

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

Ana Carolina Schmitz Da Silva

Caíque Lacerda Martel

Leonardo Thiesen

Natália Heinzen

HENDRIX-BOT

UM *CHATBOT* INTEGRADO COM IA PARA OTIMIZAR A COMUNICAÇÃO ENTRE OS ALUNOS E *LEADS* COM A GESTÃO ESCOLAR DE UMA ESCOLA DE MÚSICA

Palhoça
2024

Ana Carolina Schmitz Da Silva

Caíque Lacerda Martel

Leonardo Thiesen

Natália Heinzen

HENDRIX-BOT

UM *CHATBOT* INTEGRADO COM IA PARA OTIMIZAR A COMUNICAÇÃO ENTRE OS ALUNOS E *LEADS* COM A GESTÃO ESCOLAR DE UMA ESCOLA DE MÚSICA

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

Palhoça

2024

Ficha de identificação da obra

H498 Hendrix-bot: um *chatbot* integrado com IA para otimizar a comunicação entre os alunos e *leads* com a gestão escolar de uma escola de música / Ana Carolina Schmitz Da Silva... [et al.] – Palhoça (SC): Faculdade Senac Palhoça, 2024.
62 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. Inteligência artificial. 2. *Chatbot*. 3. Comunicação I. Silva, Ana Carolina Schmitz Da. II. Martel, Caíque Lacerda. III. Thiesen, Leonardo. IV. Heinzen, Natália. V. Vieira, Jaime Elias de VI. Silva, Telma Alessandra Correa da. VII. Felipe, Wildson Caio VIII. Figueiredo, Luciano Coelho. IX. Título.

CDD 22 ed.: 006.3

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



Creative Commons - Atribuição - Não Comercial CC BY-NC

Ana Carolina Schmitz Da Silva

Caique Lacerda Martel

Leonardo Thiesen

Natália Heinzen

HENDRIX-BOT

UM *CHATBOT* INTEGRADO COM IA PARA OTIMIZAR A COMUNICAÇÃO ENTRE OS ALUNOS E *LEADS* COM A GESTÃO ESCOLAR DE UMA ESCOLA DE MÚSICA

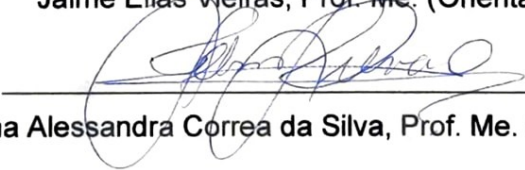
Trabalho apresentado à Faculdade Senac Palhoça como requisito parcial para obtenção do título de Tecnólogo em Análise de Desenvolvimento de Sistemas.

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

Avaliadores:



Jaime Elias Vieiras, Prof. Me. (Orientador)



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



Wildson Caio Felipe, Prof. Esp. (Orientador)



Luciano Figueiredo Coelho, Me. (Coorientador)

Palhoça, 25 de Junho de 2024

RESUMO

Este trabalho detalha o desenvolvimento do Hendrix-bot, um sistema de atendimento automatizado com integração a uma inteligência artificial (IA) para uma escola de música. O objetivo é fornecer respostas rápidas e eficientes para perguntas frequentes dos alunos. O *backend* foi desenvolvido com Java e Spring Boot, enquanto o *frontend* foi construído com Angular. O banco de dados utilizado foi MySQL, e o treinamento da inteligência artificial foi realizado com RASA, permitindo diálogos complexos e naturais. A arquitetura modular e escalável do sistema facilita a manutenção e expansão futuras. A metodologia ágil *Scrum* e um quadro kanban foram adotados para gerenciar o desenvolvimento do projeto. Essa abordagem facilitou a comunicação entre a equipe e permitiu ajustes rápidos às mudanças de escopo. As funcionalidades incluem agendamento de aulas, consultas sobre eventos e cursos, e atendimento a dúvidas comuns, acessíveis através de um *chatbot* que pode ser integrado ao sistema prévio da escola. Os testes de usabilidade mostraram que o sistema melhora a comunicação entre alunos e administração e reduz a carga de trabalho da equipe de atendimento. A personalização do *chatbot* pelo administrador garante atendimento atualizado e relevante. O sucesso do Hendrix-bot demonstra o potencial dos *chatbots* para melhorar a eficiência do atendimento ao cliente e pode ser replicado para diversos setores. Este trabalho exemplifica como a integração de inteligência artificial com uma arquitetura de software bem projetada pode criar sistemas de atendimento automatizados eficazes, personalizados e de fácil manutenção.

Palavras-chave: Inteligência Artificial. *Chatbot*. Comunicação. Automatização. Eficiência.

ABSTRACT

This work details the development of Hendrix-bot, an automated support system with artificial intelligence (AI) integration for a music school. The main goal is to provide quick and efficient answers to frequently asked questions from students. The backend was developed with Java and Spring Boot, while the frontend was built with Angular. The database used was MySQL, and the AI training was conducted with RASA, enabling complex and natural dialogues. The system's modular and scalable architecture facilitates future maintenance and expansion. The agile Scrum methodology and a kanban board were adopted to manage the project's development. This approach facilitated team communication and allowed for quick adjustments to scope changes. The features include scheduling classes, inquiries about events and courses, and addressing common questions, accessible through a chatbot that can be integrated into the school's existing system. Usability tests showed that the system improves communication between students and administration and reduces the workload of the support team. The chatbot's customization by the administrator ensures updated and relevant responses. The success of Hendrix-bot demonstrates the potential of chatbots to improve customer service efficiency and can be replicated across various sectors. This work exemplifies how the integration of artificial intelligence with well-designed software architecture can create effective, personalized, and easily maintainable automated support systems.

Keywords: Artificial Intelligence. Chatbot. Communication. Automation. Efficiency.

LISTAS DE ILUSTRAÇÕES

Figura 1 – Revisão Sprint e Retrospectiva - PHAM (2011)	22
Figura 2 – Estrutura básica de uma arquitetura em camada.....	24
Figura 3 – Diagrama de Requisitos Não-Funcionais	26
Figura 4 – Diagrama de Caso de Uso	37
Figura 5 – Diagrama de Atividade Alunos e <i>leads</i>	39
Figura 6 – Diagrama de Atividade Admin	40
Figura 7 – Quadro Kanban	41
Figura 8 – Lista de atividades.....	42
Figura 9 – Log de treinamento do framework Rasa	44
Figura 10 – Identificação de erro em teste	47
Figura 11 – Identificação de erro em teste	47
Figura 12 – Identificação de erro em teste	48
Figura 13 – Tela Home (Chat).....	49
Figura 14 – Tela de Login.....	50
Figura 15 – Tela de Dashboard.....	50
Figura 16 – Tela de Cadastro.....	51
Figura 17 – Tela do gerenciador de perguntas.....	52
Figura 18 – Tela do gerenciador de respostas	52
Figura 19 – Tela de Gerenciar Hendrix → CRUD	53
Figura 20 – Tela de histórico de conversa	53
Figura 21 – Tela de atendimentos.....	54

LISTAS DE TABELAS

Tabela 1 – Requisitos Funcionais	34
Tabela 2 – Requisitos Não Funcionais	35

LISTA DE SIGLAS

CRUD - *Create Read Update Delete* (criar ler atualizar apagar).

IDE - *Integrated Development Environment* (ambiente de desenvolvimento integrado).

JIT - *Just-in-time* (no tempo exato).

Kb – *Kilobyte*.

Mb – *Megabyte*.

NLP - *Natural Language Processing* (processamento de linguagem natural)

NLU - *Natural Language Understanding* (compreensão de linguagem natural)

VPS - Servidor Virtual Privado

SUMÁRIO

1	INTRODUÇÃO	10
1.1	OBJETIVOS	11
1.1.1	Objetivo geral	11
1.1.2	Objetivos específicos	11
2	FUNDAMENTAÇÃO TEÓRICA	12
2.1	PROCESSAMENTO DE LINGUAGEM NATURAL.....	12
2.2	<i>CHATBOT</i>	12
2.3	INTELIGÊNCIA ARTIFICIAL	13
2.4	COMUNICAÇÃO	14
2.5	ATENDIMENTO AO CLIENTE	15
2.6	GESTÃO ESCOLAR	16
2.7	GESTÃO COMERCIAL	17
3	METODOLOGIA DE PESQUISA	19
3.1	TÉCNICA DE COLETA DE DADOS	19
4	METODOLOGIA DO DESENVOLVIMENTO DO SISTEMA	21
4.1	METODOLOGIA ÁGEIS	21
4.2	ARQUITETURA.....	23
4.3	REQUISITOS	25
4.3.1	Requisitos Funcionais (RF)	25
4.3.2	Requisitos Não-Funcionais (RNF)	26
4.4	DIAGRAMAS	27
4.4.1	Caso de Uso	27
4.4.2	Atividades	27
4.5	AMBIENTE DE DESENVOLVIMENTO	28
4.5.1	Ambiente Integrado de Desenvolvimento (IDE)	28
5	RESULTADOS E DISCUSÃO	34
5.1	PLANEJAMENTO DO PROJETO	34
5.1.1	Diagrama de Caso de Uso	36
5.1.2	Diagrama de Atividade	38
5.2	GERENCIAMENTO DO PROJETO.....	40
5.3	DESENVOLVIMENTO DO SISTEMA.....	42
5.4	IMPLANTAÇÃO E OPERAÇÃO DO SISTEMA	48
6	CONCLUSÃO	55
	REFERÊNCIAS	55
	APÊNDICE A – entrevista semiestruturada	55

1 INTRODUÇÃO

Na contemporaneidade, observa-se um aumento na exigência e impaciência das pessoas, o que leva as empresas a buscarem maior agilidade em seus processos. Contudo, é essencial que essa busca por eficiência não resulte na exclusão do contato físico e personalizado, tão importante para lidar com as necessidades individuais de cada cliente.

Há bastante tempo, é perceptível em sites de empresas, especialmente aquelas que oferecem serviços, a presença de uma seção de "perguntas frequentes", com o intuito de reduzir a demanda por questões comuns e facilitar o acesso à informação para os interessados.

Atualmente, surge o *Chatbot*, uma ferramenta com o intuito similar às “perguntas frequentes” supracitadas, porém ainda limitada em sua funcionalidade. A Inteligência Artificial (IA) desponta como uma contribuição significativa nesse sentido, possibilitando uma gestão mais eficiente dessas demandas. Com a IA, torna-se viável desenvolver diversas ferramentas que beneficiam tanto os usuários, que encontram maior facilidade de acesso às informações, quanto os provedores de serviço, que conseguem automatizar tarefas rotineiras, como por exemplo a confirmação ou cancelamento de aulas.

Nesse objetivo de automação e agilidade, é fundamental que haja cautela para que as ferramentas não se tornem fonte de frustração, podendo, conseqüentemente, gerar mais incômodos do que facilidades. As demandas devem ser cuidadosamente avaliadas, garantindo sempre a possibilidade de interação humana quando solicitado.

Como a integração da Inteligência Artificial em um *chatbot* pode aprimorar a qualidade da comunicação entre empresas e clientes, permitindo respostas mais precisas e personalizadas às necessidades individuais dos usuários?

Neste contexto, o trabalho propõe a criação de um *chatbot* personalizado e integrado com a Inteligência Artificial, conforme solicitado pelo entrevistado durante o levantamento de necessidades realizado por meio de uma entrevista para coleta de dados. Sua premissa é automatizar o que for passível de melhoria e agilização, ao mesmo tempo em que assegura a possibilidade simples de interação humana, demonstrando assim um diferencial em sua prestação de serviço.

1.1 OBJETIVOS

Melhorar a comunicação e eficiência entre alunos e coordenação de curso no ensino não regular, em uma escola de música, com a implementação de um *chatbot*.

1.1.1 Objetivo geral

Desenvolver um sistema de *chatbot* integrado à inteligência artificial com o propósito de otimizar e personalizar a comunicação entre os alunos e a coordenação de uma escola de música no âmbito do ensino não regular, visando aprimorar a eficiência e a experiência colaborativa.

1.1.2 Objetivos específicos

Os objetivos específicos deste trabalho são:

- Analisar o uso do Processamento de Linguagem Natural integrado a IA.
- Entender as vantagens no uso de uma IA integrada a um sistema de *chatbot*.
- Demonstrar os benefícios do uso desta aplicação para o usuário, no que diz respeito ao custo-benefício da aplicação para empresas deste porte.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, serão abordados os conceitos principais que sustentam a pesquisa. Primeiramente, será explorado o Processamento de Linguagem Natural (PLN) e os *chatbots*, que utilizam PLN para interações automatizadas. Em seguida, será discutida a Inteligência Artificial (IA) e sua aplicação. Também será analisada a importância da comunicação, especialmente no atendimento ao cliente. Finalmente, serão examinadas as práticas de gestão escolar e comercial e seu impacto nas organizações.

2.1 PROCESSAMENTO DE LINGUAGEM NATURAL

Segundo Gonçalves (2023), o Processamento de Linguagem Natural (PLN ou NLP) é uma disciplina que combina ciência da computação, inteligência artificial e linguística para a geração e compreensão automáticas da linguagem natural. Ele se dedica ao desenvolvimento de programas computacionais capazes de analisar, reconhecer e/ou gerar textos em linguagens humanas.

Os objetivos do PLN incluem a recuperação de informações a partir de textos, tradução automática, interpretação de textos e realização de inferências a partir de textos. Esses objetivos podem ser vistos de forma macro, como compreensão da, geração e processamento de linguagem humana. Do ponto de vista computacional, o PLN passa por várias etapas. Primeiro, há o pré-processamento de texto, que envolve a remoção de pontuações, transformação de letras maiúsculas em minúsculas para padronização e remoção de palavras desnecessárias, conhecidas como *stop words*. Em seguida, ocorre a *tokenização*, onde o texto é dividido em unidades menores chamadas "*tokens*", que podem ser palavras, frases ou subpalavras, facilitando a análise e compreensão do texto. Finalmente, podem ser realizadas análises morfológicas, sintáticas ou semânticas (Gonçalves, 2023).

2.2 CHATBOT

Segundo Cruz, Alencar e Schmitz (2019), os *chatbots* são programas de computador com os quais é possível conversar em linguagem natural, mantendo um diálogo coerente, similar ao que teríamos com um interlocutor humano.

Um sistema de *chatbot* tem como objetivo emular uma conversa humana, utilizando uma tecnologia que combina um modelo de linguagem e algoritmos computacionais para simular a interação de bate-papo entre humanos e computadores, utilizando linguagem natural (Magalhães; Castro, 2019).

Embora os *chatbots* sejam amplamente utilizados, muitas vezes têm limitações evidentes, pois geralmente operam com base em regras fixas estabelecidas por humanos, sem incorporar aprendizado de máquina. Eles não têm a capacidade de interpretar respostas fora do padrão do usuário, resultando na ausência de processamento de linguagem natural (NLP) ou compreensão de linguagem natural (NLU) (Magalhães; Castro, 2019).

Gedalha (2019) resume os conceitos de NLP, que é a maneira como os *chatbots* entendem e processam a linguagem humana. Essa abordagem auxilia na interpretação e geração de respostas com base no texto fornecido pelos usuários. Por outro lado, a NLU é uma parte específica da NLP que se concentra em compreender o significado subjacente às palavras. Isso possibilita aos *chatbots* entender a intenção do usuário e oferecer respostas relevantes e úteis.

2.3 INTELIGÊNCIA ARTIFICIAL

Segundo Cruz, Alencar e Schmitz (2019), as primeiras discussões sobre mecanismos virtuais inteligentes surgiram quando o matemático Alan Mathison Turing (1912-1954) propôs um teste a Isaac Asimov (1920-1992), conhecido por seus livros sobre robôs. O objetivo do teste era verificar se um ser humano poderia conversar com um computador sem perceber que estava interagindo com uma máquina.

O conceito da IA foi muito bem esclarecido por Gurkaynak; Yilmaz; Hakserver, (2016 *apud* Valdati, 2020, p. 08).

O termo Inteligência Artificial é originário da área da ciência da computação. Foi especificamente criado em 1956, por J. McCarthy, um dos fundadores da área, definindo-a como ciência e engenharia de construir e tornar máquinas inteligentes, especialmente “computadores inteligentes”.

Não havendo consenso sobre o tema de IA dentro da comunidade científica, Medeiros (2018) analisa diferentes conceitos entre os estudiosos e parametriza alguns padrões, que incluem a capacidade de resolução de problemas, aprendizado com o

ambiente, desenvolvimento de estruturas cognitivas e definição de orientações e metas.

Mesmo com tantas conquistas e avanços no mundo da IA, incluindo grandes progressos na capacidade de processamento dos computadores e sua aplicação em diversos campos, ainda há muito a ser desenvolvido em seu potencial (Valdati ,2020). Ela cita como exemplo a falta de capacidade das IA em perceber emoções e ações humanas.

A inteligência artificial tem avançado e se expandido para diversas áreas, muitas vezes sem se parecer com um programa ou um chat de suporte. Hoje em dia, para pessoas mais leigas no ramo da tecnologia ficou bem evidenciado o que é a IA com o uso do ChatGPT, que conquistou a aceitação das pessoas em diversos setores. Ele é utilizado por advogados para consultar legislações, por médicos para auxiliar em dúvidas e até mesmo na programação, onde se tornou uma ferramenta importante e diretamente relacionada à IA.

O uso do ChatGPT tem desempenhado um papel fundamental na demonstração dos benefícios da IA para um público mais amplo, incluindo aqueles menos familiarizados com a tecnologia. Esse avanço tem sido evidente em diversos setores, conquistando a aceitação de profissionais e usuários, como por exemplo, médicos, engenheiros, advogados, contadores e outros.

2.4 COMUNICAÇÃO

Gross (2013) apresenta a perspectiva do psicólogo Abraham Maslow, que afirma que todas as pessoas possuem necessidades básicas, sem considerar sua origem étnica, religiosa ou cultural. De acordo com Maslow, as pessoas sempre buscam maneiras de satisfazer essas necessidades para garantir sua sobrevivência, e isso geralmente acontece por meio da comunicação. Assim, fica claro que a comunicação desempenha um papel fundamental ao atender às necessidades básicas da vida.

No contexto corporativo, essa importância da comunicação é ainda mais relevante. É amplamente reconhecido que muitos problemas cotidianos surgem devido a ruídos na comunicação ou à sua ausência.

Torquato (2015), afirma que a comunicação dá vida a empresa, e entendendo isso, é possível concluir que uma empresa se organiza, desenvolve e cresce graças

ao sistema de comunicação que ela cria e mantém seguindo essa mesma lógica, fica evidente a importância de uma comunicação eficaz em todas as esferas, seja com os clientes, fornecedores ou até mesmo entre os colaboradores internos.

2.5 ATENDIMENTO AO CLIENTE

O cenário empresarial atual está sujeito a mudanças constantes e significativas, as quais têm um impacto profundo na forma como as empresas se posicionam no mercado. Essas mudanças, influenciadas pela globalização, demandam uma adaptação rápida das organizações, visando manter sua competitividade elevada (Chiavenato, 2012).

No atual mercado competitivo, as empresas comerciais enfrentam o desafio contínuo de exceder as expectativas dos consumidores. Nesse sentido, é essencial que permaneçam focadas e sejam capazes de atender e satisfazer as necessidades dos clientes. Um atendimento de alta qualidade não apenas se destaca como um diferencial, mas também desempenha um papel fundamental no êxito da organização (Souza *et al.*, 2019).

Um estudo foi conduzido para investigar os motivos pelos quais as pessoas mudam de fornecedor, revelando que 45% das respostas apontaram o mau atendimento e 20% indicaram a falta de atenção pessoal como razões principais (Whiteley, 1992). Esses resultados destacam a importância do atendimento ao cliente como um dos principais produtos a serem oferecidos para garantir a fidelidade do cliente. Com frequência, empresas direcionam seus recursos para atrair novos clientes, às vezes negligenciando os já existentes, embora seja mais econômico manter um cliente do que conquistar um novo.

Pesquisadores da Universidade do Texas A&M, conforme mencionado por Seleme (2016), identificaram cinco dimensões essenciais para avaliar a experiência do cliente durante um atendimento. São elas: confiabilidade, que se refere à capacidade de cumprir de forma segura e precisa o que foi prometido; convicção, relacionada ao conhecimento e cortesia dos colaboradores para transmitir confiança e segurança; aspectos tangíveis, que dizem respeito à qualidade da instalação física, equipamento e aparência geral do ambiente; empatia, demonstrando cuidado e atenção ao cliente durante o atendimento; e receptividade, evidenciando a prontidão em oferecer um atendimento imediato e auxiliar o cliente prontamente.

2.6 GESTÃO ESCOLAR

O termo "gestão" tem sua origem nas palavras gregas que significam trazer, produzir, criar, executar, administrar e levar. No Brasil, a primeira organização educacional baseou-se no Ratio Studiorum, documento da Companhia de Jesus de 1599, que influenciou a administração, currículos e métodos de ensino nas escolas jesuítas. Esse documento estabeleceu regras, processos didáticos e promoveu a observância da doutrina católica. Na estrutura jesuíta, o Reitor era responsável pelas atividades burocráticas e administrativas, enquanto o Prefeito de Estudos lidava com questões pedagógicas (Sinhorelli, 2013).

No ano de 1759, foi estabelecido no Brasil o Alvará Régio, documento que organizou o sistema educacional, introduzindo o cargo de Diretor Geral de Estudos. Suas principais responsabilidades incluíam a seleção de professores, a supervisão das condições escolares e o controle do ensino ministrado. Com a chegada da família real em 1807, foi instituída a estrutura de ensino imperial, composta pelos níveis primário, secundário e superior, cada um supervisionado pelo Diretor Geral de Estudos. No nível superior, os diretores eram os próprios professores, subordinados ao governo imperial. Durante o período Pombalino, o ensino passou a ser regulado pelo Estado, resultando na criação das Aulas Régias, onde um único professor ministrava todas as disciplinas, sem um currículo estabelecido. Essas mudanças na estrutura educacional afetaram principalmente as elites locais. A expulsão dos jesuítas em 1759 levou à extinção dos colégios da Ordem Jesuítica e à criação das Aulas Régias. O cargo de Diretor Geral de Estudos foi introduzido, embora ainda estivesse mais envolvido em questões administrativas e burocráticas do que pedagógicas (Sinhorelli, 2013).

A sociedade, em constante transformação, influencia e é influenciada pelo Estado, especialmente no contexto educacional dentro do sistema capitalista. Nesse cenário, a gestão democrática na escola emerge como uma abordagem essencial, caracterizada pela democracia, participação e autonomia. A democracia, valorizada crescentemente, requer compreensão dos elementos como participação e responsabilidade social. A eleição de diretores por processos eleitorais é uma estratégia para democratizar a gestão escolar, embora ainda prevaleçam indicações políticas em muitas regiões. A gestão democrática promove a participação de todos os segmentos da escola, considerando questões locais como a sustentabilidade na

Amazônia e a valorização da diversidade cultural. Destaca-se a importância do Conselho Escolar como um meio de envolver os membros da comunidade na tomada de decisões escolares, promovendo uma gestão transparente e participativa (Ferreira, 2013).

2.7 GESTÃO COMERCIAL

Gestão, significa gerenciamento e administração com o propósito de crescimento, desempenha um papel central no comportamento organizacional, direcionando suas ações. Reconhecida como uma prática aprendida com a experiência e integrada ao contexto, não é considerada uma ciência devido à sua natureza prática e foco nos resultados organizacionais. Os gestores precisam estar aptos a lidar com diversas situações, analisando e mobilizando recursos e esforços para alcançar metas estabelecidas. As funções dos gestores envolvem o planejamento, análise, resolução de problemas, organização de recursos, comunicação, liderança, tomada de decisões e avaliação do desempenho global. (Lissarassa, 2014).

A gestão comercial é composta por um conjunto de atividades e estratégias utilizadas para administrar as operações comerciais de uma empresa, visando alcançar metas de vendas, lucratividade e satisfação do cliente. Ela abrange várias áreas que têm impacto direto no desempenho e sucesso da empresa, tais como marketing, recursos humanos, vendas, atendimento ao cliente, logística, estoque, precificação, inteligência de mercado, expansão e aspectos financeiros.

De acordo com Lissarassa (2014), a gestão comercial, uma das áreas funcionais fundamentais em qualquer organização, é encarregada de realizar diversas tarefas no âmbito comercial e de marketing. Essas tarefas incluem o estudo do ambiente externo e das capacidades internas da organização para prever vendas e identificar ameaças e oportunidades, além de gerenciar as ações comerciais e de marketing, estabelecendo estratégias e políticas de atuação. Também é responsável pelo relacionamento com os clientes, desde a angariação até o serviço pós-venda e a gestão de reclamações, além de coordenar a força de vendas, incluindo seu dimensionamento e estruturação. A prosperidade da empresa está diretamente ligada à função comercial, já que a venda dos produtos é imprescindível para gerar resultados. Portanto, o principal objetivo da gestão comercial é atrair e reter clientes,

aumentando a rentabilidade da carteira de clientes por meio do planejamento e execução de atividades de marketing e vendas.

3 METODOLOGIA DE PESQUISA

Nesta seção, serão descritos os métodos de coleta e análise de dados. Será abordada a abordagem qualitativa por meio de entrevistas e a pesquisa descritiva utilizando questionários abertos. Também serão apresentadas as etapas de análise das respostas e como essas informações foram utilizadas para identificar as necessidades e desafios da escola de música, visando desenvolver um sistema adequado.

3.1 TÉCNICA DE COLETA DE DADOS

Utilizando uma abordagem de Pesquisa Qualitativa por meio de entrevista e Pesquisa Descritiva através de questionário aberto, a equipe coletou dados relevantes usando o *Google Forms*. O formulário foi enviado ao coordenador de uma escola de música, que respondeu detalhadamente sobre as dificuldades enfrentadas pela instituição e as possíveis problemáticas relacionadas ao desenvolvimento do produto.

A Pesquisa Qualitativa foi conduzida por meio de entrevista com o coordenador da escola de música. Essa entrevista permitiu uma compreensão profunda das operações diárias, desafios específicos e expectativas em relação ao novo sistema proposto. Já a Pesquisa Descritiva foi implementada utilizando questionário aberto via *Google Forms*. O questionário foi elaborado para complementar as informações obtidas nas entrevistas, visando coletar dados específicos sobre as dificuldades e necessidades operacionais.

Após a recepção do formulário preenchido, a equipe se dedicou para analisar as respostas. As dificuldades apontadas pelo coordenador foram mapeadas e categorizadas, destacando problemáticas relacionadas à criação e desenvolvimento do produto. As respostas claras e detalhadas do coordenador facilitaram a compreensão dos requisitos do produto, oferecendo uma visão precisa dos desafios a serem abordados.

O coordenador destacou uma alta demanda de novos leads que buscam informações preliminares e mencionou que a equipe se concentra principalmente em questões financeiras e de agenda para os alunos ativos. Essas informações evidenciaram a necessidade de melhorar a situação atual da escola, abrindo oportunidades para a implementação de um *chatbot* integrado à IA. Automatizar as

demandas iniciais dos *leads* através do *chatbot* foi identificado como uma melhoria necessária e urgente. Além disso, o desenvolvimento de novas ferramentas para gerenciar agendas e questões financeiras pode diferenciar a escola, superando as expectativas dos clientes e agregando valor ao serviço prestado.

A liberação da equipe para se concentrar em aspectos que valorizam o atendimento humano presencial, conforme mencionado pelo coordenador, é essencial para destacar o diferencial da escola. Essa estratégia justifica o investimento para os alunos e melhora a qualidade geral do serviço, proporcionando uma experiência mais personalizada e eficiente.

Com base nas metodologias aplicadas e no tempo dedicado à coleta e análise de dados, a pesquisa forneceu uma base sólida para o desenvolvimento de um sistema que atende às necessidades identificadas, garantindo que a solução proposta seja alinhada com as expectativas e demandas da escola de música.

4 METODOLOGIA DO DESENVOLVIMENTO DO SISTEMA

Neste capítulo, serão apresentados os princípios das metodologias ágeis, com foco no Scrum e no Kanban, seguidos pela descrição da arquitetura adotada, requisitos do sistema e diagramas utilizados, além do ambiente de desenvolvimento.

4.1 METODOLOGIA ÁGEIS

Os conceitos atuais sobre o desenvolvimento ágil de software tiveram sua origem em meados da década de 1990, como uma resposta negativa aos chamados "métodos pesados" de desenvolvimento de software. Estes métodos eram caracterizados por um excesso de formalismo em documentação e regulamentação, frequentemente gerenciados de forma micro gerencial pelo tradicional e burocrático modelo em cascata. Em reação a essa abordagem, surgiram novos frameworks para processos de desenvolvimento de software, inicialmente conhecidos como "métodos leves" (Sbrocco; Macedo, 2012).

De acordo com Sbrocco e Macedo (2012), em 2001, um grupo de desenvolvedores que já estavam familiarizados com os métodos ágeis criou o Manifesto Ágil, com o intuito de impulsionar a adoção de novos sistemas de software.

Paula Filho (2015, p. 102) descreve em seu livro os quatro valores criados pelo Manifesto Ágil:

Indivíduos e interações, em relação a processos e ferramentas; Software funcional, em relação a documentação abrangente; Colaboração com o cliente, em relação a negociações contratuais; Resposta à mudança em relação a seguir planos.

Conforme Pressman e Maxin (2016), a agilidade transcende a mera adaptação às mudanças, é também uma filosofia delineada no manifesto mencionado anteriormente. Ela promove a estruturação e as atitudes em equipe que facilitam a comunicação entre todos os stakeholders, inclusive incorporando o cliente como um membro da equipe, com ênfase na entrega rápida de software funcional.

O termo Scrum é derivado de uma jogada no *rugby* em que os jogadores se agrupam ao redor da bola, colaborando para avançar em direção ao gol adversário. Criado por Jeff Sutherland em 1990, o Scrum é um método de desenvolvimento ágil de software. Seus princípios estão alinhados com o manifesto ágil e são utilizados

para guiar as atividades de desenvolvimento, que incluem requisitos, análise, projeto, evolução e entrega. Cada uma dessas atividades ocorre dentro de um ciclo iterativo chamado *sprint* (Pressman; Maxin, 2016).

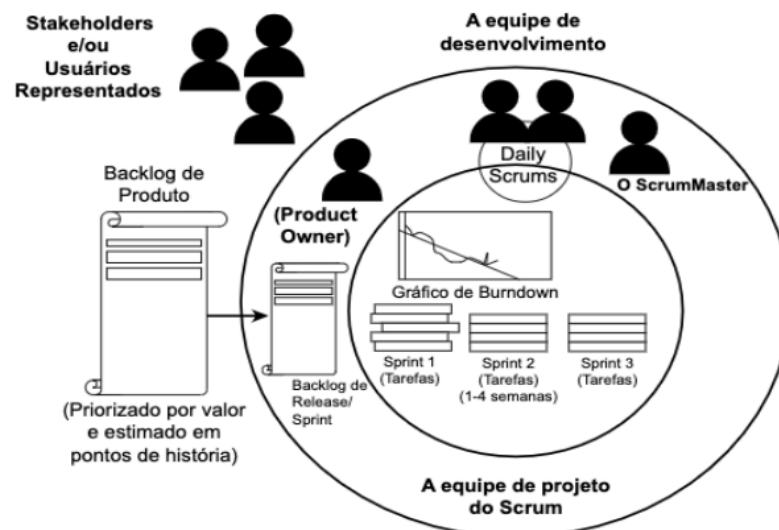
O Scrum é uma metodologia estruturada que divide o trabalho em ciclos denominados "*sprints*". Durante cada *sprint*, a equipe concentra-se em um conjunto de tarefas prioritárias, geralmente especificadas em uma lista conhecida como "*backlog do sprint*". O Scrum atribui papéis bem definidos, como o *Product Owner*, *Scrum Master* e a equipe de desenvolvimento.

Para Sbrocco e Macedo (2012, p. 160), mesmo não tendo uma solução mágica para problemas complexos, o Scrum pode ser usado para:

Desenvolvimento complexos em que os requisitos mudam rapidamente e constantemente; Gerenciar e controlar o desenvolvimento de trabalho; Tornar a equipe autogerenciável e funcional; Implementar o conceito iterativo e incremental no desenvolvimento de software e/ou produtos; Identificar causas de problemas e remover impedimentos; Valorizar os indivíduos.

Pham (2011) descreve a composição de uma equipe de Scrum com um *ScrumMaster*, *Product Owner* (dono do produto) e a equipe de desenvolvimento com todas as habilidades necessárias para construir o produto software.

Figura 1 – Revisão Sprint e Retrospectiva - PHAM (2011)



Fonte - Pham (2011).

Conforme Sbrocco e Macedo (2012), uma característica fundamental do Scrum é a premissa de que o cliente tem o direito de alterar os requisitos do projeto a qualquer

momento. Essa flexibilidade torna os desafios imprevisíveis e não convencionais. Por essa razão, a abordagem do Scrum é empírica, reconhecendo que o problema do cliente pode não ser completamente compreendido ou definido.

O conceito *Kanban* nasceu no Japão quando um grupo de executivos japoneses da empresa Toyota, em uma viagem aos EUA para observar a linha de produção das fabricantes de automóveis deles, ao parar em um supermercado, ficaram surpresos com o que viram lá dentro, onde os clientes determinavam a linha de produção com a retirada dos produtos das gôndolas (Ballester-Alvarez, 2016).

Segundo Muniz et al. (2021), o método Kanban foi criado por David J. Anderson e Don Reinertsen em 2010, o qual faz uma abordagem evolucionária em mudança de processos para as organizações. Esse método foi desenvolvido para a indústria de software, mas por sua capacidade adaptativa e evolucionária, com o passar dos anos, ele se estendeu para outros contextos que adotaram como gestão de fluxo do seu dia a dia.

O conceito básico do Kanban é que deve ser eliminado tudo que não seja estritamente necessário para produção e apenas a quantidade mínima necessária para o mesmo (Bellestero-Alvarez, 2016).

O Kanban é um sistema de gerenciamento visual que destaca o fluxo contínuo de trabalho. As tarefas são representadas visualmente em um quadro Kanban, divididas em colunas que refletem os estágios do processo, como "A Fazer", "Em Progresso" e "Concluído". Diferentemente do Scrum, o Kanban não estabelece papéis ou eventos específicos, permitindo uma adaptação mais flexível às necessidades de cada equipe. Seu principal foco é otimizar o fluxo de trabalho, limitando o trabalho em progresso e identificando e solucionando gargalos à medida que surgem.

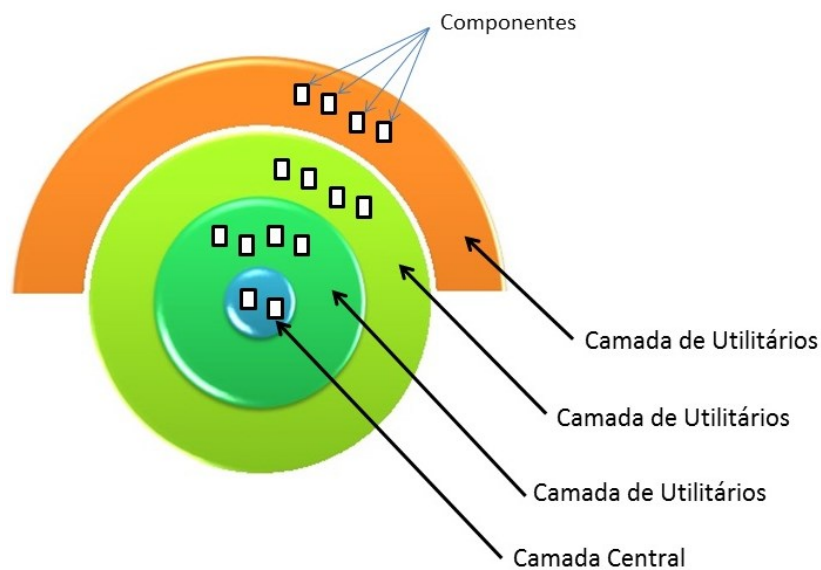
4.2 ARQUITETURA

Quando se discute sobre arquitetura, geralmente as pessoas tendem a pensar na arquitetura da construção civil, a qual abrange diversos detalhes, desde aspectos funcionais e estéticos até a consideração das necessidades e premissas do cliente. Da mesma forma, na engenharia de software, há uma abordagem similar, porém, diferentemente da arquitetura da construção civil, que é tangível e visualmente perceptível, a arquitetura de software é mais abstrata e menos visual.

De acordo com Pressman e Maxin (2016), a introdução do conceito de arquitetura na Engenharia de Software ocorreu com a divisão de programas em módulos. Os autores complementam que a arquitetura é uma representação do projeto que atende aos requisitos declarados, proporcionando uma comunicação facilitada entre todos os envolvidos.

Para o desenvolvimento do *Chatbot*, foi adotada a arquitetura em camadas. A Figura 2 apresenta a estrutura fundamental dessa arquitetura em camadas.

Figura 2 – Estrutura básica de uma arquitetura em camada



Fonte – Pressman e Maxin (2016).

Pressman e Maxin (2016, p. 261) descrevem:

São definidas várias camadas diferentes, cada uma realizando operações que progressivamente se tornam mais próximas do conjunto de instruções de máquina. Na camada mais externa, os componentes atendem às operações da interface do usuário. Na camada mais interna, fazem a interface com o sistema operacional. As camadas intermediárias fornecem serviços utilitários e funções de software de aplicação.

Segundo Sommerville (2018), à medida que as camadas são desenvolvidas, alguns serviços já podem ser disponibilizados aos usuários. A arquitetura também é mutável e móvel. Caso a interface permaneça inalterada, é possível introduzir uma nova camada com funcionalidades expandidas, substituindo uma camada existente sem afetar outras partes do sistema. Adicionalmente, quando ocorrem mudanças nas interfaces de uma camada ou quando novos recursos são incorporados, apenas a

camada adjacente é impactada. Essa abordagem em camadas reduz a dependência de máquina, facilitando a implementação multiplataforma de um sistema de aplicação. Nesse sentido, somente as camadas dependentes da máquina precisam ser reimplementadas para se ajustar aos recursos de diferentes sistemas operacionais ou bancos de dados.

4.3 REQUISITOS

A engenharia de requisitos desempenha um papel fundamental no processo de desenvolvimento de *software*, na análise de negócios e na gestão de processos. Ao levantar requisitos, é possível estimar prazos e orçamentos, bem como garantir a entrega de um sistema conforme as expectativas dos clientes. Seu objetivo é definir as funcionalidades necessárias do sistema e as restrições relacionadas à sua operação e desenvolvimento (Kerr, 2015).

Em resumo, os “Requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece, seus atributos e as restrições a seu funcionamento” (Kerr, 2015, p. 3).

Segundo Machado (2016), os requisitos constituem o ponto de partida para toda a definição do sistema e, conseqüentemente, desempenham um papel primordial no desenvolvimento do produto final.

4.3.1 Requisitos Funcionais (RF)

Os requisitos funcionais frequentemente representam um desafio significativo e contribuem para altas taxas de falha nos projetos de sistemas. Eles descrevem o comportamento do sistema, suas ações para cada entrada e as funcionalidades esperadas do sistema. A especificação de um requisito funcional deve identificar o que o *software* deve fazer, sem se preocupar com o como ele faz. É necessário distinguir entre a atividade de especificar requisitos e a atividade de especificações que ocorre durante o design do *software*. Enquanto na especificação de requisitos define-se o que o *software* deve fazer para atender às necessidades dos usuários ou do processo de negócio, no design do *software* decide-se quais funções o sistema terá para atender a essas exigências (Machado, 2016).

Conforme Kerr (2015), os requisitos funcionais abrangem as expectativas relacionadas ao desempenho do sistema, incluindo suas funcionalidades e respostas a diferentes entradas e cenários. Geralmente derivados das necessidades e experiências dos usuários, esses requisitos são comumente representados em diagramas de casos de uso. Além disso, os requisitos funcionais podem não apenas definir o que o sistema deve realizar, mas também delinear as restrições ou limitações nas operações do sistema.

4.3.2 Requisitos Não-Funcionais (RNF)

Conforme mencionado por Kerr (2015), os requisitos Não Funcionais dizem respeito a características abrangentes do *software*, que atuam como critérios para complementar os requisitos funcionais e afetam o sistema como um todo. Eles especificam detalhes e atributos internos do sistema relacionados à sua parte técnica, como linguagem de programação, características de interface, métodos de desenvolvimento e restrições de tempo de resposta, entre outras características.

De acordo com Machado (2016), os requisitos não funcionais não se relacionam diretamente com as funcionalidades oferecidas pelo sistema, mas sim com preocupações mais amplas, como padrões de qualidade, confiabilidade, desempenho, robustez, segurança, usabilidade, portabilidade, legibilidade, qualidade, manutenibilidade e outros aspectos. Ele exemplifica um requisito não funcional ao mencionar: "O tempo de resposta do sistema não deve exceder 30 segundos".

Figura 3 – Diagrama de Requisitos Não-Funcionais



Fonte – Machado (2016).

4.4 DIAGRAMAS

O Draw.IO é uma ferramenta de criação de diagramas baseada na plataforma do Google Drive, permitindo aos usuários elaborar diversos tipos de diagramas, fluxogramas e esquemas diretamente no navegador. Com uma ampla variedade de elementos categorizados por temas, como formas, ícones e modelos de interface, os usuários têm acesso a *templates* ajustáveis e a capacidade de importar conteúdos via URL. Além disso, os diagramas podem ser exportados em vários formatos, como PDF, PNG e HTML, e incorporados em páginas web (PPLWare, 2015).

Integrado ao Google Drive, o Draw.IO oferece colaboração gratuita entre usuários, destacando-se por sua completude e disponibilidade sem custos adicionais. Embora não possua a opção de exportação para VDX (Visio), é possível editar utilizando a extensão Draw.IO no Chrome, tornando-se uma escolha atrativa para quem busca uma ferramenta versátil e acessível para criação e compartilhamento de diagramas (PPLWare, 2015).

4.4.1 Caso de Uso

Segundo Rogers, Sharp e Preece (2013), o diagrama de casos de uso é considerado um dos elementos mais cruciais na representação de sistemas UML. Isso se deve ao fato de que suas categorias, assim como outros elementos da notação de diagramação UML, representam os principais conjuntos de casos de uso (agrupamentos de funcionalidades semelhantes) na fase inicial do desenvolvimento do sistema.

O Diagrama de Caso de Uso ilustra as funcionalidades e características de um sistema, bem como a maneira pela qual esses elementos se inter-relacionam com os usuários e entidades externas envolvidas em um determinado processo. Ele apresenta atores (como pessoas ou outros usuários do sistema), casos de uso (os cenários nos quais eles utilizam o sistema) e os relacionamentos entre eles (Souza, 2016).

4.4.2 Atividades

O diagrama de atividades permite a visualização das etapas sequenciais de um processo, dando ênfase ao fluxo de controle entre as atividades. Cada atividade é

considerada uma execução que não pode ser interrompida ou dividida em partes menores enquanto está em execução, resultando em uma ação específica, como uma mudança de estado do sistema ou o retorno de um valor. Após a análise do sistema, a modelagem se concentra no projeto ou conceito do sistema, refinando as representações das atividades e identificando os objetos que farão parte do sistema e suas relações. Em resumo, o diagrama de atividades na modelagem UML é uma ferramenta essencial para compreender e representar processos empresariais, proporcionando uma visão clara do fluxo de controle e das ações resultantes de cada atividade, além de servir como base para o desenvolvimento posterior do sistema (Oliveira, 2019).

O projeto Hendrix-bot contou com dois diagramas de atividade para auxiliar na visualização dos fluxos internos do sistema, fragmentados para os usuários externos e internos, da mesma forma que foi utilizado para gerar os diagramas de caso de uso. O diagrama de atividade desenvolvido para os usuários externos compreende a sequência de atividades possíveis tanto para usuários do tipo aluno quanto *leads*, sendo que, para os alunos, uma gama maior de atividades são possíveis dentro do sistema, visto que o *chatbot* consulta dados dos alunos de acordo com as demandas apresentadas via diálogo. Já a sequência de atividades disponibilizadas para usuários externos e que não possuem vínculo com a escola são mais restritas, mas ainda assim, agregam valor ao serviço prestado, principalmente com relação a funcionalidade de agendar um horário de visita à escola.

4.5 AMBIENTE DE DESENVOLVIMENTO

Nesta seção, é apresentado o ambiente de desenvolvimento utilizado no projeto, abordando ferramentas e tecnologias essenciais para a criação do sistema.

4.5.1 Ambiente Integrado de Desenvolvimento (IDE)

Para o desenvolvimento do sistema a IDE utilizada foi o IntelliJ IDEA e ele se destaca pela atenção meticulosa aos detalhes, automatizando tarefas repetitivas e tediosas para aumentar a produtividade dos desenvolvedores. Desde o início, oferece um conjunto robusto de ferramentas, incluindo suporte para controle de versão e uma ampla gama de idiomas. Para o desenvolvimento empresarial, móvel e web, o IntelliJ

IDEA oferece recursos excepcionais, como assistência de codificação para Java, Scala e Groovy, além de suporte para tecnologias modernas e *frameworks*. Com integração a sistemas de controle de versão e ferramentas de banco de dados, o IntelliJ IDEA facilita o desenvolvimento eficiente. Para o desenvolvimento móvel, oferece ferramentas de ponta para Android, PhoneGap, Cordova e Ionic, enquanto no desenvolvimento web, proporciona suporte avançado para JavaScript, HTML e CSS, incluindo AngularJS e Node.js (L3SOFTWARE, 2024).

Para criação do Hendrix-bot foi utilizado a plataforma de código aberto Rasa, que é destinada à construção de *chatbots* e assistentes virtuais com inteligência artificial. O texto explana sobre seus três principais componentes: Rasa NLU: Funciona como o "ouvido" do assistente, compreendendo a linguagem humana não estruturada. Extrai informações estruturadas, como intenções (objetivos ou significados de entradas do usuário) e entidades (palavras-chave relevantes). Por exemplo, em "Meu nome é João", "Meu nome é" é uma intenção e "João" é a entidade; Rasa Core: Responsável pelo gerenciamento do diálogo, decide como o assistente deve responder com base no contexto da conversa e no estado atual. Aprende padrões observando interações entre usuários e o assistente; e Rasa X: Um conjunto de ferramentas para desenvolvedores construírem, melhorarem e implantarem assistentes contextuais com o *framework* Rasa. Permite visualizar e anotar conversas, obter *feedback* de testadores, versionar e gerenciar modelos. Além disso, facilita o compartilhamento do assistente com usuários reais e a coleta de conversas para melhorias contínuas (Rasa, 2019).

A linguagem de programação utilizada foi o Java. A primeira versão do Java foi lançada em 1996, causando grande alvoroço entre as mídias especializadas e a imprensa. A promessa era de que ele se tornaria a "cola universal" que conecta usuários a informações, independentemente de sua origem, seja de servidores *web*, bancos de dados, provedores de informações ou outras fontes (Horstmann; Cornell, 2009).

O Spring Boot, um *framework open source* derivado do Spring, simplifica configurações de aplicações, sendo uma ferramenta altamente opinativa, guiando desenvolvedores através de uma abordagem de convenção sobre configuração. Isso permite que os desenvolvedores se concentrem mais no código do que na configuração. Ao eliminar a necessidade de configuração extensiva, o Spring Boot ampliou a acessibilidade e popularidade do ecossistema Java corporativo. Iniciando

com o *Spring* Inicializar, desenvolvedores podem criar projetos facilmente, aproveitando recursos como *Boot Starter* para unificar configurações, *AutoConfigurator* para coordenar configurações padrão e personalizadas, e *Spring Boot Actuator* para fornecer métricas de desempenho úteis em projetos de grande porte. O Spring Boot, hospedado no GitHub e suportado pela comunidade, tem como objetivos oferecer uma introdução rápida ao desenvolvimento *Spring*, ser opinativo enquanto oferece flexibilidade, e fornece uma variedade de recursos não funcionais comuns, sem a necessidade de geração de código ou configuração XML (Coodesh, 2024).

Um servidor é um computador ou sistema de computadores que fornece serviços, recursos ou dados para outros dispositivos, conhecidos como clientes, em uma rede. Ele é responsável por processar solicitações e distribuir informações, como páginas da web, arquivos, e-mails, entre outros, para os usuários que acessam a rede.

Em termos de hospedagem de sites, um servidor remoto é utilizado para armazenar os arquivos que compõem um site e disponibilizá-los na internet para acesso dos visitantes. Esses servidores podem ser compartilhados, onde várias empresas compartilham os recursos de um mesmo servidor, ou dedicados, onde uma empresa tem acesso exclusivo aos recursos do servidor (Locaweb, 2024).

O servidor utilizado para o desenvolvimento do projeto foi o Locaweb, por meio de um Servidor Virtual Privado (VPS): Neste tipo de hospedagem, por meio de virtualização, um servidor físico robusto é dividido em vários servidores virtuais isolados uns dos outros virtualmente. Apesar de compartilharem recursos físicos, processamento, tráfego, espaço em disco e memória RAM são totalmente dedicados a cada servidor virtual privado (Locaweb, 2024).

Segundo Andrei L (2023), o ReactJS é uma das bibliotecas de JavaScript mais populares para desenvolvimento de aplicativos web ou para dispositivos móveis. Criado pelo Facebook, ele contém uma coleção de trechos de código de JavaScript reutilizáveis chamados componentes, usados para a construção da interface do usuário (UI). É importante observar que o ReactJS não é um framework JavaScript, pois é responsável apenas por renderizar os componentes da camada de visualização de uma aplicação. Ele representa uma alternativa a *frameworks* como Angular e Vue, que permitem criar funções complexas.

Costa (2022), explica que o controle de versão consiste basicamente em um sistema que permite registrar alterações feitas no desenvolvimento de um *software*. A partir dele, toda a equipe envolvida no projeto tem acesso ao histórico das versões anteriores do *software*, podendo recuperar uma versão específica ou compreender quais mudanças foram feitas por outras pessoas. O versionamento é ótimo para trabalho em equipe e solo, pois permite registrar as versões de um arquivo, codar em simultâneo a outras pessoas e juntar trabalhos de forma segura e documentada.

O GitHub é um site que permite hospedar repositórios (projetos) usando o Git. Nele, outras pessoas podem ver, baixar e revisar seu código, além de você poder fazer o mesmo com projetos de outros desenvolvedores. Funciona como uma rede social para desenvolvedores, onde é possível publicar e compartilhar projetos pessoais e colaborar com outros ao redor do mundo. Oferece uma interface gráfica que facilita a visualização e gerenciamento dos repositórios. Os projetos podem ser salvos de forma pública (acessíveis a todos) ou privada (acessíveis apenas a membros autorizados). Resumidamente, você trabalha em sua máquina, salvando versões do código no GitHub e também pode baixar cópias do código hospedado no GitHub para sua máquina (Costa, 2022).

Carvalho (2020) observa que atualmente praticamente todas as pessoas têm contato com bancos de dados, que servem como locais de armazenamento de informações para consulta quando necessário. Um exemplo prático disso é a agenda telefônica de um celular.

O MySQL foi escolhido como gerenciador de banco de dados adotado para o projeto Hendrix-bot, pois, de acordo com HTI (2022), é amplamente reconhecido como um dos sistemas de gerenciamento de banco de dados mais populares e eficientes do mundo. Sua gratuidade e praticidade contribuem para sua ampla adoção, oferecendo aos usuários uma experiência prática e acessível. Além disso, o acesso a informações úteis em fóruns online e outros recursos disponibilizados tanto por desenvolvedores quanto pelo Oracle enriquecem a experiência dos usuários.

Com uma estrutura relacional baseada na Linguagem de Consulta Estruturada (MySQL), este sistema permite o acesso simultâneo de vários usuários e a execução de múltiplas tarefas de forma eficiente. Inicialmente concebido para projetos de pequeno e médio porte, o MySQL evoluiu significativamente ao longo do tempo, com uma capacidade de armazenamento que passou de milhões para *terabytes*. Sua natureza de código aberto e atualizações constantes garantem não apenas

acessibilidade, mas também facilidade de uso e melhorias contínuas para os usuários, independentemente de seu nível de expertise (HTI, 2022).

O Flyway representa uma ferramenta de migração de banco de dados de código aberto, concedendo aos desenvolvedores a capacidade de administrar a evolução do banco de dados de forma consistente, automatizada e progressiva. Ao invés de efetuar modificações manuais no banco de dados, os desenvolvedores têm a opção de redigir scripts SQL básicos que são executados automaticamente pelo Flyway. Esses *scripts* passam por versionamento e são gerenciados pela ferramenta, possibilitando que a equipe de desenvolvimento monitore as modificações no banco de dados (Cod3r, 2023).

Para Batista (2024), o Angular, um *framework* de código aberto desenvolvido pelo Google, é amplamente adotado por empresas e desenvolvedores devido à sua robustez e variedade de recursos. Baseado em TypeScript, ele adota uma abordagem orientada a componentes para a construção de interfaces dinâmicas e interativas na web. Com ferramentas poderosas para gerenciamento de estados, manipulação de formulários e roteamento, o Angular oferece uma estrutura sólida para o desenvolvimento de aplicações complexas, facilitando a manutenção por meio de sua arquitetura modular e sistema de injeção de dependência.

Um banco de dados relacional constitui-se como uma categoria de banco de dados que armazena e disponibiliza acesso a conjuntos de dados interligados. Estes bancos de dados são construídos com base no modelo relacional, o qual oferece uma abordagem intuitiva e clara para a representação dos dados em forma de tabelas. Dentro de um banco de dados relacional, cada linha na tabela representa um registro identificado por uma chave exclusiva, denominada chave primária. As colunas da tabela contêm os atributos dos dados, sendo que cada registro geralmente possui um valor correspondente para cada atributo, simplificando assim o estabelecimento das relações entre os pontos de dados.

Nos primeiros estágios dos bancos de dados, cada aplicativo mantinha seus próprios dados em estruturas únicas, resultando em dificuldades para acessar e otimizar essas informações. O modelo de banco de dados relacional surgiu como solução para esse problema, oferecendo uma abordagem padronizada e eficiente baseada em tabelas. Com o tempo, a Linguagem de Consulta Estruturada (SQL) emergiu como uma ferramenta essencial, simplificando a consulta e a otimização de desempenho em bancos de dados, graças à sua base na álgebra relacional.

A validação e os testes de *software* representam atividades de extrema importância para garantir que um *software* seja desenvolvido de maneira funcional e atenda às expectativas dos usuários. Esse processo é trabalhoso e geralmente envolve a participação de várias pessoas em sua execução. Os testes de *software* têm o objetivo de garantir que um programa atenda às necessidades dos usuários e também de identificar defeitos em seu funcionamento antes de disponibilizá-lo para uso (Polo, 2024).

Os testes de usabilidade visam simular as condições de uso do *software* do ponto de vista do usuário final, abordando aspectos como facilidade de navegação, clareza de textos e mensagens, acesso simplificado a recursos de apoio, eficiência na realização de funções, e padronização visual. O objetivo principal é avaliar e melhorar a usabilidade do *software*, garantindo que seja simples e intuitivo, e que atenda aos requisitos estabelecidos, incluindo a correção de erros e a prevenção de ações que possam causar perda de dados (Bartié, 2022).

Para Bartié (2022), os testes podem ser executados da seguinte forma:

- Entrar em cada tela e avaliar a facilidade de navegação entre elas.
- Realizar n operações e depois desfazê-las.
- Realizar procedimentos críticos e avaliar mensagem de alerta.
- Avaliar o número de passos para realizar as principais operações.
- Avaliar a existência de ajuda em todas as telas.
- Realizar n buscas no manual de ajuda e validar os procedimentos sugeridos.

5 RESULTADOS E DISCUSÃO

A seguir, o texto detalha o ciclo completo do projeto, desde o planejamento estratégico inicial até a operação contínua do sistema desenvolvido. Ele enfatiza o gerenciamento organizado e eficaz do projeto para atender aos requisitos do cliente, além de descrever os testes realizados para validar e otimizar o sistema, garantindo sua conformidade com as expectativas e especificações estabelecidas.

5.1 PLANEJAMENTO DO PROJETO

O projeto Hendrix-bot foi baseado em requisitos funcionais e não funcionais, bem como o escopo de produto gerado a partir da pesquisa realizada junto à escola de música. Portanto, para o melhor entendimento e visualização dos requisitos, as Tabela 1 e Tabela 2 foram desenvolvidas no período de planejamento do projeto, descrevendo em ordem de prioridade os requisitos funcionais e não funcionais do sistema. Os requisitos foram nomeados com um identificador padrão, sendo as iniciais “RF” para requisitos funcionais e “RNF” para requisitos não funcionais, seguidos de um valor numérico definido por ordem de prioridade para a realização do mesmo.

Tabela 1 – Requisitos Funcionais

IDENTIFICADOR	REQUISITO	DESCRIÇÃO
RF01	Interface de Chat Intuitiva	Deve ter uma interface de chat fácil de usar, permitindo que os alunos façam perguntas e recebam respostas de forma intuitiva (Pergunta aberta IA)
RF02	Respostas Precisas e Relevantes	Deve ser capaz de compreender as perguntas dos alunos e fornecer respostas precisas e relevantes sobre o curso, procedimentos administrativos, entre outros.
RF03	Parametrizável pela Coordenação	Possibilidade da coordenação inserir informações, como por exemplo, calendário acadêmico, cronograma de aulas e professores e avisos. (manual o integração)
RF04	Gestão de Conversação	Capacidade de gerenciar a conversação de forma eficiente, mantendo o contexto da interação e possibilitando múltiplas perguntas em uma mesma sessão.
RF05	Encaminhamentos e Agendamentos	Caso o aluno ainda necessite de informações complementares estas devem ser reportadas por email automático a coordenação ou ainda com agendamento de horário via whats zap
RF06	Relatórios	Atendimentos Realizados (Finalizados via chat Encaminhamentos de e-mail Agendamento de atendimentos
RF07	Autonomia da Coordenação	Possibilidade de desativação do bot, para resposta manual

Fonte – Elaborado pelos autores (2024).

Tabela 2 – Requisitos Não Funcionais

RNF01	Segurança e Privacidade	Garantir a segurança e privacidade das informações dos alunos, implementando controle de acesso (matricula ou e-mail)
RNF02	Desempenho e Escalabilidade	O chatbot deve ser capaz de lidar com um grande volume de interações de forma eficiente, garantindo tempos de resposta rápidos mesmo em momentos de alta demanda.
RNF03	Disponibilidade	Deve estar disponível para os alunos 24 horas por dia, 7 dias por semana, para que possam acessar informações.
RNF04	Compatibilidade	O chatbot deve ser compatível com uma variedade de dispositivos, incluindo computadores, tablets e smartphones, para atender às necessidades dos alunos em diferentes contextos.

Fonte – Elaborado pelos autores (2024).

A pesquisa realizada junto ao coordenador de uma escola de música apontou indicadores e níveis de importância para as demandas mais urgentes da escola no período e que podem ser tratadas através do uso de um *chatbot* integrado à IA. A pesquisa na íntegra está incluída no

Apêndice A, contudo, dentre os resultados enfatizados pelo entrevistado, destacam-se as funcionalidades que segundo o mesmo, serão as que trarão maiores benefícios à dinâmica administrativa da escola como:

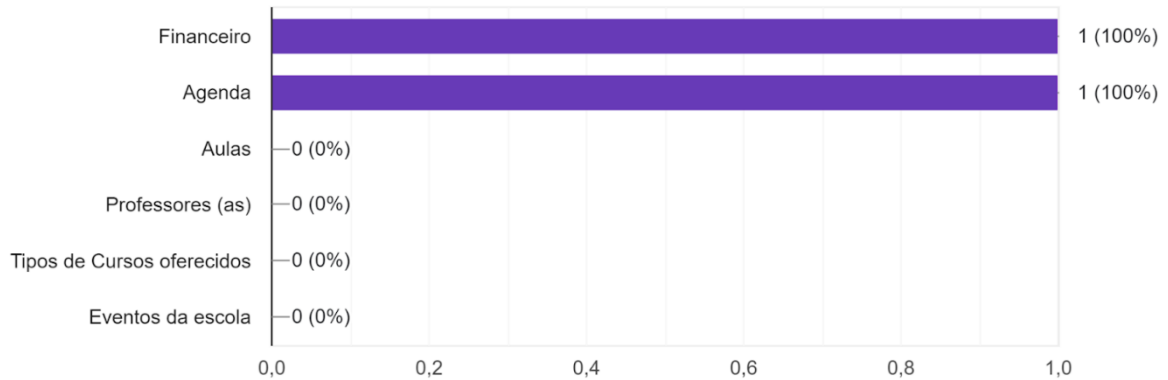
- A possibilidade de o aluno obter a confirmação de aulas ou solicitar o reagendamento ou o cancelamento;
- Verificação da agenda de eventos da escola;
- Atendimento inicial aos *leads*, que demandam desde informações mais generalistas sobre as escolas até questões de carga horária e currículo de determinado curso;

Em relação ao uso do *chatbot* por alunos matriculados, o coordenador destacou ainda, que os temas financeiro e agenda são os de maior recorrência no atendimento pela equipe de *help desk* no dia a dia da escola, conforme Gráfico 1.

Gráfico 1 – Demandas

1.4 - Qual é a área específica da interação com a escola que gera a maior demanda de dúvidas por parte dos alunos ativos?

1 resposta



Fonte – Elaborado pelos autores (2024).

Com base nos resultados trazidos pela entrevista realizada, o escopo de produto foi desenvolvido, incluindo os requisitos supracitados para a criação de um sistema, que traz vantagens relacionadas à produtividade da equipe da escola de música, apontada também como um gargalo produtivo pelo coordenador. Dessa forma, foram desenvolvidos dois tipos de diagramas para o melhor entendimento do fluxo de dados no sistema e de ações possíveis pelos usuários envolvidos.

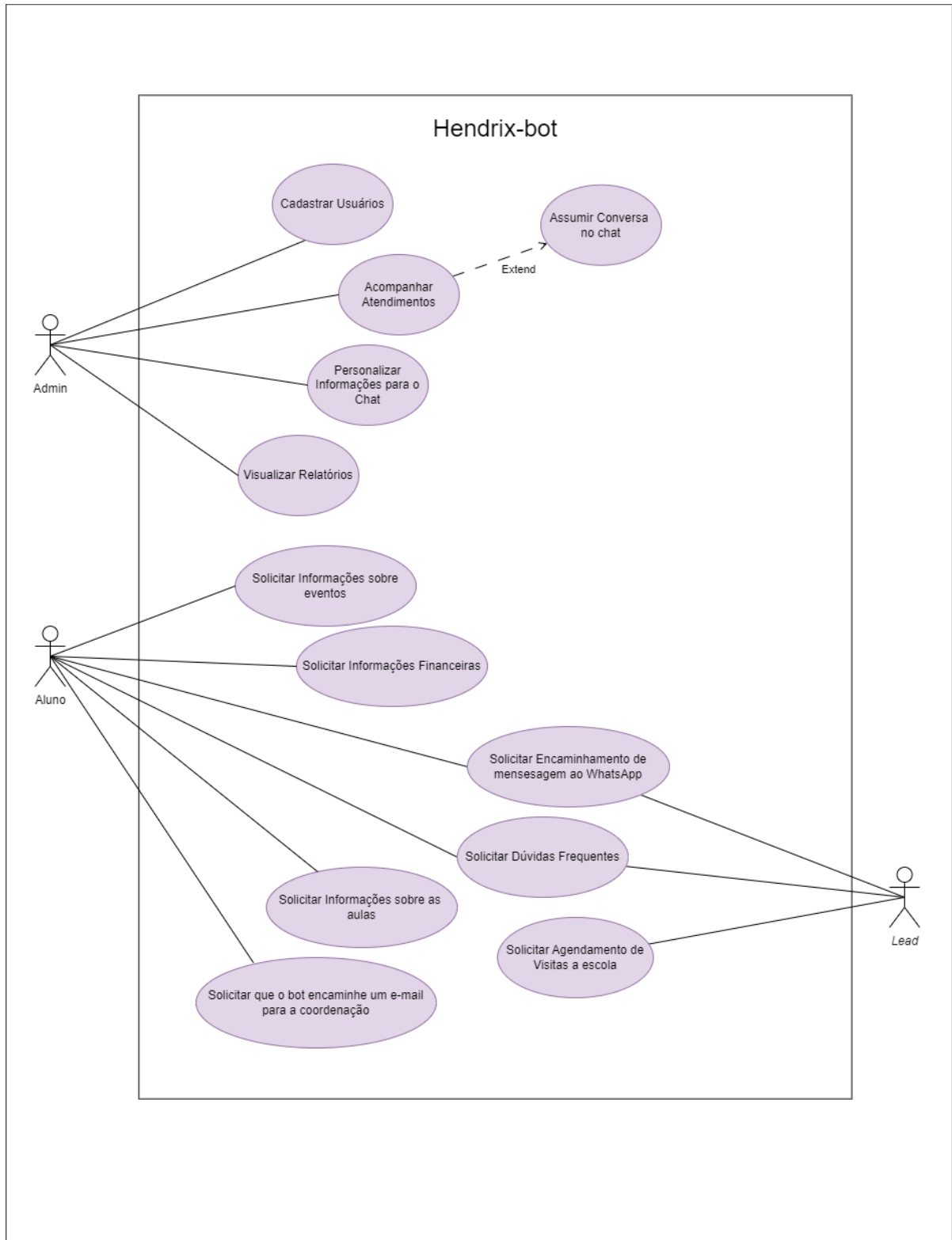
5.1.1 Diagrama de Caso de Uso

Considerando a importância de se desenvolver um Diagrama de Caso de Uso na fase inicial do desenvolvimento do sistema, reforçado por Rogers, Sharp e Preece (2013), a equipe de desenvolvimento produziu uma versão deste diagrama, juntando as ações possíveis pelo usuário principal (*Admin*), secundário (aluno) e terceiro (*lead*). Este último tipo de usuário externo se refere, de forma geral, a pessoas interessadas em angariar informações sobre a escola, agendar uma visita presencial e ou conversar diretamente com a equipe de *help desk* da instituição.

O diagrama apresentado na Figura 4 possibilitou à equipe de desenvolvimento um melhor entendimento visual das ações possíveis por tipo de ator, além de facilitar a visualização dos requisitos funcionais fundamentais, como os pressionamentos e controle de sessão. A utilização do diagrama auxiliou ainda na validação de cenários de testes realizados através dos testes de usabilidade, reforçando a sua importância

para a interpretação da dinâmica entre usuário e o sistema computacional desenvolvido.

Figura 4 – Diagrama de Caso de Uso



Fonte – Elaborado pelos autores (2024).

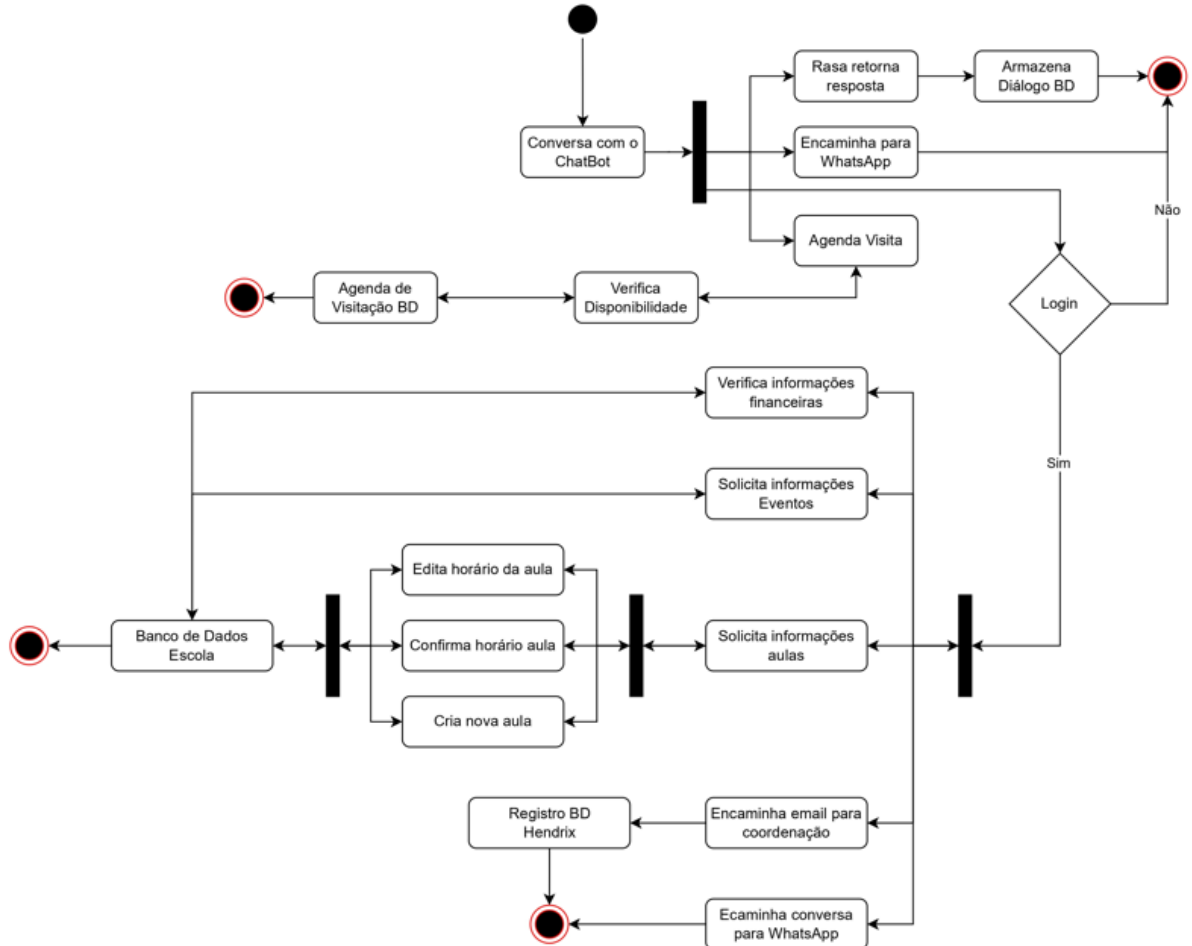
5.1.2 Diagrama de Atividade

O projeto Hendrix-bot contou com dois diagramas de atividade para auxiliar na visualização dos fluxos internos do sistema, fragmentados para os usuários externos e internos. O diagrama de atividade desenvolvido para os usuários externos (Figura 5) compreende a sequência de atividades possíveis tanto para usuários do tipo aluno quanto *lead*, sendo que, para os alunos, uma gama maior de atividades são possíveis dentro do sistema, visto que o *chatbot* consulta dados dos alunos de acordo com as demandas apresentadas via diálogo. Já a sequência de atividades disponibilizadas para usuários externos e que não possuem vínculo com a escola são mais restritas, mas ainda assim, agregam valor ao serviço prestado, principalmente com relação a funcionalidade de agendar um horário de visita à escola, ponto de atenção levantado pelo entrevistado, pois a cultura da empresa em questão valoriza a aproximação do público com a escola.

O diagrama apresentado na Figura 6 descreve as atividades inerentes ao administrador do sistema. Esse tipo de usuário, será majoritário no que tange toda a configuração personalizada do *chatbot*, os cadastros de alunos da escola, a visualização de relatórios e históricos de conversas registradas. Além disso, esse usuário poderá assumir o atendimento em substituição ao robô a qualquer momento, caso seja de sua preferência.

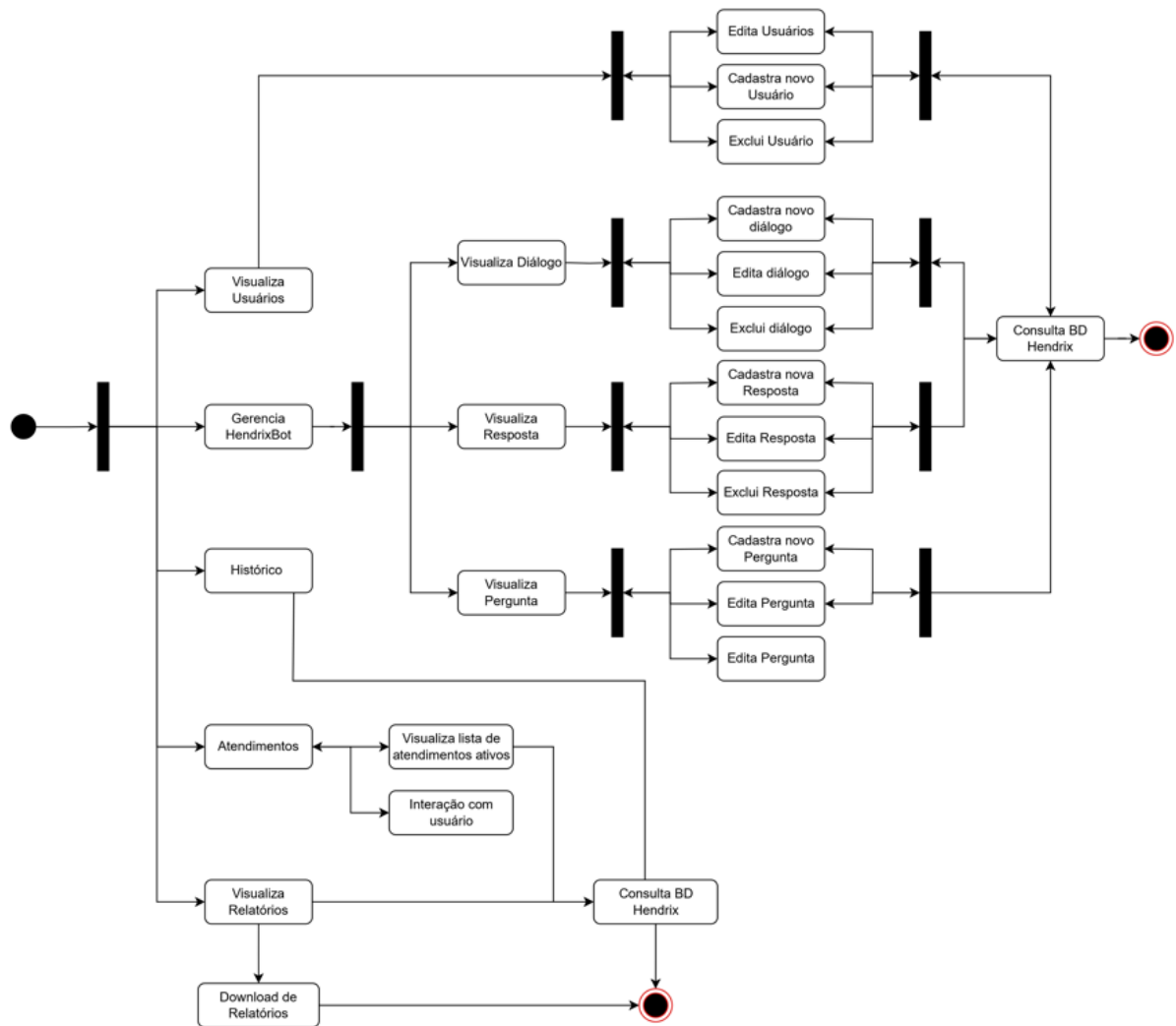
A produção de dois diagramas de atividades distintos para tipos de usuários diferentes, se mostrou vantajoso durante as tomadas de decisões inerentes ao desenvolvimento do sistema. Dessa forma, as tarefas de desenvolvimentos puderam ser melhor descritas e alinhadas entre os membros da equipe, principalmente por estabelecer limites de permissionamento e acesso a diferentes funcionalidades.

Figura 5 – Diagrama de Atividade Alunos e leads
Hendrix-bot



Fonte – Elaborado pelos autores (2024).

Figura 6 – Diagrama de Atividade Admin
Hendrix-bot



Fonte – Elaborado pelos autores (2024).

5.2 GERENCIAMENTO DO PROJETO

O time de desenvolvimento do projeto foi baseado e estruturado nos princípios da metodologia ágil, aplicando uma variação do *framework* Scrum, conforme viabilidade técnica, temporal e financeira disponível aos autores deste trabalho. Desta forma, a equipe foi previamente dividida com atividades e papéis específicos para abranger as necessidades de execução do projeto, mas, sem limitações hierárquicas definidas, tendo sido registrado variações de responsabilidades entre os membros do time ao longo do processo.

Para que os artefatos fossem entregues e concluídos conforme as solicitações avaliativas pré-definidas no âmbito da disciplina de Desenvolvimento de Aplicação, a equipe utilizou o quadro kanban (ferramenta Monday) para organizar as demandas de forma temporal e prioritária (Figura 7). A Figura 8 mostra a listagem de atividades criadas a serem desenvolvidas, e a ferramenta possibilitou visualizar de forma clara, o responsável pela atividade, status, prioridade e prazo para finalização, facilitando o trabalho individual e coletivo.

Figura 7 – Quadro Kanban



Fonte – Elaborado pelos autores (2024).

Figura 8 – Lista de atividades

<input type="checkbox"/>	Tarefa	Owner	Status	Priority	Timeline
<input type="checkbox"/>	Diagrama de Atividade		Working on it	Medium	10 - 18 mar
<input type="checkbox"/>	Diagrama de Caso de Uso		Working on it	Medium	10 - 18 mar
<input type="checkbox"/>	[CA] Escopo do Sistema		Done	Critical Δ	10 - 18 mar
<input type="checkbox"/>	Primeira Entrega		Done	Critical Δ	26 fev - 5 mar
<input type="checkbox"/>	Definição de Arquitetura		Done	High	26 fev - 2 mar
<input type="checkbox"/>	Pesquisa junta a Coordenadores de curso		Done	Critical Δ	27 fev - 9 mar
<input type="checkbox"/>	> Logotipo e nome do produto		Done	Low	27 fev - 11 mar
<input type="checkbox"/>	TL - Definição de: problema, solução, objetivos: geral e específicos. Justifica...		Done	High	5 - 12 mar
<input type="checkbox"/>	TL - Metodologia de Pesquisa : tpo de pesquisa, método de coleta de dados, ...		Done	Low	5 - 26 mar
<input type="checkbox"/>	TL - Metodologia da criação do produto: ferramentas, análise e definição de r...		Done	Low	5 mar - 2 abr
<input type="checkbox"/>	TL - Fundamentação teórica e apresentação de bibliografia		Done	Medium	5 mar - 16 abr
<input type="checkbox"/>	TL - Material (slides) da 1ª apresentação intermediária		Done	High	5 mar - 7 mai
<input type="checkbox"/>	TL - Entrega metodologia e resultados e discussões		Working on it	Critical Δ	5 mar - 7 mai
<input type="checkbox"/>	TL - Entrega final do TCS escrito (com referências bibliográficas)		Working on it	Critical Δ	5 mar - 4 jun
<input type="checkbox"/>	+ Adicionar tarefa				26 fev - 4 jun

Fonte – Elaborado pelos autores (2024).

A conjunção entre o Scrum e o Kanban é popularmente conhecida como “ScrumBan” (KANBANTOOL), e ao longo do processo de gerenciamento do projeto, foi verificadas vantagens interessantes nesta conjunção, no que tange aos indicadores de sucesso na implementação do projeto. Como por exemplo, a questão da facilitação na comunicação entre os membros da equipe, que por ter ocorrido diariamente, facilitou a resolução de problemas ou a tomada de decisões mais pontuais ao longo do período. A definição de papéis, ainda que de forma flexibilizada, equilibrou a distribuição de tarefas e otimizou os processos a partir de experiências prévias de cada membro da equipe. Bem como, a organização das sprints baseadas nos prazos de entregas de artefatos aos professores responsáveis pela disciplina e priorizadas de forma visual no quadro kanban.

5.3 DESENVOLVIMENTO DO SISTEMA

O projeto Hendrix-bot foi desenvolvido através de um conjunto de tecnologias definidas em acordo com a arquitetura de *software* utilizada. Além disso, as ferramentas que possibilitaram a produção do sistema foram definidas de forma a viabilizar o melhor desempenho individual, considerando a atuação dos(as) desenvolvedores(as) e em equipe, facilitando o versionamento e as ações inerentes à manutenção e registro do código fonte. Já as escolhas das tecnologias envolvidas

no desenvolvimento foram baseadas em conhecimentos prévios da equipe, gratuidade, maior disponibilidade de tutoriais relacionados e frequência de uso no mercado de trabalho atualmente.

A arquitetura em Camadas foi utilizada no sistema, o que garantiu a modularidade e escalabilidade do produto. O *backend* foi implementado em Java, utilizando o *framework* Spring Boot, o qual facilita o desenvolvimento de aplicações robustas e seguras (Coodesh, 2024). O *frontend* foi desenvolvido em Angular, uma plataforma para a criação de interfaces de usuário dinâmicas (Batista, 2024). A comunicação entre o *frontend* e o *backend* é realizada por meio de uma API RESTful, permitindo uma integração eficiente e a troca de dados em tempo real. Esse tipo de comunicação facilita a transferência de grandes volumes de dados, o que é essencial para a manutenção das conversas do *chatbot*, considerando que as respostas podem variar em tamanho e complexidade

O banco de dados utilizado foi o MySQL, um sistema de gerenciamento de banco de dados relacional que armazena informações críticas, incluindo todos os registros das conversas executadas através do *chatbot*, passíveis de serem utilizadas na geração de relatórios para o administrador do sistema. O armazenamento de informações inerentes aos alunos e a escola foi executado no banco de dados vinculado ao sistema, para possibilitar a execução e demonstração do mesmo. Contudo, para um cenário onde o sistema será utilizado de forma integrada a algum outro produto externo, esses dados devem ser originados e coletados na base existente do cliente através de regras de negócio pré-definidas a serem estabelecidas.

Para o treinamento da inteligência artificial integrado ao *chatbot*, utilizou-se o *framework* RASA, especializado em processamento de linguagem natural (PLN) e gestão de diálogos. O RASA permite que o *chatbot* compreenda e responda às perguntas dos usuários de maneira natural e eficaz. A Figura 9 representa a execução do treinamento do *chatbot*, programada para ser realizada através de um *script* automatizado a cada trinta minutos, 24hrs por dia.

Figura 9 – Log de treinamento do framework Rasa

```

declare_namespace(pkg)
/usr/local/lib/python3.10/dist-packages/urllib3/poolmanager.py:316: DeprecationWarning: The 'strict' parameter is no longer needed on Python 3+. This will raise an error in urllib3 v2.1.0.
warnings.warn(
/usr/local/lib/python3.10/dist-packages/urllib3/poolmanager.py:316: DeprecationWarning: The 'strict' parameter is no longer needed on Python 3+. This will raise an error in urllib3 v2.1.0.
warnings.warn(
/usr/local/lib/python3.10/dist-packages/rasa/shared/utils/io.py:98: UserWarning: Intent 'nova_pergunta' has only 1 training examples! Minimum is 2, training may fail.
/usr/local/lib/python3.10/dist-packages/rasa/shared/utils/io.py:98: UserWarning: Intent 'inf_bolsa' has only 1 training examples! Minimum is 2, training may fail.
/usr/local/lib/python3.10/dist-packages/rasa/shared/utils/io.py:98: UserWarning: Found a rule-based policy in your configuration but no rule-based training data. Please add rule-based stories to your train
to or remove the rule-based policy ('RulePolicy') from your configuration.
More info at https://rasa.com/docs/rasa/rules
2024-06-04 20:55:57 rasa.engine.training.hooks - Starting to train component 'RegexFeaturizer'.
2024-06-04 20:55:57 rasa.engine.training.hooks - Finished training component 'RegexFeaturizer'.
2024-06-04 20:55:57 rasa.engine.training.hooks - Starting to train component 'LexicalSyntacticFeaturizer'.
2024-06-04 20:55:57 rasa.engine.training.hooks - Finished training component 'LexicalSyntacticFeaturizer'.
2024-06-04 20:55:57 rasa.nlg.featurizers.opener - Starting to train component 'CountVectorsFeaturizer'.
2024-06-04 20:55:57 rasa.nlg.featurizers.opener - Finished training component 'CountVectorsFeaturizer'. - 709 vocabulary items were created for text attribute.
2024-06-04 20:55:57 rasa.engine.training.hooks - Starting to train component 'CountVectorsFeaturizer'.
2024-06-04 20:55:57 rasa.nlg.featurizers.opener - Finished training component 'CountVectorsFeaturizer'. - 178 vocabulary items were created for text attribute.
2024-06-04 20:55:57 rasa.engine.training.hooks - Starting to train component 'DIETClassifier'.
/usr/local/lib/python3.10/dist-packages/rasa/shared/utils/train_utils.py:525: UserWarning: constrain_similarities is set to 'False'. It is recommended to set it to 'True' when using cross-entropy loss.
rasa.shared.utils.io.raise_warning(
Epochs: 100% | 100/100 [00:23<00:00, 4.25It/s, t_loss=1.53, t_acc=0.964]
2024-06-04 20:56:21 rasa.engine.training.hooks - Finished training component 'DIETClassifier'.
2024-06-04 20:56:21 rasa.engine.training.hooks - Starting to train component 'EntitySynonymMapper'.
2024-06-04 20:56:21 rasa.engine.training.hooks - Finished training component 'EntitySynonymMapper'.
Processed story blocks: 100% | 16/16 [00:00<00:00, 5991.33It/s, # trackers=5]
Processed story blocks: 100% | 16/16 [00:00<00:00, 646.98It/s, # trackers=16]
Processed story blocks: 100% | 16/16 [00:00<00:00, 175.81It/s, # trackers=50]
Processed story blocks: 100% | 16/16 [00:00<00:00, 148.60It/s, # trackers=5]
2024-06-04 20:56:22 rasa.engine.training.hooks - Starting to train component 'MemorizationPolicy'.
Processed trackers: 100% | 16/16 [00:00<00:00, 5028.01It/s, # action=53]
Processed actions: 53It [00:00, 22905.52It/s, # examples=53]
2024-06-04 20:56:22 rasa.engine.training.hooks - Finished training component 'MemorizationPolicy'.
2024-06-04 20:56:22 rasa.engine.training.hooks - Starting to train component 'RulePolicy'.
Processed trackers: 0It [00:00, 0It/s]
Processed trackers: 100% | 16/16 [00:00<00:00, 5077.85It/s, # action=53]
Processed trackers: 0It [00:00, 0It/s]
2024-06-04 20:56:22 rasa.engine.training.hooks - Finished training component 'RulePolicy'.
2024-06-04 20:56:22 rasa.engine.training.hooks - Starting to train component 'TEDPolicy'.
Processed trackers: 100% | 516/516 [00:00<00:00, 4818.59It/s, # action=839]
Epochs: 100% | 100/100 [00:18<00:00, 5.42It/s, t_loss=0.67, loss=0.392, acc=0]
2024-06-04 20:56:41 rasa.engine.training.hooks - Finished training component 'TEDPolicy'.

Processed trackers: 100% | 16/16 [00:00<00:00, 3752.03It/s]
2024-06-04 20:56:22 rasa.engine.training.hooks - Finished training component 'RulePolicy'.
2024-06-04 20:56:22 rasa.engine.training.hooks - Starting to train component 'TEDPolicy'.
Processed trackers: 100% | 516/516 [00:00<00:00, 4818.56It/s, # action=839]
Epochs: 100% | 100/100 [00:18<00:00, 5.42It/s, t_loss=0.67, loss=0.392, acc=0]
2024-06-04 20:56:41 rasa.engine.training.hooks - Finished training component 'TEDPolicy'.

```

Fonte – Elaborado pelos autores (2024).

No Rasa, o termo "Epochs" refere-se ao número de passagens completas pelo conjunto de dados durante o treinamento do modelo de *machine learning*. Cada época (ou "epoch") é uma rodada completa onde o modelo vê todos os exemplos de treinamento uma vez. Durante essa rodada, o modelo ajusta seus parâmetros para melhorar seu desempenho com base no *feedback* obtido (Rasa, 2019).

Os valores destacados na Figura 9 são utilizados para monitorar o progresso do treinamento do modelo e ajustar parâmetros conforme necessário. Se a perda de validação começa a aumentar enquanto a perda de treinamento continua a diminuir, isso pode indicar um problema de *overfitting*, onde o modelo está memorizando os dados de treinamento, mas não está generalizando bem para novos dados. Conforme os dados apresentados os valores de perda de validação e treinamento estão equilibrados, não indicando um cenário de *overfitting*. A velocidade temporal verificada foi aplicada para uma média de 350 Kb (*kilobyte*) de arquivos de treinamento, o que sugere que o valor de crescimento da base de dados de treinamento pode gerar um aumento médio de 5,5 iterações por segundo a cada 350 Kb. Segundo a documentação do *framework* (Rasa, 2019), o desempenho do treinamento pode variar de acordo com as configurações de *hardware* do servidor e a média de tempo para

modelos com uma média de 1 Mb (*megabyte*) de dados fica entre 5 e 10 minutos. Esses dados são importantes para estimativas financeiras do produto, considerando que a personalização do *chatbot* irá aumentar a demanda de armazenamento e processamento do treinamento da IA.

Os resultados apresentados demonstram um alto desempenho do treinamento do *framework* RASA, integrado ao *chatbot*, em termos de tempo de processamento e sucesso das iterações do modelo. Além disso, a capacidade do RASA de personalizar as ações do sistema, por meio do consumo de *endpoints* ou consultas diretas ao banco de dados, reforça sua vantagem na utilização em produtos escaláveis. Isso se deve ao fato de que a responsabilidade de executar algoritmos de processamento de linguagem natural é gerida de forma eficiente pelo próprio *framework* segundo Sharma e Joshi (2020).

Segundo Jiao (2020), o uso do *framework open source* RASA, quando integrado a um *chatbot*, traz melhores resultados no que diz respeito a precisão do entendimento de sentenças maiores fornecidas pelo usuário, conseguindo extrair inclusive, maior número de entidades de uma única entrada, quando comparado ao método de redes neurais. Contudo, para sentenças com palavras isoladas, o método de redes neurais apresentou uma diferença superior se comparado com o método de NLU. Desta forma, a personalização de ações disponibilizada pelo RASA foi utilizada para pré-definir um menu *default* disponibilizado ao usuário conforme entradas esperadas, facilitando a interação com o usuário e evitando perdas de comunicação pelo retorno de informações distorcidas.

A utilização da IA através do *framework* RASA, no âmbito do produto desenvolvido, apresentou vantagens diretamente relacionadas ao cumprimento dos requisitos funcionais estabelecidos e das necessidades do cliente. O principal ponto foi a capacidade de contextualização da IA sobre assuntos contidos em seu modelo de treinamento, o que proporciona uma forte aproximação com a linguagem humana natural, durante a interação entre o usuário e o *chatbot*. Essa característica atende uma das principais necessidades do cliente em manter a aproximação humana nos serviços prestados pela escola, bem como os serviços de redirecionamento do atendimento para o *WhastApp* e agendamento de visitas à escola.

Outra vantagem analisada sobre o uso da inteligência artificial integrada, e mais especificamente, ao uso do *framework* RASA, foi a possibilidade de personalização dos contextos pelo próprio cliente. Dessa forma, o produto mantém o seu valor ao

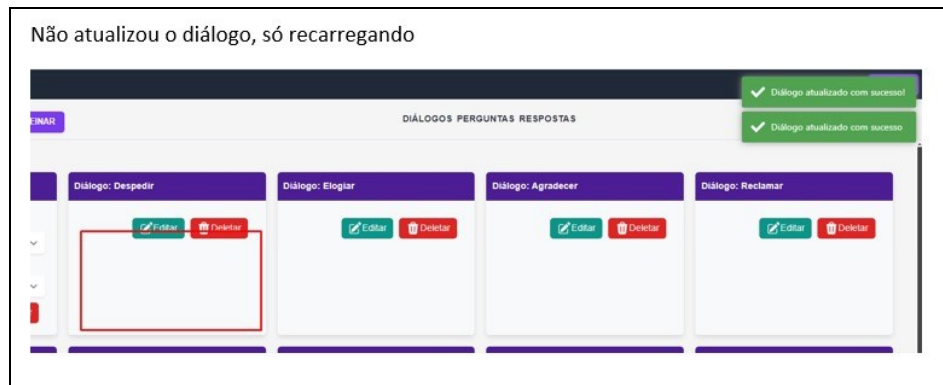
longo do tempo e proporciona maior independência ao usuário administrador conforme a necessidade de mudanças nas respostas disponibilizadas aos alunos. Além disso, o produto, inicialmente direcionado a um nicho de mercado específico pode ser facilmente aplicável a outros setores com interesse em utilizar um sistema de atendimento automatizado e personalizável de acordo com suas demandas específicas.

O Hendrix-bot foi construído de forma integrada à ferramenta Flyway, com o intuito de gerenciar e controlar as migrações do banco de dados, facilitando a manutenção e evolução da estrutura dos dados ao longo do tempo. A utilização dessa ferramenta é de suma importância, considerando que todos os arquivos de treinamento e chats realizados são armazenados em base dados, evitando possíveis perdas durante o desenvolvimento e utilização do sistema. O sistema está hospedado em um servidor privado fornecido pela Localweb, garantindo performance, segurança e disponibilidade.

A integração entre esses componentes foi planejada para garantir que todas as partes do sistema trabalhem harmoniosamente. O *backend*, desenvolvido com Spring Boot, gerencia as requisições do *frontend* e se comunica com o banco de dados MySQL para obter e armazenar informações. A API RESTful expõe os *endpoints* necessários para que o *frontend* Angular possa interagir com o *backend*, oferecendo uma experiência de usuário fluida e responsiva. O RASA, como motor de IA, processa as mensagens dos usuários e gera respostas inteligentes ou se comunica com os *endpoints* disponibilizados pelo *backend* para processar as ações solicitadas pelos usuários e, que são então, enviadas de volta para o *frontend* através da API. O uso do Flyway assegura que qualquer alteração no esquema do banco de dados seja aplicada de maneira controlada e consistente.

Durante o desenvolvimento do sistema e após a finalização dos artefatos que incluíam o funcionamento da parametrização do *chatbot*, foram iniciados os testes de usabilidade conforme as páginas foram finalizadas. A atividade ficou sob responsabilidade de um membro da equipe, contudo, as anotações e visualizações de defeitos foram sendo constantemente realizadas durante todo o processo de desenvolvimento e por todos da equipe. O repasse para a equipe de desenvolvimento foi realizado de forma dinâmica através de evidências da tela no momento em que o erro ocorria e com a descrição breve do problema. Os arquivos gerados foram mantidos em repositório de arquivos para melhor controle dos problemas averiguados.

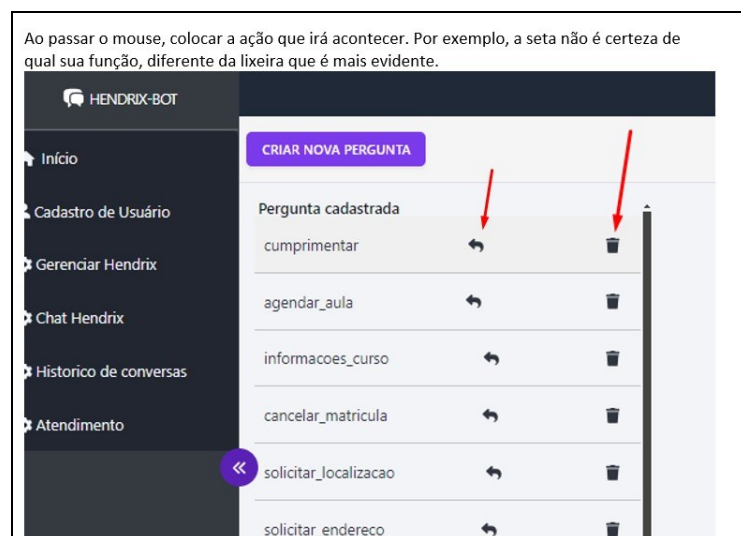
Figura 10 – Identificação de erro em teste



Fonte – Elaborado pelos autores (2024).

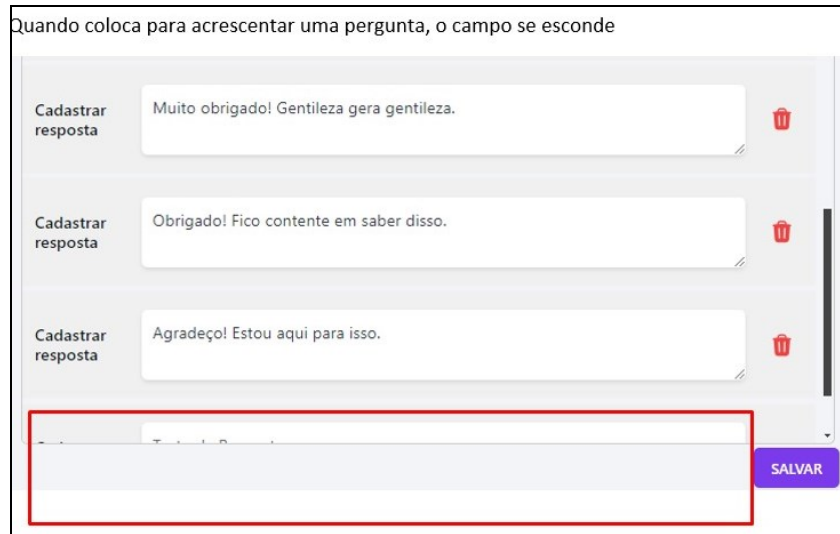
A Figura 10 apresenta um problema de carregamento dos diálogos já registrados, causado por uma falha na rota de redirecionamento após a inclusão de um novo diálogo, e o problema foi corrigido adicionando a ação de busca na base de dados após o usuário realizar alguma alteração de CRUD (*create, read, update and delete*). Já as Figura 11 e Figura 12 se referem a problemas relacionados ao *FrontEnd* do sistema, principalmente no que se refere a estilização das páginas. Como por exemplo, na Figura 11, foi apontado a ausência de ação em dois ícones da página de perguntas cadastradas, e como forma de ajuste, foi retirado um dos ícones e adicionado a ação de deletar junto ao ícone de lixeira apontada na imagem. A Figura 12 também demonstrou um erro de estilização da janela de diálogo disponibilizada para que o usuário inclua sua mensagem e que foi melhorada através de alterações de estilo.

Figura 11 – Identificação de erro em teste



Fonte – Elaborado pelos autores (2024).

Figura 12 – Identificação de erro em teste



Fonte – Elaborado pelos autores (2024).

5.4 IMPLANTAÇÃO E OPERAÇÃO DO SISTEMA

O Hendrix-bot é um sistema projetado para ser integrado ao sistema do cliente, no que tange a base de dados de acordo com as regras de negócio solicitadas pelo mesmo. Contudo, para demonstração do produto e funcionamento do sistema de forma integrada, dados simulados foram inseridos à base de dados do Hendrix-bot, como informações de agenda de eventos, aulas e dados financeiros. Além disso, *scripts* iniciais foram adicionados ao treinamento da IA com temas generalistas e entradas básicas possíveis pelo usuário ou respostas gerais dadas retornadas pelo *chatbot*, com frases como “olá, tudo bem?” ou “como posso ajudar você?”.

Quando da implantação do sistema, um ou mais usuários do tipo administrador (*admin*) serão criados conforme demanda, e esse tipo de usuário poderá então, realizar o cadastro de alunos no sistema ou gerenciar esses cadastros. Esse cadastro é independente do sistema do cliente, oferece ao aluno a possibilidade de acessar informações referentes ao seu relacionamento com a escola através do *chatbot* e foi estruturado para ser feito por um usuário administrador, pelo fato de que a escola possui alunos com idade menor à idade legal.

Para possibilitar que o aluno consiga obter as informações desejadas e relacionadas à escola, um menu é integrado aos arquivos de treinamento do RASA (.yml), através das ações personalizadas e regras que sinalizam um *checkpoint* no *framework*. As implantações dessas configurações são realizadas diretamente com o cliente de acordo com as necessidades apontadas e para a simulação do

funcionamento destas configurações, um menu foi incluído previamente. A etapa de configuração inicial e integração com a base de dados da escola, demonstra um bom custo benefício ao cliente acerca da personalização de um sistema cujo serviço de atendimento automatizado traz valor aos seus produtos ou serviços oferecidos.

A operação do sistema pode variar de acordo com o tipo de usuário logado ou não. A página principal do mesmo compreende a página do *chabot* (Figura 13) e poderá ser utilizada por qualquer usuário, mesmo ele não estando logado, pois o sistema oferecerá informações sobre dúvidas frequentes, o redirecionamento ao *whatsapp* da escola e o agendamento de visitas à escola.

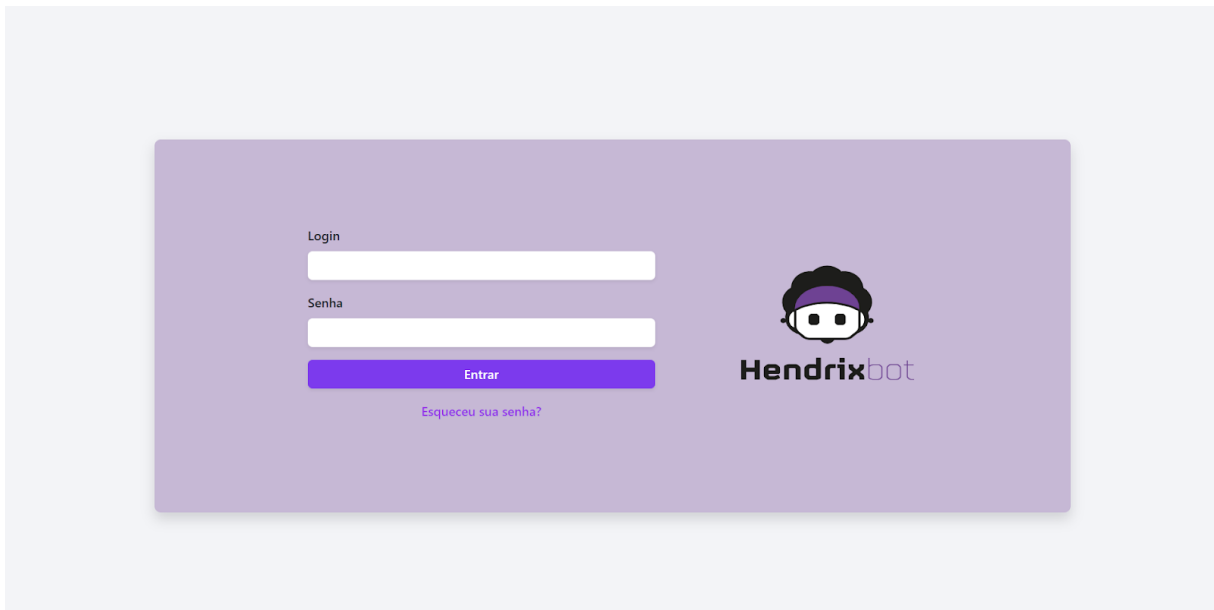
Figura 13 – Tela Home (Chat)



Fonte – Elaborado pelos autores (2024).

Caso o aluno deseje realizar o login no sistema para solicitar informações sobre o seu relacionamento com a escola, ele poderá acessar a página de login através de um botão contido na nave-bar da página principal (Figura 14). O acesso para os usuários administradores se dará da mesma forma.

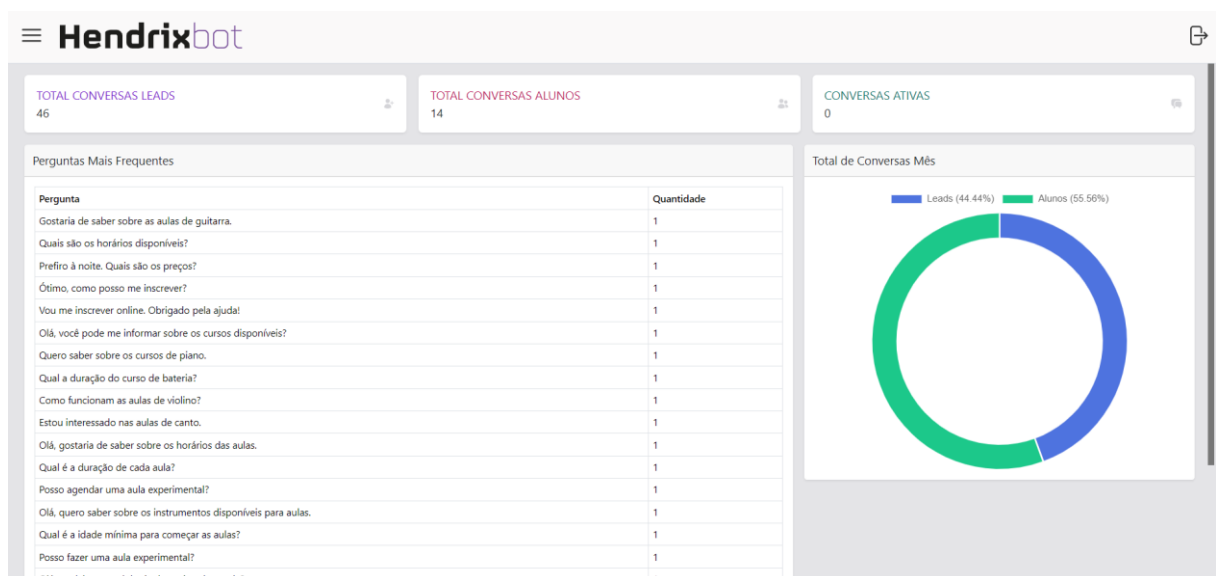
Figura 14 – Tela de Login



Fonte – Elaborado pelos autores (2024).

O usuário principal do sistema é o usuário com mais atividades disponíveis a serem realizadas e todas elas estarão expostas no menu lateral do sistema. Seguindo a ordem de opções, a tela inicial na visão do administrador após o *login* será a home com os *dashboards* de indicadores (Figura 15), disponibilizados ao administrador com três tipos de relatórios iniciais: “número de conversas totais realizadas”, “números totais de interações com o chat” e “número de alunos logados que utilizaram o chat”.

Figura 15 – Tela de Dashboard



Fonte – Elaborado pelos autores (2024).

Conforme supracitado, o administrador será responsável em realizar o cadastro dos alunos no Hendrix-bot, portanto, o sistema disponibiliza uma página onde o *admin* pode consultar todos os alunos cadastrados, verificar suas informações e realizar as operações de CRUD sobre esses cadastros, conforme a Figura 16.

Figura 16 – Tela de Cadastro

The image shows a web interface for user registration. On the left, there is a modal form titled 'Cadastrar Usuário:' with the following fields: 'Nome completo', 'E-mail', 'Senha', 'Confirme a senha', 'CPF', 'Data de nascimento' (with a date picker), 'Contato', and 'Responsável'. Below the form are 'Salvar' and 'Cancelar' buttons. On the right, a table displays a list of registered users with columns for 'Data de Nascimento', 'Contato', 'Responsável', and 'Tipo de Usuário'. A button 'ADICIONAR USUÁRIO' is visible in the top right corner of the table area.

	Data de Nascimento	Contato	Responsável	Tipo de Usuário
	01/11/1994	555-6789	Mário Castro	ALUNO
	06/12/1986	555-3456	André Oliveira	ALUNO
	10/01/1997	555-7890	José Silva	ALUNO
	22/02/1984	555-4567	Raimundo Santos	ALUNO
	29/03/1999	555-8901	Roberto Costa	ALUNO
	03/04/1983	555-2345	Fernando Almeida	ALUNO
Camila Lima	13/05/1996	789.012.345-67	Antônio Lima	ALUNO

Fonte – Elaborado pelos autores (2024).

O item “Gerenciar Hendrix” do menu lateral, direciona para a página onde o usuário poderá realizar a personalização do *chatbot*, possibilitada através de uma interface intuitiva e em acordo com as necessidades do RASA para geração dos modelos de treinamento. Essa página contém basicamente três abas para o treinamento, onde o administrador poderá criar respostas, perguntas e montar diálogos possíveis com essa listagem de perguntas e respostas criadas. O RASA utiliza três arquivos para a geração de treinamento do modelo, um deles é o *nlu.yml* que inclui as possíveis entradas de dados dos usuários quando forem utilizar o chat, e para melhor entendimento do *admin* na personalização, essas entradas estão descritas como “perguntas”, conforme Figura 17. Já as mensagens de retornos do *chatbot* são inseridas no modelo de treinamento pelo arquivo *domain.yml* e para o usuário administrador, a entrada desses dados está definida como “respostas” e as possibilidades de CRUD para elas estão demonstradas na Figura 18. A partir do cadastro das perguntas e respostas, o administrador do sistema poderá utilizá-las para montar diálogos possíveis entre os alunos e o chatbot (Figura 19). Esses diálogos são cadastrados junto ao arquivo *stories.yml* pelo sistema e todos são disponibilizados ao RASA para o próximo treinamento agendado.

Figura 17 – Tela do gerenciador de perguntas

Hendrixbot

+ Criar Nova Pergunta

Perguntas Respostas Diálogos

Perguntas Cadastradas

- aula_de_canto
- aulas_bateria_data
- aulas_individuais
- aulas_individuais_grupo
- aulas_violao
- cumprimentar
- despedir
- endereco_escola
- escolha_menu_1
- escolha_menu_2
- escolha_menu_3
- escolha_menu_4

Título da pergunta

aula_de_canto **Salvar**

a escola oferece aula de canto

aulas de canto

aulas de voz

canto

curso de canto

Fonte – Elaborado pelos autores (2024).

Figura 18 – Tela do gerenciador de respostas

Hendrixbot

+ Criar Nova Resposta

Perguntas Respostas Diálogos

Respostas Cadastradas

- cumprimentar_response
- despedir_response
- salas_de_aula_acustica
- aulas_violao
- valor_aula_violão
- aula_bateria_data
- endereco_escola
- aulas_individuais
- aulas_individuais_grupo
- periodo_funcionamento_escola
- nivel_experiencia
- aula_de_canto

Título da resposta

cumprimentar_response **Salvar**

Alô! Em que posso ser útil?

Beleza! E com você?

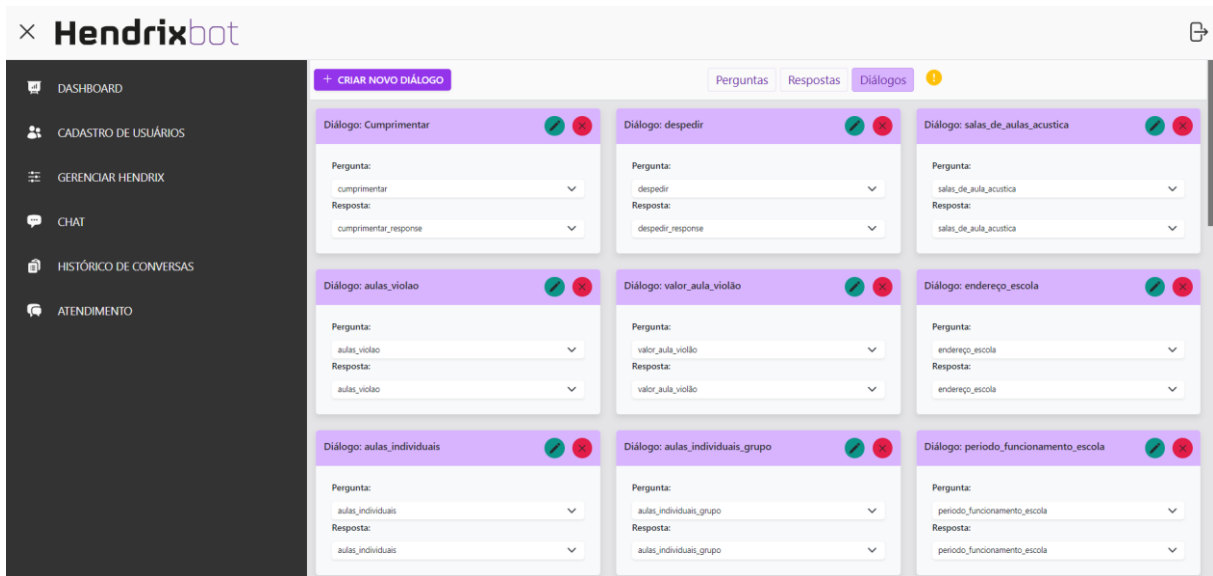
Boa noite! Em que posso te ajudar?

Boa tarde! Precisa de alguma ajuda?

Bom dia! Como você está?

Fonte – Elaborado pelos autores (2024).

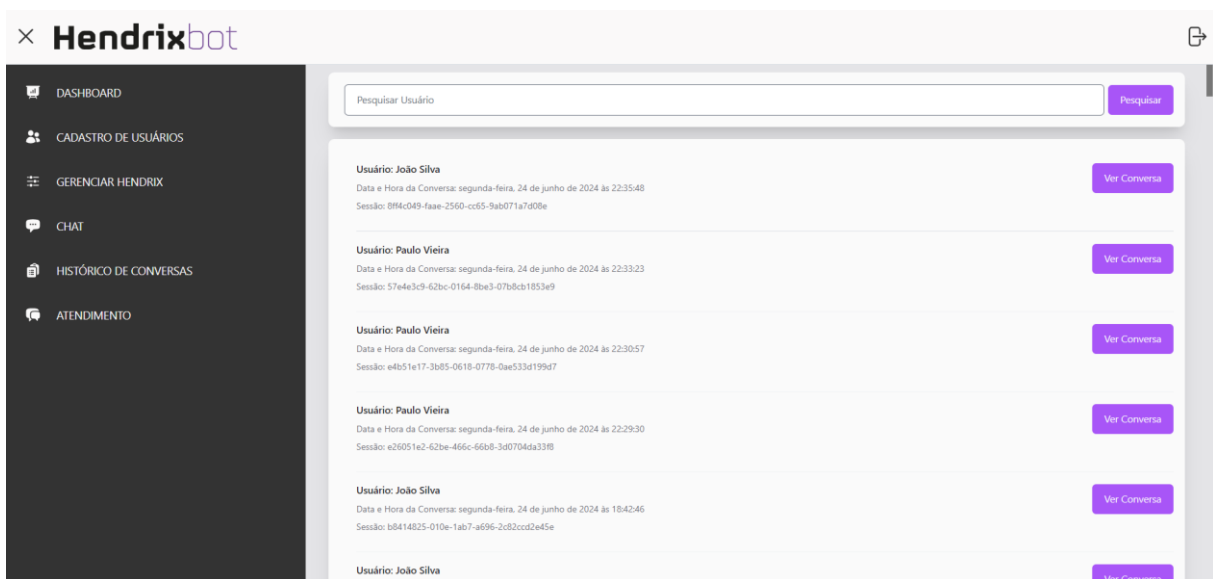
Figura 19 – Tela de Gerenciar Hendrix → CRUD



Fonte – Elaborado pelos autores (2024).

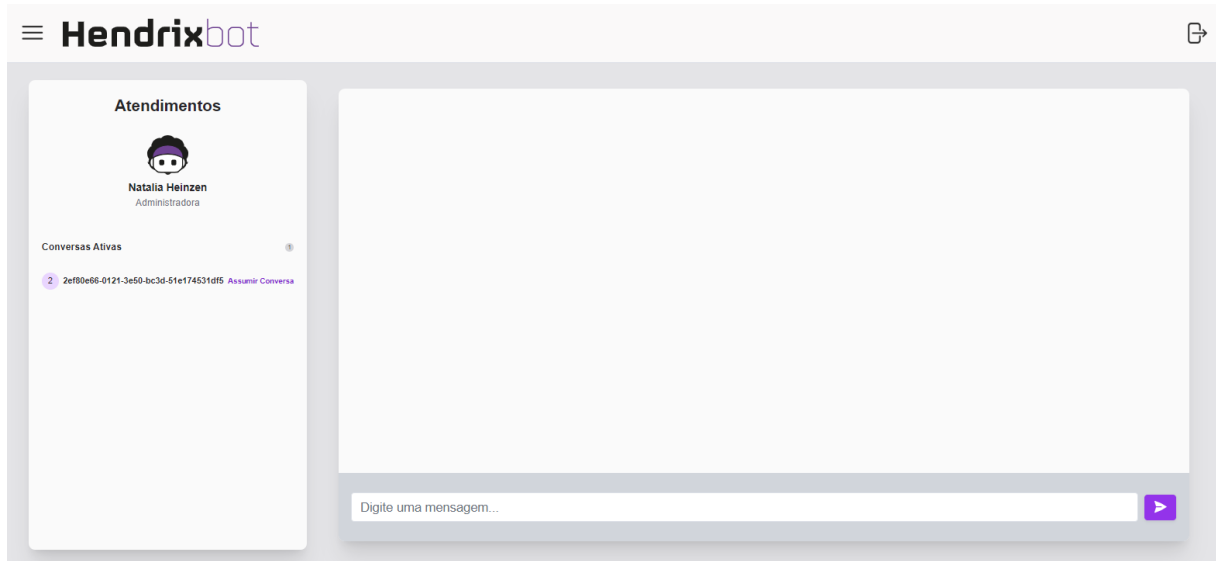
O Hendrix-bot oferece a possibilidade de o administrador visualizar o histórico de todas as conversas realizadas e filtrar por nome do usuário (Figura 20). Por fim, o sistema possibilita ao administrador, a visualização de todas as conversas que estão em andamento no *chatbot*, sejam elas de alunos cadastrados ou *leads*. Além disso, o usuário pode ou não assumir um dos diálogos em andamento para conversar diretamente com o usuário externo (aluno ou *lead*) conforme demonstrado na Figura 21.

Figura 20 – Tela de histórico de conversa



Fonte – Elaborado pelos autores (2024).

Figura 21 – Tela de atendimentos



Fonte – Elaborado pelos autores (2024).

6 CONCLUSÃO

O projeto Hendrix-bot, voltado para a automação de atendimento de uma escola de música, demonstrou uma série de vantagens significativas ao longo de seu desenvolvimento e testes. Principalmente, no que se refere ao alcance dos objetivos principais, proporcionando um sistema eficiente e customizável que atende às demandas específicas da escola. Além disso, a integração de tecnologias como Java, Spring Boot, Angular, MySQL e o framework RASA mostrou-se acertada, garantindo robustez, segurança e escalabilidade ao sistema.

Um dos pontos mais relevantes foi a capacidade do RASA em personalizar e contextualizar interações, oferecendo uma experiência de atendimento próxima à linguagem humana natural. Isso foi essencial para manter a proximidade e qualidade do atendimento desejadas pela escola. Ainda, a possibilidade de personalização dos contextos pelo próprio cliente assegura a longevidade e adaptabilidade do sistema, permitindo ajustes conforme novas demandas surgem.

Os resultados de desempenho apresentaram fortes evidências de escalabilidade para o produto, através do uso de tecnologias de alta qualidade e de uma interface intuitiva de configuração para que o cliente adicione personalizações aos atendimentos. A eficiência do chat, contudo, no que diz respeito às respostas esperadas, pode ter um crescimento ainda maior através da união de método de redes neurais em artefatos futuros.

Por fim, o Hendrix-bot provou ser uma solução escalável e aplicável a outros setores que necessitam de um sistema de atendimento automatizado e personalizável. As tecnologias e métodos utilizados garantem não apenas a eficiência operacional, mas também a capacidade de adaptação contínua às necessidades dos usuários. O sucesso do projeto demonstra o potencial da combinação de inteligência artificial com uma arquitetura de *software* bem planejada, oferecendo uma base sólida para futuros desenvolvimentos e expansões.

REFERÊNCIAS

- ANDREI L. **React Descomplicado: O Que é e Como Começar Já!**. 2023. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-react-javascript>. Acesso em: 13 abr. 2024.
- BALLESTERO-ALVAREZ, María Esmeralda. **Gestão de qualidade, produção e operações**. 2. ed. São Paulo, SP: Atlas, 2012.
- BARTIÉ, Alexandre. **Garantia da qualidade de software**. Rio de Janeiro: Elsevier, 2022.
- BATISTA, Nyanne. **Angular: o que é, para que serve e um Guia para iniciar no framework JavaScript**. 2024. Disponível em: <https://www.alura.com.br/artigos/angular-js>. Acesso em: 11 mai. 2024.
- CARVALHO, Vinicius. **MySQL: Comece com o principal banco de dados open source do mercado**. São Paulo: Casa do Código, 2020.
- CHIAVENATO, Idalberto. **Empreendedorismo: dando asas ao espírito empreendedor**. 4ª ed. Barueri: Manole, 2012.
- COD3R. **Introdução ao Flyway usando Spring Boot**. 2023. Disponível em: <https://blog.formacao.dev/introducao-ao-flyway-usando-spring-boot/>. Acesso em: 13 abr. 2024.
- COODESH. **O que é Spring Boot?**. 2024. Disponível em: <https://coodesh.com/blog/dicionario/o-que-e-spring-boot/#:~:text=O%20Spring%20Boot%20%C3%A9%20um,importante%20no%20ecosistema%20Java%20corporativo>. Acesso em: 08 abr. 2024.
- COSTA, Louise. **introdução: Conceitos Iniciais + Versionamento de Código**. 2022. Disponível em: <https://github.com/reprograma/on19-tet-s1-git>. Acesso em: 13 abr. 2024.
- CRUZ, Leoncio Teixeira. ALENCAR, Antônio Juarez. SCHMITZ, Eber Assis. **Assistentes Virtuais Inteligentes e Chatbots: Um guia prático e teórico sobre como criar experiências e recordações encantadoras para os clientes da sua empresa**. 1. ed. Rio de Janeiro: Brasport, 2019.
- FERREIRA, Francinara Silva. **Gestão Escolar na perspectiva democrática: estudo a partir dos Projetos de Intervenção desenvolvidos em escolas públicas da Região Oeste do Pará**. 2017. Dissertação (Mestrado em Educação) – Universidade Federal do Oeste do Pará, Santarém, 2017. Disponível em: https://www.ufopa.edu.br/ppge/images/dissertacoes/turma_2015/francinara_silva_ferreira.pdf. Acesso em: 06 abr. 2024.
- GADELHA, Igor Bruno Liz. **O uso de chatbots no atendimento de clientes de revenda por catálogo**. 2019. Dissertação (Pós-Graduação em Computação

Aplicada do Núcleo de Desenvolvimento da Amazônia em Engenharia) - Universidade Federal do Pará, Tucuruí, 2019. Disponível em: https://www.repositorio.ufpa.br/jspui/bitstream/2011/12417/1/Dissertacao_UsoChatbotsAtendimento.pdf. Acesso em: 23 mar. 2024.

GONÇASVES, Thiago. **PLN: o que é Processamento de Linguagem Natural?** 2023. Disponível em: https://www.alura.com.br/artigos/o-que-e-pln?utm_term=&utm_campaign=%5BSearch%5D+%5BPerformance%5D+-+Dynamic+Search+Ads+-+Artigos+e+Conte%C3%BAdos&utm_source=adwords&utm_medium=ppc&hsa_acc=7964138385&hsa_cam=11384329873&hsa_grp=111087461203&hsa_ad=687448474447&hsa_src=g&hsa_tgt=aud-409949667484:dsa-425656816943&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gad_source=1&gclid=CjwKCAjwuJ2xBhA3EiwAMVjkVNGrOu1iXgH8FN7JXNbxhhVp0D4bOs4V23mgVxKA8O2OoABFdSJGTxoCPnkQAvD_BwE. Acesso em: 29 abr. 2024.

GROSS, Marcos. **Dicas práticas de comunicação: boas ideias para os relacionamentos e os negócios.** 1ª ed. São Paulo: Trevisan, 2023. *E-book*. Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/212716>. Acesso em: 22 mar. 2024.

HORSTMANN, Cay S. CORNELL, Gary. **CoreJava.** 8ª ed. São Paulo: Pearson, 2009.

HTI, Tecnologia. **As principais funções MySQL.** 2022. Disponível em: <https://www.hti.com.br/blog-mobile/544-as-principais-funcoes-mysql#:~:text=com%20a%20HTI!-,Principais%20caracter%C3%ADsticas,para%20proporcionar%20melhorias%20ao%20usu%C3%A1rio..> Acesso em: 17 abr. 2024.

JIAO, Anran. An Intelligent Chatbot System Based on Entity Extraction Using RASA NLU and Neural Network. **Journal of Physics: Conf. Series 1487 (2020).** Nankai University, Tianjin, China 2020.

KERR, Eduardo Santos. **Gerenciamento de requisitos.** 1ª ed. São Paulo: Pearson, 2015. *E-book*. Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/22130>. Acesso em: 30 mar. 2024.

L3SOFTWARE. **JetBrains – IntelliJ IDEA.** 2024. Disponível em: <https://l3software.com.br/produto/jetbrains-intellij-idea/>. Acesso em: 08 abr. 2024.

LISSARASSA, Maurita Machado. **Gestão comercial: uma análise da coordenação comercial do jornal Zero Hora no interior do RS.** 2014. Dissertação (Mestrado) - Universidade Regional Integrada do Alto Uruguai e das Missões, Santo Ângelo, 2014. Disponível em: https://san.uri.br/mestrado_gestao/dissertacao/Dissertacao.Mauria.pdf. Acesso em: 06 abr. 2024.

LOCAWEB. **O que é e como funciona um Servidor de Hospedagem de Sites?.**

2024. Disponível em: <https://www.locaweb.com.br/conteudos/servidor-de-hospedagem/>. Acesso em: 08 abr. 2024.

MACHADO, Felipe Nerys Rodrigues. **Análise e Gestão de Requisitos de Software**: Onde Nascem os Sistemas. 3ª ed. São Paulo: Érica, 2016.

MAGALHÃES, Marcela Roméro. CASTRO, Rayssa Bastos de. **Avaliação do uso de Chatbots por parte das empresas como meio de atendimento ao consumidor**. 2019. Dissertação (Pós-graduação e Pesquisa de Engenharia) - Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2019. Disponível em: <https://www.crie.ufrj.br/assets/Centro-de-estudos/wida-04-chatbots-final-2.pdf>. Acesso em: 23 mar. 2024.

MEDEIROS, Luciano Frontino de, **Inteligência Artificial Aplicada: Uma abordagem Introdutória**. 1.ed. Curitiba: InterSaberes, 2018.

MUNIZ, Antônio *et al.* **Jornada Kanban na prática**: Unindo teoria e prática para acelerar o aprendizado para quem está iniciando. 1ª ed. Brasport, 2021. Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/187432>. Acesso em: 25 mar. 2024.

OLIVEIRA, Gabriel Riso. **Proposta de modelagem empresarial uml de planejamento e controle da produção e o desenvolvimento de um protótipo de ensino-aprendizagem**. 2019. Dissertação (Pós-Graduação em Engenharia de Produção) - Universidade Estadual Do Norte Fluminense Darcy Ribeiro, Campos dos Goytacazes, 2019. Disponível em: <https://uenf.br/posgraduacao/engenharia-de-producao/wp-content/uploads/sites/13/2019/07/Disserta%C3%A7%C3%A3o-Gabriel-Riso.pdf>. Acesso em: Acesso em: 01 abr. 2024.

PAULA FILHO, Wilson de Pádua. **Engenharia de Software**: Fundamentos, Métodos e Padrões. 3ª ed. Rio de Janeiro: LTC, 2015.

PHAM, Andrew. **Scrum em Ação**: Gerenciamento e Desenvolvimento Ágil de Projetos de Software. 1ª ed. São Paulo: Novatec, 2011.

POLO, Rodrigo Cantú. **Validação e teste de software**. 1ª ed. Curitiba: Contentus, 2020. Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/188186>. Acesso em: 17 abr. 2024.

PPLWARE. **Draw.IO** – Desenhar diagramas nunca foi tão fácil. 2015. Disponível em: <https://pplware.sapo.pt/internet/draw-io-desenhar-diagramas-nunca-foi-tao-facil/>. Acesso em: 08 abr. 2024.

PRESSMAN, Roger S; MAXIN, Bruce R. **Engenharia de Software**: uma abordagem profissional. 8ª ed. Porto Alegre: AMGH, 2016.

RASA. **The RasaMasterclass Handbook**. 2019. Disponível em: <https://info.rasa.com/masterclass-ebook>. Acesso em: 22 abr. 2024.

ROGERS, Yvonne; SHARP, Helen; PREECE, Jennifer. **Design de Interação**: além

da interação humano-computador. 3ª ed. Porto Alegre: Bookman, 2013.

SBROCCO, José Henrique Teixeira de Carvalho; MACEDO, Paulo Cesar de. **Metodologias Ágeis: Engenharia do Software sob medida.** 1ª ed. São Paulo: Ética, 2012.

SELEME, Robson. **Gestão de operações de serviços: planejando o sucesso no atendimento ao cliente.** 1ª ed. Curitiba: Intersaberes, 2016. Disponível em: <https://plataforma.bvirtual.com.br/Leitor/Publicacao/37469/pdf/0?code=BDi8VpbbJHN8jHDaIVFQggjpOW2LWbdZn+o/Hv2vPX9FpkVYbywlgMyWruv8GaL3nmM7m9BP2jl0HdHV+/Og5w==>. Acesso em: 23 set. 2024.

SHARMA, Rakesh Kumar; JOSHI, Manoj. An Analytical Study and Review of open Source Chatbot framework, RASA. **International Journal of Engineering Research & Technology (IJERT).** v. 9, p. 1011-1014, jun. 2020.

SINHORELLI, Cristiane Bartunek. **Gestão escolar: uma análise sobre o fazer do gestor no cotidiano escolar.** 2023. Dissertação (Mestrado em Educação) – Faculdade de Educação da Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2013. Disponível em: <https://tede2.pucrs.br/tede2/bitstream/tede/9386/2/Disserta%C3%A7%C3%A3o%20Cristiane%20Bartunek%20Sinhorelli.pdf>. Acesso em: 06 abr. 2024.

SOMMERVILLE, Ian. **Engenharia de Software.** 10ª ed. São Paulo: Pearson Education do Brasil, 2018.

SOUZA, Brunno Wagner Lemos de. **Convergência entre BPM-SOA e UML-SOA: uma análise comparativa de integração e de cenários.** 2016. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Pernambuco, Recife, 2016. Disponível em: https://repositorio.ufpe.br/bitstream/123456789/18365/1/Disserta%C3%A7%C3%A3o%20Final-Biblioteca_Brunno.pdf. Acesso em: 01 abr. 2024.

SOUZA, Valeria Ferreira de *et al.* A importância da qualidade do atendimento ao cliente. **Ricadi.** v.07. 1, ago/dez;2019. Disponível em: <https://urisaoluiz.com.br/site/wp-content/uploads/2020/01/Revista-7%C2%AA-edi%C3%A7%C3%A3o-1-Artigo-06.pdf>. Acesso em: 23 set. 2024.

TORQUATO, Gaudêncio. **Comunicação nas organizações: empresas privadas, instituições e setor público.** 1ª ed. São Paulo: Summus, 2015.

VALDATI, Aline de Brittos. **Inteligência Artificial - AI.** 1. ed. Curitiba: Contentus, 2020. *E-book.* Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/191624>. Acesso em: 18 mar 2024.

WHITELEY, Richar C. **A Empresa Totalmente Voltada Para O Cliente.** 27ª ed. Brasil: Campus, 1992.

Apêndice A – Entrevista semiestruturada

Entrevista - parte1

Olá, somos alunos da última fase do curso de Análise e Desenvolvimento de Sistemas (ADS) e estamos iniciando o projeto do nosso trabalho final. Foi nos solicitado para que desenvolvêssemos um sistema de ChatBot que facilite a comunicação entre alunos(as) e a coordenação de curso/secretaria, principalmente no que tange as perguntas frequentes, que são repetidamente respondidas pela instituição, dificultando os processos administrativos ou de melhorias.

Pensamos na possibilidade de aplicar esse sistema em um contexto de ensino não regular, como a School of Rock e visitando o site da rede, descobrimos que já existe um serviço similar, mas pouco personalizável e que não dá autonomia para o aluno (a)/futuro aluno (a) tirar dúvidas específicas.

Então ficamos curiosos em saber se a ideia de um ChatBot onde o aluno (a) possa de fato escrever as suas dúvidas e ser respondido diretamente pela unidade de ensino desejada, e de forma automatizada seria interessante para vocês.

Lembrando que, a ideia é implementar um ChatBot com inteligência artificial integrada, possibilitando a assertividade das respostas de forma cada vez mais ágil ao longo do tempo.

OBS: As perguntas numeradas como 1 referem-se ao cenário atual da escola, enquanto que as perguntas numeradas como 2 são sobre a criação do Chatbot.

Contato para tirar mais dúvidas e mandar sugestões: 48-991905708 Natália Heinzen

1.1 - De que forma ocorre atualmente a interação do aluno ativo com a coordenação da escola?

1 resposta

Principal canal é o WhatsApp com o frontdesk da escola, mediado por aplicativo Chat Bot, para gerenciar conversas com alunos e leais. Existem grupos também com os pais de alunos de determinados programas para que se passe informações de shows, eventos e ensaios. Também estamos implementando o Band para a comunicação mais segura e controlada, sem a exposição do contato dos alunos, para a interação entre alunos menores de idade e os instrutores de ensaios para trocará informações relativas aos ensaios.

1.2 - Quais dificuldades você enfrenta na comunicação atual com esses alunos?

1 resposta

A dificuldade maior é familiares leiam e respondam as mensagens de avisos importantes, como datas e confirmações em eventos e etc. Também ultimamente tem havido muitos contatos de novos leads que sobrecarregam a equipe da escola, que precisam de mais horas dedicadas apenas a responder mensagens.

1.3 - A maior parte da integração da secretaria de ensino ocorre principalmente com alunos ativos ou outros usuários?

1 resposta

A maior parte acontece com pessoas pedindo informações sobre a escola, chamamos de "leads", buscando marcar aulas experntais, consultando preços e programas. Mas isso é sazonal, varia com a época do ano.

Entrevista - parte 2

1.3 - A maior parte da integração da secretaria de ensino ocorre principalmente com alunos ativos ou outros usuários?

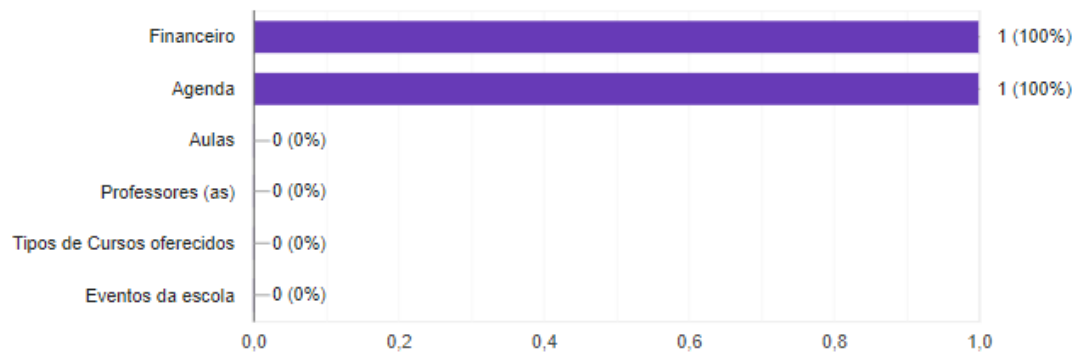
1 resposta

A maior parte acontece com pessoas pedindo informações sobre a escola, chamamos de "leads", buscando marcar aulas experntais, consultando preços e programas. Mas isso é sazonal, varia com a época do ano.

1.4 - Qual é a área específica da interação com a escola que gera a maior demanda de dúvidas por parte dos alunos ativos?

 Copiar

1 resposta



2.1 - Quais oportunidades você identifica nesse projeto para a escola?

1 resposta

Liberar a equipe para tarefas mais estratégicas, e melhorar o tempo de resposta para alunos e familiares.

2.2 - Quais oportunidades você identifica nesse projeto para os alunos?

1 resposta

Mais agilidade e conformidade nas respostas das dúvidas. Mas acredito, a principio, que a maior vantagem seria percebida pela equipe.

Entrevista - parte 3

2.3 - Como a empresa encara a implementação de novas ferramentas e tecnologias? A Rede costuma ser aberta para novas propostas e inovações?

1 resposta

Sim, especialmente para resolver questões que são tão latentes. Como é uma rede grande é sempre um desafio implementar novas ferramentas. Especialmente na parte didática, mas acredito que uma ferramenta de atendimento ao cliente é mais desejada e também mais fácil de implementar.

2.4 - Qual é a principal desvantagem que você percebe nesse projeto?

1 resposta

A desumanização dese processo, uma vez que a interação com a equipe preparada é carismática da escola é um fator importante da experiência dos clientes da escola, e acredito que um fator de fidelização. No entanto esse processo ocorre principalmente de forma presencial nas interações na sede e nos eventos da escola.

2.5 - A IA (Inteligência Artificial) é algo que desperta interesse nessa interação?

1 resposta

Sim. Mas como nossa escola não é tradicional e também exige um investimento maior do aluno, a gente já percebeu que para que às pessoas entendam nossa proposta e entendam porque ela custa o que custa, é indispensável que esteja presencialmente na escola, observe ensaios, a estrutura física e faça aulas. Portanto a dimensão da nossa entrega precisa da interação presencial e humana. Mas usar IA para resolver questões operacionais e pontuais seria ótimo, até para liberar a equipe para estar ainda mais plena na experiência da nossa escola.

2.6 - Como você se sentiria ao interagir com um chatbot para obter informações sobre aulas de música ou um novo serviço do seu interesse?

1 resposta

Não me incomodaria em obter informações preliminares sobre um serviço por um chatbot. Desde que uma vez que me interesse pelo serviço, ainda mais no caso de aulas de música e serviços de alto custo e de grande grau de personalização, passasse a ser atendido por pessoas em seguida.

Entrevista - parte 4

2.7 - Como você se sentiria ao interagir com um *chatbot* para resolver algum problema ou tirar uma dúvida num relacionamento de prestação de serviço ?

1 resposta

Eu não me sinto confortável interagindo com chatbot, mas talvez seja uma questão de tempo para me acostumar. Definitivamente não gosto de interagir quando existem opções limitadas de interação e sei de antemão que não vou conseguir a informação que estou procurando, vou ter que pedir um tempo com o robô até que me encaminhe para um atendente humano. Acredito que pode haver uma certa rejeição ao chatbot por parte dos alunos, e dos leads não acho que é interessante para a comunicação da experiência da escola a interação com o chatbot.

2.8 - Quais recursos ou funcionalidades você gostaria que o *chatbot* tivesse?

1 resposta

Confirmação de aulas. Desmarcar aulas, notificações e Confirmação de eventos. Agenda de eventos, ensaios. Atendimento inicial aos leads. Perguntas frequentes e informações sobre os programas, como proposta, carga-horária e currículos.

2.9 - O que você não gosta quando utiliza um *chatbot* para resolver algum problema ?

1 resposta

Que tenham opções limitadas de resposta e que não resolvam de fato os problemas. Não gosto de não ser atendido por pessoas serviços muito personalizados ou quanto existe muita personalização no serviço.