

FACULDADE SENAC PALHOÇA
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Celso Ronaldo de Paiva Moura Junior

Gabriel Dutra Alves

João Vinicius de Aguiar

Marcos João Nunes Júnior

Matheus da Silva Amaral

EDUBOT

**CHATBOTS EM ESCOLAS INFANTIS: UMA FERRAMENTA PARA MELHORAR A
COMUNICAÇÃO ENTRE PAIS E INSTITUIÇÃO**

Palhoça

2024

Celso Ronaldo de Paiva Moura Junior

Gabriel Dutra Alves

João Vinicius de Aguiar

Marcos João Nunes Júnior

Matheus da Silva Amaral

EDUBOT

**CHATBOTS EM ESCOLAS INFANTIS: UMA FERRAMENTA PARA MELHORAR A
COMUNICAÇÃO ENTRE PAIS E INSTITUIÇÃO**

Trabalho apresentado à [Faculdade Senac Palhoça] como
requisito parcial para obtenção do título de [Tecnólogo em
Análise e Desenvolvimento de Sistemas].

Orientador:

Wildson Caio Felipe, Esp.

Jaime Elias Vieira, Me.

Telma Alessandra Correa da Silva, Me.

Coorientador: Luciano Figueiredo Coelho, Me.

Palhoça

2024

Ficha de identificação da obra

E24 Edubot: *chatbots* em escolas infantis: uma ferramenta para melhorar a comunicação entre pais e instituição / Celso Ronaldo de Paiva Moura Junior... [et al.] – Palhoça (SC): Faculdade Senac Palhoça, 2024.

75 p.: il. color.

Orientadores: Jaime Elias Vieira, Telma Alessandra Correa da Silva, Wildson Caio Felipe. Coorientador: Luciano Coelho Figueiredo.

Trabalho de Conclusão de Curso (Graduação em Análise e Desenvolvimento de Sistemas) – Faculdade Senac Palhoça, 2024.

1. Educação infantil. 2. *Chatbot*. 3. Comunicação escolar I. Moura Junior, Celso Ronaldo de Paiva. II. Alves, Gabriel Dutra. III. Aguiar, João Vinicius de. IV. Nunes Júnior, Marcos João Nunes. V. Amaral, Matheus da Silva. VI. Vieira, Jaime Elias de. VII. Silva, Telma Alessandra Correa da. VIII. Felipe, Wildson Caio. IX. Figueiredo, Luciano Coelho. X. Título.

CDD 22 ed.: 371.3078

Ficha catalográfica elaborada pela bibliotecária Cristiane de Melo – CRB 14/1413



Creative Commons - Atribuição - Não Comercial CC BY-NC

Celso Ronaldo de Paiva Moura Junior

Gabriel Dutra Alves

João Vinicius de Aguiar

Marcos João Nunes Júnior


Matheus da Silva Amaral

EDUBOT

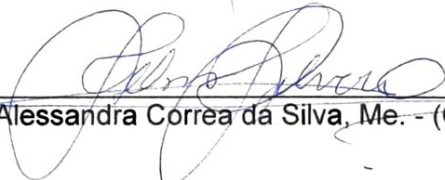
**CHATBOTS EM ESCOLAS INFANTIS: UMA FERRAMENTA PARA MELHORAR A
COMUNICAÇÃO ENTRE PAIS E INSTITUIÇÃO**

Trabalho apresentado à Faculdade Senac Palhoça como
requisito parcial para obtenção do título de Tecnólogo em
Análise e Desenvolvimento de Sistemas.

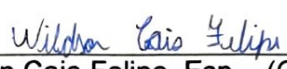
Orientador: Prof. Jaime Elias Vieiras, Me.
Prof. Telma Alessandra Correa da Silva, Me.
Prof. Wildson Caio Felipe, Esp.
Coorientador: Prof. Luciano Figueiredo Coelho, Me.



Jaime Elias Vieira, Me. - (Orientador)



Telma Alessandra Correa da Silva, Me. - (Orientador)



Wildson Caio Felipe, Esp. - (Orientador)



Luciano Figueiredo Coelho, Me. - (Coorientador)

Palhoça, 25 de Junho de 2024

RESUMO

Este trabalho aborda a implementação e o uso de *chatbots* em contextos educacionais, com foco em escolas infantis. O objetivo principal é analisar como essa tecnologia pode melhorar a comunicação entre pais e a instituição, oferecendo suporte e informações de maneira eficiente e acessível. Para isso, foi desenvolvido um sistema de chatbot que permite a interação dos pais com a escola, possibilitando ações como a consulta de listas de material escolar e merenda, visualização de datas de matrícula e lista de espera, além da interação direta com o coordenador escolar. Utilizamos uma metodologia de pesquisa exploratória e qualitativa para obter conhecimentos em uma área desconhecida e identificar os principais desafios enfrentados. A criação do produto começou com um levantamento de requisitos e a escolha das melhores tecnologias, utilizando a metodologia Scrum para o desenvolvimento do projeto. O método inclui uma revisão bibliográfica sobre tecnologias de chatbot na educação, seguida de um plano para implementação e avaliação em uma escola infantil. Espera-se que os resultados mostrem uma significativa melhoria na satisfação dos pais com a comunicação escolar, redução do tempo de resposta às suas dúvidas e maior facilidade no acesso a informações relevantes. Conclui-se que a adoção de chatbots em escolas infantis tem o potencial de otimizar a comunicação e o relacionamento entre pais e instituição, contribuindo para um ambiente escolar mais integrado e eficiente.

Palavras-chave: Chatbots. Educação infantil. Comunicação escolar. Tecnologia educacional. Interação escola-pais.

ABSTRACT

This paper addresses the implementation and use of chatbots in educational contexts, focusing on kindergarten schools. The main objective is to analyze how this technology can improve communication between parents and the institution, providing support and information efficiently and accessibly. For this purpose, a chatbot system was developed that allows parents to interact with the school, enabling actions such as checking school supplies and lunch lists, viewing enrollment dates and waiting lists, and direct interaction with the school coordinator. We used an exploratory and qualitative research methodology to gain knowledge in an unfamiliar area and to identify the main challenges faced. The product creation began with a requirements gathering and selection of the best technologies, using the Scrum methodology for project development. The method includes a literature review on chatbot technologies in education, followed by a plan for implementation and evaluation in a kindergarten school. It is expected that the results will show a significant improvement in parent satisfaction with school communication, a reduction in response time to their inquiries, and greater ease in accessing relevant information. It is concluded that the adoption of chatbots in kindergarten schools has the potential to optimize communication and the relationship between parents and the institution, contributing to a more integrated and efficient school environment.

Keywords: Chatbots, Early Childhood Education, School Communication, Educational Technology, Parent-School Interaction.

LISTAS DE ILUSTRAÇÕES

Figura 1 - Arquitetura.....	21
Figura 2 - Requisitos Funcionais.....	29
Figura 3 - Requisitos Não Funcionais.....	31
Figura 4 - Diagrama de Caso de Uso.....	33
Figura 5 - Tabela de caso de Uso.....	34
Figura 6 - Diagrama de Sequência.....	36
Figura 7 - Diagrama de atividade.....	37
Figura 8 - Diagrama de Classe (Banco de Dados).....	38
Figura 9 - Tela de login.....	39
Figura 10 - Tela de Cadastro.....	40
Figura 11 - Tela Home Usuário.....	41
Figura 12 - Tela Home Coordenador.....	42
Figura 13 - Tela de Parametrização 1.....	43
Figura 14 - Tela de Parametrização 2.....	43
Figura 15 - Tela de Relatórios.....	44
Figura 16 - Tela de Agendamentos.....	44
Figura 17 - Tela de treinamento do Bot.....	46
Figura 18 - Tela do Domínio.....	47
Figura 19 - Tela das Histórias.....	48
Figura 20 - Tela das Regras.....	49
Figura 21 - Tela das Ações.....	50
Figura 22 - Trello.....	52
Figura 23 - GitHub.....	53
Figura 24 - Tela de Registro.....	54
Figura 25 - Tela de Login.....	55
Figura 26 - Tela Home.....	55
Figura 27 - Tela de Ajuda.....	56
Figura 28 - Tela Home (Conversando).....	57
Figura 29 - Tela de Registro.....	58
Figura 30 - Tela de Login.....	58
Figura 31 - Tela Home Coordenador.....	59

Figura 32 - Tela de Relatórios.....	60
Figura 33 - Tela de Agendamentos.....	61
Figura 34 - Tela de Parametrização 1.....	62
Figura 35 - Tela de Parametrização 2.....	63

LISTA DE SIGLAS

ABNT – Associação Brasileira de Normas Técnicas

ADS – Análise e Desenvolvimento de Sistemas

CSS - Cascading Style Sheets

C# - C Sharp.

DB - Database

GIT - Get In Touch

HTML - Hyper Text Markup Language

IDE - Integrated Development Environment

JS - Java Script

PDF- Portable Document Format.

UML - Unified Modeling Language

UI - Design de interface

.NET - .NET Framework

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVOS	12
1.1.1	Objetivo geral	12
1.1.2	Objetivos específicos	12
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	INTRODUÇÃO AOS CHATBOTS	14
2.2	APLICAÇÕES DE CHATBOTS NA EDUCAÇÃO	14
2.3	BENEFÍCIOS DE CHATBOTS NA EDUCAÇÃO	15
2.4	CAPACITAÇÃO HUMANA NO USO DE IA E CHATBOTS	15
2.5	CHATBOTS E A HUMANIZAÇÃO DO ATENDIMENTO AO CLIENTE	15
2.6	DESAFIOS E CONSIDERAÇÕES NA IMPLEMENTAÇÃO	16
3	METODOLOGIA	17
3.1	METODOLOGIA DE PESQUISA	17
3.1.1	Técnica de Coleta de Dados	18
3.2	METODOLOGIA DE CRIAÇÃO DE PRODUTO	18
3.2.1	Análise e Definição dos Requisitos	18
3.2.2	Ambiente de Desenvolvimento	18
3.2.3	Metodologia Ágil	19
3.2.4	Projeto de sistema e Software	19
3.2.3.1	<i>Arquitetura</i>	20
3.2.3.2	<i>UML</i>	22
3.2.3.3	<i>Diagrama de Caso de Uso</i>	22
3.2.3.4	<i>Diagrama de Atividade</i>	23
3.2.3.5	<i>Prototipação</i>	23
3.2.4	Tecnologias	23
3.2.4.1	<i>React JS</i>	24
3.2.4.2	<i>Python</i>	24
3.2.4.3	<i>C#</i>	24
3.2.4.4	<i>.NET</i>	24
3.2.4.5	<i>JavaScript</i>	25
3.2.4.6	<i>CSS</i>	26
3.2.4.7	<i>HTML</i>	26
3.2.4.8	<i>Material UI</i>	26
3.2.4.9	<i>MongoDB</i>	27
3.2.4.10	<i>Rasa</i>	27
3.2.4.11	<i>Conclusão: Aplicação das Tecnologias no Edubot</i>	27
4	RESULTADOS E DISCUSSÃO	29
4.1	ANÁLISE DE REQUISITOS	29

4.1.2	Requisitos Funcionais	29
4.1.3	Requisitos Não Funcionais	30
4.2	ARQUITETURA DO SOFTWARE	32
4.2.1	Modelagem.....	32
4.2.1.1	<i>Diagrama de Caso de Uso</i>	32
4.2.1.2	<i>Diagrama de Sequência</i>	35
4.2.1.3	<i>Diagrama de Atividade</i>	36
4.2.2	Estrutura do Banco de Dados	37
4.2.3	Prototipação	38
4.2.4	Treinamento e configuração RASA: IA do chatbot	45
4.2.4.1	<i>NLU (Natural Language Understanding)</i>	45
4.2.4.2	<i>Domain (Domínio)</i>	46
4.2.4.3	<i>Stories (Histórias)</i>	47
4.2.4.4	<i>Rules (Regras)</i>	48
4.2.4.5	<i>Actions (Ações)</i>	49
4.2.5	Arquitetura em Microserviço	50
4.3	AMBIENTE DE DESENVOLVIMENTO	51
4.4	FUNCIONAMENTO DO SOFTWARE	53
4.4.1	Passo a Passo Usuário Regular.....	53
4.4.1.1	<i>Passo 1</i>	54
4.4.1.2	<i>Passo 2</i>	54
4.4.1.3	<i>Passo 3</i>	55
4.4.1.4	<i>Passo 4</i>	56
4.4.1.5	<i>Passo 5</i>	56
4.4.2	Passo a Passo Usuário Coordenador.....	57
4.4.2.1	<i>Passo 1</i>	57
4.4.2.2	<i>Passo 2</i>	58
4.4.2.3	<i>Passo 3</i>	59
4.4.2.4	<i>Passo 4</i>	59
4.4.2.5	<i>Passo 5</i>	60
4.4.2.6	<i>Passo 6</i>	61
4.4.2.7	<i>Passo 7</i>	62
5	CONCLUSÃO	64
	REFERÊNCIAS.....	66
	APÊNDICE A – QUESTIONÁRIO.....	70

1 INTRODUÇÃO

No cenário contemporâneo, embora a tecnologia da informação e comunicação tenha avançado significativamente, ainda nos deparamos com desafios substanciais que afetam adversamente o desempenho das organizações. Um dos principais fatores que influenciam diretamente esses resultados é o processo de comunicação. A forma como as informações são trocadas, compartilhadas e compreendidas dentro e fora das organizações desempenha um papel crucial na eficácia operacional e no alcance dos objetivos estratégicos. Neste contexto, aprimorar os sistemas de comunicação torna-se não apenas uma necessidade, mas uma prioridade para as organizações que buscam se manter competitivas e adaptáveis em um ambiente em constante evolução.

De acordo com Comarella (2008):

Chatterbots são aplicativos que simulam uma conversa de um ser humano. Essa característica tem despertado o interesse das diversas áreas do conhecimento, inclusive da Ciência da Informação, na busca por uma interface que se aproxime da linguagem natural humana.

Pode ser a resposta para sanar a dificuldade que muitas famílias enfrentam ao lidar com os processos administrativos de uma escola de educação infantil. Ao tornar mais fácil a obtenção de informações e resolução de problemas do dia-a-dia através de um *chatbot* acessível e amigável, espera-se reduzir a sobrecarga na coordenação e melhorar a experiência de todos os envolvidos na comunidade escolar.

Este trabalho visa apresentar uma solução para uma instituição de educação infantil a qual apresenta um problema recorrente que tem afetado tanto famílias quanto à coordenação: a dificuldade em lidar com os processos de matrícula, transferências e lista de espera através do site da prefeitura. Muitas famílias se veem obrigadas a recorrer pessoalmente à secretaria da escola em busca de informações, sobrecarregando os funcionários e resultando em repetições das informações passadas aos pais. Diante desse cenário desafiador, surge a questão: Como podemos simplificar e tornar mais acessíveis esses processos burocráticos?

Propomos criar um *chatbot* que possa auxiliar a coordenação escolar na distribuição de informações e auxílio aos pais que estão lidando com questões administrativas, como, solicitação de vagas, processo de matrícula, resolução de

dúvidas sobre transferências, etc. O objetivo é tornar a comunicação mais rápida e fácil para as famílias obterem as informações que precisam, reduzindo a sobrecarga na secretaria e melhorando a comunicação com a escola.

Por meio de uma entrevista, foi feito um levantamento específico dessas necessidades, as quais poderão ser sanadas por meio do *chatbot*.

Com esta solução proposta, espera-se não apenas resolver os problemas atuais enfrentados pela instituição de educação infantil, mas também estabelecer um modelo de comunicação eficiente que possa ser replicado em outras instituições, contribuindo para a melhoria contínua da comunicação e dos processos administrativos escolares.

1.1 OBJETIVOS

Este trabalho visa trazer eficiência e praticidade para a entrega de informações aos pais dos alunos e à gestão escolar. Nosso objetivo principal é garantir que as informações necessárias sejam entregues de forma precisa e prática, atendendo às necessidades de ambas as partes. Além disso, buscamos utilizar o EDUBOT como solução para retirar a sobrecarga de trabalho braçal enfrentada pela gestão escolar, contribuindo para uma operação mais eficiente e focada em atividades estratégicas.

1.1.1 Objetivo geral

Desenvolver um *chatbot* para uma instituição de educação infantil com o intuito de melhorar a comunicação entre a coordenação e pais, facilitando o acesso à informações acerca da instituição e questões administrativas.

1.1.2 Objetivos específicos

- Criar um *chatbot* amigável que possa lidar com essas necessidades, facilitando o atendimento entre a coordenação da instituição e pais de alunos, ajudando a obterem respostas rápidas e precisas;
- Promover e disponibilizar informações do site da prefeitura no produto criado;

- Testar o *chatbot* para garantir que ele seja eficaz e fácil de usar.
- Realizar testes no *chatbot* para garantir que os requisitos sejam atendidos.
- Disponibilizar uma tela prática e eficaz para o usuário.
- Disponibilizar a criação de relatórios para conhecimento da coordenação.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão discutidos os principais aspectos teóricos relacionados ao desenvolvimento e implementação de *chatbots* em contextos educacionais, especialmente em escolas infantis. Serão exploradas as aplicações de chatbots na educação, destacando suas funcionalidades e como podem ser integrados para melhorar a interação entre pais e coordenadores. Além disso, serão analisados os benefícios que essa tecnologia pode trazer para o ambiente escolar, como a automatização de respostas a perguntas frequentes e a facilitação do acesso a informações. Por fim, serão discutidos os desafios e considerações na implementação de *chatbots*, incluindo questões técnicas, éticas e práticas que precisam ser abordadas para garantir seu uso eficaz e seguro nas escolas.

2.1 INTRODUÇÃO AOS CHATBOTS

Segundo Sengupta e Lakshman (2017), os *chatbots* são programas de computador que utilizam Inteligência Artificial para manter conversas em linguagem natural com os utilizadores. Eles entendem a mensagem do usuário, fornecem respostas com base nas regras de negócio e nos dados da organização. Podem compreender linguagem verbais e ou escritas e respondem tanto em caixas de diálogo quanto verbalmente a partir de áudios.

2.2 APLICAÇÕES DE CHATBOTS NA EDUCAÇÃO

A aplicação de *chatbots* na educação tem se mostrado promissora para facilitar a comunicação entre escolas, alunos e pais. Em particular, no âmbito dos sistemas de gestão da aprendizagem, os *chatbots* são implantados para executar diversas tarefas, tais como esclarecer dúvidas dos estudantes sobre o curso (Paschoal *et al.*, 2022), promover a participação dos alunos em sala de aula, entre outros.

2.3 BENEFÍCIOS DE *CHATBOTS* NA EDUCAÇÃO

Ao implantar *chatbots* em escolas infantis, diversos benefícios podem ser alcançados. Polatidis (2014) explora a perspectiva de que os *chatbots* têm a capacidade de adquirir novos padrões e conceitos de interação com os usuários, resultando em uma melhoria automática da ferramenta. Durante as interações, esses sistemas podem identificar semelhanças entre conversas e, conseqüentemente, determinar a abordagem mais adequada para atender às expectativas dos usuários.

2.4 CAPACITAÇÃO HUMANA NO USO DE IA E *CHATBOTS*

De acordo com Júnior (2023), as máquinas são programadas para imitar o comportamento humano, uma vez que são ensinadas por pessoas que orientam o seu desenvolvimento. Portanto, é cada vez mais essencial capacitar as pessoas para que saibam utilizar e gerenciar adequadamente cada uma das novas tecnologias.

A principal tendência na área tecnológica é que ela se torne uma grande aliada nos setores de gamificação do sistema educacional e da comunicação, sendo cada vez mais explorada para auxiliar no desenvolvimento de habilidades personalizadas, tornando-se assim uma facilitadora do processo de aprendizagem (Júnior, 2023).

2.5 *CHATBOTS* E A HUMANIZAÇÃO DO ATENDIMENTO AO CLIENTE

Moreira (2021) destaca que os *chatbots* têm sido amplamente utilizados para tornar o atendimento ao cliente mais humanizado, mesmo sendo automatizado. Com essa tecnologia, os consumidores podem se comunicar diretamente com a empresa, sem a necessidade de intervenção humana por parte dela. O uso de avatares e personas em *chatbots* facilita o atendimento e o relacionamento, proporcionando aos clientes a sensação de estarem sendo atendidos por uma pessoa, mesmo que seja um robô.

2.6 DESAFIOS E CONSIDERAÇÕES NA IMPLEMENTAÇÃO

Entretanto, a implementação de *chatbots* em escolas infantis também apresenta desafios. De acordo com Aguiar (2023), um dos principais desafios reside na disparidade de disponibilidade de recursos tecnológicos e acesso à internet, o que pode acarretar desigualdades no processo de aprendizagem e na participação dos estudantes, contribuindo para um aumento da exclusão social.

3 METODOLOGIA

Nesta seção, será apresentada a metodologia utilizada para conduzir a pesquisa e desenvolver o produto proposto. Serão descritos os métodos e técnicas empregados para alcançar os objetivos propostos, garantindo a robustez e a consistência do estudo. Esta metodologia inclui desde a definição dos procedimentos de coleta de dados até a análise e interpretação dos resultados obtidos. Essa estrutura metodológica é fundamental para assegurar a validade dos achados e a relevância das conclusões apresentadas neste trabalho.

3.1 METODOLOGIA DE PESQUISA

Para conduzir este trabalho, adotamos uma abordagem qualitativa, a fim de compreender profundamente as necessidades e percepções dos usuários em relação ao produto em desenvolvimento. Inicialmente, realizamos uma entrevista com uma diretora de uma creche pública da cidade da Palhoça - Santa Catarina durante o período de 13 de Março de 2024 à 20 de março de 2024, visando explorar suas experiências, opiniões e expectativas.

Além disso, para validar a eficácia das soluções propostas, conduzimos uma segunda fase de pesquisa qualitativa. Nessa etapa, nos concentramos em verificar se as funcionalidades concebidas atendiam adequadamente às necessidades identificadas durante as entrevistas. Utilizamos métodos como grupos focais e observação participante para coletar dados adicionais e obter insights mais aprofundados.

Ao adotar essa abordagem qualitativa em duas etapas distintas, buscamos garantir uma compreensão abrangente das necessidades dos usuários e a validação efetiva das soluções propostas, contribuindo assim para a qualidade e relevância do trabalho desenvolvido.

3.1.1 Técnica de Coleta de Dados

Nesta etapa, a coleta de dados foi aplicada através de entrevista com perguntas elaboradas pelo grupo, realizada via *WhatsApp*. A pesquisa foi dividida em duas etapas, ambas qualitativas. Na primeira etapa, foram elaboradas perguntas mais abertas a fim de explorar e conhecer as necessidades do ambiente, iniciando o escopo do projeto. Na segunda etapa, foi feita uma entrevista mais detalhada após o início do desenvolvimento, explicando os requisitos ao entrevistado para verificar se o sistema estava atendendo à necessidade do produto. O conteúdo detalhado das perguntas e respostas das entrevistas pode ser encontrado no apêndice deste trabalho.

3.2 METODOLOGIA DE CRIAÇÃO DE PRODUTO

A metodologia de criação do produto, destacada no desenvolvimento do EDUBOT, é crucial para garantir uma comunicação eficaz entre pais e escolas infantis. O levantamento de requisitos e testes asseguram a funcionalidade e a adequação do sistema às necessidades dos usuários.

3.2.1 Análise e Definição dos Requisitos

Para obter os requisitos funcionais e não funcionais do sistema, o formulário contendo perguntas para validar o produto e as entrevistas desempenharam um papel fundamental. Essas abordagens foram essenciais para compreender e perceber a complexidade do problema, especialmente considerando a participação ativa de coordenadores e diretores escolares, que atuam na área da educação.

3.2.2 Ambiente de Desenvolvimento

Durante o desenvolvimento do sistema EduBot, foram utilizadas duas ferramentas principais para o ambiente de codificação: *Visual Studio Code* (VS Code) e *Visual Studio*. O VS Code foi utilizado tanto para integrar e desenvolver a inteligência

artificial do *chatbot* quanto para o desenvolvimento do *front-end*. Este editor de código-fonte, leve e altamente versátil, desenvolvido pela Microsoft, é reconhecido por sua interface amigável e por uma vasta gama de recursos avançados, que aumentam a qualidade e a eficiência do desenvolvimento. Paralelamente, o *Visual Studio* foi empregado no desenvolvimento da aplicação. Esta poderosa ferramenta oferece um ambiente de desenvolvimento integrado (IDE) abrangente, que simplifica a codificação, depuração e implantação da aplicação. A combinação dessas duas ferramentas permitiu um desenvolvimento ágil e eficiente do sistema EduBot, aproveitando as vantagens de cada uma para atender às diversas necessidades do projeto.

3.2.3 Metodologia Ágil

Antes de explorar a origem e a influência do Manifesto Ágil, é fundamental entender o conceito e a gestão de projetos. De acordo com as diretrizes do Guia PMBok 6ª edição, um projeto é uma empreitada temporária realizada para criar um produto, serviço ou resultado específico. Em outras palavras, possui um período definido de início e término, visando alcançar objetivos por meio da entrega de resultados. O encerramento de um projeto ocorre quando os objetivos são alcançados ou quando se determina que não são viáveis, levando ao encerramento do projeto ou à sua descontinuação (Project Management Institute, 2017).

Ainda conforme o autor, a definição de gestão de projetos envolve a aplicação de conhecimentos, habilidades, ferramentas e técnicas para conduzir as atividades dentro de um projeto, visando atender aos seus requisitos. Através da gestão de projetos, uma organização é capacitada a executar um projeto de maneira eficaz e eficiente, auxiliando tanto indivíduos quanto a própria organização a alcançarem os objetivos do negócio, resolverem desafios e aumentarem suas chances de sucesso.

3.2.4 Projeto de sistema e Software

O processo de concepção e desenvolvimento de *software* desempenha um papel crucial na criação de um sistema, assumindo a responsabilidade pelo estudo e

planejamento abrangente do projeto. Isso implica na elaboração de um plano diretor que orientará todas as etapas do desenvolvimento.

3.2.3.1 Arquitetura

A arquitetura de microsserviços é uma abordagem arquitetural que se propõe como uma alternativa à arquitetura monolítica. Diferente desta, a abordagem de microsserviços visa isolar um único contexto da aplicação em uma unidade autossuficiente de *software*, com seus próprios recursos, como banco de dados, serviços de cache, etc. De acordo com Fowler (2014), o termo Arquitetura de Microsserviços ganhou destaque recentemente como uma forma alternativa de estruturar e construir aplicações como unidades de serviços independentes e autônomas. Embora não exista uma definição formal que descreva essa forma de arquitetar aplicações, há algumas características comuns entre as aplicações que utilizam essa arquitetura.

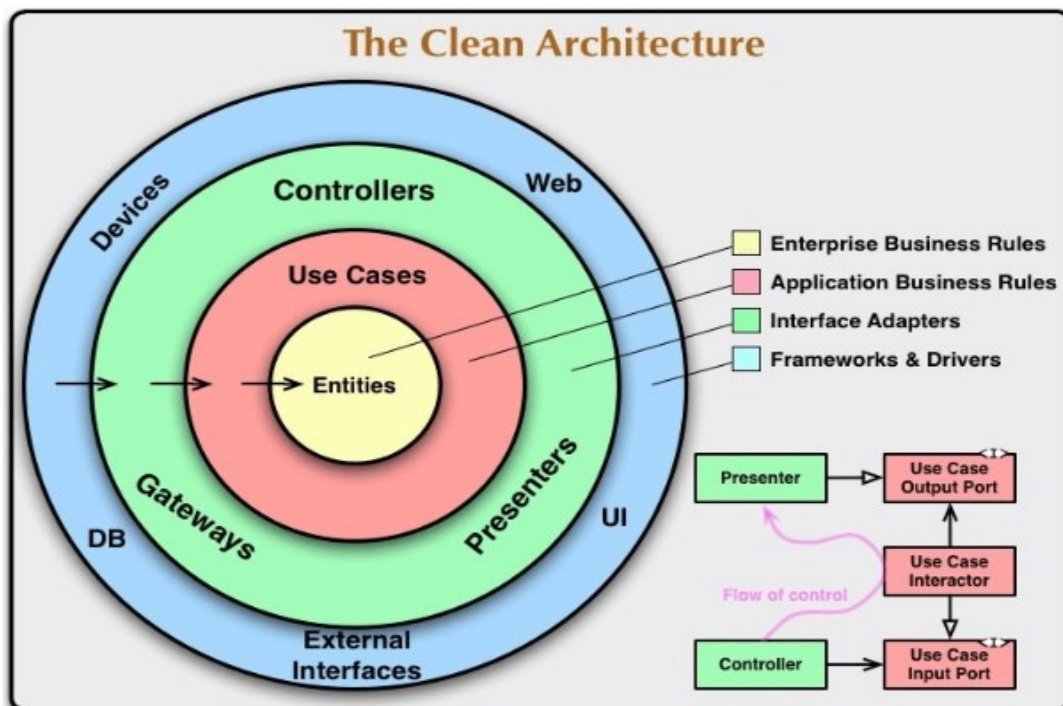
A arquitetura de microsserviços é uma alternativa relativamente nova às abordagens monolíticas. A ideia central dessa arquitetura é isolar os contextos que compõem as funcionalidades do sistema em pequenos serviços que, isolados, não têm muita utilidade, mas, quando integrados, conseguem gerar o valor esperado para os usuários do sistema. Segundo Stoiber (2017), um microsserviço é uma unidade autocontida que, junto com muitas outras, compõem um grande aplicativo. Ao dividir seu aplicativo em pequenas unidades, cada parte é implantável e escalável de forma independente, pode ser desenvolvida por diferentes equipes e em diferentes linguagens de programação, e pode ser testada individualmente.

A *Clean Architecture* (Arquitetura Limpa) foi proposta por Martin (2019) em seu livro *Arquitetura Limpa: O Guia do Artesão para Estrutura e Design de Software*. O objetivo desse modelo de arquitetura é ser utilizado no *design* de codificação para facilitar a manutenção, testes e evolução do *software* em desenvolvimento, através de um sistema de camadas e um baixo grau de dependência.

A arquitetura representa as decisões tomadas durante o desenvolvimento de um *software*, resultando em sua estrutura atual, como a divisão em camadas e a organização do código, com o propósito de facilitar seu desenvolvimento, implementação, operação e manutenção. O principal objetivo da arquitetura de

software é minimizar os recursos humanos necessários para construir e manter um determinado sistema (Martin, 2019).

Figura 1 - Arquitetura



Fonte: Martin (2019)

Segundo Bueno (2021), a Figura 1 ilustra como deve ser feita a divisão das camadas do *software* em desenvolvimento. Essas camadas são divididas da seguinte forma:

Entities (Entidades): As entidades do sistema são responsáveis por encapsular as regras de negócio e outras funções relacionadas à manipulação de dados no sistema. Elas são as menos propensas a mudanças caso algo seja solicitado.

Use Cases (Casos de Uso): Nesta camada, todos os casos de uso que serão invocados pelas camadas externas são encapsulados e implementados. Esses casos de uso administram o fluxo de dados para as regras de negócio, validação e persistência. Modificações nesta camada não devem provocar alterações na camada

de entidades, e mudanças externas não devem resultar em modificações nesta camada.

Interface Adapters (Adaptadores de Interface): Esta camada é responsável por fornecer o código necessário para converter o formato de dados utilizado pelas entidades e casos de uso.

Frameworks & Drivers (*Frameworks* e *Drivers*): A camada mais externa, geralmente composta por bancos de dados, *frameworks*, e outras ferramentas.

3.2.3.2 UML

A *UML*, Linguagem Unificada de Modelagem, é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de software. A UML proporciona uma forma-padrão para a preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais tais como processos de negócios e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, esquemas de bancos de dados e componentes de software reutilizáveis (Booch, 2006).

Chatterbots são aplicativos que simulam uma conversa de um ser humano. Essa característica tem despertado o interesse das diversas áreas do conhecimento, inclusive da Ciência da Informação, na busca por uma interface que se aproxime da linguagem natural humana.

3.2.3.3 Diagrama de Caso de Uso

Conforme enfatizado por Bezerra (2007), o diagrama de caso de uso desempenha um papel crucial na visualização e na compreensão das funcionalidades externamente observáveis de um sistema, bem como dos diversos elementos externos que interagem com ele. Este diagrama, sendo uma ferramenta fundamental na engenharia de requisitos de *software*, não só ilustra as interações entre o sistema em questão e os agentes externos, mas também captura de forma abrangente as ações e os eventos que ocorrem em um ambiente operacional específico. Assim, ele

não apenas esboça as funcionalidades oferecidas pelo sistema, mas também traça os limites do sistema e identifica os pontos de contato com os usuários e outros sistemas, tornando-se uma representação essencial para a análise e o projeto de sistemas de *software* robustos e eficientes.

Oferece uma linguagem clara e acessível, permitindo aos usuários uma visão geral do comportamento do sistema. O objetivo é identificar os atores (sejam usuários, outros sistemas ou até mesmo hardware especial) que interagem com o software, bem como os serviços que o sistema disponibilizará a esses atores, denominados casos de uso neste diagrama (Guedes,2009).

3.2.3.4 Diagrama de Atividade

Segundo (Guedes, 2009), o diagrama de atividade era anteriormente considerado uma variação especial do antigo diagrama de gráfico de estados, que agora é conhecido como diagrama de máquina de estados, conforme detalhado na seção anterior. O diagrama de atividade tem como objetivo descrever os passos necessários para concluir uma atividade específica, que pode ser desde um método com certo nível de complexidade, um algoritmo, até mesmo um processo completo.

3.2.3.5 Prototipação

Um protótipo é uma versão preliminar de um design, que pode incluir um esboço em papel de uma tela ou conjunto de telas, uma imagem eletrônica, uma simulação em vídeo de uma tarefa, uma maquete tridimensional em papel ou cartolina, ou até mesmo um conjunto simples de telas interligadas por *hyperlinks* (Rosemberg, 2008).

3.2.4 Tecnologias

Neste tópico, serão apresentadas as tecnologias utilizadas no desenvolvimento do produto, incluindo linguagens de programação, *frameworks*, ferramentas de prototipagem e outras tecnologias relevantes para a implementação do sistema.

3.2.4.1 React JS

O React.js é uma biblioteca JavaScript de código aberto amplamente utilizada para a criação de interfaces de usuário dinâmicas. Inicialmente concebido pela equipe do *Facebook* e agora mantido pela Meta e pela comunidade de código aberto, o React.js é elogiado por seu modelo de programação declarativa e sua eficácia na construção de interfaces interativas. Sua arquitetura baseada em componentes facilita a reutilização de código, o que, por sua vez, aprimora a eficiência no desenvolvimento e na manutenção de *software* (Watmore, 2020).

3.2.4.2 Python

Desenvolvido por Guido van Rossum nos meados da década de 1990, Python emergiu como uma linguagem de programação de alto nível, notável por sua proximidade com a linguagem natural, em oposição à linguagem de máquina. Reconhecida por sua natureza multiparadigma e pela extensa coleção de bibliotecas destinadas à criação de aplicativos computacionais (Rodrigues *et al*, 2019).

3.2.4.3 C#

A utilização do C# na construção de APIs oferece robustez e eficiência, características intrínsecas a essa linguagem de programação. A ampla gama de recursos disponíveis no .NET *Framework* acelera e otimiza o processo de desenvolvimento. Além disso, o C# é uma linguagem de tipagem forte, o que contribui para evitar diversos erros de programação, resultando em uma melhoria significativa na qualidade do código (Microsoft, 2023).

3.2.4.4 .NET

Segundo (El Saifi, 2006) O Dotnet *Framework*, uma tecnologia desenvolvida pela Microsoft, atua como uma interface entre os programas de aplicação e o sistema

operacional. Ele é composto por duas partes principais: o *Common Language Runtime* (CLR) e a *Framework Class Library* (FCL).

O *Dotnet Framework* oferece uma série de funcionalidades que o tornam uma plataforma robusta para o desenvolvimento de aplicações. Abaixo, destacam-se alguns aspectos do CLR e do FCL conforme descrito por Sharp e Jagger (2003):

- Modelo de programação consistente: Diferentemente das abordagens tradicionais que utilizam funções de bibliotecas dinâmicas (DLLs no Windows) e objetos COM (Common Object Model), o *Dotnet Framework* fornece todas as funcionalidades e serviços por meio de um modelo de programação orientado a objetos, resultando em um modelo de programação unificado.
- Modelo de programação simplificado: O CLR simplifica diversas construções internas do Win32 e COM. Ele elimina a necessidade do programador compreender conceitos como o registro do sistema, identificadores globais únicos (GUIDs), *IUNKNOWN*, *ADDREF*, *Release*, *HRESULTS*, entre outros. Esses conceitos não existem no CLR, exceto quando há necessidade de interoperabilidade com códigos não-Dotnet existentes.

3.2.4.5 JavaScript

A linguagem JavaScript teve sua origem na Netscape Communications Corporation⁴, inicialmente desenvolvida sob o nome de Mocha. Posteriormente, foi renomeada para LiveScript antes de ser oficialmente lançada como JavaScript em 1995, integrada à versão 2.0B3 do navegador Netscape. Seu propósito era implementar uma tecnologia de processamento em modo cliente (Grillo, 2008).

Ainda conforme o autor, JavaScript viabiliza a criação de pequenos programas incorporados diretamente no código de uma página HTML. Esses programas são capazes de executar diversas tarefas, como a geração de números, processamento de dados, validação de formulários, alteração de valores de elementos HTML e criação de novos elementos HTML. Tudo isso ocorre localmente no computador cliente, eliminando a necessidade de troca de informações com o servidor e reduzindo a dependência do tempo de resposta à latência da rede.

3.2.4.6 CSS

CSS é a sigla para Cascading Style Sheet, que em português significa folhas de estilo em cascata. Uma definição clara e concisa para folha de estilo pode ser encontrada em Silva (2007): "Folha de estilo em cascata é um método simples para adicionar estilos (como fontes, cores, espaçamentos) a documentos *web*".

As CSS têm como objetivo devolver ao (X)HTML seu propósito original. A HTML foi criada como uma linguagem exclusivamente para marcação e estruturação de conteúdo. Isso significa que, de acordo com seus idealizadores, não é função da HTML fornecer informações ao usuário sobre a apresentação dos elementos. Por exemplo, cores de fontes, tamanhos de textos, posicionamentos e todos os aspectos visuais de um documento não devem ser responsabilidade do HTML. "Cabe à CSS todas as funções de apresentação de um documento, e esta é sua finalidade maior" (Silva, 2007).

3.2.4.7 HTML

HTML é a sigla em inglês para Hyper Text Markup Language, que, em português, significa linguagem para marcação de hipertexto. O conceito de hipertexto admite um sem-número de considerações e discussões que fogem ao escopo deste livro. Para o bom entendimento das definições, podemos resumir hipertexto como todo o conteúdo inserido em um documento para a *web* e que tem como principal característica a possibilidade de se interligar a outros documentos da *web*. O que torna possível a construção de hipertextos são os *links*, presentes nas páginas dos sites que estamos acostumados a visitar quando entramos na internet.

3.2.4.8 Material UI

De acordo com Soares (2022) "O *framework* de interface Material UI foi criado em 2014 com licença *open source* para tornar o *design*, coincidentemente, mais acessível". Usar um *framework* já popular e construir funcionalidades de acessibilidade sobre ele traz benefícios de adoção uma vez que um *framework* já maduro e popular é capaz de aumentar o nível de acessibilidade das páginas que o utilizam mais rapidamente que construir um *framework* exclusivamente focado em

acessibilidade e convencer toda uma comunidade de usuários a adotar uma nova ferramenta (Soares, 2022).

3.2.4.9 MongoDB

O MongoDB, um sistema não-relacional, é uma ferramenta utilizada para o gerenciamento de bancos de dados. Ele difere dos sistemas relacionais como o MySQL em sua abordagem de armazenamento e recuperação de dados, sendo especialmente adequado para aplicações com grande volume de dados não-estruturados ou semiestruturados. Sua flexibilidade e escalabilidade o tornam uma escolha popular para aplicativos modernos, especialmente em cenários onde a estrutura dos dados pode mudar frequentemente ou precisa ser altamente adaptável (Souza,2019).

3.2.4.10 Rasa

O RASA é uma plataforma de desenvolvimento de ChatBots de código aberto, baseada em PLN (Processamento de Linguagem Natural) e AM (ML, do inglês Machine Learning). Por ser de código aberto, o RASA oferece diversas formas de configurar suas pipelines, como, por exemplo, adicionando ou removendo etapas de processamento (Correa, 2021).

3.2.4.11 Conclusão: Aplicação das Tecnologias no Edubot

As tecnologias apresentadas neste tópico desempenham um papel crucial na construção e no funcionamento do Edubot. O uso do React.js como base para a interface do usuário permite a criação de uma experiência interativa e responsiva, facilitando a interação dos usuários com o sistema. Além disso, o Python, especialmente utilizado com a IA proporcionada pelo Rasa, traz inteligência ao sistema, permitindo interações naturais e eficientes com os usuários. Por sua vez, o C# é fundamental para o desenvolvimento do *backend*, garantindo a robustez e a eficiência necessárias para o processamento de dados e a comunicação com o banco

de dados MongoDB. A integração do .NET Framework proporciona uma estrutura sólida para a construção de APIs, contribuindo para a eficácia geral do sistema. O Material UI e o CSS são empregados para garantir uma apresentação visual atrativa e intuitiva, enquanto o HTML serve como base para a estruturação do conteúdo. Ao aplicarmos essas tecnologias de forma integrada, conseguimos criar uma solução completa e eficiente que atende às necessidades específicas do ambiente escolar, proporcionando uma experiência de usuário aprimorada e facilitando a comunicação entre pais, alunos e coordenadores escolares.

4 RESULTADOS E DISCUSSÃO

Neste capítulo, apresentamos e analisamos os resultados obtidos a partir da implementação do sistema EduBot. Discutimos as implicações dos dados coletados, avaliando o impacto da utilização do *chatbot* na comunicação entre pais e a instituição escolar. Os resultados são interpretados à luz dos objetivos propostos, e são exploradas as possíveis melhorias e desafios identificados durante o processo. A seguir, detalhamos as principais descobertas e suas respectivas análises.

4.1 ANÁLISE DE REQUISITOS

A Análise de requisitos é um processo que envolve o estudo, a compreensão, o detalhamento e a identificação de restrições dos usuários e de suas necessidades em relação ao sistema. O objetivo é captar, documentar e validar as funcionalidades e características do sistema.

4.1.2 Requisitos Funcionais

Em relação aos requisitos funcionais, a figura 2 apresenta uma lista desses requisitos juntamente com as atribuições específicas que o sistema deverá realizar:

Figura 2 - Requisitos Funcionais

IDENTIFICADOR	DESCRIÇÃO
RF1	O sistema deve permitir que usuários registrados façam login utilizando suas credenciais.
RF2	Deve ser possível autenticar usuários com sucesso no sistema.
RF3	O sistema deve oferecer a funcionalidade de criar uma nova conta para usuários que ainda não estão registrados.

RF4	O sistema deve permitir que usuários acessem determinadas funcionalidades sem a necessidade de fazer login, oferecendo a opção de continuar como visitante.
RF5	Usuários devem poder interagir com o chatbot de forma limitada, acessando apenas funcionalidades básicas ou recebendo informações simples.
RF6	Deve ser possível aos usuários interagir livremente com o chatbot, obtendo assistência em diversas áreas e realizando consultas.
RF7	O sistema deve permitir que usuários administrativos solicitem a geração de relatórios específicos, como desempenho sistemático e educacional
RF8	Os usuários devem ter acesso à lista de espera para determinados serviços ou atividades, como matrícula em cursos concorridos.
RF9	O sistema deve fornecer informações sobre as datas de matrícula, incluindo prazos e procedimentos necessários.
RF10	Deve ser possível para os usuários iniciar uma conversa direta com o coordenador da instituição para obter orientação ou resolver questões específicas.
RF11	Deve ser possível o usuário solicitar o agendamento para o chatbot

Fonte: Elaborado pelos autores (2024)

4.1.3 Requisitos Não Funcionais

Os requisitos não funcionais são definidos como requisitos que descrevem as qualidades e atributos do sistema. A figura 3 apresenta esses requisitos não funcionais juntamente com as competências específicas que o sistema deve possuir:

Figura 3 - Requisitos Não Funcionais

IDENTIFICADOR	DESCRIÇÃO
RNF1	O sistema deve ser intuitivo e fácil de usar, mesmo para usuários iniciantes.
RNF2	O sistema deve garantir a segurança das informações dos usuários, protegendo dados pessoais e de login.
RNF3	Deve apresentar uma interface amigável e bem organizada.
RNF4	O sistema deve ter um desempenho rápido e responsivo, mesmo durante períodos de pico de uso.
RNF5	Deve ser minimizado o tempo de resposta para todas as operações do sistema.
RNF6	O sistema deve estar disponível e acessível a qualquer momento, com um tempo de inatividade mínimo.
RNF7	Deve ser projetado para escalar facilmente conforme necessário.
RNF8	Deve ser implementado um controle de acesso granular para garantir que apenas usuários autorizados tenham acesso a informações sensíveis.
RNF9	O sistema deve ser capaz de lidar com um aumento no número de usuários e de dados sem comprometer o desempenho.

Fonte: Elaborado pelos autores (2024)

4.2 ARQUITETURA DO SOFTWARE

A arquitetura de *software* envolve a criação de uma estrutura que especifica a organização dos componentes do sistema e suas interações. Seu objetivo é garantir um *software* de alta qualidade, proporcionando uma base sólida para a implementação e a manutenção do sistema em larga escala.

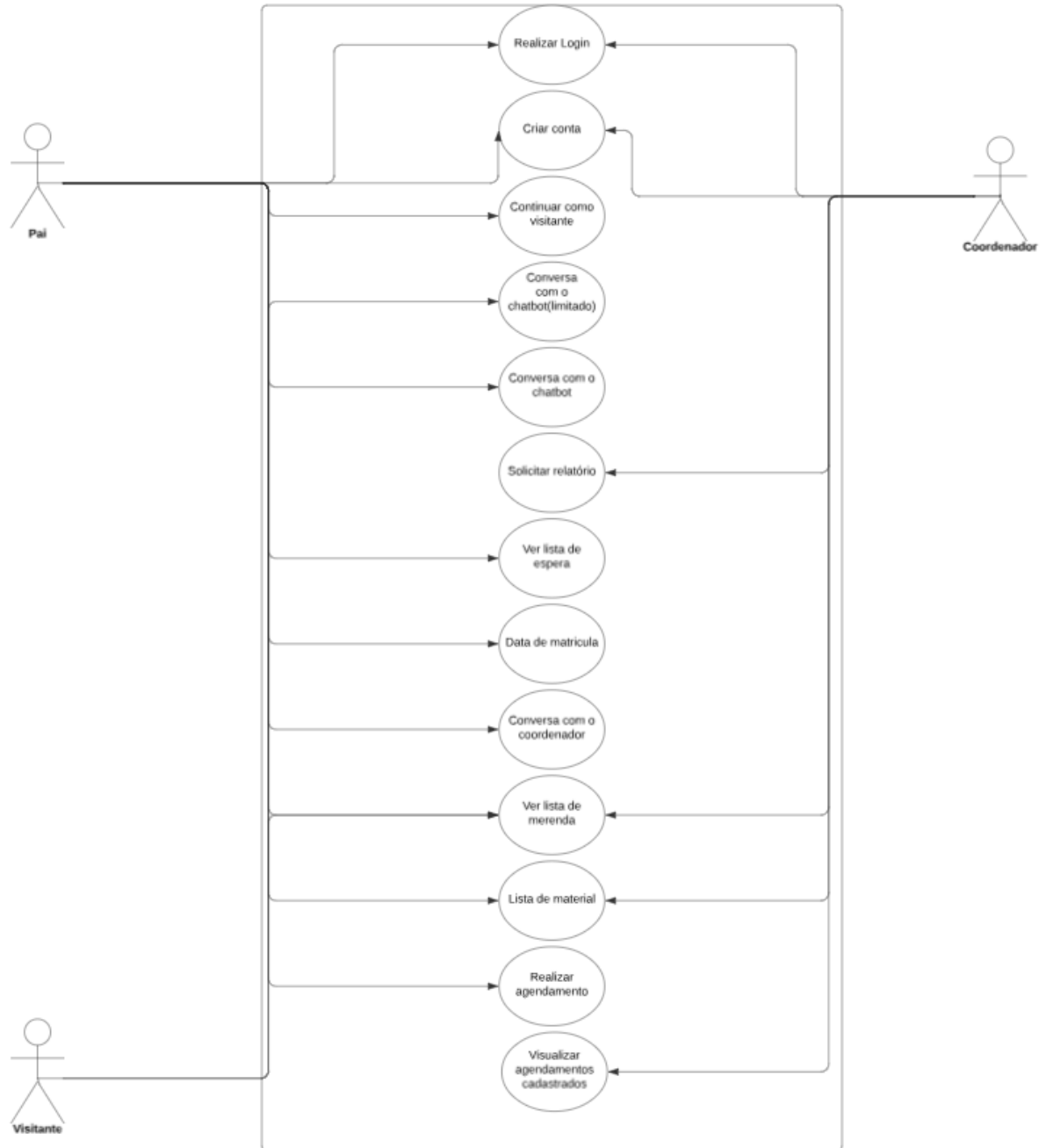
4.2.1 Modelagem

A modelagem de dados é crucial em sistemas de informação porque organiza e estrutura os dados de forma que sejam compreensíveis e utilizáveis pelos sistemas. Ela define como os dados são armazenados, manipulados e relacionados entre si, garantindo consistência, integridade e eficiência nas operações. Uma boa modelagem de dados facilita a tomada de decisões, melhora a precisão das análises e promove a comunicação eficaz entre as partes interessadas.

4.2.1.1 Diagrama de Caso de Uso

Os diagramas de caso de uso são uma ferramenta essencial na UML para a análise inicial de requisitos. Eles ajudam a identificar os atores que interagem com o sistema e os serviços oferecidos, proporcionando uma visão clara e acessível do comportamento (Bezerra,2007).

Figura 4 - Diagrama de Caso de Uso



Fonte: Elaborado pelos autores (2024)

A seguir na figura 5, são elencadas as ações correspondentes ao usuário 'Pai' conforme representadas no diagrama de caso de uso (Figura 4):

Figura 5 - Tabela de caso de uso

Usuário	Ação	Descrição
Visitante	Continuar como visitante	Acessar o sistema como visitante, com acesso limitado ao chatbot para obter informações básicas e redirecionamento para sites de educação pública.
Pais e Coordenador	Criar conta	Criar uma nova conta de usuário para acessar funcionalidades exclusivas do sistema.
Pais e Coordenador	Realizar Login	Autenticar-se no sistema para acessar funcionalidades personalizadas.
Pais e Coordenador	Conversa com o chatbot	Iniciar uma conversa com o chatbot para obter informações e suporte.
Pais, Coordenador e visitante	Ver lista de espera	Visualizar a lista de espera para matrícula em determinado período.
Pais, Coordenador e visitante	Ver lista de matrícula	Consultar a lista de matrícula das escolas no município da Palhoça.
Pais	Conversa com o coordenador	Iniciar uma conversa direta com o coordenador da escola para tratar de assuntos específicos relacionados à educação do filho.
Pais	Solicitar agendamento	Quando o bot não conseguir atender a necessidade do pai, o bot irá disponibilizar a função do pai agendar um atendimento presencial na data disponível mais próxima
Pais, Coordenador e visitante	Ver lista de merenda	Acessar a lista de alimentos disponíveis na merenda escolar.
Pais, Coordenador e visitante	Ver lista de material escolar	Consultar a lista de materiais escolares necessários para o ano letivo.
Pais e Coordenador	Acesso às funcionalidades parametrizadas	Acesso às funcionalidades do chatbot que foram parametrizadas pelo coordenador da escola
Coordenador	Parametrizar o chatbot	Configurar e ajustar o chatbot de acordo com as necessidades de atendimento da escola, definindo respostas automáticas e fluxos de

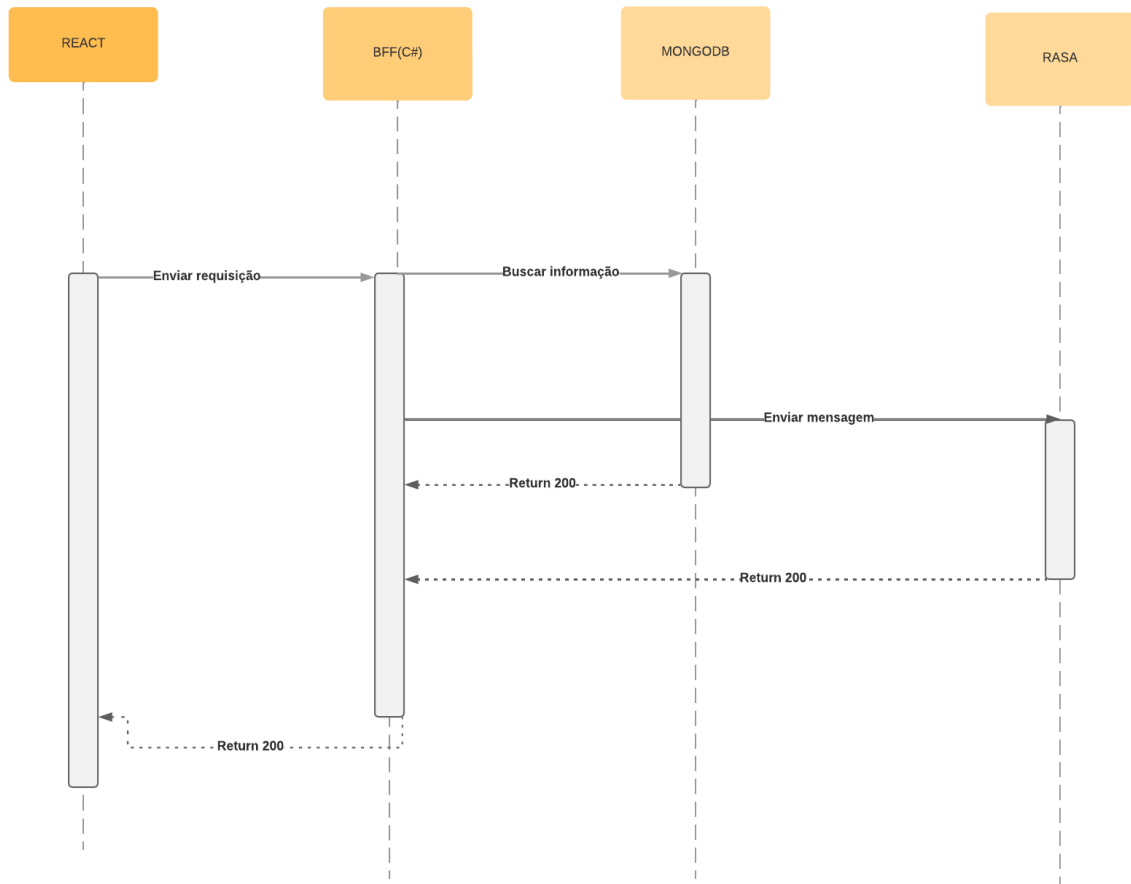
		conversação.
Coordenador	Gerar relatórios	Gerar relatórios sobre o uso do chatbot
Coordenador	Acessar a conversa com o pai	Intervir diretamente nas conversas entre o chatbot e os pais, podendo assumir o controle da comunicação para tratar de assuntos específicos.
Coordenador	Visualizar agendamentos	Visualizar a lista de agendamentos feitos pelo bot

Fonte: Elaborado pelos autores (2024)

4.2.1.2 Diagrama de Sequência

O diagrama de sequência é um tipo de diagrama comportamental que se concentra na ordem temporal das mensagens trocadas entre os objetos em um processo específico. Normalmente, ele se fundamenta em um caso de uso identificado pelo mesmo diagrama e utiliza o diagrama de classes para identificar os objetos das classes envolvidas, conforme figura 6.

Figura 6 - Diagrama de Sequência

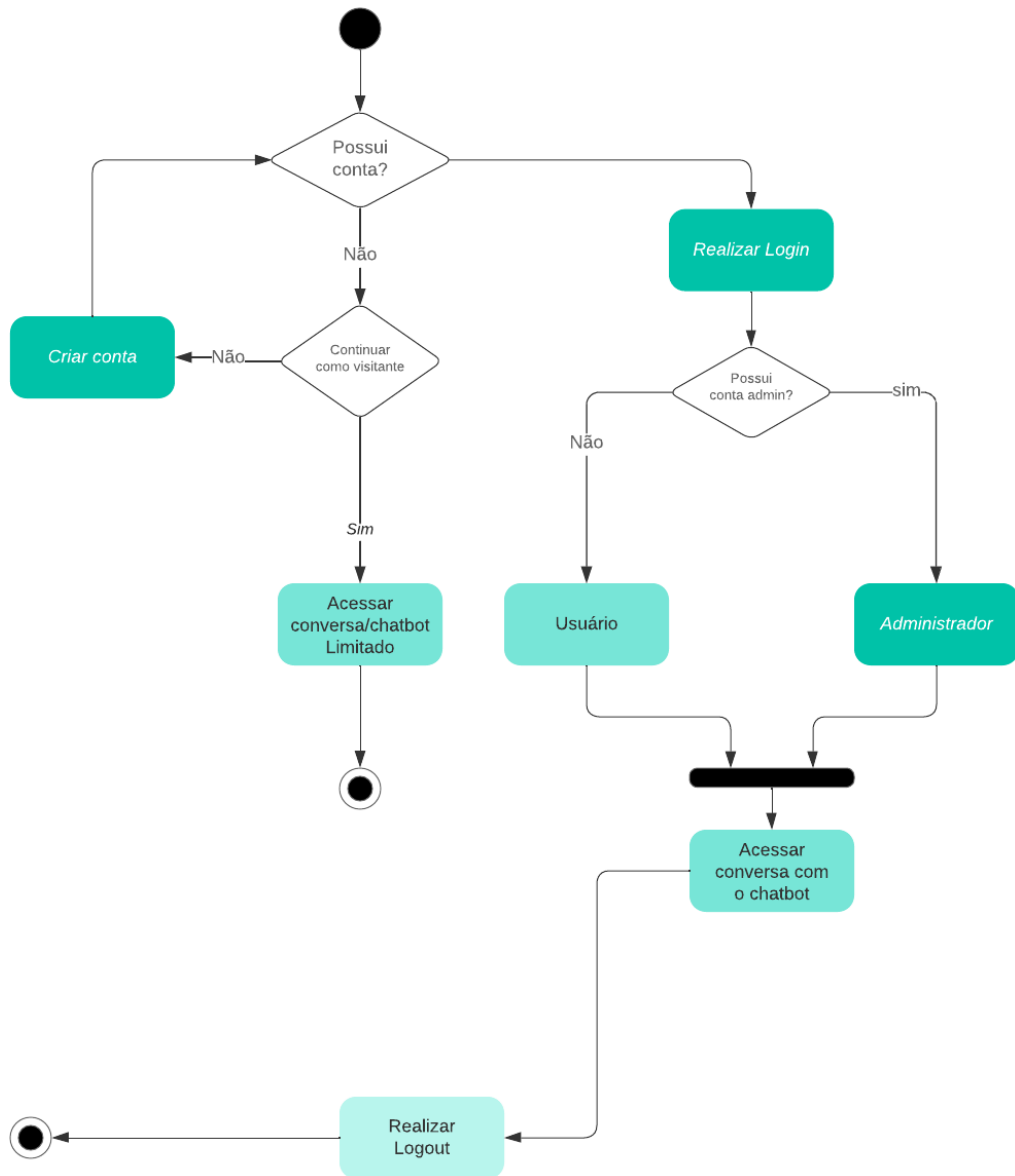


Fonte: Elaborado pelos autores (2024)

4.2.1.3 Diagrama de Atividade

De acordo com Guedes (2009), os diagramas de atividade detalham os passos necessários para completar uma atividade (figura 7), oferecendo uma representação visual de processos complexos, desde métodos e algoritmos até processos complexos. Essa representação visual é essencial para entender e documentar a dinâmica de processos e fluxos de trabalho dentro do sistema, facilitando a análise e a comunicação entre os membros da equipe de desenvolvimento.

Figura 7 - Diagrama de atividade



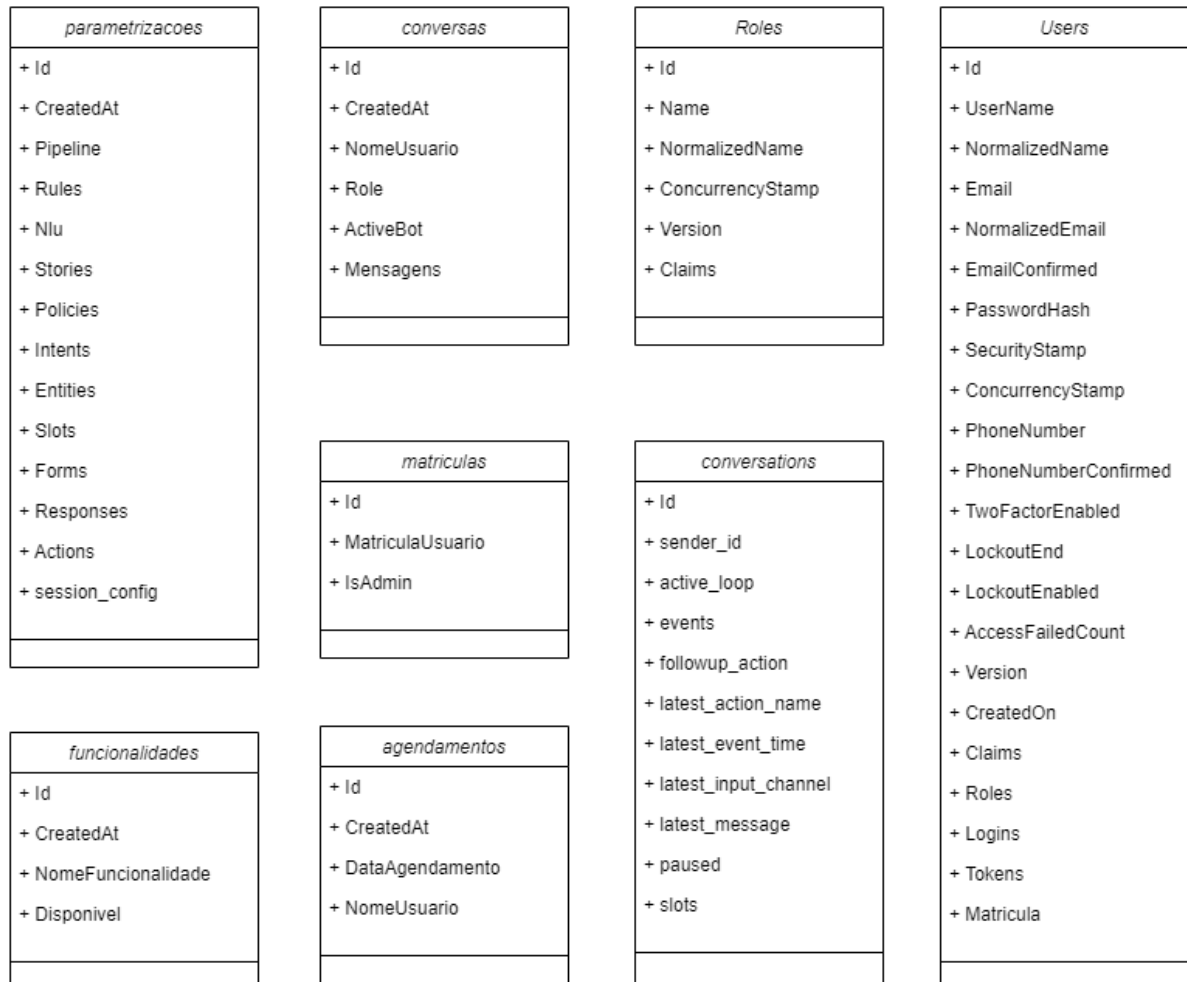
Fonte: Elaborado pelos autores (2024)

4.2.2 Estrutura do Banco de Dados

Segundo Souza (2019), o MongoDB é um sistema de banco de dados não-relacional, ideal para gerenciar grandes volumes de dados não-estruturados ou semiestruturados devido à sua flexibilidade e escalabilidade. No projeto EduBot,

utilizamos o MongoDB para armazenar dados de interação com o *chatbot*, permitindo adaptações frequentes na estrutura dos dados e garantindo a eficiência na recuperação de informações, como mostra a figura 8.

Figura 8 - Diagrama de Classe (Banco de Dados)



Fonte: Elaborado pelos autores (2024)

4.2.3 Prototipação

A prototipação do projeto foi conduzida utilizando a plataforma *web Figma*, escolhida devido à sua acessibilidade, robustez e facilidade de uso. As telas foram prototipadas seguindo uma abordagem embasada nos requisitos funcionais e não funcionais, bem como nos diagramas previamente concebidos. Nas figuras abaixo (9

à 16), podemos visualizar como foi possível desenvolver protótipos interativos de alta fidelidade, proporcionando uma representação realista do produto final:

Figura 9 - Tela de *login*



Tela de Login

Seja bem vindo ao EDU website, onde você acompanha a vida escolar de seu filho(a).

Entre em sua conta

Email

Senha

Entrar

Não Possui Uma Conta? [Crie Sua Conta](#)

[Continuar Como Visitante](#)

Fonte: Elaborado pelos autores (2024)

Figura 10 - Tela de Cadastro

Tela de Registro Coordenação

Crie sua conta

Membro Da Coordenação

Email

Mátricula

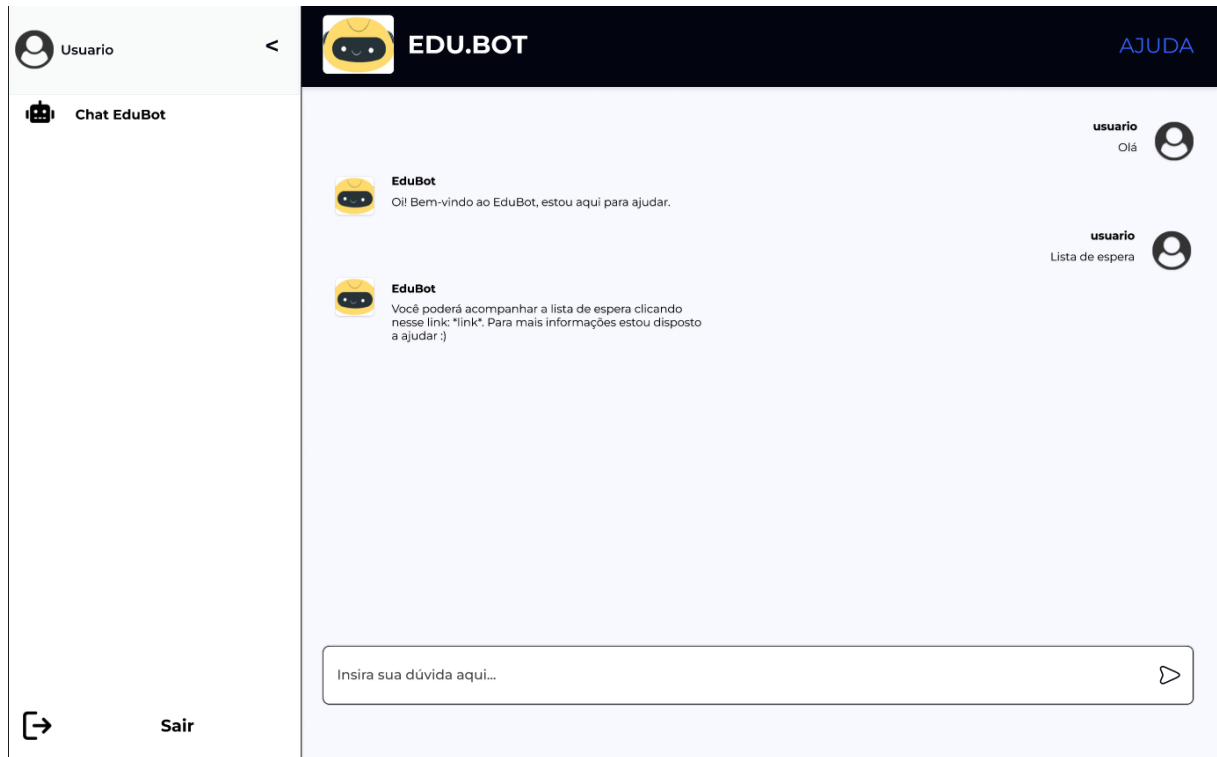
Senha

Confirme Sua Senha

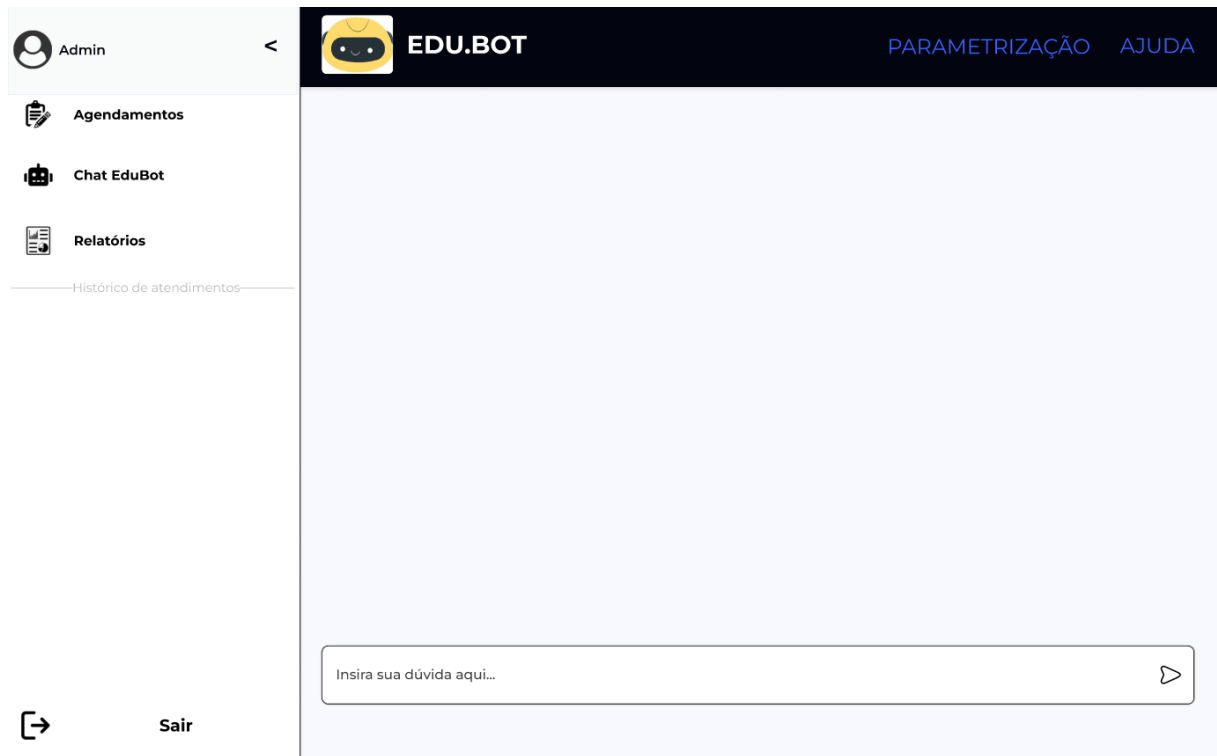
[Criar conta](#)

Já Possui Uma Conta? [Entrar](#)

Fonte: Elaborado pelos autores (2024)

Figura 11 - Tela *Home* Usuário

Fonte: Elaborado pelos autores (2024)

Figura 12 - Tela *Home* Coordenador

Fonte: Elaborado pelos autores (2024)

Figura 13 - Tela de Parametrização 1

Titulo das perguntas *

Bem-vindo à etapa de parametrização do seu chatbot!
Para começar, Você irá cadastrar o título da sua nova pergunta e as possíveis abordagens que os seus usuários terão para fazer a pergunta. Por exemplo:
Como faço para verificar minha matrícula?
Qual é o procedimento para descobrir minha matrícula?
Pode me informar qual é a minha matrícula?

Adicione novas perguntas aqui

BACK NEXT

Fonte: Elaborado pelos autores (2024)

Figura 14 - Tela de Parametrização 2

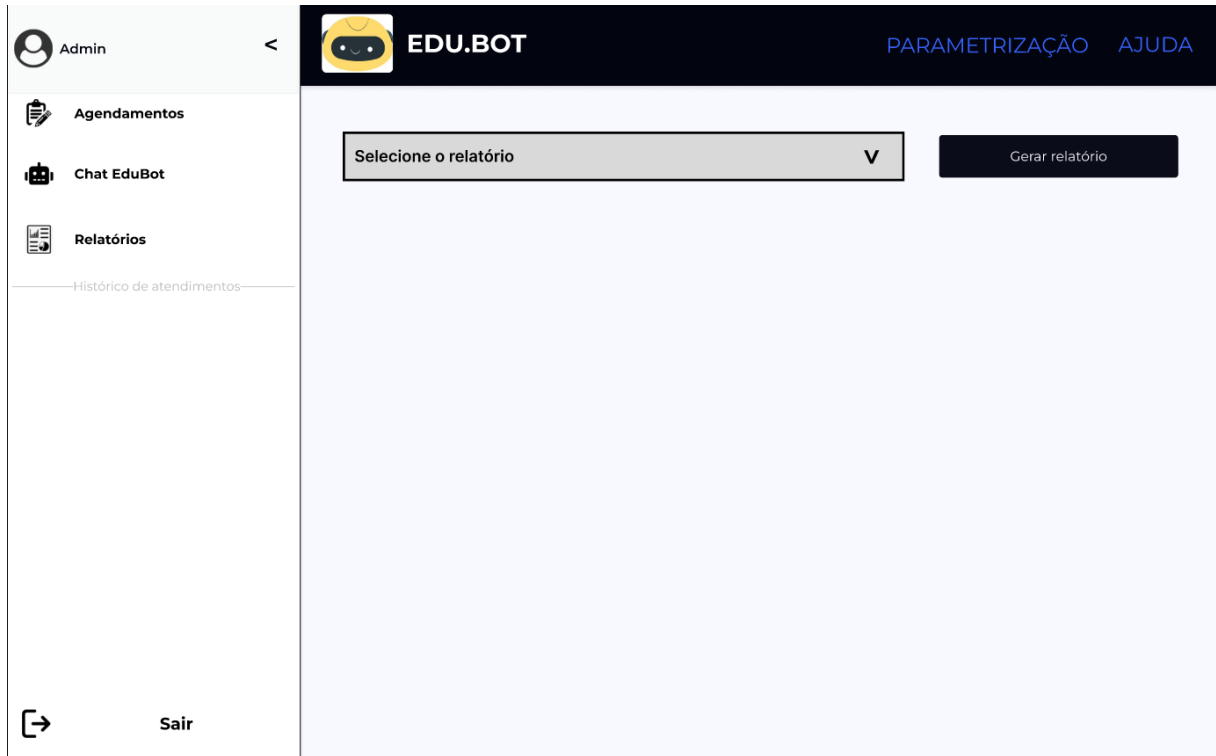
Nessa etapa você irá cadastrar a(s) resposta(s) da sua nova pergunta.

Adicione novas respostas aqui

BACK FINISH

Fonte: Elaborado pelos autores (2024)

Figura 15 - Tela de Relatórios



Fonte: Elaborado pelos autores (2024)

Figura 16 - Tela de Agendamentos



Fonte: Elaborado pelos autores (2024)

4.2.4 Treinamento e configuração RASA: IA do chatbot

Para construir um *chatbot* eficaz com o RASA, é essencial entender e configurar corretamente o treinamento e os componentes principais. Abaixo, apresentamos uma abordagem prática para configurar e treinar o seu *chatbot*.

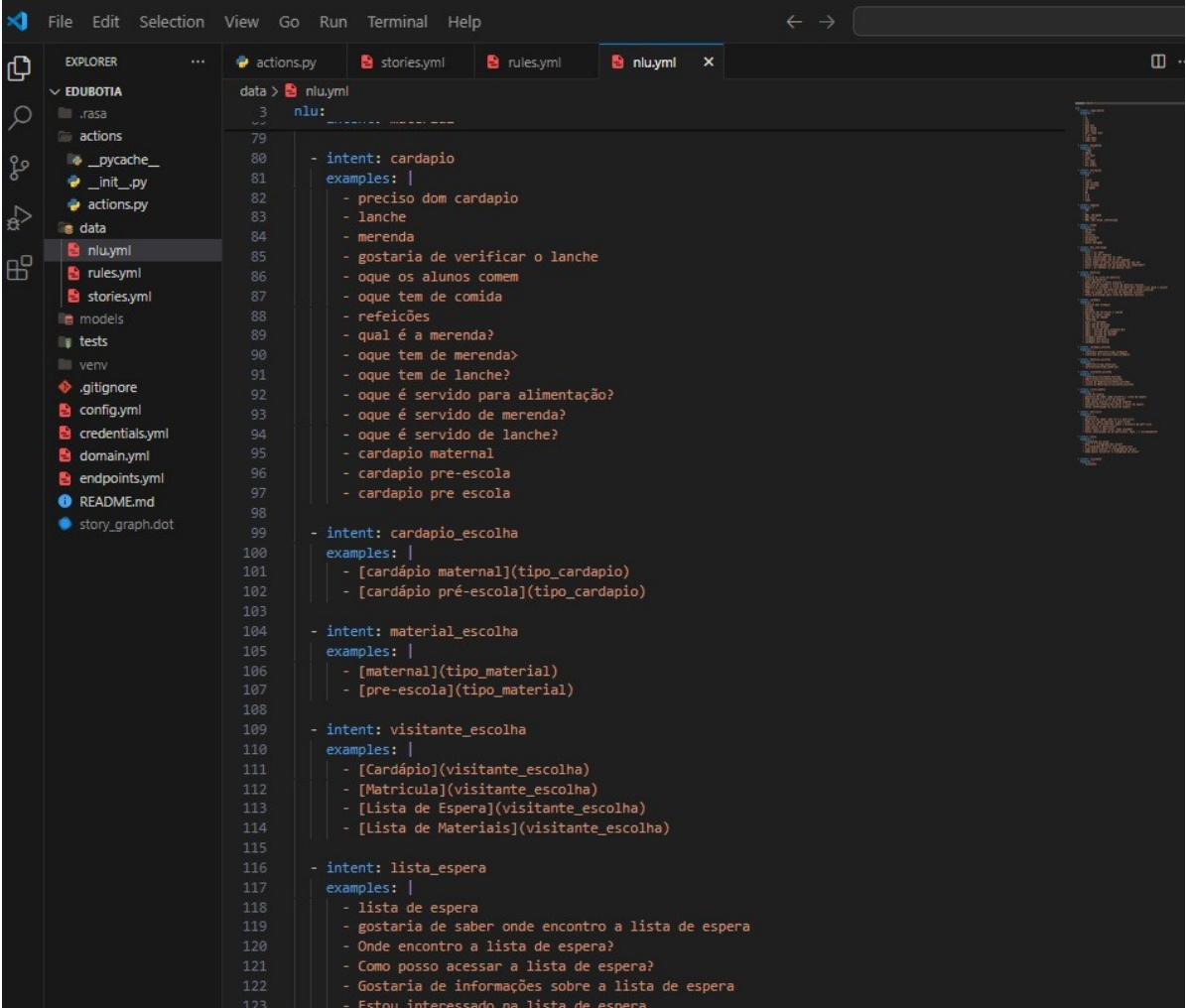
4.2.4.1 NLU (*Natural Language Understanding*)

A Compreensão da Linguagem Natural (NLU) é um componente essencial da arquitetura do RASA, responsável por interpretar as entradas dos usuários e extrair informações relevantes. A configuração da NLU envolve a definição de intenções, entidades e exemplos de treinamento.

A NLU é responsável pela interpretação das entradas textuais dos usuários, permitindo ao chatbot identificar intenções (como saudações, perguntas, comandos, etc.) e entidades (informações específicas como datas, nomes, locais). A configuração adequada da NLU é crucial para garantir que o chatbot compreenda com precisão as consultas dos usuários e responda de maneira apropriada.

Abaixo (figura 17) está um exemplo de configuração do arquivo de treinamento da NLU (`nlu.yml`), que inclui a definição de intenções e exemplos de frases que o *chatbot* deve reconhecer.

Figura 17 - Tela de treinamento do Bot



```

3 nlu:
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
- intent: cardapio
  examples: |
    - preciso dom cardapio
    - lanche
    - merenda
    - gostaria de verificar o lanche
    - oque os alunos comem
    - oque tem de comida
    - refeições
    - qual é a merenda?
    - oque tem de merenda?
    - oque tem de lanche?
    - oque é servido para alimentação?
    - oque é servido de merenda?
    - oque é servido de lanche?
    - cardapio maternal
    - cardapio pre-escola
    - cardapio pre escola

- intent: cardapio_escolha
  examples: |
    - [cardápio maternal](tipo_cardapio)
    - [cardápio pré-escola](tipo_cardapio)

- intent: material_escolha
  examples: |
    - [maternal](tipo_material)
    - [pre-escola](tipo_material)

- intent: visitante_escolha
  examples: |
    - [Cardápio](visitante_escolha)
    - [Matricula](visitante_escolha)
    - [Lista de Espera](visitante_escolha)
    - [Lista de Materiais](visitante_escolha)

- intent: lista_espera
  examples: |
    - lista de espera
    - gostaria de saber onde encontro a lista de espera
    - Onde encontro a lista de espera?
    - Como posso acessar a lista de espera?
    - Gostaria de informações sobre a lista de espera
    - Estou interessado na lista de espera
  
```

Fonte: Elaborado pelos autores (2024)

4.2.4.2 Domain (Domínio)

O arquivo domain.yml define as intenções, entidades, *slots*, ações e respostas que o *chatbot* pode usar.

O arquivo de domínio especifica as intenções e entidades reconhecidas pela NLU, além das ações e respostas do *chatbot*. Ele também configura os *slots*, que armazenam informações das mensagens dos usuários. Abaixo está um exemplo de configuração do arquivo domain.yml, com definições de intenções, entidades, *slots*, ações e respostas, assim como mostra na figura 18.

Figura 18 - Tela do Domínio

```

75 responses:
93   utter_goodbye:
94     - text: "Tchau"
95
96   utter_lista_espera:
97     - text: "
98     você poderá acompanhar a lista de espera clicando nesse link:
99     \nhttps://palhoca.educarweb.net.br/servicoexterno/matricula-2024/listaespera
100     \nPara mais informações estou disposto a ajudar :)
101     "
102   utter_material_maternal:
103     - text: "
104     \nLista de materiais para o Maternal G1, G2 e G3:
105     \n1 Caderno brochura 60 folhas para agenda
106     \n1 Brinquedo (adequado para a faixa etária)
107     \nObservação: Os materiais de higiene pessoal serão solicitados pela Unidade Escolar em que a cr
108     \nPara mais informações estou disposto a ajudar :)
109     "
110
111   utter_lista_material_pre:
112     - text: "
113     \nLista de materiais para o Pré-escola G4 e G5 :
114     \n1 Caderno brochura 60 folhas para agenda
115     \n1 Caderno de desenho
116     \n1 Penal/ estojo
117     \n1 Lápis de cor - 12 cores
118     \n1 Lápis preto
119     \n1 Borracha
120     \n1 Apontador
121     \n1 Cola líquida infantil 35 gramas
122     \n1 Tesoura escolar ponta arredondada
123     \nPara mais informações estou disposto a ajudar :)
124     "
125
126   utter_cardapio:
127     - text: "
128     \nAguarde o download do arquivo.\n
129     "
130
131   utter_cronograma_aulas:
132     - text: "cronograma de aulas"
133
134   utter_matricula:
135     - text: "Você Precisar trazer: \n
136     xerox certidão nascimento ou carteira de identidade \n
137     1 foto 3/4 \n

```

Fonte: Elaborado pelos autores (2024)

4.2.4.3 Stories (Histórias)

As histórias (stories.yml) são usadas para treinar o *chatbot* em diferentes caminhos de conversação, simulando interações reais com os usuários. As histórias são sequências de interações que descrevem como o *chatbot* deve responder com base nas entradas do usuário. Elas são essenciais para treinar o modelo de diálogo, garantindo que o *chatbot* possa lidar com várias situações de forma consistente.

A figura 19 temos um exemplo de configuração do arquivo stories.yml, mostrando diferentes cenários de interação entre o usuário e o chatbot.

Figura 19 - Tela das Histórias

```

84 - story: quando faz elogio ao edubot path 5
85   steps:
86     - intent: otimo
87     - action: utter_otimo
88
89 - story: quando faz negacao ao edubot path 6
90   steps:
91     - intent: negacao
92     - action: utter_negacao
93
94
95 - story: interactive_story_1
96   steps:
97     - intent: material
98     - action: tipo_material_form
99     - active_loop: tipo_material_form
100    - slot_was_set:
101      - requested_slot: tipo_material
102    - intent: material_escolha
103
104
105 - story: interactive_story_2
106   steps:
107     - intent: cardapio
108     - action: tipo_cardapio_form
109     - active_loop: tipo_cardapio_form
110     - slot_was_set:
111       - requested_slot: tipo_cardapio
112     - intent: cardapio_escolha
113
114 - story: interactive_story_3
115   steps:
116     - intent: visitante
117     - action: tipo_visitante_form
118     - active_loop: tipo_visitante_form
119     - slot_was_set:
120       - requested_slot: tipo_visitante
121     - intent: visitante_escolha
122
123 - story: interactive_story_3
124   steps:
125     - intent: visitante
126     - action: tipo_visitante_form
127     - active_loop: tipo_visitante_form
128     - slot_was_set:
129       - requested_slot: tipo_visitante

```

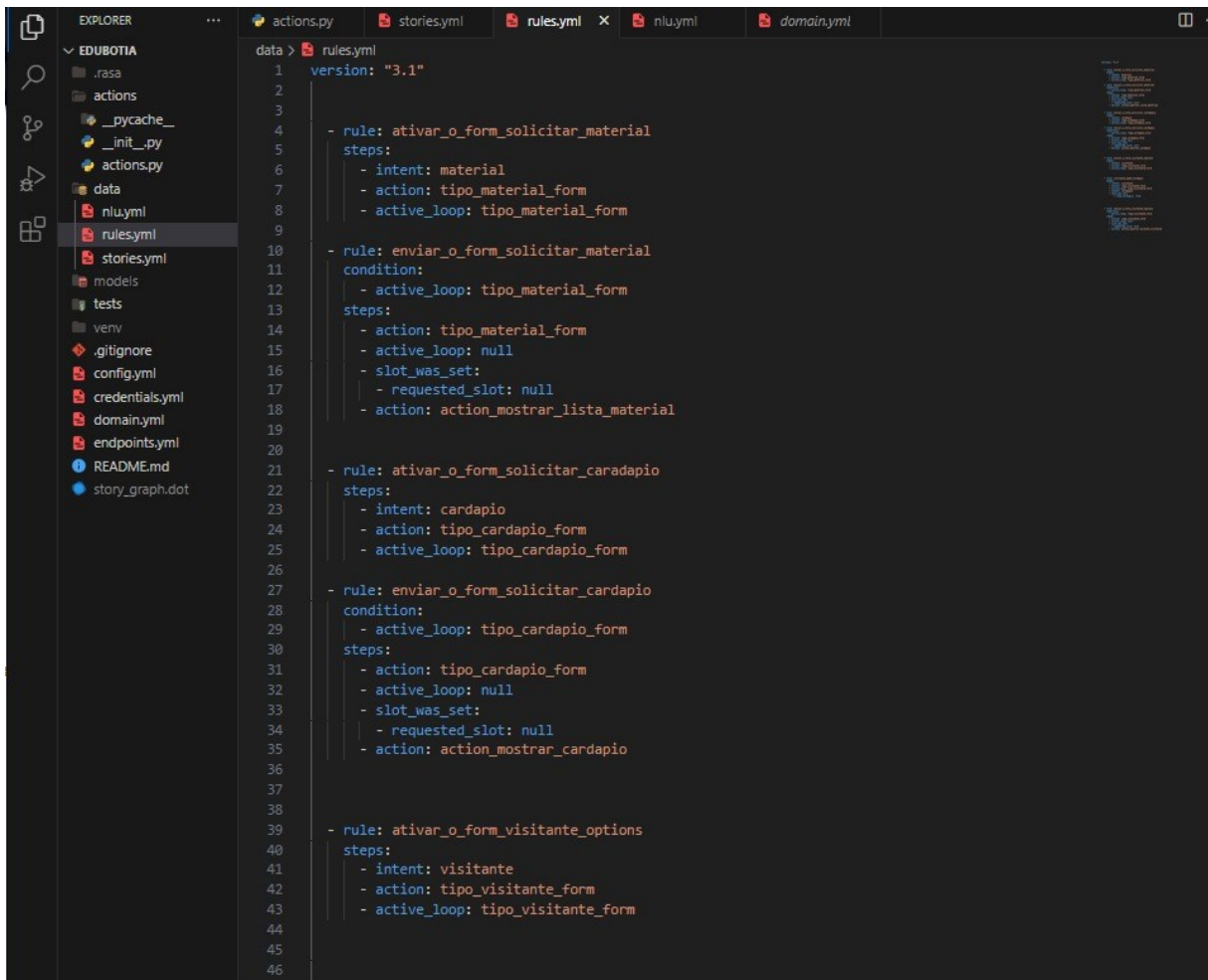
Fonte: Elaborado pelos autores (2024)

4.2.4.4 Rules (Regras)

O arquivo de regras (`rules.yml`) define regras específicas de conversação que o *chatbot* deve seguir para responder a certos gatilhos ou eventos. As regras são utilizadas para especificar respostas do *chatbot* em situações pré-definidas. Elas permitem definir comportamentos determinísticos e ações específicas que o *chatbot* deve executar quando certos eventos ocorrem ou certas condições são atendidas. As regras são fundamentais para garantir respostas consistentes e previsíveis em situações específicas.

A figura 20 corresponde a um exemplo de configuração do arquivo `rules.yml`, ilustrando como definir regras de conversação para o *chatbot*.

Figura 20 - Tela das Regras



```

1  version: "3.1"
2
3
4  - rule: ativar_o_form_solicitar_material
5    steps:
6      - intent: material
7      - action: tipo_material_form
8      - active_loop: tipo_material_form
9
10 - rule: enviar_o_form_solicitar_material
11   condition:
12     - active_loop: tipo_material_form
13   steps:
14     - action: tipo_material_form
15     - active_loop: null
16     - slot_was_set:
17       - requested_slot: null
18     - action: action_mostrar_lista_material
19
20
21 - rule: ativar_o_form_solicitar_cardapio
22   steps:
23     - intent: cardapio
24     - action: tipo_cardapio_form
25     - active_loop: tipo_cardapio_form
26
27 - rule: enviar_o_form_solicitar_cardapio
28   condition:
29     - active_loop: tipo_cardapio_form
30   steps:
31     - action: tipo_cardapio_form
32     - active_loop: null
33     - slot_was_set:
34       - requested_slot: null
35     - action: action_mostrar_cardapio
36
37
38
39 - rule: ativar_o_form_visitante_options
40   steps:
41     - intent: visitante
42     - action: tipo_visitante_form
43     - active_loop: tipo_visitante_form
44
45
46

```

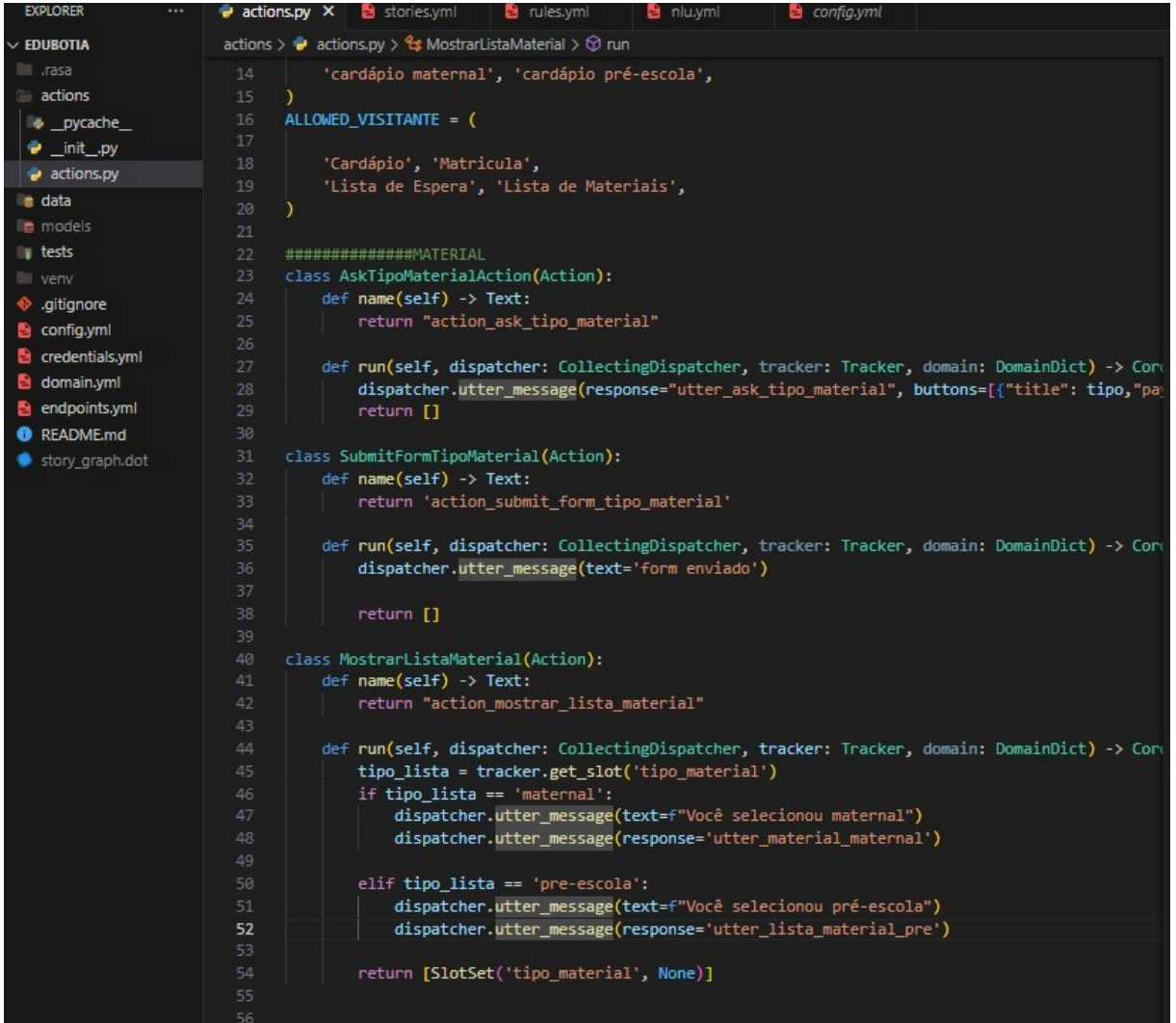
Fonte: Elaborado pelos autores (2024)

4.2.4.5 Actions (Ações)

O arquivo `actions.py` permite a criação de ações personalizadas, como botões interativos e envio de formulários, que não podem ser configurados diretamente no arquivo de domínio. Essas ações são fundamentais para melhorar a interatividade e a funcionalidade do *chatbot*, permitindo uma resposta mais dinâmica e personalizada às entradas dos usuários.

Abaixo na figura 20 está um exemplo de código no arquivo actions.py que cria botões e gerencia o envio de formulários.

Figura 21 - Tela das Ações



```

14     'cardápio maternal', 'cardápio pré-escola',
15 )
16 ALLOWED_VISITANTE = (
17     'Cardápio', 'Matricula',
18     'Lista de Espera', 'Lista de Materiais',
19 )
20 )
21
22 #####MATERIAL
23 class AskTipoMaterialAction(Action):
24     def name(self) -> Text:
25         return "action_ask_tipo_material"
26
27     def run(self, dispatcher: CollectingDispatcher, tracker: Tracker, domain: DomainDict) -> Cor:
28         dispatcher.utter_message(response="utter_ask_tipo_material", buttons=[{"title": "tipo", "pa
29         return []
30
31 class SubmitFormTipoMaterial(Action):
32     def name(self) -> Text:
33         return 'action_submit_form_tipo_material'
34
35     def run(self, dispatcher: CollectingDispatcher, tracker: Tracker, domain: DomainDict) -> Cor:
36         dispatcher.utter_message(text='form enviado')
37
38         return []
39
40 class MostrarListaMaterial(Action):
41     def name(self) -> Text:
42         return "action_mostrar_lista_material"
43
44     def run(self, dispatcher: CollectingDispatcher, tracker: Tracker, domain: DomainDict) -> Cor:
45         tipo_lista = tracker.get_slot('tipo_material')
46         if tipo_lista == 'maternal':
47             dispatcher.utter_message(text=f"Você selecionou maternal")
48             dispatcher.utter_message(response='utter_material_maternal')
49
50         elif tipo_lista == 'pre-escola':
51             dispatcher.utter_message(text=f"Você selecionou pré-escola")
52             dispatcher.utter_message(response='utter_lista_material_pre')
53
54         return [SlotSet('tipo_material', None)]
55
56

```

Fonte: Elaborado pelos autores (2024)

4.2.5 Arquitetura em Microsserviço

Conforme a Figura 1, a arquitetura do EduBot é baseada em microsserviços, proporcionando uma estrutura modular e escalável. Esta arquitetura é composta por quatro componentes principais:

React: Este é o componente responsável pelo *front-end* da aplicação. Ele gerencia a interface do usuário, garantindo uma experiência interativa e responsiva.

O React lida com as requisições ao *back-end* e também possui a funcionalidade de gerar relatórios. A escolha do React permite uma rápida atualização da interface e uma fácil manutenção do código.

BFF (Back-end for Front-end): Este componente atua como intermediário entre o front-end e os serviços de *back-end*. Ele é responsável por receber todas as requisições provenientes do React e direcioná-las para o serviço apropriado, seja o RASA ou o banco de dados. O BFF é construído seguindo a *Clean Architecture*, que promove a separação de responsabilidades e facilita a manutenção do código, garantindo que cada camada tenha uma função bem definida.

MongoDB: O MongoDB serve como o banco de dados da aplicação. É um serviço crucial no ciclo de microsserviços, armazenando todas as informações necessárias para o funcionamento do EduBot. O uso do MongoDB, um banco de dados NoSQL, oferece flexibilidade e desempenho, permitindo que a aplicação gerencie grandes volumes de dados com eficiência.

RASA: Este é o componente responsável pelo serviço do bot. Desenvolvido com o *framework* Rasa em Python, ele gerencia a lógica de conversação e interação com os usuários. O Rasa permite a construção de *chatbots* sofisticados e personalizados, utilizando técnicas avançadas de processamento de linguagem natural (NLP). Ele é essencial para a comunicação eficaz entre o usuário e o EduBot, interpretando comandos e fornecendo respostas apropriadas.

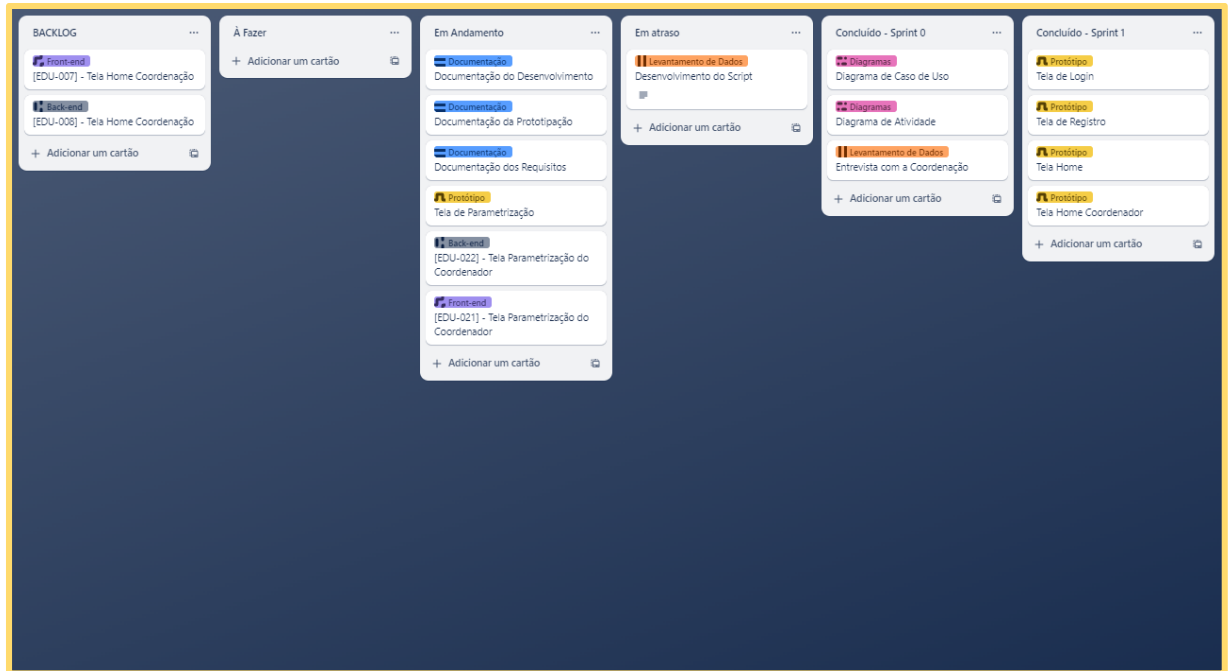
A arquitetura de microsserviços do EduBot promove uma separação clara de responsabilidades, facilitando a manutenção e a escalabilidade da aplicação. Cada componente pode ser desenvolvido, implantado e escalado independentemente, o que melhora a eficiência do desenvolvimento e a flexibilidade para futuras atualizações.

4.3 AMBIENTE DE DESENVOLVIMENTO

Durante o desenvolvimento do software, utilizou-se a metodologia ágil para gerenciar o processo, especificamente através dos frameworks Scrum e Kanban. Isso permitiu criar uma lista de pequenas entregas, utilizando a ferramenta web Trello para gerenciamento. O Trello é utilizado para administrar o Kanban e suas respectivas entregas. Abaixo estão as ilustrações do Trello do desenvolvimento do sistema

EduBot, onde as tarefas foram organizadas em cards e tarefas, entregas de objetivos específicos, Sprints, Backlog, Em Desenvolvimento, A Fazer, Em Atraso e Concluídos, conforme mostrado nas figuras abaixo:

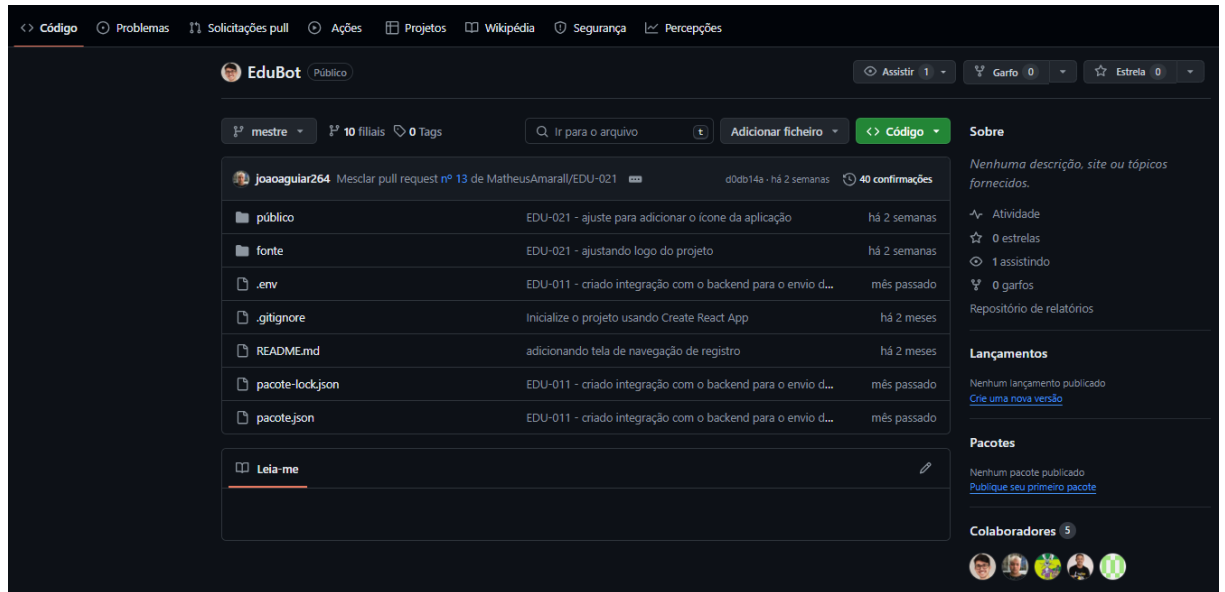
Figura 22 - Trello



Fonte: Elaborado pelos autores (2024)

Para o controle de versão, utilizou-se a ferramenta GitHub, que permitiu acompanhar e gerenciar todas as versões e repositórios durante o desenvolvimento de maneira automatizada e simplificada, conforme ilustrado na Figura 23.

Figura 23 - GitHub



Fonte: Elaborado pelos autores (2024)

4.4 FUNCIONAMENTO DO SOFTWARE 1

Este tópico apresenta um passo a passo de como utilizar o sistema, acompanhado de imagens das telas para facilitar o entendimento. O guia detalha as principais funcionalidades e operações que podem ser realizadas pelos usuários, como consultar listas de material escolar e merenda, visualizar datas de matrícula e listas de espera, e interagir diretamente com o coordenador escolar. Além disso, o coordenador poderá parametrizar o chatbot, intervir em conversas com os pais dos alunos, e gerar relatórios detalhados sobre as interações. Cada etapa é explicada de maneira clara e objetiva, com ilustrações que mostram exatamente como navegar e utilizar cada recurso do sistema.

4.4.1 Passo a Passo Usuário Regular

A seguir, serão apresentadas ilustrações detalhadas que guiarão o usuário comum através de cada etapa do processo de criação de uma conta, chegando no acesso ao sistema. Este passo a passo foi cuidadosamente elaborado para garantir

uma experiência tranquila e intuitiva, facilitando o entendimento de cada ação necessária para completar o registro e iniciar o uso dos serviços oferecidos.

4.4.1.1 Passo 1

Os usuários terão a opção de criar uma conta, na qual deverão inserir sua função, indicando se são membros da coordenação ou não, além de fornecerem seu e-mail, número de matrícula e uma senha.

Figura 24 - Tela de Registro

Crie sua conta

Membro Da Coordenação

Email *

Matricula *

Crie uma senha *

Confirme sua senha *

CRIAR CONTA

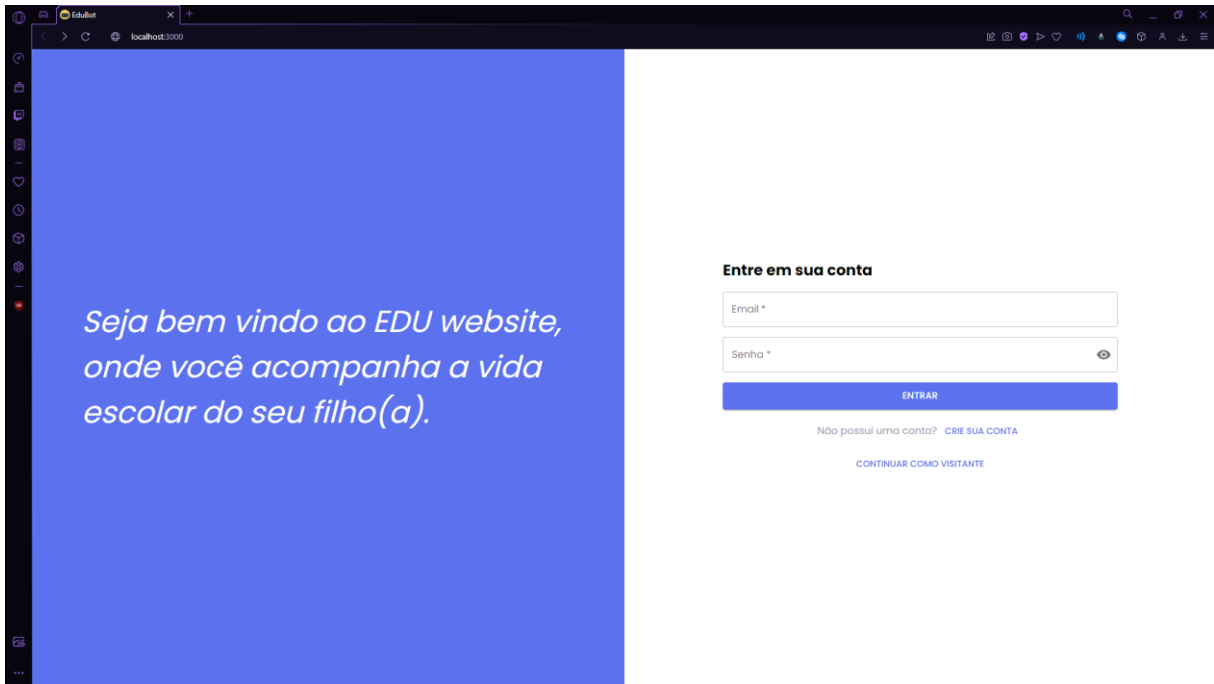
Já possui uma conta? [ENTRAR](#)

Fonte: Elaborado pelos autores (2024)

4.4.1.2 Passo 2

Após criar sua conta, o usuário deve acessá-la através da página de login, onde deverá inserir seu e-mail e senha cadastrados.

Figura 25 - Tela de Login

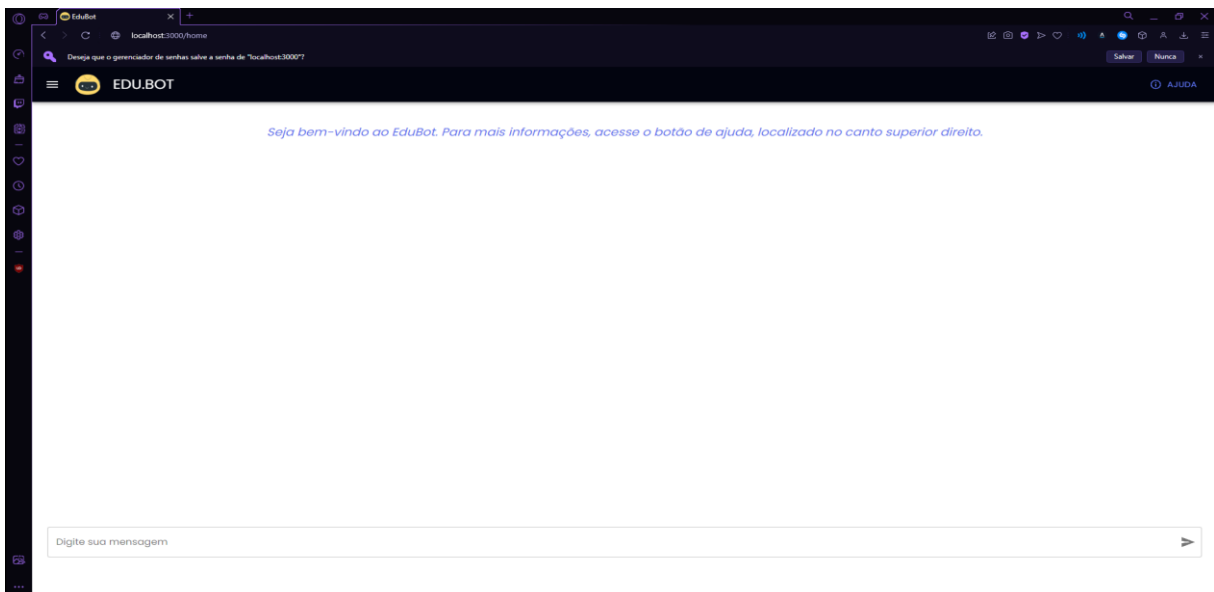


Fonte: Elaborado pelos autores (2024)

4.4.1.3 Passo 3

Após entrar em sua conta, o usuário será direcionado para a tela principal da aplicação, onde terá acesso à conversa com o chatbot.

Figura 26 - Tela Home

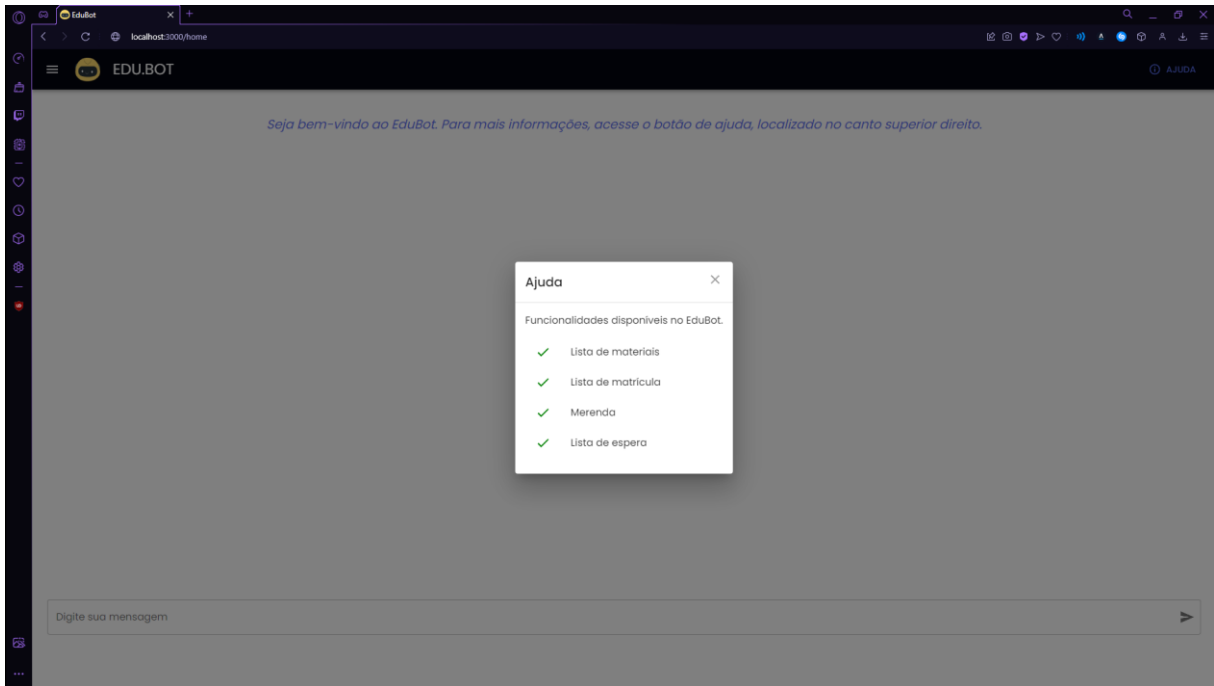


Fonte: Elaborado pelos autores (2024)

4.4.1.4 Passo 4

Ao clicar na opção 'Ajuda', localizada no canto superior direito da tela, o usuário terá acesso a uma lista de possíveis dúvidas que o chatbot pode auxiliar a resolver.

Figura 27 - Tela de Ajuda

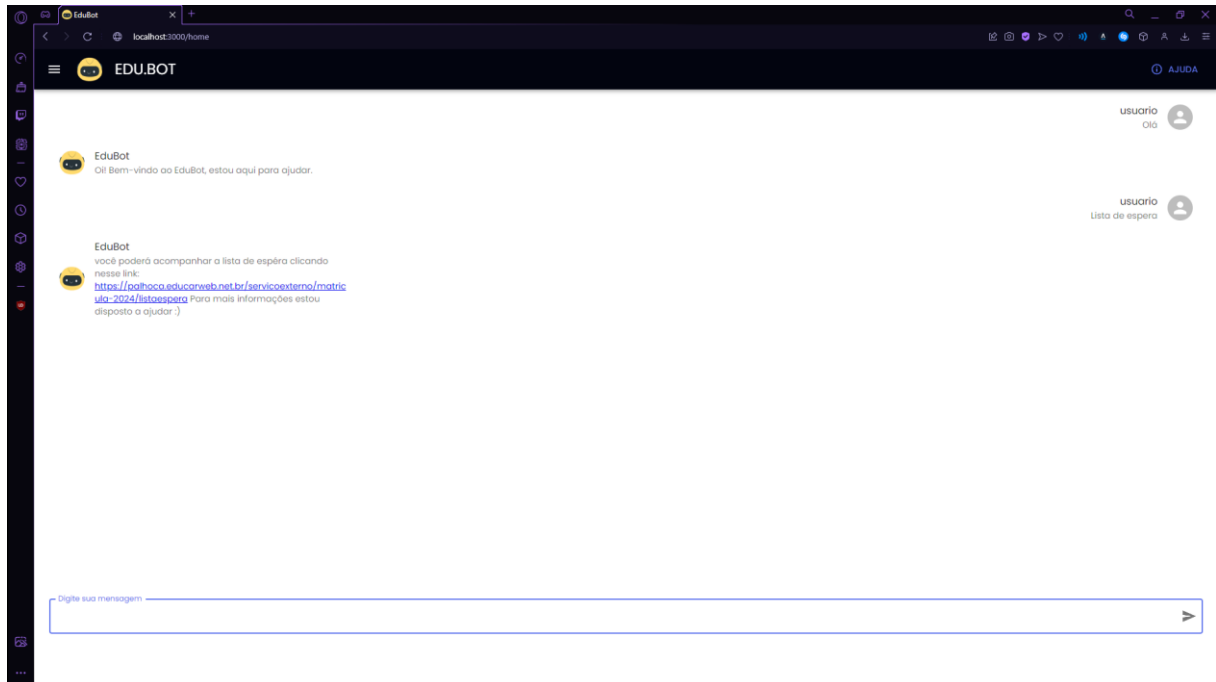


Fonte: Elaborado pelos autores (2024)

4.4.1.5 Passo 5

Após o usuário identificar sua dúvida, ele iniciará a interação com o bot, digitando sua pergunta desejada no campo abaixo de "Digite sua mensagem". Após alguns instantes, o bot responderá de acordo.

Figura 28 - Tela Home (Conversando)



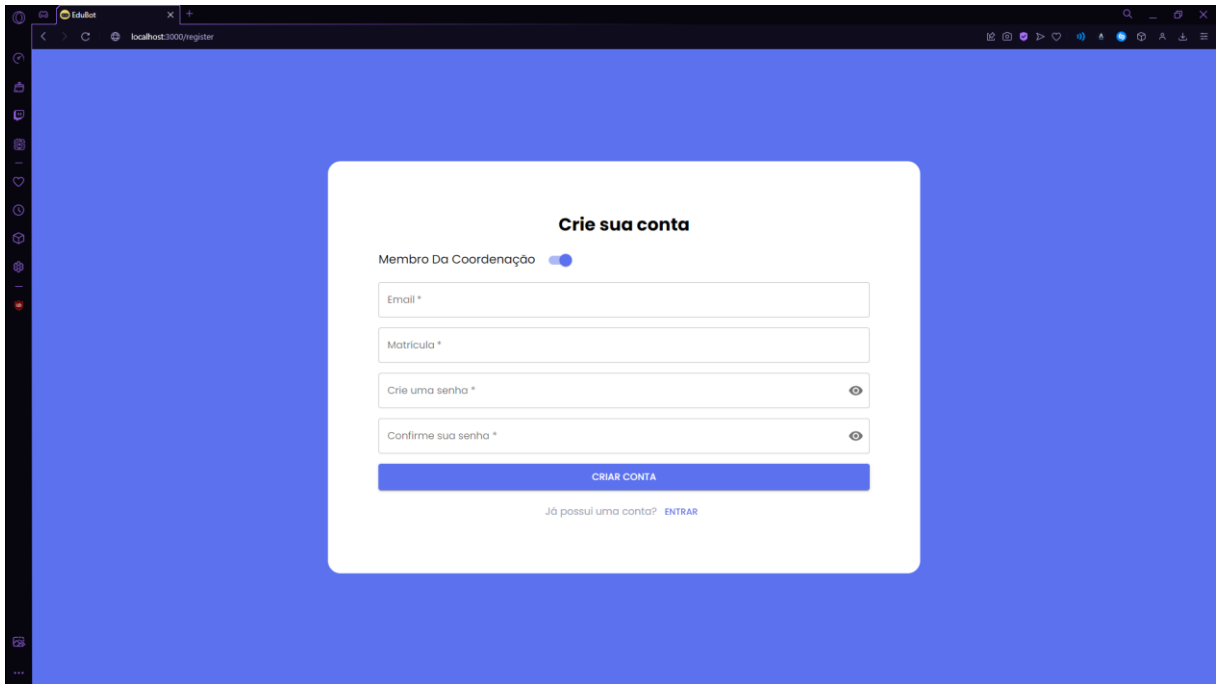
Fonte: Elaborado pelos autores (2024)

4.4.2 Passo a Passo Usuário Coordenador

4.4.2.1 Passo 1

O coordenador deve selecionar o botão 'Membro da Coordenação' para criar acesso como coordenador, inserindo sua função, e-mail, número de matrícula e senha.

Figura 29 - Tela de Registro



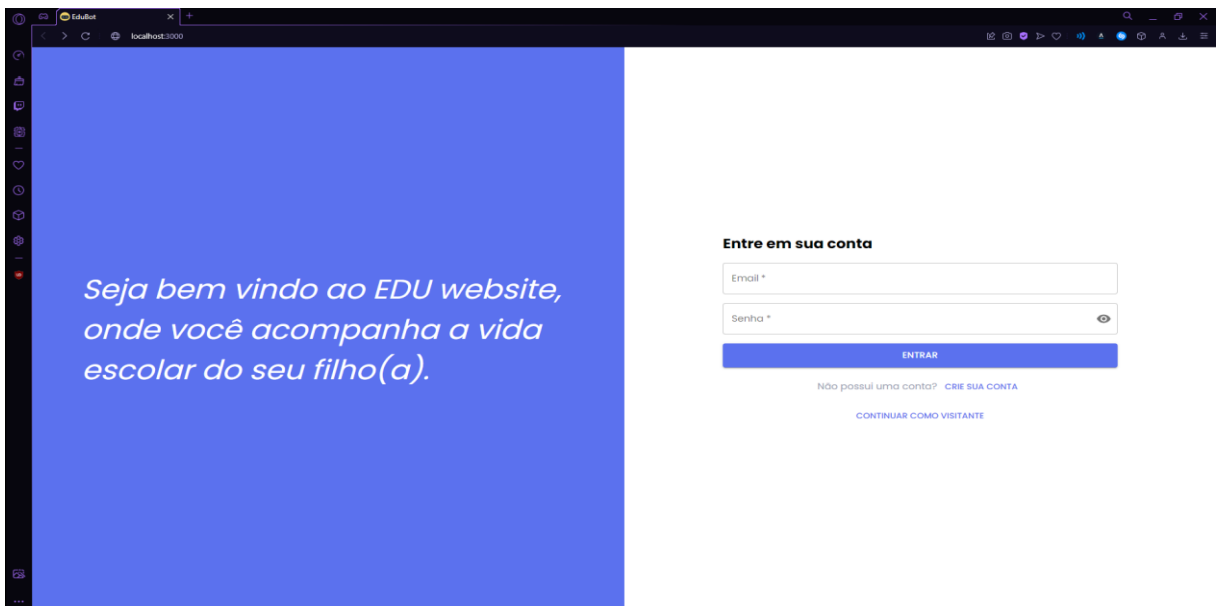
The screenshot shows a web browser window with the URL `localhost:3000/register`. The page has a blue background and a white registration form titled "Crie sua conta". The form includes a toggle switch for "Membro Da Coordenação" (checked), and four input fields: "Email *", "Matricula *", "Crie uma senha *", and "Confirme sua senha *". Below the fields is a blue "CRIAR CONTA" button and a link "Já possui uma conta? ENTRAR".

Fonte: Elaborado pelos autores (2024)

4.4.2.2 Passo 2

Após criar sua conta, o coordenador deve acessá-la através da página de login, inserindo seu email e senha cadastrados.

Figura 30 - Tela de Login



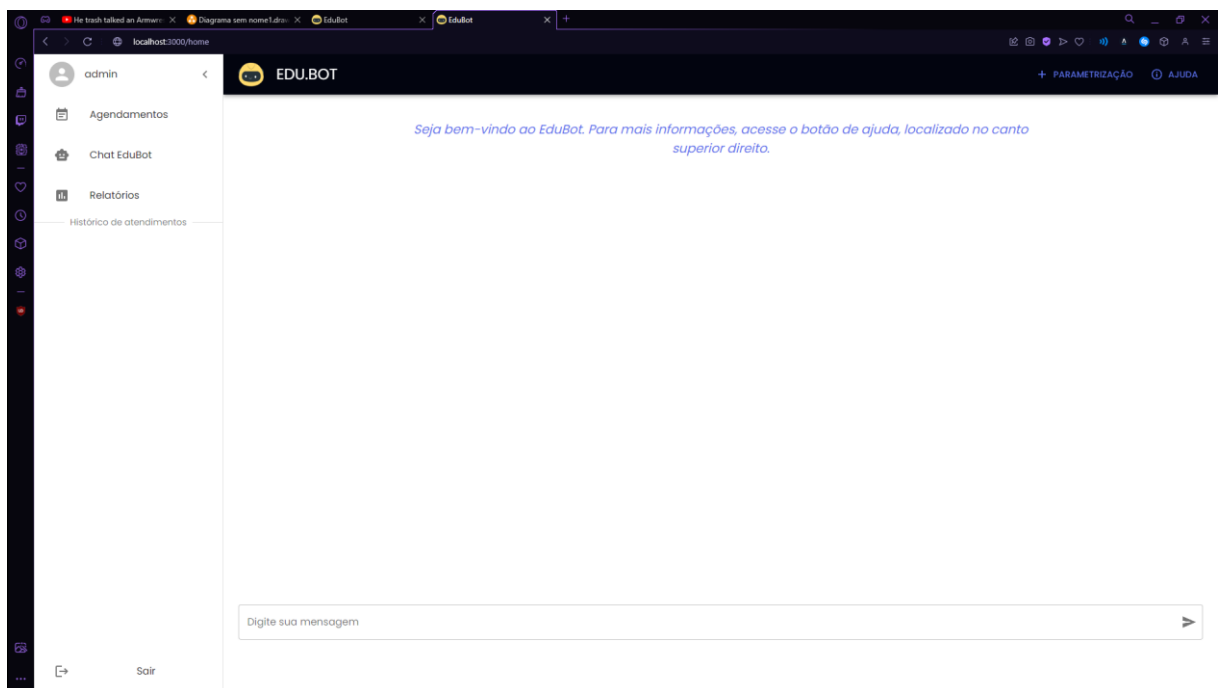
The screenshot shows a web browser window with the URL `localhost:3000`. The page is split into two sections. On the left, a blue vertical banner contains the text: "Seja bem vindo ao EDU website, onde você acompanha a vida escolar do seu filho(a)". On the right, there is a white login form titled "Entre em sua conta". It has two input fields: "Email *" and "Senha *". Below the fields is a blue "ENTRAR" button. At the bottom of the form, there are two links: "Não possui uma conta? CRIE SUA CONTA" and "CONTINUAR COMO VISITANTE".

Fonte: Elaborado pelos autores (2024)

4.4.2.3 Passo 3

Após entrar em sua conta, o coordenador será direcionado para a tela principal da aplicação, onde terá acesso à conversa com o chatbot, além das opções específicas para coordenadores, tais como agendamentos, relatórios e parametrização.

Figura 31 - Tela Home Coordenador

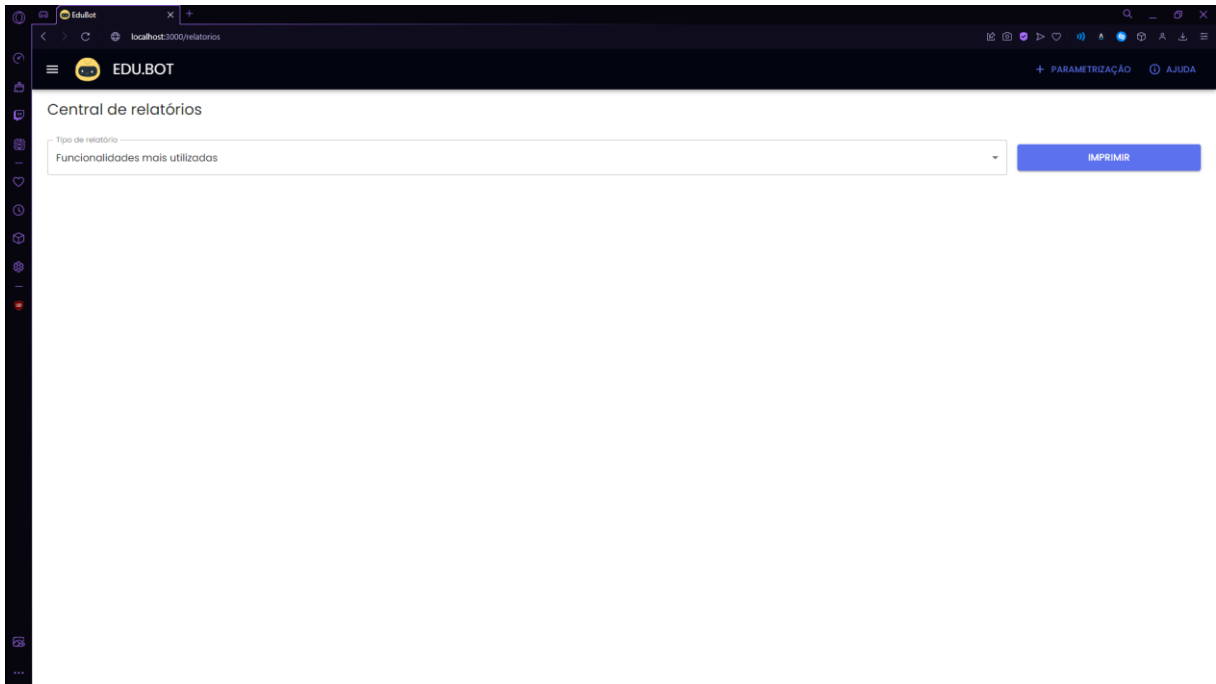


Fonte: Elaborado pelos autores (2024)

4.4.2.4 Passo 4

Ao selecionar o botão "Relatórios", localizado no lado esquerdo da tela, abaixo da opção "Chat EduBot", o coordenador será direcionado à tela de relatórios, onde poderá gerar métricas referentes aos atendimentos do chatbot.

Figura 32 - Tela de Relatórios

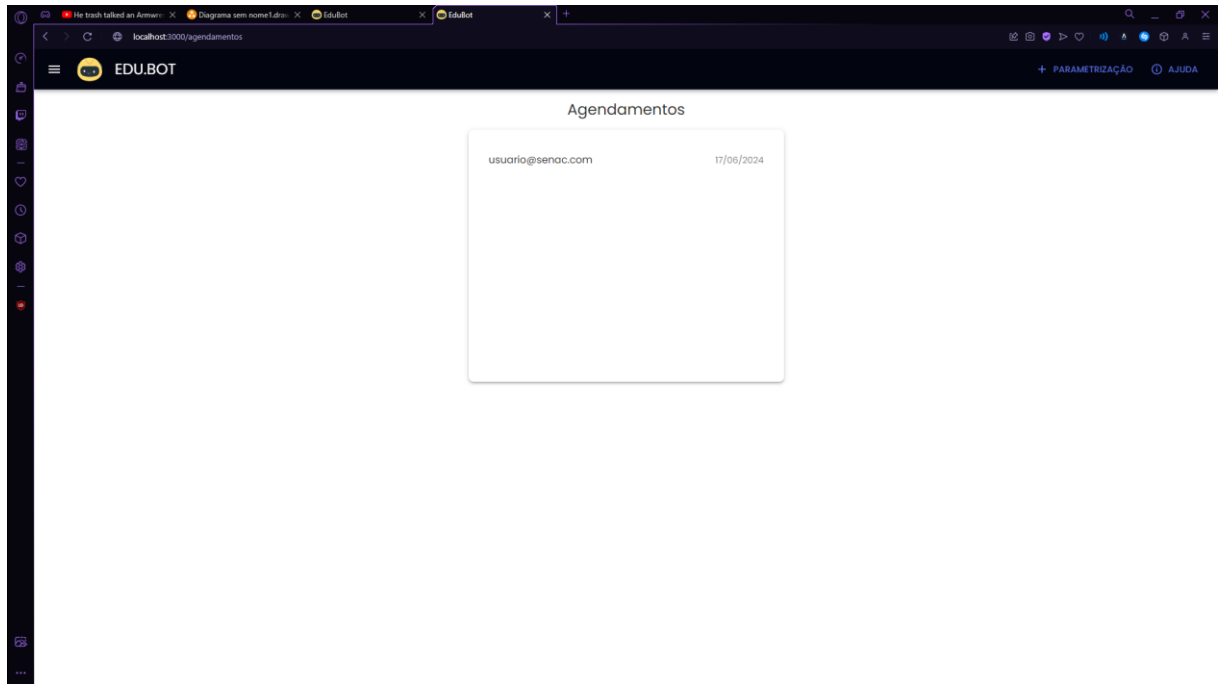


Fonte: Elaborado pelos autores (2024)

4.4.2.5 Passo 5

Ao selecionar o botão "Agendamentos", localizado no lado esquerdo da tela, acima da opção "Chat EduBot", o coordenador será direcionado à tela de agendamentos, onde poderá visualizar todas as reuniões com pais de alunos agendadas.

Figura 33 - Tela de Agendamentos



Fonte: Elaborado pelos autores (2024)

4.4.2.6 Passo 6

Ao selecionar o botão "Parametrização", localizado no canto superior direito da tela, ao lado da opção "Ajuda", o coordenador será direcionado à primeira tela de parametrização. Lá, ele poderá adicionar novas parametrizações para o bot, dando um título para ela na caixa de texto no topo da tela e adicionando as possíveis variações da pergunta na caixa de texto mais abaixo. Após adicionar o título e as perguntas, o coordenador deve clicar no botão "NEXT" localizado no canto direito da tela.

Figura 34 - Tela de Parametrização 1

1 Perguntas

Titulo das perguntas *

Bem-vindo à etapa de parametrização do seu chatbot!
Para começar, Você irá cadastrar o título da sua nova pergunta e as possíveis abordagens que os seus usuários terão para fazer a pergunta. Por exemplo:
Como faço para verificar minha matrícula?
Qual é o procedimento para descobrir minha matrícula?
Pode me informar qual é a minha matrícula?

Adicione novas perguntas aqui

BACK

NEXT

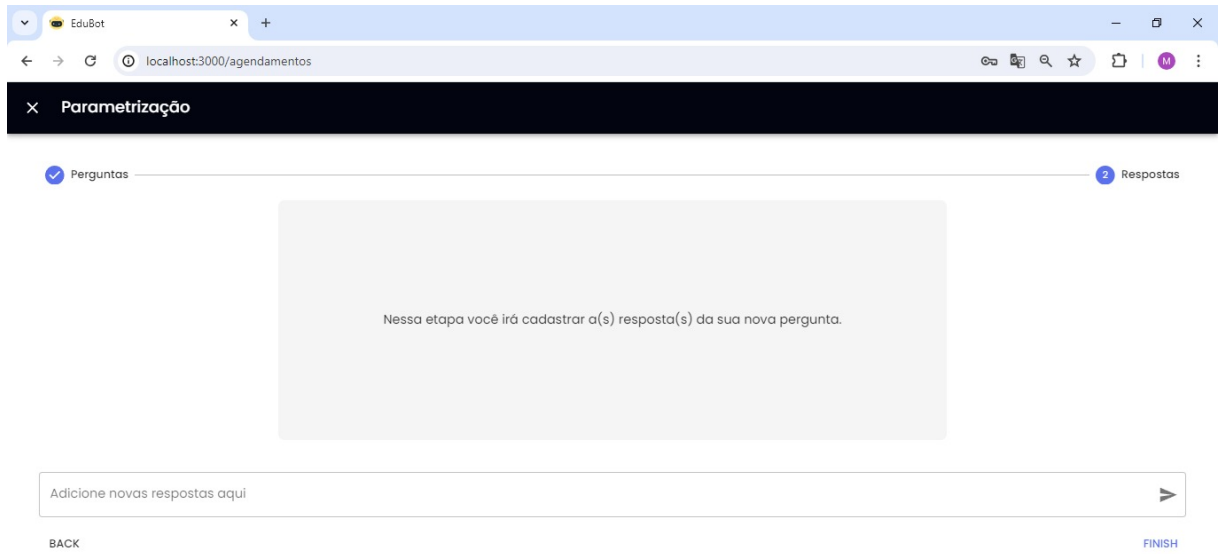
2 Respostas

Fonte: Elaborado pelos autores (2024)

4.4.2.7 Passo 7

Após selecionar "NEXT", o coordenador será levado para a segunda etapa da parametrização, que é a criação da resposta. Nessa tela, o coordenador poderá criar uma resposta para a pergunta que foi criada anteriormente. Após adicionar a resposta desejada, o coordenador deve clicar no botão "FINISH", localizado no canto direito da tela, para finalizar a parametrização e carregá-la no bot.

Figura 35 - Tela de Parametrização 2



Fonte: Elaborado pelos autores (2024)

5 CONCLUSÃO

As conclusões deste trabalho visam responder às questões de pesquisa, reforçando as descobertas e conclusões obtidas ao longo do desenvolvimento do projeto. Partindo da necessidade identificada em uma entrevista com a diretora de uma escola infantil, observou-se que a repetição constante das mesmas informações sobrecarregava a coordenação. Em resposta, foi criado um escopo focado nas informações frequentemente repetidas e adicionando outras questões que poderiam ser levantadas pelos pais.

Foi desenvolvido um *chatbot*, o EduBot, acessível por qualquer dispositivo com conexão à internet e navegador, possibilitando que tanto visitantes quanto pais obtenham informações relevantes. Para os pais, o EduBot fornece informações mais específicas e permite a comunicação direta com o coordenador da escola. Além disso, o coordenador pode personalizar diálogos para responder automaticamente às perguntas mais comuns, otimizando ainda mais o atendimento.

Nosso projeto automatiza atendimentos repetitivos, liberando tempo para que a coordenação possa se dedicar a tarefas mais complexas e importantes. Com isso, o objetivo deste projeto de reduzir a carga de trabalho da coordenação e melhorar a eficiência do atendimento escolar foi alcançado.

Avaliação dos Objetivos

Os objetivos estabelecidos no início do projeto foram alcançados de maneira satisfatória. Identificamos a necessidade específica da escola, desenvolvemos uma solução tecnológica apropriada e implementamos o EduBot com sucesso. A coordenação da escola agora dispõe de mais tempo para se concentrar em atividades mais críticas e personalizadas, enquanto o chatbot lida com questões repetitivas de forma eficaz.

Recomendações e Sugestões para Trabalhos Futuros

Para futuros trabalhos relacionados a esta pesquisa, recomenda-se:

Expansão de Funcionalidades: Adicionar novas funcionalidades ao EduBot, como integração com sistemas de gestão escolar para fornecer informações ainda mais detalhadas e personalizadas.

Análise de Dados: Implementar análises de dados das interações do chatbot para identificar novas áreas de melhoria e adaptar as respostas de acordo com as necessidades dos usuários.

Estudo de Usabilidade: Conduzir estudos de usabilidade com pais e coordenadores para aprimorar a interface e a experiência do usuário.

Integração com Redes Sociais: Expandir o alcance do EduBot integrando-o com plataformas de redes sociais, facilitando ainda mais o acesso à informação para os pais.

Este trabalho não só proporciona uma solução prática para a comunicação entre pais e escola infantil específica, mas também serve como base para o desenvolvimento de chatbots em outras instituições educacionais, contribuindo para a evolução do atendimento escolar automatizado. Ao seguir as normas da ABNT descritas no "Manual para Formatação de Trabalhos Acadêmicos" do Senac, garantimos a conformidade e a qualidade na apresentação do nosso trabalho.

REFERÊNCIAS

- AGUIAR, Janderson Jason Barbosa. **Inteligência Artificial e tecnologias digitais na educação: oportunidades e desafios**. Open Minds International Journal, v. 4, n. 2, p. 183-188, 2023. Disponível em: https://www.researchgate.net/publication/373371969_INTELIGENCIA_ARTIFICIAL_E_TECNOLOGIAS_DIGITAIS_NA_EDUCACAO_OPORTUNIDADES_E_DESAFIOS. Acesso em: 09 abril. 2024.
- BEZERRA, Eduardo. **Princípios de Análise e Projeto de Sistemas com UML**. Rio de Janeiro: Elsevier, 2007. Disponível em: https://d1wqtxts1xzle7.cloudfront.net/36964560/AMS_Cap5_Diagrama_Classe-libre.pdf?1426271904=&response-content-disposition=inline%3B+filename%3DModelagem_de_Classes_de_Analise.pdf&Expires=1718660785&Signature=Npl2LX8bt6ize2587Bxhngkdz6DNO51Cy4aC24ZqHwwlyvxmTaRLLLe2Wil7Jfworb9SBHgcA5B~VDb8AROU~tajgcWgl1Isdsjop182MJ2usIODoCkorfiXVEbQ6ri7pa1Ni6qQ4NxFCH8~zgRwwNeUEezEdimVIHmO-Xuq6UIx0R846-pQr8-xMr04-aZiT2RCTxpc59KkXv8s~wwwBgSnDS5sHS3n~sAzP2rAkL3cyXhBaoBVzDzWctlnCgKjkZlZJ8fDOPSYpXodWU2SnLRi9AJeKzaHohTJ~Ygyr6e719ePgejGRegezK0QrQtllkJuNBaGisOJeT59l3CRg__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA. Acesso em 15 Junho. 2024.
- BOOCH, Grady. **UML: Guia do usuário**. Rio de Janeiro: Elsevier, Brasil, 2006. Disponível em: https://books.google.com.br/books?hl=pt-BR&lr=lang_pt&id=ddWqxcDKGF8C&oi=fnd&pg=PR13&dq=linguagens+gr%C3%A1ficas+de+modelagem+uml&ots=fgwLje8LLH&sig=dWyp-mX9o8SxgTTIMNFx3FJB67c#v=onepage&q&f=false. Acesso em: 01 maio. 2024.
- BUENO, C. E. d. O. **Desenvolvimento de um aplicativo utilizando o framework flutter e arquitetura limpa**. 2021. Monografia. (Graduação em Ciência da Computação) - Pontifícia Universidade Católica de Goiás, Goiás, 2021. Disponível em: <https://repositorio.pucgoias.edu.br/jspui/handle/123456789/1861>. Acesso: 01 junho. 2024.
- COMARELLA, Rafaela Lunardi; CAFÉ, Lúgia Maria Arruda. **Chatterbot: conceito, características, tipologia e construção**. Informacao & sociedade, v. 18, n. 2, p. 55-67, 2008. Disponível em <https://cip.brapci.inf.br/autoridade/v/92052>. Acesso em: 10 abril. 2024.
- CORREA, Joeckson; VIANA, Davi; TELES, Ariel. **Desenvolvendo chatbots com o dialogflow**. Sociedade Brasileira de Computação, 2021. Disponível em: <https://sol.sbc.org.br/livros/index.php/sbc/catalog/download/76/321/577-1>. Acesso em: 16 abril. 2024.
- EL SAIFI, Mohamad Maamoun. **PMPI: uma implementação MPI multi-plataforma, multi-linguagem**. 2006. Dissertação (Mestrado em Engenharia Elétrica) - Universidade de São Paulo, São Paulo, 2006. Disponível em:

<https://www.teses.usp.br/teses/disponiveis/3/3141/tde-08122006-154811/en.php>
Acesso em: 05 Junho. 2024.

FOWLER, Martin. **Microservices** - a definition of this new architectural term. 25 mar. 2014. Disponível em: <https://martinfowler.com/articles/microservices.html>. Acesso em: 31 maio. 2024.

GRILLO, Filipe Del Nero; FORTES, Renata Pontin de Mattos. **Aprendendo JavaScript**. 2008. Disponível em: <https://repositorio.usp.br/bitstreams/4cd7f9b7-7144-40f4-bfd0-7a1d9a6bd748>. Acesso em: 16 abril. 2024.

GUEDES, Gilleanes TA. **UML 2**. Uma Abordagem Prática. São Paulo, Novatec, 2009. Disponível em: <https://s3.novatec.com.br/capitulos/capitulo-9788575222812.pdf>. Acesso em: 03 junho. 2024.

JÚNIOR, João Fernando Costa et al. **A inteligência artificial como ferramenta de apoio no ensino superior**. Revena-Revista Brasileira de Ensino e Aprendizagem, v. 6, p. 246-269, 2023. Disponível em: <https://revena.emnuvens.com.br/revista/article/view/111/101>
Acesso em: 15 junho. 2024.

MARTIN, Matheus Nunes. **Nobullying api**: uma API REST que auxilie a prevenção e combate ao bullying e assédio no âmbito acadêmico do IFPB-campus Cajazeiras. 2019. Disponível em: <https://repositorio.ufrn.br/bitstream/123456789/56033/1/tcc-gradua%c3%a7%c3%a3o-final.pdf>
Acesso em: 03 Junho. 2024.

MICROSOFT. **Tour of C#**. 2023. Disponível em: <https://learn.microsoft.com/pt-br/dotnet/csharp/tour-of-csharp/>. Acesso em: 15 abril. 2024.

MOREIRA, Diorginis Ormond; MIGNONI, Maria Eloisa. **Inteligência artificial**: o uso de chatbots no atendimento ao cliente. Revista Ibero-Americana de Ciências Ambientais, v. 12, n. 12, p. 157-165, 2021. Disponível em: <https://www.sustenere.inf.br/index.php/rica/article/view/6313/3420>
Acesso em 15 junho. 2024.

PASCHOAL, Leo Natan; CONTE, Tayana Uchôa; DE SOUZA, Simone RS. **O que revelam os estudos secundários sobre chatbots na educação?**. In: Anais do XXXIII Simpósio Brasileiro de Informática na Educação. SBC, 2022. p. 1126-1137. Disponível em: <https://sol.sbc.org.br/index.php/sbie/article/view/22487>. Acesso: em 09 abril. 2024.

POLATIDIS, Nikolaos. Chatbot para admissões. arXiv preprint arXiv:1408.6762 , 2014. Disponível em: <<https://arxiv.org/ftp/arxiv/papers/1408/1408.6762.pdf>>. Acesso em: 01 jun. 2024.

PROJECT MANAGEMENT INSTITUTE. Um guia de conhecimento em gerenciamento de projetos. **Guia PMBOK. 6.** ed. Pensilvânia, Project Management Institute, 2017. Disponível em: <https://seer.uniacademia.edu.br/index.php/cesRevista/article/view/2026>. Acesso em: 15 abril. 2024.

RODRIGUES, Jadeilson Leonardo Trizotti; ALVES, Celeny Fernandes; OSSHIRO, Marcio. **Anetha-Desenvolvimento de Chatbot em Python.** 2019. Disponível em: https://www.academia.edu/41255426/Anetha_Desenvolvimento_de_Chatbot_em_Python. Acesso em: 15 abril. 2024.

ROSEMBERG, Carlos et al. **Prototipação de software e design participativo: uma experiência do atlântico.** *In: VIII Simpósio Brasileiro de Fatores Humanos em Sistemas Computacionais, VIII, 2008, Porto Alegre. Anais [...].* Porto Alegre, 2008. Disponível em: https://www.researchgate.net/profile/Carlos-Carvalho-41/publication/220737394_Prototipacao_de_software_e_design_participativo_uma_e_xperiencia_do_atlantico/links/5c984bc592851cf0ae95ecbc/Prototipacao-de-software-e-design-participativo-uma-experiencia-do-atlantico.pdf. Acesso em: 01 maio. 2024.

SENGUPTA, R.; LAKSHMAN, S. Conversational chatbots—let’s chat. Deloitte, 2017. Disponível em: <https://www2.deloitte.com/content/dam/Deloitte/in/Documents/strategy/in-strategy-innovation-conversational-chatbots-lets-chat-final-report-noexp.pdf>. Acesso em: 23 abril. 2024.

SHARP, John; JAGGER, Jon. **Microsoft Visual C#. Net.** McGraw-Hill, Inc., 2003. Disponível em: <https://archive.org/details/mictvisualc00shar/page/n111/mode/2up>. Acesso em 16 Junho. 2024.

SILVA, Maurício Samy. **Criando sites com HTML: sites de alta qualidade com HTML e CSS.** Novatec Editora, São Paulo. 2007. Disponível em: [https://books.google.com.br/books?hl=pt-BR&lr=lang_pt&id=BxKWAwAAQBAJ&oi=fnd&pg=PA10&dq=Construindo+sites+com+CSS+e+\(X\)HTML&ots=kliY6OCEjK&sig=miY7c6qRknT9a1D3fUYHyCXSJd0#v=onepage&q&f=false](https://books.google.com.br/books?hl=pt-BR&lr=lang_pt&id=BxKWAwAAQBAJ&oi=fnd&pg=PA10&dq=Construindo+sites+com+CSS+e+(X)HTML&ots=kliY6OCEjK&sig=miY7c6qRknT9a1D3fUYHyCXSJd0#v=onepage&q&f=false). Acesso em: 31 maio. 2024.

SOARES, Italo Ribeiro et al. **Conformidade da biblioteca material UI para react com a web accessibility content guidelines.** 2022. Disponível em: <https://repositorio.ufpe.br/handle/123456789/48081>. Acesso em: 16 abril. 2024.

SOUZA, Elaine Calasans; DE OLIVEIRA, Marcus Rogério. **Comparativo entre os bancos de dados mysql e mongodb: quando o mongodb é indicado para o desenvolvimento de uma aplicação.** Revista Interface Tecnológica, v. 16, n. 2, p. 38-48, 2019. Disponível em: <https://revista.fatectq.edu.br/interfacetecnologica/article/download/664/411>.

Acesso em: 16 abril. 2024.

STOIBER, Max. **Build your first Node.js microservice**. 11 jan. 2017. Disponível em: <https://mxstbr.blog/2017/01/your-first-node-microservice/>.

Acesso em: 31 maio. 2024.

WATMORE, Jason. **How to use the Facebook SDK in a React App**, 2020.

Disponível em: <https://jasonwatmore.com/post/2020/10/28/react-facebook-how-to-use-thefacebook-sdk-in-a-react-app>.

Acesso em: 2 abril. 2023.

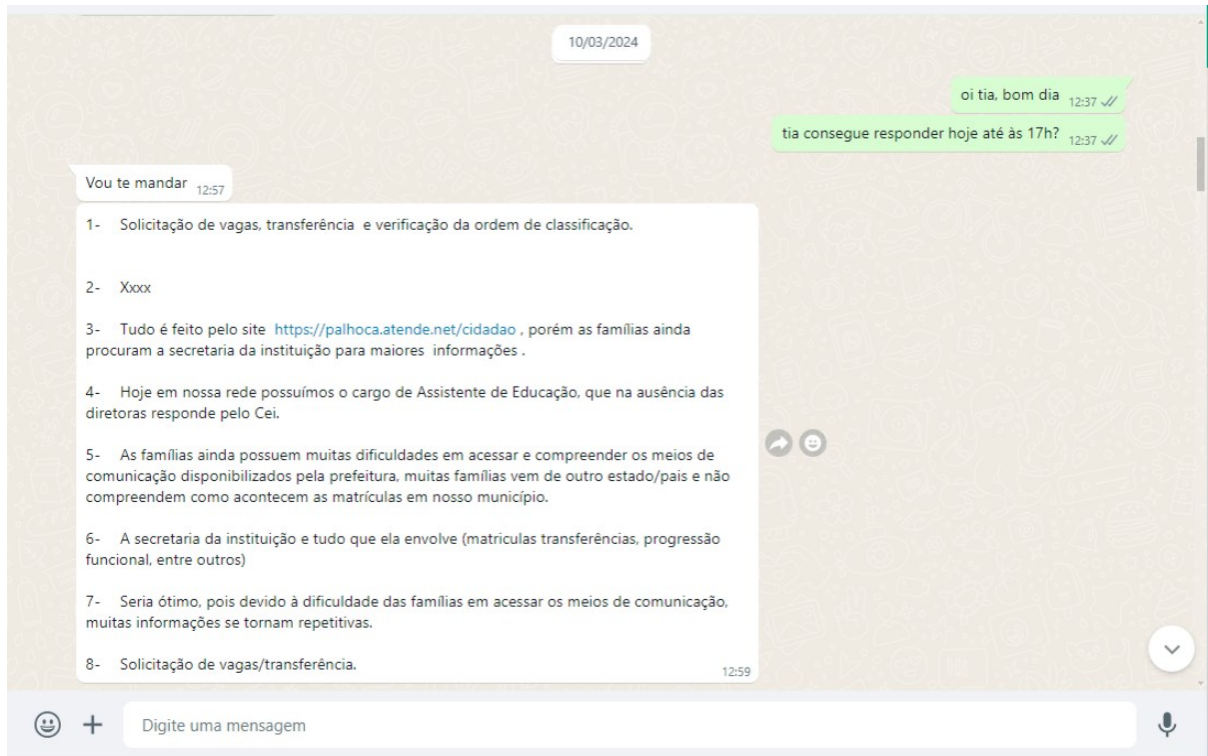
APÊNDICE A – QUESTIONÁRIO

Questionário realizado com a coordenação da instituição escolhida sobre as demandas recebidas – parte 1

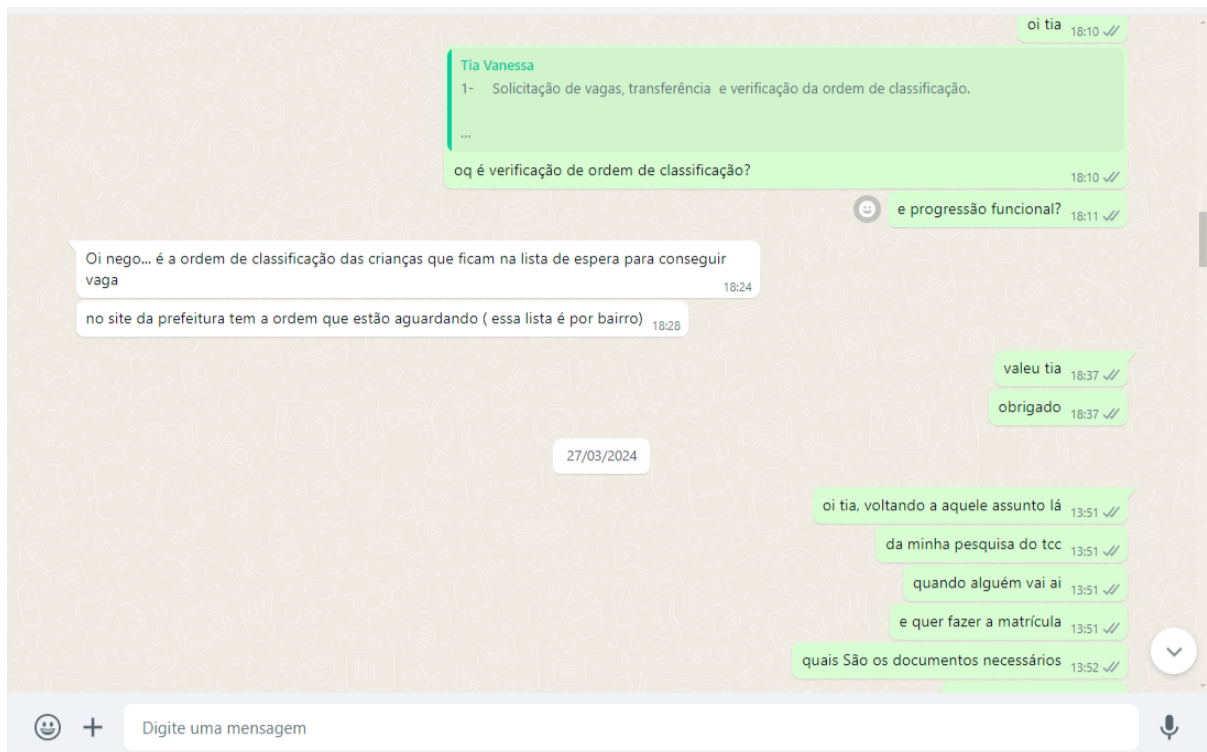
Conversa com entrevistado(a) parte 1



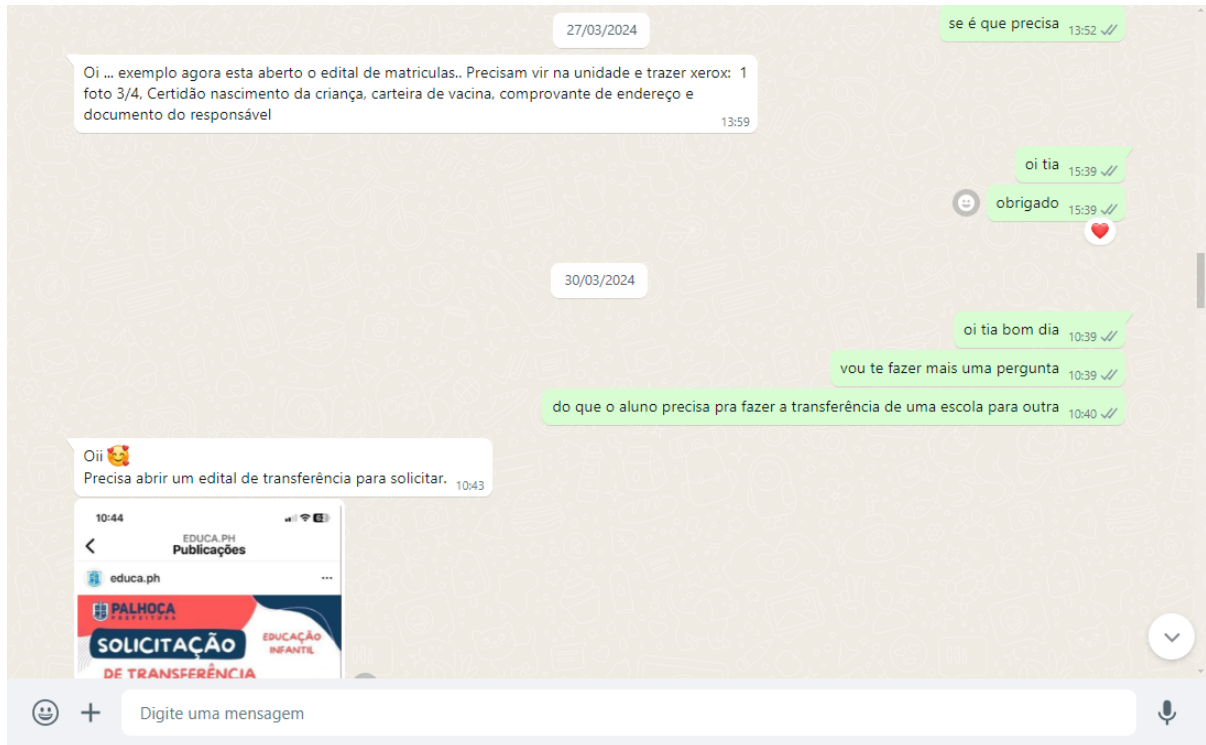
Conversa com entrevistado(a) parte 2



Conversa com entrevistado(a) parte 3



Conversa com entrevistado(a) parte 4



Conversa com entrevistado(a) parte 5



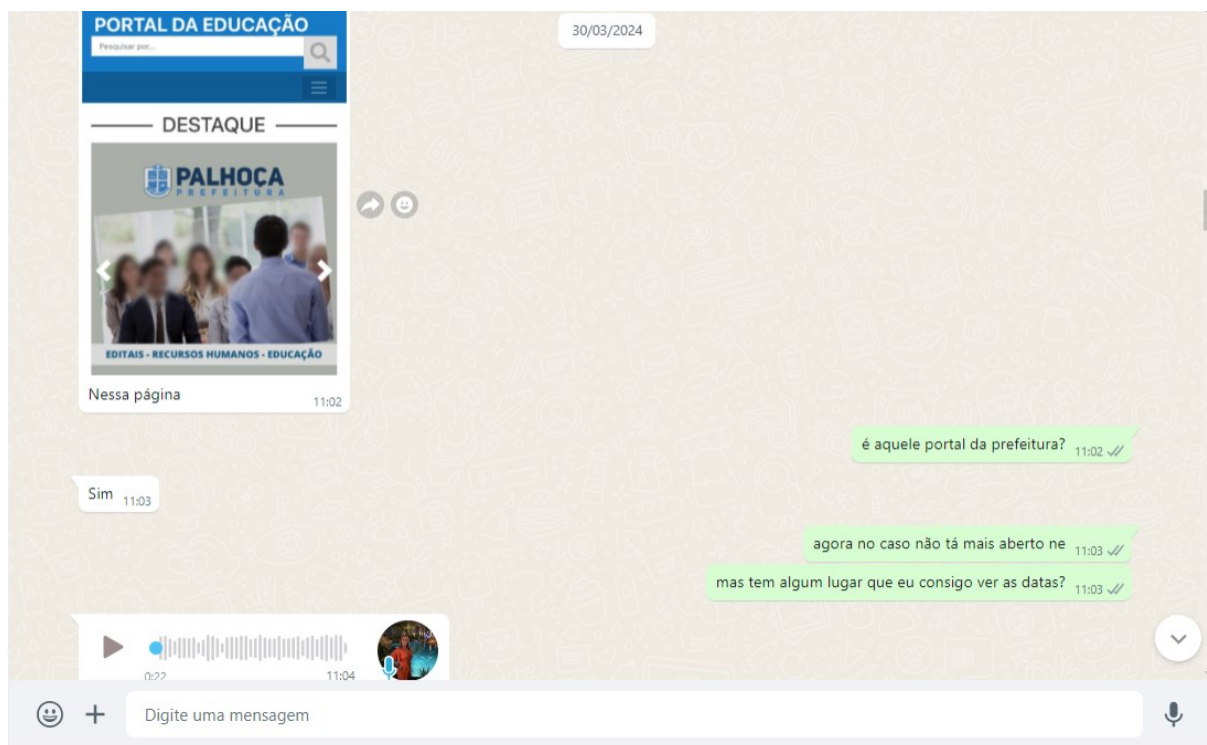
Áudio do entrevistado: Ok. Vamos ver se eu consigo te explicar melhor, vamos ver se eu entendi a pergunta. Exemplo, antes não podia pedir solicitação, uma transferência, se ela tivesse no mesmo zoneamento.

O zoneamento que a gente chama é tudo próximo. Exemplo, onde eu trabalho hoje é o Bela Vista. Hoje nós temos quatro unidades, que ficam quase na mesma rua, assim. Umas um pouco mais longe, outras mais próximas.

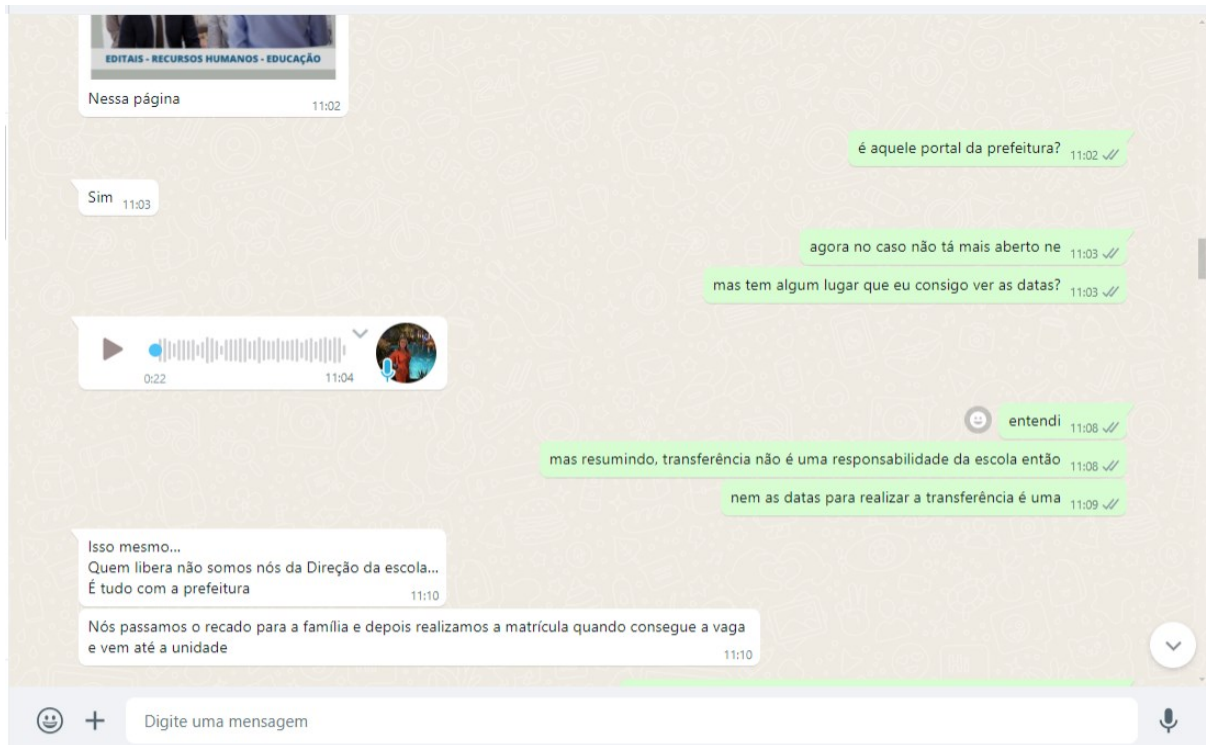
Quatro unidades, vamos ter cinco agora. Por ser Bela Vista, ninguém podia pedir dentro do Bela Vista para sair de um e ir para outro. Agora pode. Por quê? Porque a gente fica mais próximo para um do que para outro chegar.

Quem mora mais perto da BR é melhor no meu. Quem mora no Terra Nova é melhor no Vale Verde. Então, eles podem pedir.

Conversa com entrevistado(a) parte 6



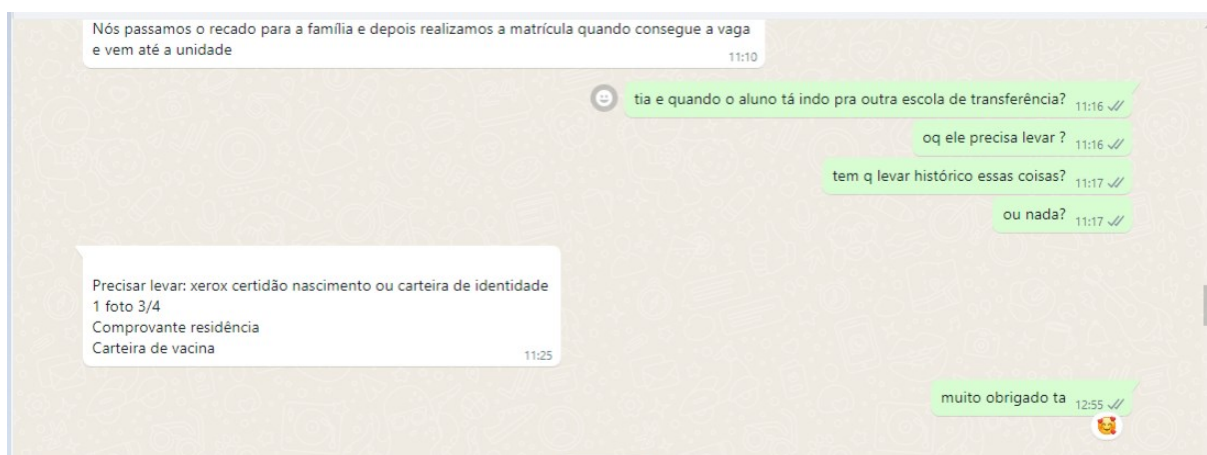
Conversa com entrevistado(a) parte 7



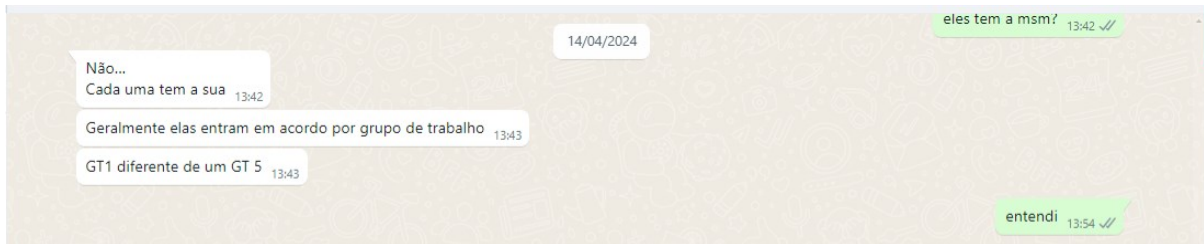
Áudio do entrevistado: Entra na página da prefeitura, no portal, vai lá em educação e desce a página. Sabe onde tem a palavra destaque? Então, vai pra baixo ali.

Aí você consegue ver os últimos que saíram, o edital de quem foi classificado. Ali saem algumas informações. Eu não sei se eles deixam por muito tempo, mas ali tu consegue ver.

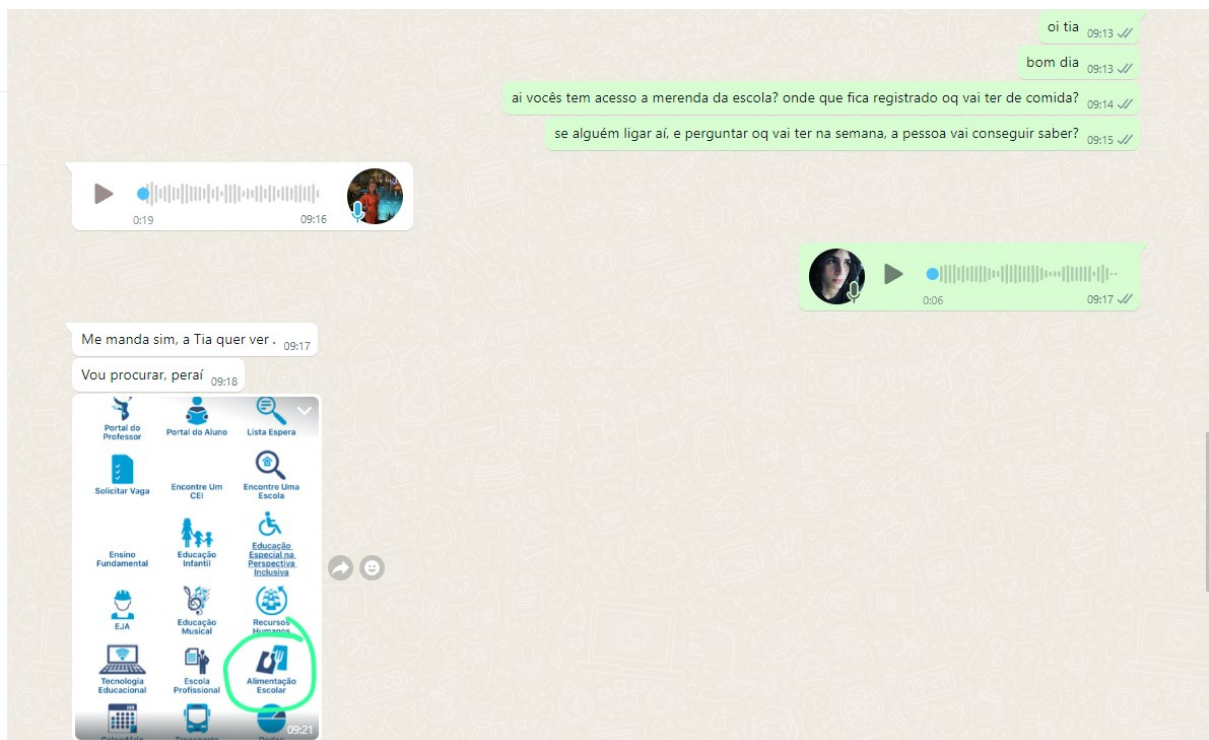
Conversa com entrevistado(a) parte 8



Conversa com entrevistado(a) parte 9



Conversa com entrevistado(a) parte 10



Áudio do entrevistado: Oi. Nós temos sim, a gente tem um cardápio. Esse cardápio é liberado pela nutricionista todo mês.

Ele muda. Ele está no portal do site da educação, ele está lá. E nós deixamos fixado também na entrada para as famílias poderem ter acesso.

Áudio do entrevistador: Ah, ok está no portal da Educação. Você consegue me mandar esse link? Da parte onde está o cardápio?