

Adicionalmente, eu gostaria de dedicar a Jussara Santos e Júlia Sofia, pessoas que me deram ânimo e força de vontade para finalizar mais essa etapa.

AGRADECIMENTOS

Gostaria de expressar minha gratidão a todas as pessoas que fizeram parte dessa jornada!

RESUMO

A Universidade Federal da Paraíba (UFPB) é uma instituição autárquica de regime especial, dedicada ao ensino, pesquisa e extensão. Dada a diversidade de setores que prestam serviços à comunidade acadêmica, cada um com seu próprio sistema de solicitação de atendimento, observou-se a necessidade de elaborar uma ferramenta para centralizar e simplificar a abertura de chamados para esses setores. Com base nessa necessidade, foi desenvolvido um sistema de chatbot destinado à abertura de chamados via WhatsApp, com o objetivo de simplificar e unificar os sistemas utilizados por alguns setores. Devido à familiaridade dos usuários com essa ferramenta, conseguimos simplificar e centralizar o processo de abertura de chamados, expandindo a gama de dispositivos que podem realizar solicitações de serviços, já que o WhatsApp está disponível em plataformas web e mobile. Todo esse processo é transparente para o servidor responsável pelo atendimento no setor, eliminando a necessidade de treinamento para o uso de uma nova plataforma, o que representa um ganho significativo em eficiência. Este trabalho realiza um estudo de caso dessa aplicação, partindo da engenharia de requisitos, passando pelo seu projeto e desenvolvimento e finalizando com os resultados obtidos.

Palavras-chave: Aplicação Web, Sistema Web, Chatbot.

ABSTRACT

The Federal University of Paraíba (UFPB) is an autonomous institution with a special status, dedicated to teaching, research, and outreach. Given the diversity of departments that provide services to the academic community, each with its own system for requesting service, the need arose to create a tool to centralize and simplify the process of opening service tickets for these departments. Based on this need, a chatbot system was developed to open service tickets via WhatsApp, aiming to simplify and unify the systems used by some departments. Due to the users' familiarity with this tool, we were able to streamline and centralize the process of opening service tickets, expanding the range of devices that can request services, since WhatsApp is available on web and mobile platforms. This entire process is transparent to the staff responsible for service in the department, eliminating the need for training to use a new platform, which represents a significant gain in efficiency. This work presents a case study of this application, starting from requirements engineering, through its design and development, and concluding with the results obtained.

Keywords: Web Application, Web System, Chatbot.

LISTA DE FIGURAS

1	Visão do sistema de chamados da gtic	22
2	Visão do sistema de chamados da subprefeitura	23
3	Visão do sistema de chamados da ASCOM	24
4	Diagrama de casos de uso	28
5	Fluxo de coleta de informações do usuário	29
6	Fluxo com informações necessárias para conectar-se a internet sem fio	29
7	Visão Contextual do CCAE Bot	30
8	Visão Contêiners do CCAE Bot	31
9	Visão de Componentes do CCAE Bot	32
10	Tela inicial	40
11	Tela de coleta de dados do usuário	41
12	Tela dos setores disponíveis para abertura de chamado	41
13	Tela de filtro de opções para usuários do tipo professor	42
14	Tela de filtro de opções para usuários do tipo aluno	43
15	Tela ação de voltar	44
16	Tela de ação de "sair"	44
17	Tela de reenvio de mensagem ao receber resposta com opção inexistente	45

LISTA DE TABELAS

1	Persona A	36
2	Persona B	37
3	Persona C	37
4	Persona D	37

LISTA DE ABREVIATURAS

JS – *JavaScript*

API - *Application Programming Interface*

App - Aplicativo

UFPB - Universidade Federal da Paraíba

LLM - *Large language model*

Sumário

1 INTRODUÇÃO	17
1.1 Tema	17
1.2 Problema	17
1.3 Objetivo geral	18
1.4 Objetivos específicos	18
1.5 Estrutura do relatório técnico	19
2 ELICITAÇÃO E ANÁLISE DE REQUISITOS	20
2.1 Reuniões com stakeholders	20
2.1.1 Primeiro encontro	20
2.1.2 Segundo encontro	20
2.2 Análise dos sistemas atuais para solicitação de chamados	21
2.2.1 Gerência de Tecnologia da Informação e Comunicação (GTIC)	21
2.2.2 Subprefeitura Universitária	22
2.2.3 Assessoria Administrativa	23
2.2.4 Assessoria de Comunicação	23
2.3 Análise de sistemas concorrentes	24
2.3.1 Funcionalidades compartilhadas entre sistemas de <i>chatbot</i>	25
2.4 Análise dos requisitos	25
2.4.1 Requisitos Funcionais	25
2.4.2 Requisitos não funcionais	26
2.5 Diagrama de casos de uso	27
2.6 Fluxos de usuário	28
2.6.1 Coleta de informações do usuário	28
2.6.2 Conectar-se a internet sem fio	29
2.7 Conclusão do capítulo	29
3 MODELAGEM DO SISTEMA	30
3.1 Visão Arquitetural	30

3.1.1	C1 - Visão Contextual	30
3.1.2	C2 - Visão de Containers	31
3.1.3	C3 - Visão de Componentes	32
3.2	Tecnologias envolvidas	33
3.2.1	NodeJS	33
3.2.2	Typescript	34
3.2.3	WppConnect	34
3.2.4	Whatsapp e Whatsapp Web	34
3.2.5	Estrutura Map e Expiry Map	35
3.2.6	Jest	35
3.2.7	Nodemailer	35
4	PLANO DE TESTES	36
4.1	Personas	36
4.1.1	Persona A	36
4.1.2	Persona B	36
4.1.3	Persona C	37
4.1.4	Persona D	37
4.2	Cenários de Testes	38
4.2.1	Caso de teste 1 - Solicitação de chamado para a GTIC	38
4.2.2	Caso de teste 2 - Solicitação de chamado para a ASCON	38
4.2.3	Caso de teste 3 - Solicitação de chamado para a Assessoria Admi- nisitrativa	39
4.2.4	Caso de teste 4 - Solicitação de chamado para a Prefeitura Univer- sitária	39
5	APRESENTAÇÃO E ANÁLISE DOS RESULTADOS	40
5.1	Interface da aplicação	40
5.1.1	Tela Inicial	40
5.1.2	Coleta de dados do usuário	41
5.1.3	Setores disponíveis para abertura de chamado	41

5.1.4	Filtro de opções personalizado por usuário	42
5.1.5	Ação de voltar	43
5.1.6	Ação de sair	44
5.1.7	Reenvio de mensagem ao receber resposta com opção inexistente	45
5.2	Análise dos resultados	45
6	CONCLUSÕES E TRABALHOS FUTUROS	46
6.1	Conclusões	46
6.1.1	Limitações e Dificuldades enfrentadas	46
6.2	Trabalhos Futuros	47
	ANEXO A – ANEXOS E APÊNDICES 1	49

1 INTRODUÇÃO

1.1 Tema

Diante da crescente popularização de ferramentas de comunicação como o WhatsApp, juntamente com sua facilidade de uso e acessibilidade em dispositivos móveis, surgiu o desejo de diminuir a carga cognitiva relacionada ao manuseio de alguns sistemas de chamados utilizados por setores da Universidade Federal da Paraíba (UFPB). Essa demanda motivou a implementação de um *chatbot*, uma vez que os sistemas de chamados estão dispersos em diversas plataformas, como sites, formulários e e-mails.

Segundo a definição de (LOKMAN; AMEEDEN, 2019), o *chatbot* é um tipo de aplicação que possibilita com que humanos interajam com computadores utilizando linguagem natural.

Dessa forma a ferramenta simplifica o processo de abertura de chamados, tornando-o mais acessível e eficaz. Com a capacidade de processar mensagens escritas por humanos através do WhatsApp, proporcionando uma forma direta e ágil para os usuários se comunicarem com os departamentos.

1.2 Problema

A Universidade Federal da Paraíba (UFPB), inicialmente conhecida como Universidade da Paraíba, é uma instituição autônoma de ensino superior fundada em 1955 por meio da Lei Nº 1.366 sancionada pelo Governo do Estado da Paraíba. A UFPB é sustentada pelos pilares do ensino, pesquisa e extensão. Conforme estipulado na referida legislação (Palácio do Governo do Estado da Paraíba, 1955), a universidade opera de acordo com a legislação federal, as disposições da lei em questão, além de seu Estatuto, regulamentos e regimentos aprovados.

A UFPB é vinculada ao Ministério da Educação (MEC) e possui uma estrutura de múltiplos campi, com presença nas cidades de João Pessoa e Santa Rita, Areia, Bananeiras e Rio Tinto e Mamanguape.

Com o objetivo de desempenhar suas funções com excelência, a universidade está organizada em setores estratégicos e claramente definidos, tais como, reitoria, prefeitura, procuradoria jurídica e centros acadêmicos.

O Campus IV é o 4º Campus da UFPB. Está dividido nas cidades de Mamanguape e Rio tinto. A sua atuação é voltada às necessidades da vida educacional, cultural, social e econômica por meio de seu único centro acadêmico, o Centro de Ciências Aplicadas e Educação (CCAIE).

De forma semelhante ao Campus I, o Campus IV também possui subdivisões em seus setores administrativos, tais como, coordenações, Gerência de Tecnologia da Informação e Comunicação (GTIC), Subprefeitura universitária e Assessoria Administrativa.

A GTIC é o setor do CCAE encarregado de fornecer serviços de Tecnologia da Informação e Comunicação (TIC) que apoiam as atividades de ensino, pesquisa e extensão realizadas neste centro. Além disso, a GTIC é responsável pela gestão e melhoria contínua dos serviços de TI e de toda a infraestrutura tecnológica.

A Subprefeitura tem como responsabilidade a manutenção dos edifícios e das demais estruturas físicas do Campus IV. Suas atribuições incluem a execução de diversos serviços, como manutenção hidrossanitária e elétrica, pintura e limpeza das áreas externas.

A Assessoria Administrativa desempenha um papel auxiliar crucial para a Direção do Centro, sendo encarregada de funções específicas delegadas pela Direção nas áreas de administração, contabilidade, finanças, gestão de materiais, gestão de patrimônio, gestão de recursos humanos e outras atividades relacionadas à administração do centro.

Para acionar os setores que fornecem serviço faz-se necessário a abertura de um chamado, uma solicitação realizada ao setor competente, onde o usuário passa por uma etapa de identificação e fornece informações relacionadas ao problema em questão.

Cada setor possui o seu próprio sistema de chamados, alocados em diferentes plataformas e manuseados de diferentes formas. Podemos encontrar sistemas de abertura de chamados que funcionam por meio de formulários ou páginas web. Também há dificuldade quanto a acessibilidade e facilidade com relação ao acesso a esses sistemas, alguns estão disponíveis por meio do site institucional, outros apenas por link fornecidos por integrantes do setor.

1.3 Objetivo geral

O objetivo deste trabalho é analisar o desenvolvimento de uma aplicação de chatbot, que tem o propósito de facilitar a realização de chamados, acessível tanto por dispositivos móveis quanto por navegadores web, realizando integração e comunicação com sistemas de terceiros para abertura de chamados.

1.4 Objetivos específicos

- Identificar e especificar requisitos funcionais e não funcionais do sistema, bem como a arquitetura geral.
- Explicar o processo de desenvolvimento de um chatbot

- Realizar a análise dos resultados coletados

1.5 Estrutura do relatório técnico

Este trabalho está estruturado em seis capítulos. No primeiro é feito a introdução do tema, bem como apresentação da relevância e objetivos do trabalho. O segundo capítulo fazemos a elicitação dos requisitos, onde delineamos o escopo do problema e obtemos insumos para resolvê-lo, bem como identificamos as funcionalidades das quais o sistema deve contemplar. Na sequência, temos a modelagem do sistema, onde é possível ter a visão arquitetural da aplicação além da diagramação dos componentes e tecnologia envolvidas. O quarto capítulo trata do plano de testes, identificação das personas e execução dos possíveis cenários que podem ser exibidos para os usuários. O quinto capítulo trata da apresentação da ferramenta, com imagens e análise dos resultados obtidos. Por fim, o capítulo 6 trata das conclusões deste trabalho e avalia possíveis temas para trabalhos futuros.

2 ELICITAÇÃO E ANÁLISE DE REQUISITOS

Segundo (SOMMERVILLE, 2011), por meio da elicitação de requisitos podemos derivar os requerimentos do sistema. Essa análise pode ocorrer por meio de discussões com potenciais usuários, avaliação de sistemas existentes além de modelos de protótipos de sistemas.

O processo de elicitação de requisitos começou com reuniões com os *stakeholders*, nas quais foram discutidas as necessidades e os problemas que a aplicação deve resolver. Além disso, foram realizadas análises dos sistemas de chamados atualmente utilizados por cada setor e dos sistemas de chatbot disponíveis no mercado que se comunicam via WhatsApp. Essas atividades ajudaram a definir o escopo da aplicação.

2.1 Reuniões com stakeholders

2.1.1 Primeiro encontro

Na primeira reunião estavam envolvidos, a diretoria de centro e a equipe da GTIC. Neste primeiro momento houve a explicação acerca da complexidade de acesso aos sistemas de chamados do centro, e que um possível caminho para simplificar o processo de abertura de chamados poderia ser a utilização de um chatbot, uma vez que os sistemas de tickets ficariam centralizados. Adicionalmente, a interação entre o sistema e o usuário deveria ser realizada via whatsapp, uma vez que esta é uma ferramenta tem vasta utilização parte da comunidade acadêmica do campus além de fácil integração com serviços de QRCode, links e etc.

2.1.2 Segundo encontro

Alguns meses depois, após a realização do estudo de viabilidade, ocorreu a segunda reunião. Esta aconteceu internamente entre os membros da GTIC (chefia de departamento e responsáveis pelo desenvolvimento). Neste momento, algumas *features* foram elencadas. Estas funcionalidades visam adequar o sistemas às diretrizes de alguns setores, e promover uma melhor usabilidade para o usuário. Entre as funções discutidas nesse momento estão:

- Integração com os sistemas de chamados existentes
- Adição de novas opções no menu de chamados da GTIC
- Permissão de acesso a menus de acordo com o tipo de usuário (Técnico administrativos, professores, alunos e servidores terceirizados)

- Dar ciência para o usuário que toda a comunicação será realizada através do email informado ao *chatbot*
- Armazenar as informações pessoais do usuário para evitar o reenvio destas em um segundo contato com o sistema.
- Exibir resumo da solicitação ao final do atendimento.

A comunicação entre a GTIC e os demais setores avançaram de forma casual e informal.

2.2 Análise dos sistemas atuais para solicitação de chamados

Dentre os setores administrativos do CCAE estão, a Assessoria de Comunicação (ASCOM), Gerência de Tecnologia da Informação e Comunicação (GTIC), Subprefeitura Universitária e a Assessoria Administrativa. A seguir, iremos analisar como os usuários interagem com os atuais sistemas de chamados.

O ponto de entrada para a comunicação com os três últimos setores pode ser localizado diretamente na página web do CCAE, por meio da opção "Setores Administrativos" no menu lateral do *site*.

2.2.1 Gerência de Tecnologia da Informação e Comunicação (GTIC)

A GTIC adota o *OTRS* como sua principal ferramenta para o recebimento de chamados. No site oficial, o *OTRS* é descrito como uma solução versátil, oferecendo tanto soluções prontas para uso quanto a flexibilidade de ser um software customizável, adaptável a todas as necessidades de gerenciamento de serviços.

Para abrir chamados com a GTIC, os usuários podem solicitar:

- Resolução de problemas de internet cabeada ou *wireless*
- Gerenciamento ou criação de email institucional
- Manutenção, substituição e solicitação de equipamentos de informática
- Instalação e configuração de *hardware* e *software*

A abertura de chamados com a GTIC é facilitada por meio de uma página web que se adapta às escolhas do usuário, dependendo do tipo de chamado desejado. No entanto, alguns campos são fixos e obrigatórios, como nome, e-mail e unidade de localização.

Para questões que não são abrangidas pelo sistema de chamados, é possível entrar em contato com o setor enviando um e-mail diretamente



O formulário da GTIC para solicitação de suporte contém os seguintes campos:

- Nome completo (campo de texto)
- E-mail (campo de texto)
- Telefone (campo de texto)
- Seleção a unidade (menu suspenso)
- Informe o tipo de solicitante (menu suspenso)
- Seleção o tipo de serviço (menu suspenso)
- Anexos (*.jpg, *.png, *.doc, *.docx, *.pdf):
 - Botão "Escolher arquivos"
 - Texto: "Nenhum arquivo escolhido"
 - Nota: "Tamanho máximo de 3MB por arquivo."
- Botão "Solicitar Suporte"

Figura 1: Visão do sistema de chamados da gtic

2.2.2 Subprefeitura Universitária

A subprefeitura utiliza o *HESK*, um sistema caracterizado como um software de Help Desk extremamente simples, amigável e gratuito, com a conveniência de contar com uma base de conhecimento integrada.

Ambas as ferramentas possibilitam a utilização de interfaces web próprias e customizáveis, além de serem desacopladas, o que permite a implementação de soluções de frontend personalizadas de acordo com as necessidades específicas de cada contexto. Essa flexibilidade oferecida por ambas as plataformas contribui significativamente para a adaptação e otimização dos processos de suporte e gerenciamento de serviços.

O link para abertura de chamados da subprefeitura pode ser encontrado em um documento no formato PDF localizado no menu lateral "Manutenção", em sua página web

O sistema utilizado pela subprefeitura realiza uma filtragem inicial, identificando a unidade de localização da qual necessita do serviço, após essa etapa são requeridos alguns campos obrigatórios como nome, email e matrícula, além das informações necessária para solucionar o problema corrente.

A subprefeitura atende chamados relacionados a infraestrutura do campus e manutenção em equipamentos, tais como:

- Construção Civil
- Elétrica
- Hidráulica
- Jardinagem

- Ar-condicionado
- Marcenaria



Figura 2: Visão do sistema de chamados da subprefeitura

2.2.3 Assessoria Administrativa

Diferenciando-se dos demais setores, a Assessoria Administrativa opta por receber exclusivamente as requisições por meio de seus emails institucionais, dispensando qualquer sistema de help desk disponível no mercado. Para cada tipo de solicitação há um endereço de email específico, abrangendo pedidos relacionados a materiais, compras e outras demandas correlatas.

Os emails para os quais devemos encaminhar solicitações estão descritos na página dedicada a Assessoria Administrativa, no site do CCAE.

2.2.4 Assessoria de Comunicação

A ASCOM é responsável pela divulgação de ações acadêmicas diversas, como eventos, cursos e outras. O setor recebe suas demandas através de um formulário desenvolvido através do *Microsoft Forms* e anexado à sua página dedicada dentro do *site* do CCAE, adicionalmente, as solicitações podem ser realizadas via email institucional.



Figura 3: Visão do sistema de chamados da ASCOM

2.3 Análise de sistemas concorrentes

A seguir, iremos prosseguir com a análise e validação de sistemas que compartilham objetivos semelhantes aos do nosso. Para tal, conduziremos uma pesquisa em busca de serviços que possibilitam a comunicação com o usuário via WhatsApp. O propósito é identificar funcionalidades similares entre os sistemas e encontrar oportunidades de aprimoramento.

Na sequência, conheceremos algumas empresas que fazem uso de sistemas de *chatbot* para fornecer suporte a seus usuários. Este tipo de aplicação é utilizada em diversos segmentos, como, vendas e *helpdesk*.

- Flash Card - Empresa especializada em gestão de benefícios e despesas corporativas, centralizando-as em um único cartão multibenefícios que também é cartão corporativo pré-pagor.
- RNP - A Rede Nacional de Pesquisa é uma plataforma de comunicação e colaboração digital que trabalha para promover e implementar a inovação em aplicações de tecnologia da informação. Através de educação e pesquisa conectam pessoas, instituições, ideias e soluções para democratizar o acesso ao conhecimento.
- Dell - A Dell é uma fabricante de hardware de computadores, bem como montadora de desktops, notebooks e servidores.

- Boticário - O Grupo Boticário é um ecossistema de beleza. São 10 marcas que atendem diferentes perfis brasileiros em uma cadeia completa da beleza: dos ingredientes, passando pela indústria, laboratórios, tecnologia, lojas, revenda e e-commerce.

2.3.1 Funcionalidades compartilhadas entre sistemas de *chatbot*

Em razão das singularidades de segmento comercial das empresas avaliadas, os sistemas de *chatbot* interagem de forma particular, contudo, podemos destacar algumas funcionalidades compartilhadas entre eles:

1. Ativação por mensagem de texto - A interação entre a pessoa usuária e a aplicação se dá por meio de uma mensagem de texto enviada pelo usuário.
2. Solicitação de informações pessoais inerentes ao serviço prestado - O sistema solicita ao usuário algumas informações referentes ao serviço demandado.
3. Seleção de opção através de números indo-arábicos
4. Autonomia para o usuário encerrar o atendimento
5. Encerrar atendimento após inatividade de comunicação
6. Reenvio de pergunta para os casos em que a resposta do usuário não é satisfatória com o contexto

2.4 Análise dos requisitos

A análise anterior possibilita a identificação dos requisitos funcionais (RF) e dos requisitos não funcionais (RNF) do sistema, permitindo delinear as funcionalidades e comportamentos das quais a aplicação deve contemplar.

2.4.1 Requisitos Funcionais

Segundo a afirmação de (FIGUEIREDO, 2011), os requisitos funcionais descrevem explicitamente o que a aplicação deve fazer. Dessa forma, é possível documentar como o sistema deve reagir a entradas específicas.

Os requisitos devem ser completos, garantindo que todos os serviços estejam definidos, e consistentes, assegurando que não haja definições contraditórias.

- **[RF01] Ativação por mensagem de texto:** O sistema deve ser ativado ao receber qualquer mensagem de texto do usuário.

- **[RF02] Coleta de Informações do Usuário:** A aplicação é responsável por requisitar informações pessoais do usuário, tais como nome, tipo de vínculo, localização da unidade de atendimento e endereço de e-mail.
- **[RF03] Visualização de Setores com Sistema de Chamados:** O sistema deverá apresentar ao usuário uma lista dos setores disponíveis para a abertura de chamados.
- **[RF04] Filtro de Opções Personalizado:** O usuário terá acesso somente às opções específicas para seu "tipo de usuário" (Professor, Técnico Administrativo, Aluno, Outros), proporcionando uma experiência personalizada.
- **[RF05] Ação de "Voltar":** O sistema será capaz de retornar ao menu anterior mediante a digitação da palavra "voltar".
- **[RF06] Ação de "Sair":** O sistema poderá cancelar o atendimento do usuário ao reconhecer a palavra "sair".
- **[RF07] Detecção de opções inexistentes:** O sistema será capaz de identificar respostas com opções não listadas. Ao identificar uma resposta não satisfatória.
- **[RF08] Reenvio de Mensagens ao identificar uma resposta que não reflete as opções enviadas:** O sistema detectará respostas com opções não listadas e reenviará a pergunta ao usuário.
- **[RF09] Abertura de Chamados:** O sistema deve se integrar aos sistemas de chamados externos, e.g, Hesk e OTRS.
- **[RF10] Integração com Whatsapp:** O sistema deve responder a mensagens enviadas por WhatsApp (mobile e web).

2.4.2 Requisitos não funcionais

Segundo Sommerville (2011), os requisitos não funcionais descrevem as restrições sobre os serviços ou funções oferecidos pelo sistema. Ao contrário dos requisitos funcionais, que se concentram nas funcionalidades que o sistema deve ter, os requisitos não funcionais tratam de aspectos como tempo de resposta, desempenho, usabilidade, entre outros.

2.4.2.1 Desempenho

- **[RNF01] Armazenamento de sessão do usuário:** O sistema deve armazenar a sessão do usuário em um banco de dados em memória.

2.4.2.2 Disponibilidade

- [RNF02] **Tempo de sessão do usuário:** O sistema deve manter a sessão do usuário que estão sem atividade por até 30 minutos.
- [RNF03] **Tempo de atividade:** O sistema deve estar disponível 24 horas por dia e 7 dias por semana.

2.4.2.3 Usabilidade

- [RNF04] **Facilidade de uso:** O sistema deve ser fácil de usar, independentemente do nível de experiência do usuário.
- [RNF05] **Experiência do usuário:** O sistema deve ser capaz de interagir de acordo com as escolhas do usuário.

2.4.2.4 Autenticação

- [RNF06] **Cadastro no WhatsApp:** O sistema deve utilizar um número de telefone válido para que seja possível enviar e receber mensagens na plataforma do WhatsApp.

2.5 Diagrama de casos de uso

Segundo (SOMMERVILLE, 2011), o modelo de casos de uso foi originalmente desenvolvido por Jacobson et al. (1993) em 1990 e posteriormente incorporado à UML. Os modelos de casos de uso apoiam a elicitação de requisitos ao explicitar cenários simples que descrevem o que um usuário espera do sistema. Por meio de uma representação gráfica, os casos de uso retratam tarefas que envolvem interações entre os atores da aplicação.

Baseado no diagrama de casos de uso proposto por (RODRÍGUEZ et al., 2018), podemos identificar os atores envolvidos, as ações e dependências da aplicação, proporcionando uma visão clara e concisa das interações dentro do nosso sistema.

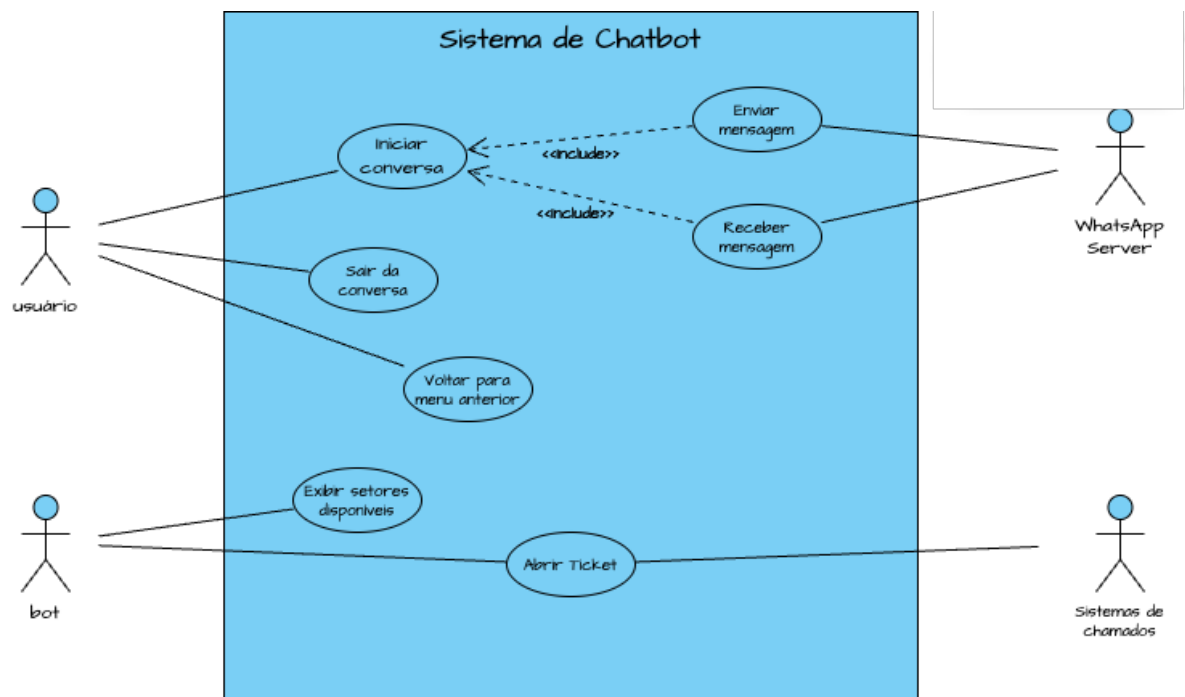


Figura 4: Diagrama de casos de uso

2.6 Fluxos de usuário

Com o objetivo de visualizar os estágios e menus dos formulários que os usuários devem seguir dentro do chatbot, foram plotados os fluxos percorridos em cada setor.

Os retângulos preenchidos representam opções selecionadas, enquanto os blocos com mais de um retângulo preenchido indicam opções que compartilham o próximo menu em comum.

Essa abordagem tem a finalidade de evidenciar uma hierarquia ou relação entre as opções de um menu, especialmente quando há seleções ou caminhos comuns para diferentes opções.

Para fins de exibição, vamos seguir dois fluxos: no primeiro, mostramos o fluxo de coleta de dados do usuário; no segundo, identificamos o fluxo para obter instruções de conexão à rede "UFPB sem fios".

2.6.1 Coleta de informações do usuário

Esta etapa é importante para identificar **quem** é o usuário que está abrindo o chamado, **qual** a sua localização e **como** essa pessoa pode ser contactada via email.

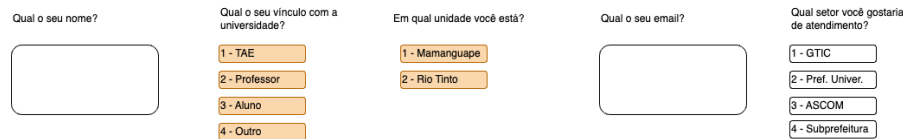


Figura 5: Fluxo de coleta de informações do usuário

2.6.2 Conectar-se a internet sem fio

Esta etapa exemplifica como foram elaborados os fluxos de serviço para usuários.

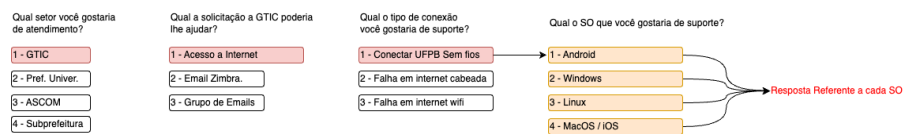


Figura 6: Fluxo com informações necessárias para conectar-se a internet sem fio

2.7 Conclusão do capítulo

A partir da identificação dos requisitos delineados pelos stakeholders e funcionalidades fundamentais habilitadas em *softwares* concorrentes, conseguimos definir as funções que a aplicação deverá incorporar, considerando tanto os requisitos funcionais quanto os não funcionais.

Adicionalmente, a interação com o sistema será realizada através do WhatsApp, proporcionando ao usuário a flexibilidade de utilizá-lo em plataformas web e mobile. Ao término de cada atendimento, o chatbot deverá integrar-se aos sistemas existentes e gerar os registros necessários.

3 MODELAGEM DO SISTEMA

Após delinear o escopo do sistema, identificar os requisitos funcionais e não funcionais, atores e aplicações das quais devem ser realizadas as integrações, neste capítulo vamos identificar o que é necessário para prosseguir no desenvolvimento do sistema.

3.1 Visão Arquitetural

3.1.1 C1 - Visão Contextual

O CCAE Bot é um *chatbot* desenvolvido com o objetivo de simplificar o processo de solicitação de chamados direcionados a diferentes setores de uma organização.

Chamados representam solicitações de materiais ou suporte técnico encaminhadas a áreas específicas.

Quando ativado por meio de uma mensagem via WhatsApp, o sistema inicia uma sequência de perguntas, aprimorando-as progressivamente até gerar o chamado para o setor desejado.

Cada setor opera seu próprio sistema de chamados. O CCAE Bot centraliza o processo de criação de tickets.

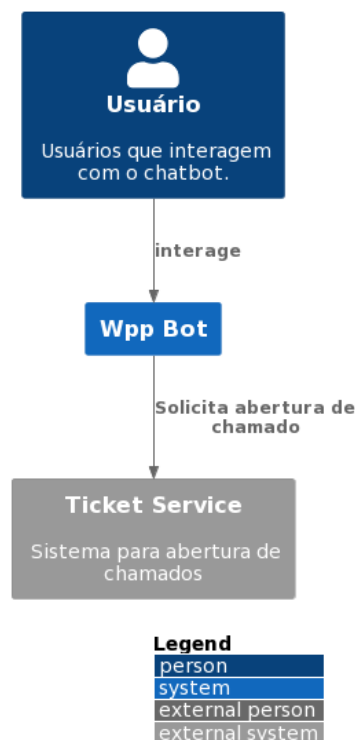


Figura 7: Visão Contextual do CCAE Bot

3.1.2 C2 - Visão de Contêiners

O CCAE Bot é acionado ao receber alguma mensagem do usuário através do Whatsapp. Estabelecida essa comunicação, acontece uma série de interações entre o chatbot e o usuário. Após a coleta de todas as informações necessárias, os dados são enviados para um serviço externo, para que a solicitação de chamado seja concluída.

Ao identificar a interação o CCAE Bot responde com uma série de questionários.

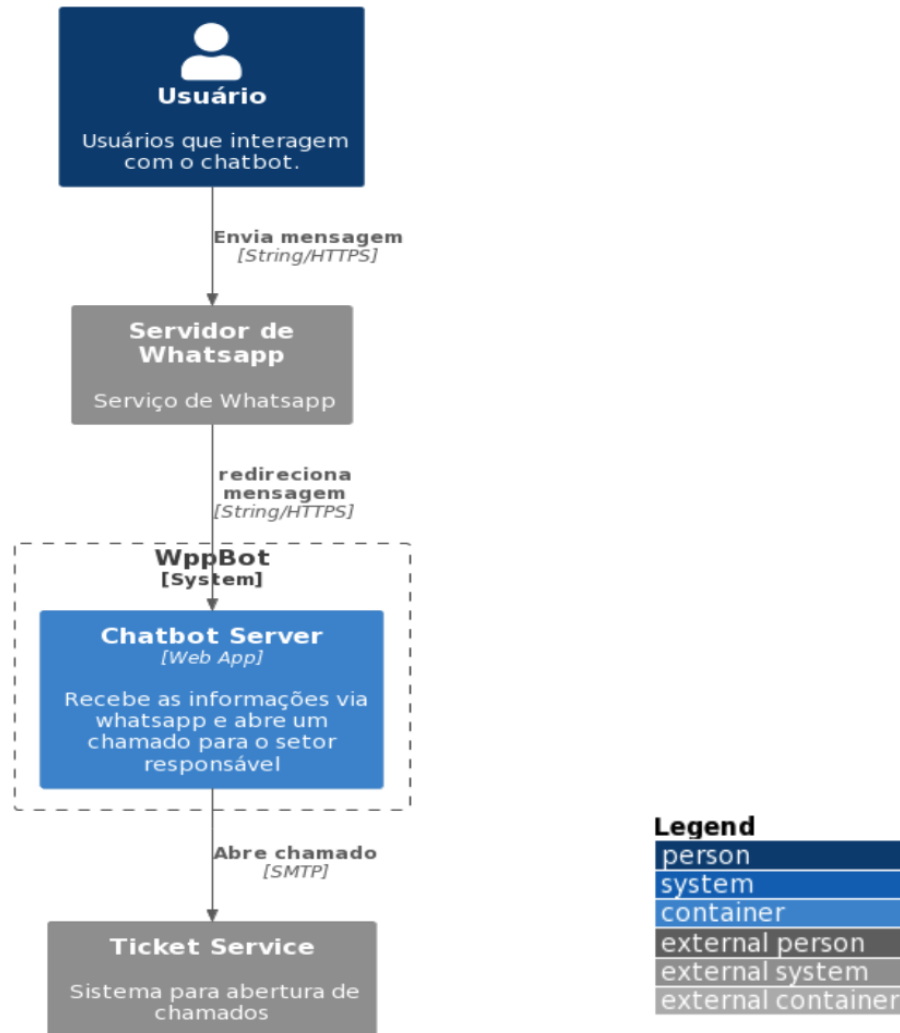


Figura 8: Visão Contêiners do CCAE Bot

3.1.3 C3 - Visão de Componentes

A visão de componentes nos mostra que a aplicação está dividida em módulos que abstraem funções e intenções entre as diversas partes do sistema.

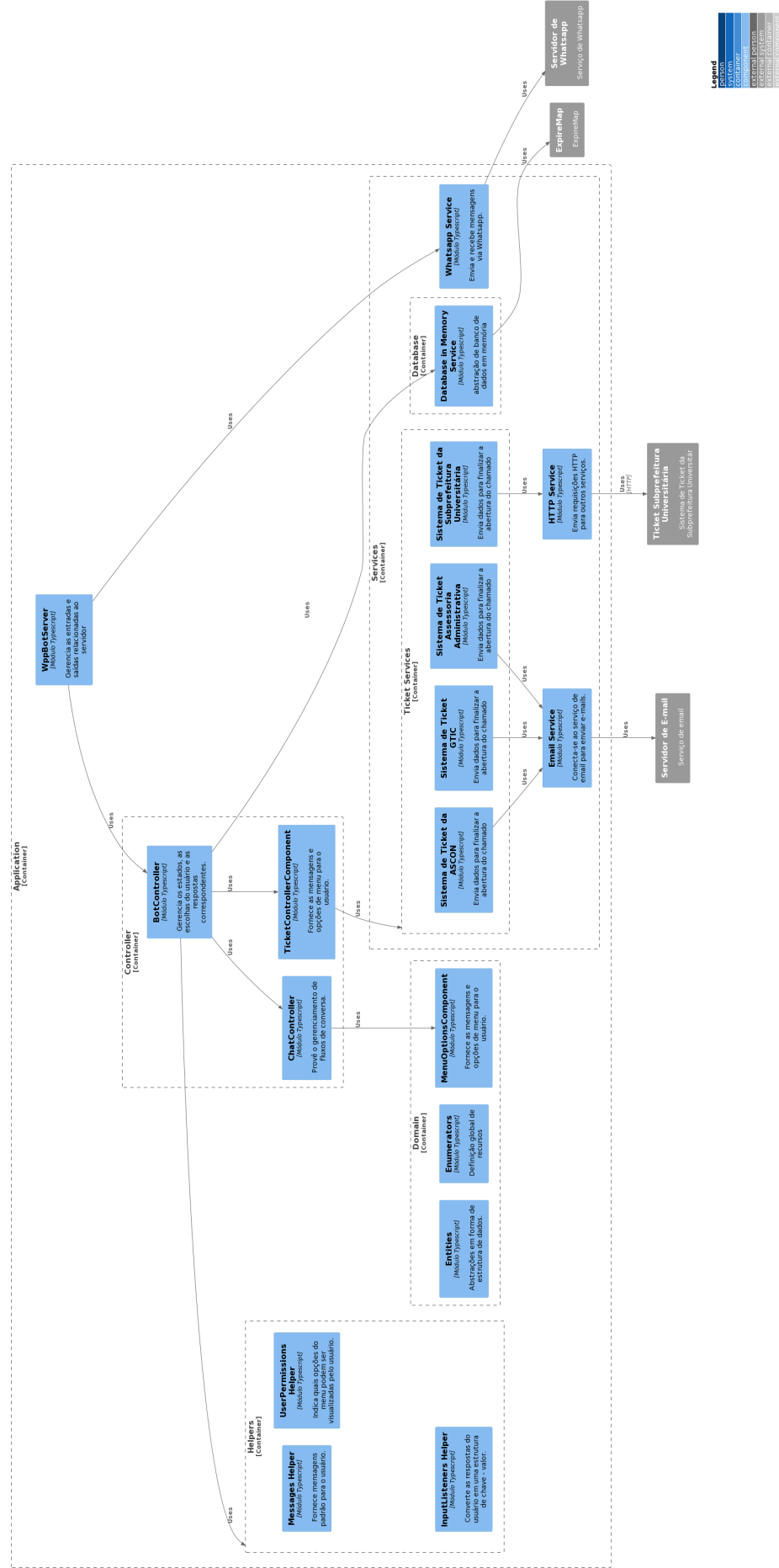


Figura 9: Visão de Componentes do CCAE Bot

Os principais módulos são:

- *Domain*
- *Services*
- *Helpers*
- *Controllers*
- *Configs*

3.1.3.1 *Domain*

A camada de domínio (*domain*) contém recursos referentes a o domínio de negócio do *app*, independente de qual tecnologia ou framework estejam sendo utilizados. Essa abordagem flexibiliza migrações para outras plataformas de comunicação. Nesta camada estão as entidades, *enumerators* e menus utilizados pelo *chatbot*.

3.1.3.2 *Services*

A *service* é responsável por abstrair comunicação com todas as bibliotecas de terceiros e sistemas externos à aplicação, tais como, sistemas de chamados, persistência e comunicação.

3.1.3.3 *Controllers*

A intenção desta camada é servir como ponte entre o domínio e a aplicação. Os *controllers* também podem realizar comunicação entre si.

3.2 Tecnologias envolvidas

3.2.1 NodeJS

Inspirado na definição de [Le \(2023\)](#), p. 22), podemos definir o *NodeJS* como um ambiente de execução JavaScript, que permite com que desenvolvedores possam construir aplicações escaláveis e de alta performance.

Na visão de [Le \(2023\)](#), p. 22) as principais características do NodeJS são: sua arquitetura orientada a eventos, o modelo não bloqueante de I/O e sua habilidade de manipular múltiplas requisições simultaneamente. Além dessas características, segundo [Stefanoski, Karadimche e Dimitrievski \(2019\)](#), p. 1), o *Nodejs* também permite a análise e execução

de código escrito em JavaScript, possibilitando sua execução em diversos ambientes ambientes e dispositivos e não apenas dentro dos *browsers*.

3.2.2 Typescript

Antes de explorarmos as definições do *TypeScript*, vamos iniciar com uma breve contextualização acerca do *JavaScript*.

De acordo com a documentação do [TypeScript \(2023\)](#), o *JavaScript* teve seu início como uma linguagem de script simples destinada a navegadores da web. Originalmente, foi concebido para ser usado em pequenos fragmentos de código incorporados em páginas da web. Hoje, no entanto, o *JavaScript* evoluiu consideravelmente e se tornou uma linguagem versátil e poderosa para o desenvolvimento de aplicativos web e muito mais.

Seguindo a documentação descrita no [Network \(2023\)](#), o *JavaScript* possui a característica de ser uma linguagem de programação interpretada com funções de primeira classe, além de ser *single threading* e multiparadígma, tais como, orientação a objetos, imperativa e declarativa.

Segundo [Bierman, Abadi e Torgersen \(2014\)](#), p. 1) o *TypeScript* é uma extensão da linguagem JavaScript. criada com finalidade de facilitar o desenvolvimento de aplicações JavaScript de larga escala. Como dito em sua documentação, "O TypeScript verifica se há erros em um programa antes da execução e o faz com base nos tipos de valores, tornando-o um verificador de tipo estático."

Além disso, [Bierman, Abadi e Torgersen \(2014\)](#), p. 1), explicam que o *TypeScript* fornece um sistema de módulos, tipagens, classes, interfaces, checagem estática de código e capacidade de identificar erros em tempo de desenvolvimento.

3.2.3 WppConnect

Conforme detalhado em sua documentação, referenciada em [Team \(2023\)](#), o *WPP-Connect* é um projeto de código aberto que surgiu da colaboração de membros da comunidade *JavaScript*. Essa biblioteca foi desenvolvida com a finalidade de oferecer as funcionalidades do *WhatsApp Web* para aplicações *Node.js*.

3.2.4 Whatsapp e Whatsapp Web

Com o WhatsApp, você pode trocar mensagens e fazer chamadas do seu celular de maneira rápida, simples e segura, em qualquer lugar do mundo.

"O WhatsApp Web e o WhatsApp para computador são extensões da conta usada

no seu celular. As mensagens enviadas e recebidas são sincronizadas entre seu celular e computador, assim, é possível acessá-las em ambos os aparelhos.” (META, 2023)

3.2.5 Estrutura Map e Expiry Map

O objeto Map contém pares de valores-chave e lembra a ordem de inserção original das chaves. Qualquer valor (objetos e valores primitivos) pode ser usado como chave ou valor. O Expiry Map é uma biblioteca *Javascript* que implementa a Estrutura Map de forma temporizada.

Pelo fato do Expiry Map permitir a persistência de dados temporizados, ou seja, após determinado período de tempo a informação é excluída da estrutura, optou-se por sua utilização para realizar a persistência das sessões de usuário, uma vez que a estrutura salva a referência de objetos em memória, essa abordagem facilita a implementação das features de ”voltar para o menu anterior” e ”sair do atendimento”.

3.2.6 Jest

Segundo (PAN, 1999) um teste de software é o procedimento de executar um programa ou sistema com o objetivo de avaliar seus atributos ou capacidades, determinando se ele atende aos requisitos estipulados ou identificando possíveis erros.

Em um ambiente de desenvolvimento JavaScript, existem vários frameworks disponíveis para a realização de testes. Um deles é o Jest, que em sua documentação se autodenomina como ”um poderoso Framework de Testes em JavaScript com um foco na simplicidade.” (Meta Platforms, Inc. and affiliates, 2023). O Jest pode ser facilmente integrado em projetos que fazem uso de uma ampla gama de frameworks e bibliotecas, incluindo Babel, TypeScript, Node, React, Angular, Vue, entre outros.

3.2.7 Nodemailer

Nodemailer é um módulo para aplicativos Node.js que permite o envio de e-mail com facilidade. O projeto começou em 2010, quando não havia uma opção sensata para enviar mensagens de e-mail. Hoje é a solução que a maioria dos usuários do Node.js recorre por padrão.

4 PLANO DE TESTES

A seguir, serão detalhados os casos de testes de pré lançamento do CCAE Bot . Por meio dessa validação, torna-se viável identificar os fluxos essenciais para a coleta de informações destinadas à abertura dos chamados.

Esse plano de testes tem por objetivo navegar por algumas funcionalidades, identificando as perguntas que o usuário precisa responder para formalizar a abertura de seu chamado. Para tal, seguimos com a criação de personas que refletem as pessoas usuárias da aplicação.

4.1 Personas

4.1.1 Persona A

Charles Xavier da Silva, tem 60 anos, caso, dois filhos e possui PhD em antropologia e mudança Social com sua formação na Universidade de *Oxford*. É professor, e já leccionou em diversas universidades como, *Oxford* e *Stanford*. Hoje prefere uma vida mais pacata, leccionando na Universidade Federal da Paraíba - UFPB, onde ministra algumas disciplinas do curso de antropologia para alunos da pós graduação (PPGA). A Tabela [1](#) apresenta as informações do professor Xavier.

Informações do usuário	
Nome	Charles Xavier da Silva
Tipo de vínculo	Professor
Localização	Rio Tinto
Email	xavier@ccae.ufpb.br
Matrícula	201501898
Telefone / Ramal	1963

Tabela 1: Persona A

4.1.2 Persona B

Roberto Martiniano, tem 28 anos, graduado em sistemas da informação pela UFPB e entrou recentemente nesta instituição como técnico de laboratório no Departamento de Ciências Exatas (DCX). Com frequência auxilia professores e alunos do curso, com deploys e manutenção de sistemas computacionais. A Tabela [2](#) apresenta as informações do Roberto.

Informações do usuário	
Nome	Roberto Martiniano
Tipo de vínculo	TAE
Localização	Rio Tinto
Email	robet.martin@ccae.ufpb.br
Matrícula	202301298
Telefone / Ramal	1981

Tabela 2: Persona B

4.1.3 Persona C

Paula Freire é estudante do curso de pedagogia, solteira, sem filhos. Cursa o quarto período do curso e faz estágio no Laboratório de Pedagogia (LABOPED). A Tabela 3 apresenta as informações da Paula.

Informações do usuário	
Nome	Paula Freire
Tipo de vínculo	Aluno
Localização	Mamanguape
Email	paula.freire@ccae.ufpb.br
Matrícula	20220145984
Telefone / Ramal	83 988440393

Tabela 3: Persona C

4.1.4 Persona D

Benjamin Franklin, 50 anos, solteiro, três filhos. Trabalha para uma empresa privada chamada "Mãos à obra", que fornece serviços de manutenção para a UFPB. A Tabela 4 apresenta as informações da Benjamin.

Informações do usuário	
Nome	Benjamin Franklin
Tipo de vínculo	Outro
Localização	Mamanguape
Email	frankin2019_2@gmail.com
Matrícula	
Telefone / Ramal	83 988440393

Tabela 4: Persona D

4.2 Cenários de Testes

A seguir, prosseguimos com a solicitação de chamados para cada setor administrativo da UFPB. Para realizar esse prosedimento fizemos uso das personas descritas na sessão anterior. Em cada teste há a descrição dos objetivos e resultados esperados.

4.2.1 Caso de teste 1 - Solicitação de chamado para a GTIC

Objetivo: Testar a funcionalidade de solicitação de chamados para a GTIC.

Passos:

1. Enviar uma mensagem para o CCAE Bot .
2. Responder as perguntas solicitadas.
3. Ao se deparar com a mensagem "Com qual setor você gostaria de falar?"selecionar a opção **GTIC**.
4. Responder as perguntas solicitadas.

Resultado esperado: O usuário solicitará o chamado para a GTIC com sucesso, e recebe a mensagem "Tudo certo! Nós registramos o seu chamado. Obrigado! Fique de olho em seu e-mail."

4.2.2 Caso de teste 2 - Solicitação de chamado para a ASCON

Objetivo: Testar a funcionalidade de solicitação de chamados para a ASCON.

Passos:

1. Enviar uma mensagem para o CCAE Bot .
2. Responder as perguntas solicitadas.
3. Ao se deparar com a mensagem "Com qual setor você gostaria de falar?"selecionar a opção **ASCON**.
4. Enviar sua dúvida ou solicitação.

Resultado esperado: O usuário solicitará o chamado para a ASCON com sucesso, e recebe a mensagem "Tudo certo! Nós registramos o seu chamado. Obrigado! Fique de olho em seu e-mail."

4.2.3 Caso de teste 3 - Solicitação de chamado para a Assessoria Administrativa

Objetivo: Testar a funcionalidade de solicitação de chamados para a Assessoria Administrativa.

Passos:

1. Enviar uma mensagem para o CCAE Bot .
2. Responder as perguntas solicitadas.
3. Ao se deparar com a mensagem "Com qual setor você gostaria de falar?"selecionar a opção **Assessoria Administrativa**.
4. Enviar sua dúvida ou solicitação.

Resultado esperado: O usuário solicitará o chamado para a Assessoria Administrativa com sucesso, e recebe a mensagem "Tudo certo! Nós registramos o seu chamado. Obrigado! Fique de olho em seu e-mail."

4.2.4 Caso de teste 4 - Solicitação de chamado para a Prefeitura Universitária

Objetivo: Testar a funcionalidade de solicitação de chamados para a Prefeitura Universitária.

Passos:

1. Enviar uma mensagem para o CCAE Bot .
2. Responder as perguntas solicitadas.
3. Ao se deparar com a mensagem "Com qual setor você gostaria de falar?"selecionar a opção **Prefeitura Universitária**.
4. Responder as perguntas solicitadas.

Resultado esperado: O usuário solicitará o chamado para a Prefeitura Universitária com sucesso, e recebe a mensagem "Tudo certo! Nós registramos o seu chamado. Obrigado! Fique de olho em seu e-mail."

5 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Esta seção visa exibir as análises dos casos de testes para validação do CCAE Bot , apresentando o método utilizado antes e depois da implementação. Não foi possível colher métricas com usuários reais, pois o sistema será lançado posteriormente a apresentação do presente relatório.

5.1 Interface da aplicação

5.1.1 Tela Inicial

A Figura 10 apresenta o primeiro contato do chatbot. Esta tela contempla o [RF01].

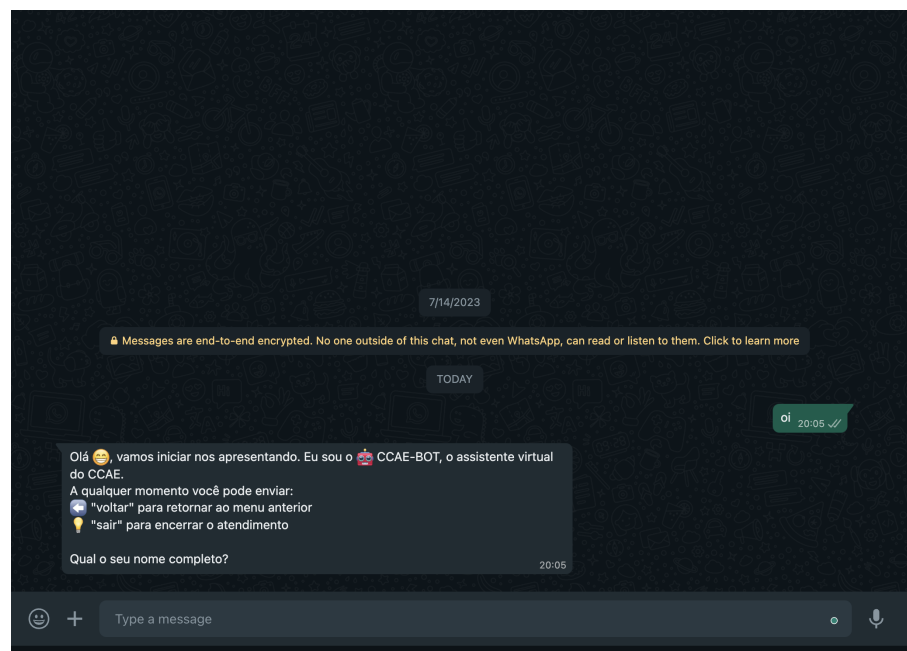


Figura 10: Tela inicial

5.1.2 Coleta de dados do usuário

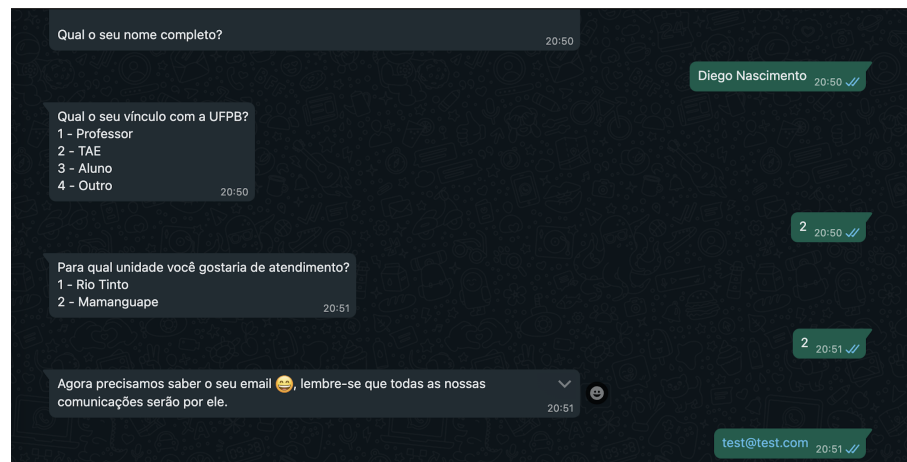


Figura 11: Tela de coleta de dados do usuário

5.1.3 Setores disponíveis para abertura de chamado

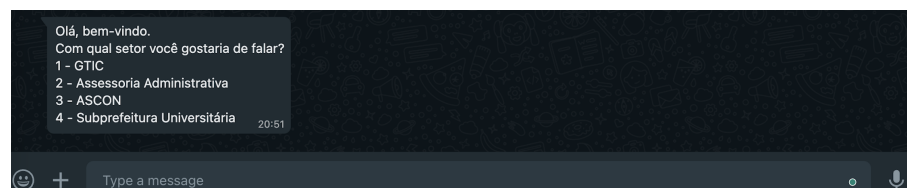


Figura 12: Tela dos setores disponíveis para abertura de chamado

5.1.4 Filtro de opções personalizado por usuário

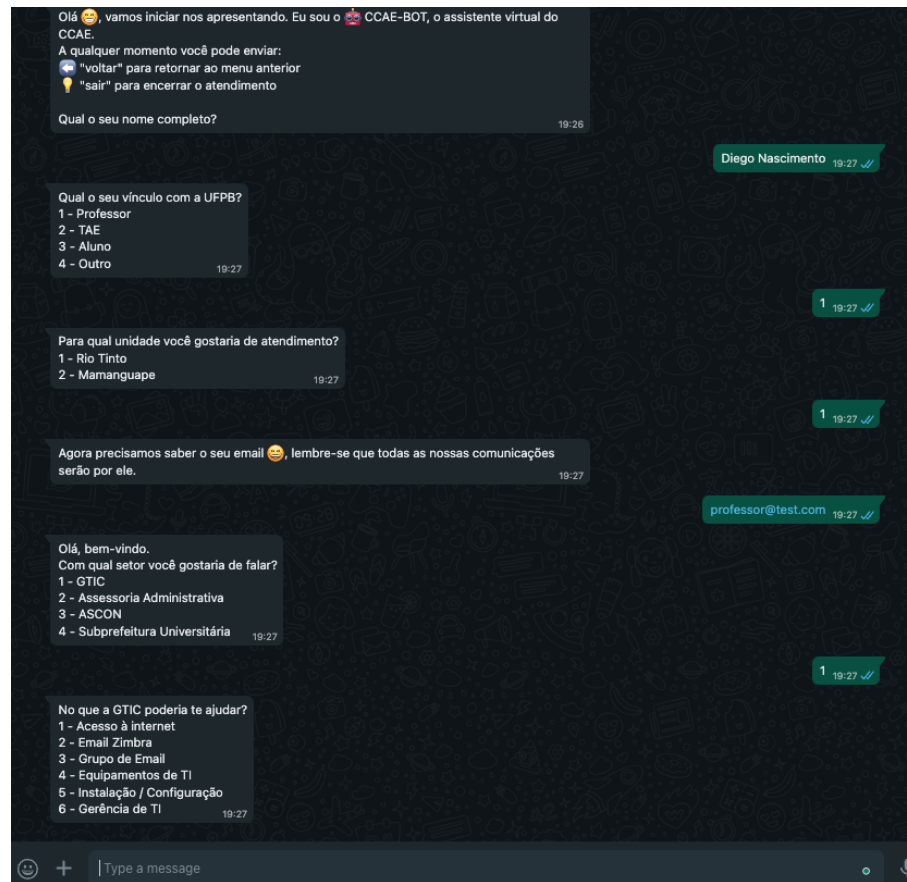


Figura 13: Tela de filtro de opções para usuários do tipo professor

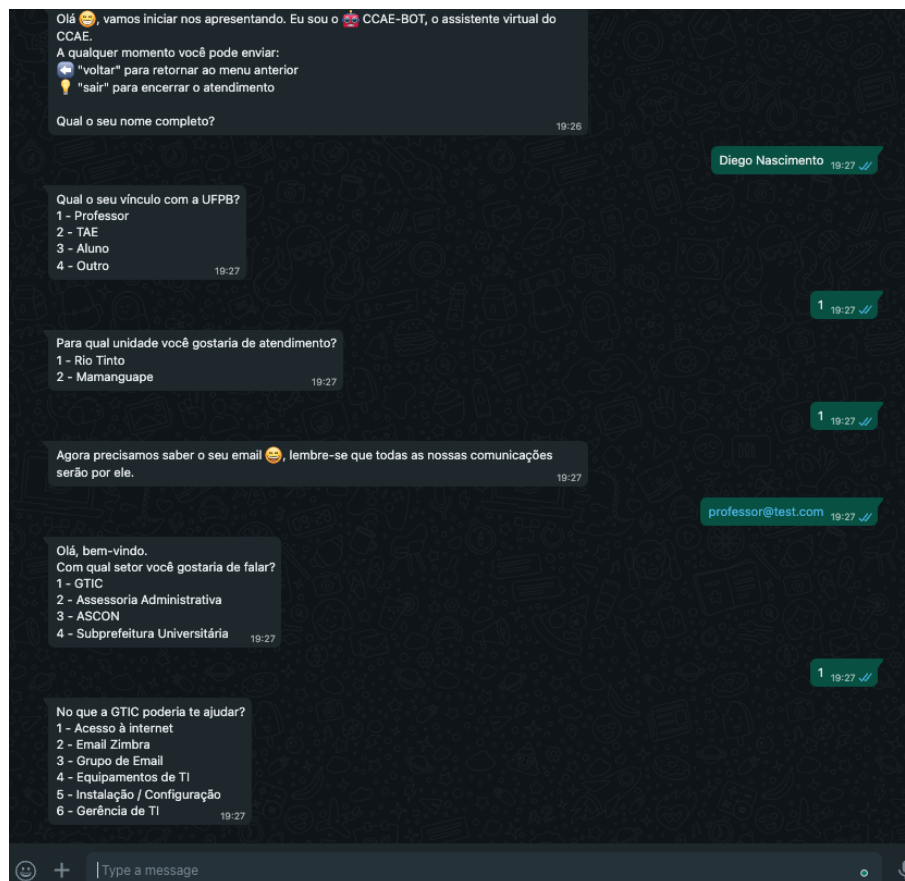


Figura 14: Tela de filtro de opções para usuários do tipo aluno

5.1.5 Ação de voltar

Ao receber a palavra "voltar" como resposta, o chatbot reenvia a pergunta anterior para que o usuário possa alterar sua resposta.

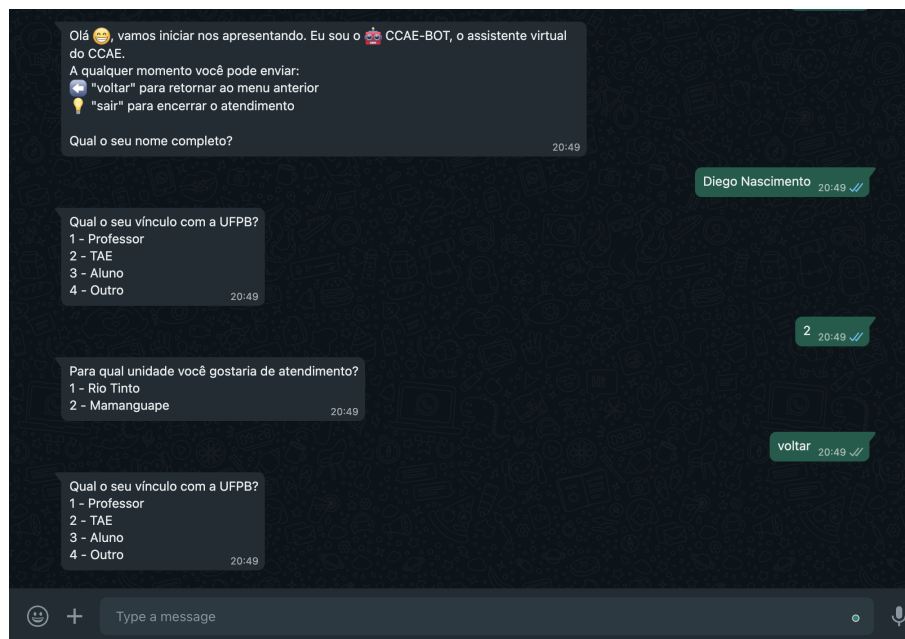


Figura 15: Tela ação de voltar

5.1.6 Ação de sair



Figura 16: Tela de ação de "sair"

5.1.7 Reenvio de mensagem ao receber resposta com opção inexistente

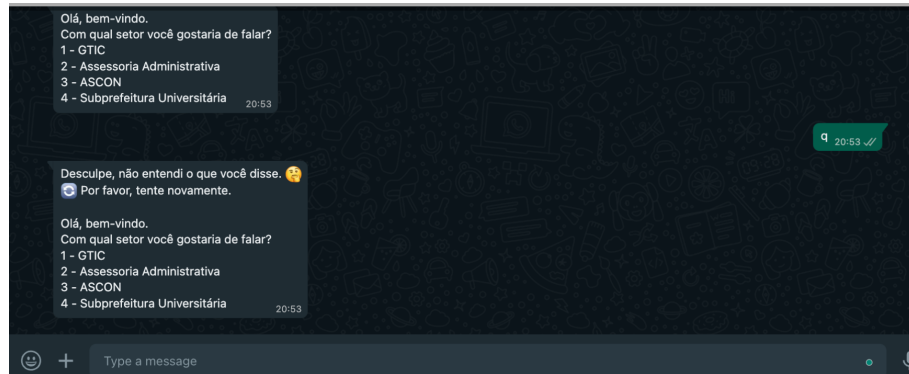


Figura 17: Tela de reenvio de mensagem ao receber resposta com opção inexistente

5.2 Análise dos resultados

As funcionalidades corresponderam como esperado ao utilizamos as personas descritas na sessão 5.1 em cada caso de testes. Estas representam os diversos de usuários, com diferentes idades e tipos de vínculos. Isso permitiu uma avaliação abrangente da aplicação em diferentes cenários.

Todos os requisitos funcionais foram contemplados no desenvolvimento do sistema. Após receber uma mensagem de texto, o CCAE Bot inicia uma conversa com os usuários. As personas foram capazes **“voltar”** para um menu anterior e cancelar o atendimento utilizando o comando **“sair”** em qualquer parte do fluxo do *chat*. Adicionalmente, o sistema conseguiu **detectar as opções incorrentas** e realizar o **reevio das perguntas**. Toda a interação ocorre via Whatsapp e ao final de cada atendimento um *ticket* é criado no sistema do setor correspondente.

A funcionalidade de **filtro personalizado** funcionou conforme esperado. As personas A e B, que são usuários dos tipos **Professor** e **TAE**, respectivamente, conseguem visualizar todas as opções do menu da GTIC. Em contrapartida, as personas C e D, que são usuários dos tipos **Aluno** e **Outros**, não têm acesso a essas opções.

6 CONCLUSÕES E TRABALHOS FUTUROS

6.1 Conclusões

Este trabalho teve como objetivo realizar um estudo de caso acerca de uma aplicação de chatbot projetada para realizar a abertura de chamados em setores prestadores de serviços no campus IV da Universidade Federal da Paraíba. O processo tem início na elicitación de requisitos, onde o problema foi delineado, seguido pela realização dos estudos de viabilidade das soluções propostas e análise dos domínios de negócio e aplicação. Adicionalmente, entendemos o design arquitetural utilizado por meio do modelo C3 e as tecnologias envolvidas durante o processo.

Cada setor da universidade possui seu próprio sistema de atendimento. Alguns utilizam ferramentas projetadas especificamente para esse propósito, como o OTRS ou Hesk, enquanto outros recorrem a soluções alternativas, como formulários *online* ou *emails*. O *chatbot* realiza a integração com todos esses sistemas, isso implica que a aplicação necessite conhecer e gerenciar cada domínio, o que pode ser um ponto de atenção ao atribuir funcionalidades semelhantes a domínios diferentes. A utilização do *WhatsApp* para a interação proporciona ganhos em usabilidade, uma vez que a aplicação fica acessível tanto em dispositivos desktop quanto móveis. Também é importante destacar a necessidade de alinhamento entre os formulários das aplicações e as perguntas do chatbot. Qualquer alteração nesses formulários deve ser refletida no CCAE Bot para que haja consistência de dados e informações.

O CCAE Bot proporciona uma comunicação eficiente entre os setores do campus IV. Os testes realizados mostraram que ela atende às expectativas, oferecendo uma experiência amigável e funcionalidades úteis. Além disso, a ferramenta cria chamados de forma transparente para os setores responsáveis pelo atendimento. A origem do chamado é indistinguível para o colaborador, seja ele criado a partir do chatbot ou do formulário. Isso é vantajoso, pois elimina a necessidade de um novo treinamento para os colaboradores.

6.1.1 Limitações e Dificuldades enfrentadas

Embora os resultados sejam positivos, é importante destacar algumas limitações deste trabalho. A principal limitação é a falta de testes com usuários reais antes da data de apresentação do relatório. A realização desses testes proporcionaria uma avaliação mais precisa das funcionalidades e dos requisitos não funcionais de usabilidade da aplicação.

A pluralidade de sistemas de chamados envolvidos no ecossistema do chatbot também se caracterizou uma dificuldade durante o processo. Uma vez que as regras de

domínio em cada aplicação são únicas. Visto isso, tomamos que abarcar diversas regras de negócio diferentes em um único sistema pode ser confuso do ponto de vista arquitetural e estrutural.

6.2 Trabalhos Futuros

A partir desse trabalho temos uma gama de oportunidades de melhorias, tanto em funcionalidades quanto em desing arquitetural. Trabalhos futuros podem incluir a implementação de funcionalidades adicionais, como menus clicáveis, envio de mídias (imagens, vídeos e áudios) e aprimoramentos na interface do usuário para tornar a aplicação ainda mais amigável. Adicionalmente, integração com departamentos e coordenações ou até mesmo a implementação de *Large Language Model* - LLM, uma vez que as inteligências artificiais tem muito a contribuir nesse domínio de aplicação.

Observando as oportunidades do ponto de vista arquitetural, podemos evoluir a aplicação para uma abordagem de microserviços, o que seria uma melhora de escalabilidade para adição de outros e disponibilidade de serviços.

Referências

- BIERMAN, G.; ABADI, M.; TORGENSEN, M. Understanding typescript. In: SPRINGER. *ECOOP 2014–Object-Oriented Programming: 28th European Conference, Uppsala, Sweden, July 28–August 1, 2014. Proceedings* 28. [S.l.], 2014. p. 257–281.
- FIGUEIREDO, E. Requisitos funcionais e requisitos não funcionais. *IceX, Dcc/Ufmg*, v. 14, p. 2–3, 2011.
- LE, D. A. E-commercial full stack web application development: with react, redux, nodejs, and mongodb. 2023.
- LOKMAN, A. S.; AMEEDEN, M. A. Modern chatbot systems: A technical review. In: SPRINGER. *Proceedings of the Future Technologies Conference (FTC) 2018: Volume 2*. [S.l.], 2019.
- META. *Sobre o WhatsApp Web e WhatsApp para computador*. 2023. Acesso em: 02 de novembro de 2023. Disponível em: https://faq.whatsapp.com/668538004658079/?locale=pt_BR&cms_platform=web&category=5245235.
- Meta Platforms, Inc. and affiliates. *Jest*. 2023. Acesso em: 02 de novembro de 2023. Disponível em: <https://jestjs.io/>.
- NETWORK, M. D. *MDN Web Docs*. 2023. Acesso em: 28 de outubro de 2023. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
- Palácio do Governo do Estado da Paraíba. Lei nº 1.366, de 2 de dezembro de 1955. Cria a Universidade Federal da Paraíba e dá outras providências. *Diário Oficial [do] Estado (DOE)*, Paraíba, PB, 1955. Acesso em: 04 de novembro de 2023. Disponível em: <https://www.ufpb.br/aci/contents/documentos/documentos-ufpb/lei-estadual-de-criacao-da-ufpb-1955.pdf>.
- PAN, J. Software testing. *Dependable Embedded Systems*, Citeseer, v. 5, n. 2006, p. 1, 1999.
- RODRÍGUEZ, R. E. et al. Chatbots en redes sociales para el apoyo oportuno de estudiantes universitarios con síntomas de trastorno por déficit de la atención con hiperactividad. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, SciELO Argentina, n. 22, p. 52–62, 2018.
- SOMMERVILLE, I. *Software Engineering*. 9th. ed. [S.l.]: Addison-Wesley, 2011. 54, 104, 141 p. ISBN 978-0137035151.
- STEFANOSKI, K.; KARADIMCHE, A.; DIMITRIEVSKI, I. Performance comparison of c++ and javascript (node. js-v8 engine). *Research Gate*, 2019.
- TEAM, W. *WPPConnect Documentation*. 2023. Acesso em: 31 de outubro de 2023. Disponível em: <https://wppconnect.io/docs/>.
- TYPESCRIPT. *TypeScript Documentation*. 2023. Acesso em: 28 de outubro de 2023. Disponível em: <https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html>.

ANEXO A – ANEXOS E APÊNDICES 1

O código do CCAE Bot , na versão em que este trabalho é escrito, encontra-se disponível em: <https://github.com/FelipeNasci/wpp-bot/tree/tcc-presentation>. Acesso em: 20/05/2024.