

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

**Hermes: Um CRM WhatsApp para Comunicação
em Massa Humanizada no Terceiro Setor**

Arthur Saddy Mograbi

PROJETO FINAL DE GRADUAÇÃO

CENTRO TÉCNICO CIENTÍFICO - CTC

DEPARTAMENTO DE INFORMÁTICA

Curso de Graduação em Ciência da Computação

Rio de Janeiro, Junho de 2025



Arthur Saddy Mograbi

**Hermes: Um CRM WhatsApp para
Comunicação em Massa Humanizada no
Terceiro Setor**

Relatório de Projeto Final, apresentado ao programa de Graduação do Departamento de Informática da PUC-Rio como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Greis Francly Mireya Silva Calpa

Rio de Janeiro
Junho de 2025

Agradecimentos

Agradeço primeiramente à minha orientadora, Greis Silva Calpa, por toda a paciência e instrução, bem como todo o corpo docente do Departamento de Informática da PUC-Rio por me ajudar a tornar o profissional que sou hoje.

Aos meus familiares, que me apoiaram em todo esse processo e sempre estiveram presentes.

À minha namorada, que é uma fonte de inspiração e carinho constante, além de corrigir incontáveis erros de ortografia que teria feito nesse texto.

Aos participantes que fizeram esse projeto possível e sempre se disponibilizaram para contribuir.

Resumo

Mograbi, Arthur Saddy. Silva Calpa, Greis Francy Mireya. Hermes: Um CRM WhatsApp para Comunicação em Massa Humanizada no Terceiro Setor. Rio de Janeiro, 2025. p. 44. Relatório de Projeto Final II - Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

Este projeto apresenta "Hermes", nomeado em homenagem do deus mensageiro grego, um CRM WhatsApp que viabiliza entidades do terceiro setor a conduzirem seus processos de comunicação de uma forma bidirecional, humanizada e transparente. Este projeto surge a partir das insuficiências dos produtos existentes e às dificuldades de usuários não especializados. Utilizou-se *análise de sentimento* e *LlamaIndex*, integradas a um projeto com foco na Experiência do Usuário (UX). Sua arquitetura inclui um website full-stack, um módulo para processamento de dados e um contêiner responsável pelos envios. O sistema foi validado com testes funcionais e através de entrevistas e testes de usabilidade. Os resultados mostraram a eficácia do projeto de UX e as contribuições geradas pelas tecnologias aplicadas.

Palavras-chave: Análise de Sentimento; UX; LlamaIndex; NLP; CRM

Abstract

Mograbi, Arthur Saddy. Silva Calpa, Greis Francy Mireya. Hermes: A WhatsApp CRM for Humanized Mass Communication in the Third Sector. Rio de Janeiro, 2025. p. 44. Final Project Report II - Informatics Department. Pontifícia Universidade Católica do Rio de Janeiro.

This project presents 'Hermes', named after the Greek messenger god, a WhatsApp CRM that enables third-sector organizations to conduct their communication processes in a bidirectional, humanized, and transparent way. This project arises from the insufficiencies of existing products and the difficulties faced by non-specialized users. It employs *sentiment analysis* and *LlamaIndex*, integrated into a project focused on User Experience (UX). Its architecture includes a full-stack website, a module for data processing, and a container responsible for message sending. The system was validated with functional tests, as well as through interviews and usability testing. The results demonstrated the effectiveness of the UX design and the contributions generated by the applied technologies

Keywords: Sentiment Analysis; UX; LlamaIndex; NLP; CRM

Sumário

1	Introdução	1
2	Situação Atual	3
3	Objetivos	5
4	Atividades Realizadas	6
4.1	Automação Inicial e Limitações identificadas	6
4.2	Transição para <i>whatsapp-web.js</i>	8
4.3	Escalabilidade com Docker	8
4.4	Desenvolvimento da Interface Web	9
4.5	Seleção de Modelos de IA	12
4.6	Investigação sobre Hospedagem	15
4.7	Cronogramas de Atividades	16
5	Projeto e especificação do sistema	18
5.1	Estrutura do Sistema	18
5.2	Contribuições	20
6	Implementação e avaliação	22
6.1	Testes Funcionais	22
6.2	Testes com usuários	22
6.3	Comentários sobre a implementação	28
7	Considerações finais	31
7.1	Aprendizados	32
7.2	Futuros Trabalhos	33
	Referencias	34
	Apêndice A Diagrama ER da Base de Dados	37
	Apêndice B Diagrama estrutura API	38
	Apêndice C Entrevista Preliminar	39
	Apêndice D Teste de Usabilidade	42
	Apêndice E Entrevista Pós-Teste	43
	Apêndice F Wireframes	44
	Apêndice G Termo de Consentimento Livre e Esclarecido (TCLE)	44

1 Introdução

De acordo com a pesquisa Panorama Mobile Time/Opinion Box de Março 2024 [28], 98% dos smartphones do Brasil possuem WhatsApp instalado. Com estimados 168 milhões de usuários brasileiros, é natural que empresas e organizações, especialmente as de pequeno e médio porte, utilizem esse para comunicação com seus clientes e contatos [23]. Existem, atualmente, dois principais tipos de soluções tecnológicas para facilitar essa comunicação: softwares de CRM (*Customer Relationship Management*) e chatbots.

Embora os chatbots apresentem um alto grau de automatização, a pesquisa Panorama Mobile Time/Opinion Box [28] indica que, dos 89% dos usuários do aplicativo que interagiram com chatbots ou agentes virtuais, a maioria demonstrou baixa satisfação, com 62% dos entrevistados manifestando insatisfação ou neutralidade sobre a experiência de uso.

Em contraponto, CRMs funcionam como uma espécie de chat com ferramentas, permitindo que um humano organize e responda as mensagens de uma forma mais eficiente. No entanto, pela escala do contato e pela natureza focada em venda e leads da maioria dos softwares existentes, torna-se difícil estabelecer um contato mais profundo com qualquer única pessoa. Para organizações do terceiro setor, como ONGs e organizações políticas, essa questão se torna ainda mais crítica. Mais do que em um cenário comercial, essas organizações precisam estabelecer um contato bidirecional com seus contatos, para verdadeiramente engajá-los e ouvi-los. No entanto, como essas organizações geralmente possuem recursos limitados dedicados à comunicação, torna-se difícil conseguir ouvir a todos sem uma ferramenta especializada.

Portanto, este projeto propõe e implementa a construção de uma ferramenta CRM para o WhatsApp, com foco nas necessidades de organizações não comerciais com o intuito de permitir uma comunicação humanizada. Isto é, uma comunicação que priorize o contato autêntico, a criação de laços de confiança e que ambos os lados consigam efetivamente escutar e entender uns aos outros. Para isto, além de englobar funcionalidades básicas de CRM como envio em massa de mensagens, é utilizado sentiment analysis, *LlamaIndex* e visualizações de dados real-time para empoderar essas instituições a operar e dialogar em um nível maior. Em um nível técnico, esses são usados, respectivamente, para o destaque visual e monitoramento das mensagens recebidas, a geração de resumos de conversas e *insights*, e painéis orientadores de operação.

Adicionalmente, em uma aplicação sensível como essa, é fundamental considerar questões éticas e legais. Apesar da comunicação em larga escala pelo WhatsApp ter o potencial de trazer informação e participação para muitos que não tinham, observa-se que já foi muito usado para fins como disseminação de notícias falsas e disparos em massa sem consentimento prévio [13]. Assim, é importante garantir que não só o projeto esteja de acordo com as normas vigentes, mas que seu uso seja alinhado com essas. Além de estabelecer termos de uso próprios para tal, espera-se

prover uma visão transparente para o usuário de quem quer e não quer receber suas mensagens.

O software da ferramenta proposta é dividido em três partes principais: 1) um container docker que efetivamente envia e captura as mensagens, 2) um website que serve de interface para o sistema e 3) um módulo para processamento das mensagens recebidas e dados coletados. O container hospeda um servidor de Node.js que irá realizar ações à medida que requisitadas pelo backend do site através de seu API. Ele utiliza de uma biblioteca open-source (whatsapp-web.js) para interagir com o WhatsApp, que usa o puppeteer para manipular o WhatsApp web. No entanto, o uso dessa biblioteca é temporário, pois a Meta, empresa gestora do WhatsApp, possui um API oficial para controle do aplicativo. Apesar disso, o acesso efetivo a esse API requer a aprovação em uma lista de espera de *Solution Partners* [10]. Dessa forma, este método não-oficial foi implementado como uma solução provisória para viabilizar a entrega de um produto funcional.

Por outro lado, o website foi desenvolvido com uma arquitetura que integra React no front-end, Node.js e Express no backend e MySQL como base de dados. Essa configuração permite o usuário a realizar ações como envio de mensagens em massa, gerir a base de contatos e organizar o chat de acordo a prioridade, quanto visualizar insumos e gráficos gerados. Para o front-end, foi utilizado em larga escala componentes da biblioteca MUI [21], que implementa o Material Design do Google em React.

Por último, o processamento de mensagens e dados é feito em um módulo Python, que aplica um modelo de análise de sentimento às mensagens recebidas e utiliza o *LlamaIndex*, um framework de orquestração de dados para organizar dados não estruturados, como históricos de conversas em índices otimizados, permitindo alimentar Modelos de Linguagem de Grande Escala (LLMs) de forma mais eficaz, a fim de gerar resumos dos aspectos relevantes das conversas e *insights* qualitativos.

Este projeto se encaixa nos requisitos de Projeto Final II por inovar na aplicação de técnicas de estado da arte em um subdomínio de software de comunicação pouco explorado. Além disso, o projeto demanda a aplicação de um conhecimento multidisciplinar de áreas como: desenvolvimento web, gerenciamento de base de dados, ciência de dados e UX (Experiência do Usuário).

Na Seção 2, são descritas soluções similares à apresentada neste estudo, comparando suas características. Será descrito também como alguns dos fatores mencionados, como o foco no processo de comunicação humanizada, aplicação de tecnologias estado da arte e projeto de UX simplificado interagem para transformar o projeto em uma ferramenta que pode contribuir positivamente a situação da área. Na Seção 3, são apresentados os objetivos deste estudo. Na Seção 4, são descritas as atividades realizadas ao longo do desenvolvimento do projeto. Depois, na seção 5, são descritos o projeto e especificação do sistema e, posteriormente, na seção 6, descreve-se o processo de avaliação. Finalmente, na seção 7, apresentam-se as considerações finais.

2 Situação Atual

Esta seção descreve o estado atual de softwares de comunicação com focos semelhantes, a aplicação de tecnologias relevantes dentro desses softwares e como o Hermes se distingue daquilo que existe.

Apesar de haver diversas empresas com produtos similares no mercado, a ferramenta proposta neste projeto se diferencia em três pontos principais: eliminação de barreiras técnicas, uso de tecnologia estado da arte e no processo de comunicação com o cliente, como descrito abaixo.

1. Acessibilidade: grande parte dos clientes prospectivos atualmente contratam não um software diretamente, mas agências intermediárias, como a BaseLab [4] e Flex [14], que operam produtos de outras empresas. Isso se deve à falta de quadros técnicos de profissionais da área e produtos pouco acessíveis àqueles que não têm esses quadros. Este trabalho, propõe, portanto, usar o feedback desses próprios profissionais para a construção de uma solução que pode ser utilizado por eles de forma simples e acessível. Além de melhorias de interface e UX, pretendem-se incluir features específicos, como remover necessidade de manter, exportar e formatar planilhas para alimentar o sistema, a fim de simplificar o uso ainda mais. Além disso, ao eliminar a dependência de serviços terceiros, a solução oferece uma vantagem financeira sobre alguns outros produtos da área.
2. Uso de Tecnologia Estado da Arte: Alguns dos produtos existentes, como a ByteBio [6], integram técnicas estado da arte, por exemplo o uso de LLMs na resposta de mensagens; apesar disso, não usam ferramentas mais prospectivas como o *sentiment analysis* e *LlamaIndex* que este trabalho propõe. Nos únicos produtos encontrados em que a integração dessas ferramentas era possível, como é o caso do Twilio [30], tal integração precisava da produção de código, que violaria a acessibilidade aqui proposta. Além do qual, mesmo no caso do Twilio [30] e outros sistemas de CRM extensíveis, como Microsoft Dynamics 365 CRM [17], estes não permitem o uso dos modelos propostos integrados à visualização das mensagens por humanos.
3. Foco no processo de comunicação: Grande parte dos CRMs existentes são focados em cenários comerciais, com várias ferramentas para captação de leads, emissão de cobranças e boletos, mas pouco focados no processo de comunicação com o cliente em si. A proposta neste trabalho é focar não somente em cenários não-comerciais, mas, dentro desse nicho, cenários não-comerciais no qual o cultivo da relação e construção de comunidade seja de fato a prioridade.

Outros produtos focados no mesmo público (do terceiro setor), contudo, tendem a ser direcionados para coletar e gerenciar doações, como é o caso da Zoho [31] e Elegis [12], ou gerenciamento de gabinete parlamentar como ConectaGabinete [7]. Outros CRMs especializados, como é o caso da TrackMob [29] ou CRMGabinete [8],

não possuem integração com WhatsApp. A confluência desses fatores pode gerar um produto único, provendo capacidade novas aos profissionais da área. Isso é ilustrado pelas Tabelas 1 e 2, que lista as características de softwares semelhantes.

Nome do Produto	Integração WhatsApp	Nenhum Conhecimento Técnico Necessário Para Uso	Especialização Não Comercial
BaseLab	Sim	Não	Não
Flex	Sim	Não	Não
ByteBio	Sim	Sim	Não
Twilio	Sim	Não	Não
Microsoft Dynamics 365	Sim	Não	Não
Zoho	Sim	Sim	Sim
Elegis	Sim	Sim	Sim
ConectaGabinete	Sim	Sim	Sim
TrackMob	Não	Sim	Sim
CRMGabinete	Não	Sim	Sim

Tabela 1: Comparação de características de softwares semelhantes (Parte 1)

Nome do Produto	Uso de Sentiment Analysis	Produto Próprio	Focado em Engajamento
BaseLab	Não	Não	Não
Flex	Não	Não	Não
ByteBio	Não	Não	Não
Twilio	Sim	Sim	Não
Microsoft Dynamics 365	Sim	Sim	Não
Zoho	Não	Sim	Não
Elegis	Não	Sim	Não
ConectaGabinete	Não	Sim	Não
TrackMob	Não	Não	Não
CRMGabinete	Não	Não	Não

Tabela 2: Comparação de características de softwares semelhantes (Parte 2)

Anteriormente ao começo desse projeto, foi realizada uma série de entrevistas informais com profissionais da área, que determinaram a direção e estrutura geral do desenvolvimento. Mas, para garantir que o projeto melhorasse concretamente a situação atual dos produtos oferecidos, realizou-se adicionalmente duas entrevistas e um teste de usabilidade da interface.

A primeira entrevista serviu tanto para formalizar os pontos elencados na pré-entrevista, quanto para investigar outras oportunidades de melhoria e prioridades de implementação. Ao término do desenvolvimento do website, foram realizados testes de interface, focados também no princípio da eliminação de barreiras técnicas mencionado, seguidos por entrevistas avaliativas sobre o que foi desenvolvido. Ambas as entrevistas foram realizadas num formato semi-estruturado, com um roteiro que

serviu de base, mas não que não restringiu rigidamente o seu conteúdo, permitindo levantar informações relevantes que surgiram durante a entrevista.

Apesar desses testes não conseguirem prover uma avaliação objetiva sobre as melhorias alcançadas, espera-se que possam ao menos validar a utilidade das tecnologias novas implementadas, e prover um teste a partir de profissionais que trabalham e trabalharam com softwares similares.

3 Objetivos

Esta seção apresenta os objetivos principais do trabalho desenvolvido, bem como o escopo do sistema, incluindo os usuários finais que pretende apoiar e a contribuição pretendida no estado da arte.

O objetivo é desenvolver uma ferramenta de comunicação que não somente seja um bom exemplo de um software CRM, mas que também use tecnologia estado da arte integrada aos processos para trazer vantagens novas aos usuários. Tendo isso em vista, espera-se gerar um produto que melhore as capacidades comunicativas de organizações da área.

Uma das inovações importantes é uso de uma aplicação de *sentiment analysis* misto, com o modelo *tabularisai/multilingual-sentiment-analysis* [1] (uma versão do distilbert) e um dicionário de palavras customizado ao cenário de aplicação. Essa aplicação é inspirada no review da literatura de Mao et al (2024) [20], usando a especificidade de domínio de abordagem de dicionários e o *awareness* de contexto de modelos BERT.

Inclui-se também o LlamaIndex, uma framework de orquestração de dados, que possibilita alimentar uma LLM ou outros modelos textuais com os dados gerados a partir dos chats do CRM de uma forma mais estruturada e eficiente. Esse framework foi utilizado para duas principais tarefas; a criação de resumos de conversas longas e a extração de insights qualitativos. A primeira tarefa, que irá gerar resumos das conversas periodicamente, poupa o usuário de ter que passar por meses de conversa para entender a história de um único contato e prover uma resposta informada. Irá ser descrito a investigação feita acerca da performance de dois principais modelos: o uso do modelo do SimCLS treinado na base CNNDM [18], e o GPT 3.5 turbo através da API da OpenAI.

Outra feature identificada como essencial é a segmentação de contatos, para pessoas com características específicas receberem tratamentos diferentes de acordo com a estratégia de comunicação de cada usuário. Integrou-se as métricas geradas pelos outros processos do sistema à visualização de cada segmento, possibilitando aos usuários analisarem sua performance com cada grupo e ajustarem suas estratégias de acordo.

Em concordância com essa customização, é necessário incluir a personalização dos envios feitos para cada contato, modificando a mensagem original com o nome, local de residência ou qualquer outra informação específica. Para evitar o uso de tokens de formatação, como %nome% que podem ser confusos para usuários sem

formação técnica, todas as informações variáveis das mensagens serão introduzidas diretamente a partir da interface de uma forma clara e simples.

Também foi implementada uma dashboard de insights e visualização de performance geral, cujo objetivo é resumir de forma concisa e humana o estado da operação naquele momento. Espera-se, através disso, conseguir ilustrar de forma mais qualitativa e aprofundada as características reais da comunicação que o usuário está conduzindo naquele momento.

Esse projeto é focado nas necessidades de profissionais de comunicação e voluntários trabalhando no terceiro setor. Esses trabalhadores, na operação do WhatsApp, costumam ser divididos em duas categorias: dos responsáveis pelo responsivo, aqueles cuja principal tarefa é de fato responder as mensagens, e dos gerentes de equipe, responsáveis pela estratégia geral da equipe e de redigir as mensagens dos envios. Pretende-se apoiar ambas categorias, com os envios customizáveis, gerenciamento de contatos e o dashboard de dados para os gerentes e o chat incrementado com modelos de resumo/LlamaIndex e o sentiment analysis para os profissionais de responsivo.

4 Atividades Realizadas

Essa seção descreve as principais atividades realizadas ao longo do desenvolvimento ao longo do projeto, bem como os protótipos criados durante tal. Essas atividades foram divididas em 6 principais eixos:

1. Automação Inicial e Limitações.
2. Transição para *whatsapp-web.js*
3. Escalabilidade com Docker
4. Desenvolvimento da Interface Web
5. Seleção de Modelos de IA
6. Investigação sobre Hospedagem

4.1 Automação Inicial e Limitações identificadas

Minha interação com a área de comunicação via WhatsApp teve início no ano 2020, quando fui convidado para operar um software existente. Este convite ocorreu devido à falta de conhecimento técnico na equipe para formatar os arquivos .csv necessários para operação do software que tinham contratado. Desde então, foi percebida uma oportunidade de grande melhoria na operação da área.

Minhas primeiras contribuições foram pequenas ferramentas, que virariam depois sub-funcionalidades de partes do projeto. Foram implementadas scripts de Python para formatar os arquivos .csv e rotinas do Google Apps Script [9] integradas

às planilhas de dados da organização, responsáveis por automatizar tarefas de rotina previamente realizadas manualmente.

Foi investigado também a possibilidade do uso de diversas técnicas, sejam de Computer Vision, como o OCR do OpenCV, ou de modelos de image-to-text mais recentes. No entanto, essa funcionalidade foi determinada como inviável devido à criticidade da aplicação; errar o nome de alguém no primeiro contato pode ser um erro grave e mandar mensagem para o número errado ainda mais, além da qualidade dos dados fontes, que costumam ser imagens de baixa resolução com campos preenchidos incorretamente, dados escritos fora das linhas e outros fatores que dificultam um resultado consistente.

Inicialmente, as tentativas de automatizar o envio por conta própria foram rudimentares, com o uso de scripts de Python que usavam o pynput [26] para controlar o cursor e entrada de texto do computador, navegando manualmente através da interface do WhatsApp Web. No entanto, essa abordagem apresentava três principais limitações:

1. Consistência: Devido a abordagem de manipulação direta da interface, a implementação sofria de questões graves de consistência. Pequenos detalhes, como o tamanho da tela utilizada, a velocidade do computador e o browser utilizado causavam erros críticos. Foi feita uma tentativa de mitigar esses problemas por meio de rotinas de calibração para computadores diferentes e o uso de OCR para reconhecimento de elementos de interface, mas estes nunca chegaram a fundamentalmente mudar a situação.
2. Uso de Recursos: Pela abordagem capturar a interface do computador, este ficava inutilizável durante o processo de envio, que às vezes demorava horas para terminar. Ademais, era necessário um computador para cada número que iria enviar mensagens simultaneamente.
3. Limitações Funcionais: Esse método possuía uma série de limitações funcionais que causavam uma grande perda de tempo e performance na operação. Por exemplo, para achar um contato específico e então mandar mensagem, o programa clicava na barra de pesquisa do WhatsApp Web e digitava o número completo dele. No entanto, isso necessita que o contato já esteja salvo no celular que possui o chip utilizado, que por sua vez necessitava a criação e importação de arquivos vCard e aguardar até que o WhatsApp registrasse os contatos novos. Ademais, essa abordagem apenas permitia o envio de texto, limitando severamente a possibilidade de conteúdo a ser usado.

Diante disso, foi identificado a necessidade do uso de uma técnica diferente. Foi estudado o uso de técnicas de web-scraping para tal, mas isso foi dificultado pela estrutura da interface web do WhatsApp, que possui um nível alto de ofuscação na página deles.

4.2 Transição para *whatsapp-web.js*

O módulo *whatsapp-web.js* foi utilizado para solucionar algumas das dificuldades encontradas anteriormente, resultando em alguns estudos feitos no decorrer da aplicação desse módulo e na análise de algumas limitações ainda presentes na implementação inicial dele.

A descoberta inicial desse módulo, e um grande passo adiante no projeto, foi feita em 2022, quando fui novamente chamado para trabalhar com uma ferramenta semelhante. Essa ferramenta tinha o formato de uma extensão web e havia sido distribuída para organização que me contratou no formato do pacote de javascript, o qual formava a extensão. Apesar de não ter muita experiência com essa linguagem, foi possível identificar que o produto se baseava no uso da biblioteca *whatsapp-web.js*.

Ao estudar, então, essa biblioteca, foi descoberto como usá-la de forma muito mais extensa do que o observado nos softwares anteriormente operados. Por meio do API fornecido, é possível acessar a grande maioria das funcionalidades do WhatsApp e a maior parte dos dados gerados por ele. Isso permitia não somente enviar tipos de conteúdo que não era possível antes, como áudios, enquetes e arquivos, mas também capturar dados e utilizá-los para orientar as decisões feitas pela equipe.

Em seguida, foi feito um estudo do puppeteer [25], a biblioteca utilizada pelo *whatsapp-web.js*, para manipulação do navegador. Foi possível, deste modo, implementar algumas funcionalidades quanto ao armazenamento e restauração de sessões, bem como a configuração do browser para o processo de envio.

Com essas técnicas aprendidas, em seguida foi desenvolvido um protótipo muito mais robusto que o anterior. Este era uma ferramenta de console que recebia um arquivo .csv, automaticamente formatava-o, fazia o envio de mensagens, segundo os parâmetros escolhidos, e registrava as mensagens recebidas para o futuro uso em visualizações de dados.

No entanto, esse protótipo tinha duas principais limitações: a escalabilidade e a usabilidade. Quanto à escalabilidade, pelo puppeteer controlar um browser no ambiente local e não-encapsulado do usuário, era somente possível manter uma sessão por vez por computador e, para cada computador, era necessário um processo de instalação e configuração lento. Quanto à usabilidade, por ser uma ferramenta de console, necessitava educar os usuários sobre o uso básico da linha de comando, e precisava da memorização de diversos parâmetros, dificultando a adoção por um usuário médio.

4.3 Escalabilidade com Docker

Em relação à questão da escalabilidade, após pesquisar alternativas, foi decidido o uso do Docker [11], uma ferramenta open-source para containerização de aplicações. Apesar de, inicialmente, ter pouca familiaridade com a tecnologia, suas vantagens a tornaram crucial para a implementação do projeto. O uso desse permite, por isolar cada processo, contornar o limite de sessões por dispositivo, mas também possibilita o uso de técnicas para execução e orquestração desses contain-

ers. Isso abre o caminho para um gerenciamento muito mais eficiente e organizado da hospedagem, possibilitando uma escalabilidade dinâmica (usando de Kubernetes ou o Elastic Container Service da AWS), bem como load balancing ou alocação controlada de recursos.

No entanto, na situação temporária no qual o projeto encontra, de usar o *whatsapp-web.js*, não é recomendável utilizar o modo headless, pois esse leva frequentemente ao bloqueio da conexão por parte do WhatsApp. No entanto, containers Docker por padrão não possuem acesso ao servidor X11 do sistema (servidor de display responsável por gerenciar o output gráfico). Seria possível configurar o *Dockerfile* (arquivo que define o ambiente e setup do container), para que se estabelecesse um *port-forwarding* para o servidor X11, no entanto, o consumo de recursos de tal, especialmente com várias instâncias rodando, seria excessivo e precisaria de um host com um display físico. Por isso, optou-se pelo uso do Xvfb (X Virtual Framebuffer), que simula um display virtual, permitindo a utilização do modo gráfico do browser mesmo em um ambiente headless.

4.4 Desenvolvimento da Interface Web

Nesse item, será tratado a escolha pela construção de uma interface web e o processo de escolha de tecnologias, bem como os princípios orientadores do projeto de UX e alguns exemplos da aplicação de tais.

Referente à questão da usabilidade, diante do perfil do usuário alvo, ficou evidente que era necessário o desenvolvimento de uma interface gráfica simples e comunicativa para o uso amplo. Por questões de modelo e escalabilidade de negócio, foi decidido utilizar um web site para hospedar essa. Após a escolha, também foi decidido incluir a base de contatos de cada usuário dentro desse site, a fim de: eliminar a necessidade dos usuários terem que lidar com exportação de contatos, diminuir o tempo de cada fluxo de uso e coletar dados úteis sobre os contatos.

Diante disso, tornou-se necessário escolher um framework web a ser utilizado. O framework com o qual tinha mais experiência era Django, mas algumas limitações o tornaram não ideal para o projeto comparado a alguns stacks web mais completos. Devido a uma série de fatores como a abrangência do ecossistema node, o uso prévio de node no projeto, e a maior adaptação de node a ao uso em websockets e aplicações realtime, foi decidido usá-lo (junto ao Express.js) no backend.

Para o frontend, foi escolhido o React por ser uma aplicação que requer um state-management relativamente complexo, o foco do React em uma arquitetura de componentes reutilizáveis e seu uso comum junto ao Node, que facilitou o processo de aprendizado dessa biblioteca nova. Para orientar o desenvolvimento e as escolhas de UX do frontend, foram feitas uma série de wireframes (Apêndice F), das telas a serem implementadas. O design dos wireframes foi, orientado pelas dez heurísticas de usabilidade do Nielsen [22]. Essas heurísticas, princípios gerais para projetar interações e interfaces, guiaram várias das escolhas de UX, como a heurística de prevenção de erro, levando a implementação de botões desabilitados até todos seus

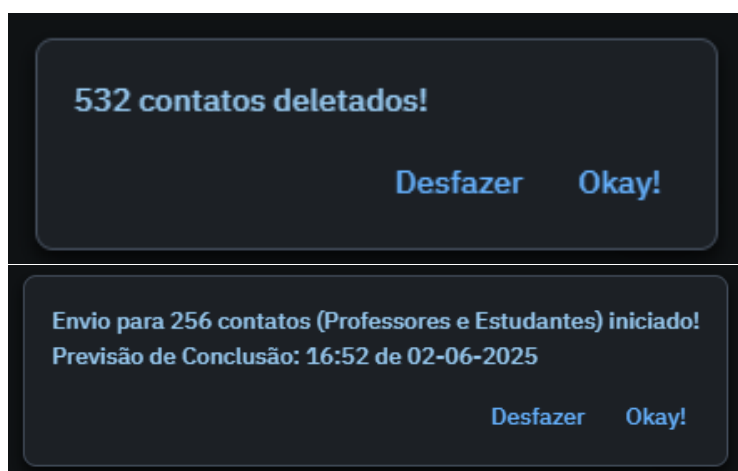


Figura 1: Exemplo de notificações de snackbar

requisitos serem cumpridos; popups de confirmação com informações sobre a ação planejada e conscientização do usuário sobre certas ações antes de possibilitá-lo de realizá-las.

Adicionalmente, os princípios de **visibilidade do estado do sistema** e **Reconhecimento ao invés de memorização** inspiraram a implementação de um sistema de notificações snackbar, visíveis na Figura 1 que, tanto informam sobre as ações realizadas, quanto retornam informações chaves sobre os processos em progresso, além da decisão de manter informação relevante de passos anteriores na tela ao lidar com fluxos com múltiplas etapas.

Dado o perfil dos usuários e um dos objetivos do projeto ser criar uma interface acessível para as funcionalidades desejadas, foi feito um estudo de Teoria de Carga Cognitiva (CLT) [27] para melhor dispor as informações apresentadas pela interface e transformar informações complexas em algo mais simples. Analisando o projeto através da lente do CLT, percebeu-se que requer fluxos com carga cognitiva intrínseca, aquela em relação à complexidade inerente da tarefa, relativamente alto. No entanto, podem ser tomadas medidas para reduzir ao máximo a carga estranha, aquela relacionada a forma que a informação é apresentada ou o fluxo é organizado, por meio da aplicação de alguns princípios apresentados na teoria, como o princípio da segmentação, que sugerem quebrar tarefas complexas em passos menores e menos demandantes. Outro desses princípios é o da contiguidade, que orienta a posicionar elementos abstratamente próximos também de uma forma próxima, tanto no espaço quanto no tempo.

Dentro também do framework do CLT, deve-se considerar a carga cognitiva germânica, uma carga considerada positiva e relacionada à aprendizagem e criação de modelos internos sobre as tarefas. É necessário ao máximo incentivar essa, por meio tanto de dar feedback constante sobre o status das tarefas e as consequências das ações dos usuários, alinhado com o princípio de visibilidade de Nielsen, quanto nas visualizações de dados, através de insights qualitativos ao invés de somente gráficos.

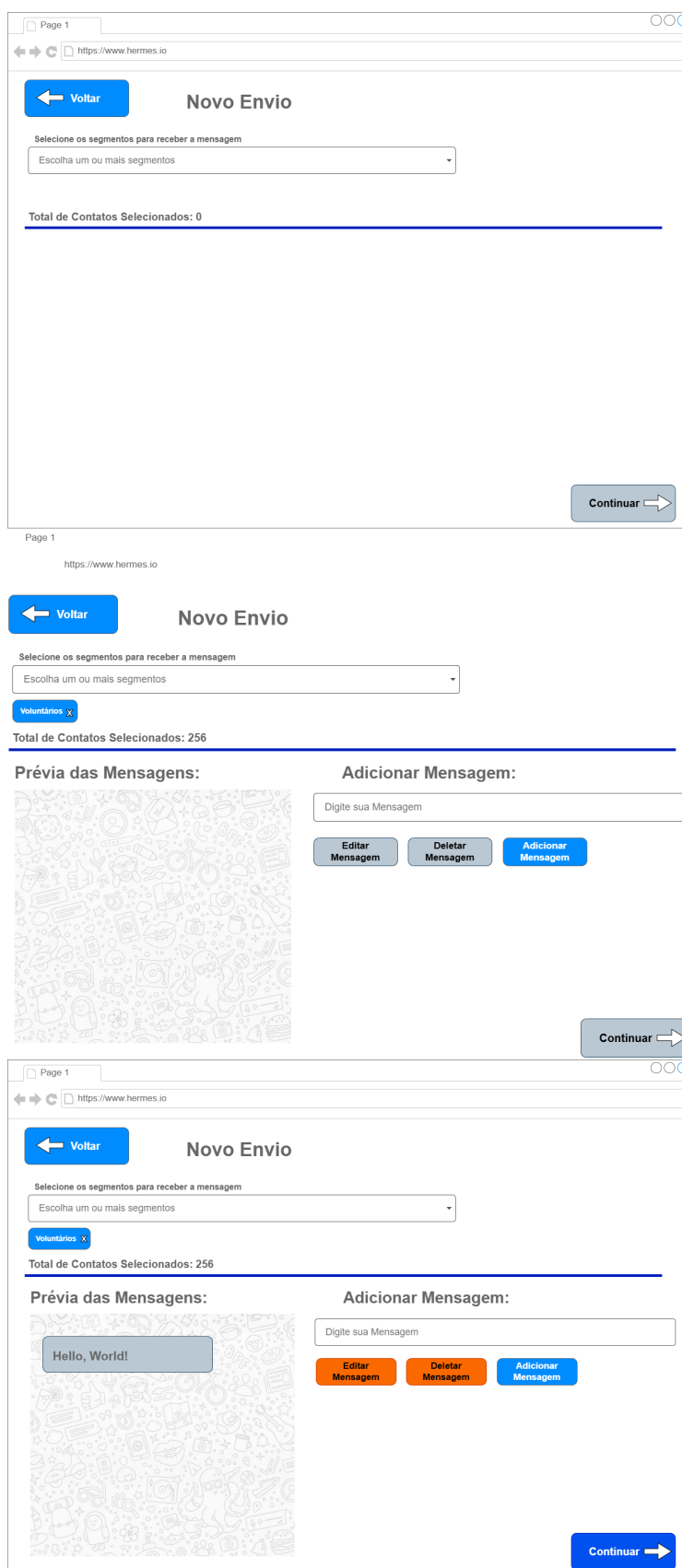


Figura 2: Exemplo do Fluxo de Novo Envio nos Wireframes, ajustado aos princípios de UX adotados. A Figura acima mostra a tela antes da seleção de ao menos um segmento válido, a do meio a tela após essa seleção e a de baixo a tela após a inserção de uma mensagem.

Pode-se observar, na Figura 2, um exemplo de como alguns desses princípios foram adotados nos wireframes. Diante do princípio da segmentação, o fluxo foi primeiramente dividido em três etapas: seleção do público alvo, adição de mensagens novas e edição e confirmação das mensagens. Apesar de só habilitar cada etapa subsequente após a prévia ser concluída, a informação das escolhas anteriores é mantida na tela e editável.

Por fim, apesar de MongoDB ser mais comumente usado junto ao resto das tecnologias escolhidas em uma stack MERN, para a base de dados foi escolhido o MySQLServer. Essa escolha foi feita pois os dados dos usuários que serão armazenados no sistema, como os contatos e segmentos, se prestam bem para o uso de tabelas relacionais mais clássicas, em contraste com a abordagem orientada a documentos do Mongo.

4.5 Seleção de Modelos de IA

Nessa seção, serão explorados os testes realizados para a escolha e validação dos modelos usados, bem como o uso do *LlamaIndex* para auxiliar na geração de outputs melhores.

Acerca de modelos usados, para a seleção do *sentiment analysis* foi feito o teste de cinco modelos diferentes:

1. *tabularisai/multilingual-sentiment-analysis* [1]: o modelo multilingual, baseado em distilbert da Tabularis.
2. *cardiffnlp/twitter-xlm-roberta-base-sentiment* [3]: o modelo multilingual XLM-RoBERTa de Barbieri et al. (2022).
3. *tbluhm/clf-sentimentos-cmts* [5]: o fine tuning do XLM-RoBERTa, feito em texto em português brasileiro em uma compilação de redes sociais, de Bluhm (2024).
4. *ggrazzioli/cls-sentimento-sebrae* [16]: um fine-tuning de distilbert, treinado em dados internos, em português brasileiro, da SEBRAE-RS.
5. *lxuyan/distilbert-base-multilingual-cased-sentiments-student*.

Como o WhatsApp pode ter particularidades de estilos de comunicação devido a suas características únicas como rede social, criou-se, por coleta de cerca de 150 mensagens reais recebidas por um cliente e anotá-las manualmente com o sentimento correto, um dataset para testes. A partir deste dataset, foram geradas matrizes de confusão para cada modelo, que podem ser vistos nas Figuras 3, 4 e 5.

A partir desses resultados optou-se pelo modelo *cardiffnlp/twitter-xlm-roberta-base-sentiment*. E, com a introdução de filtros a partir de *confidence*, ou seja, a certeza que o próprio modelo tem sobre sua resposta, é possível aumentar a acurácia do modelo a um ponto aceitável, a custo de se isentar de algumas classificações. Ao

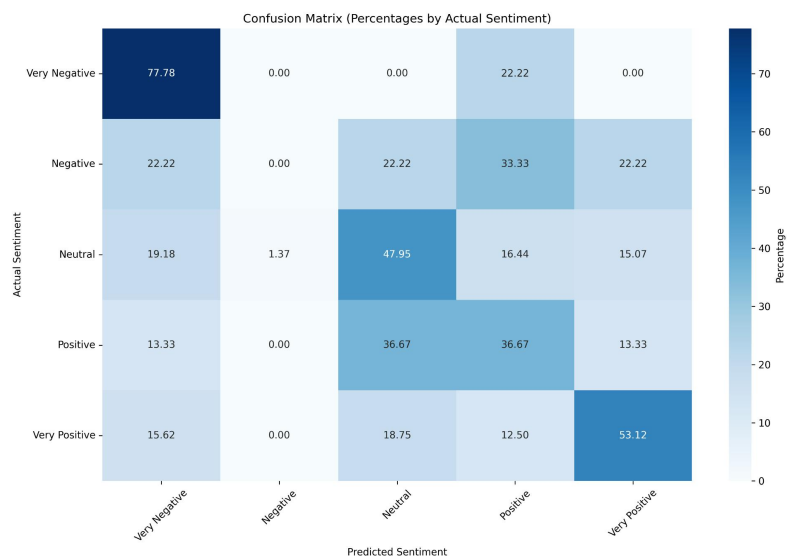


Figura 3: Matriz de Confusão do modelo *tabularisai/multilingual-sentiment-analysis*

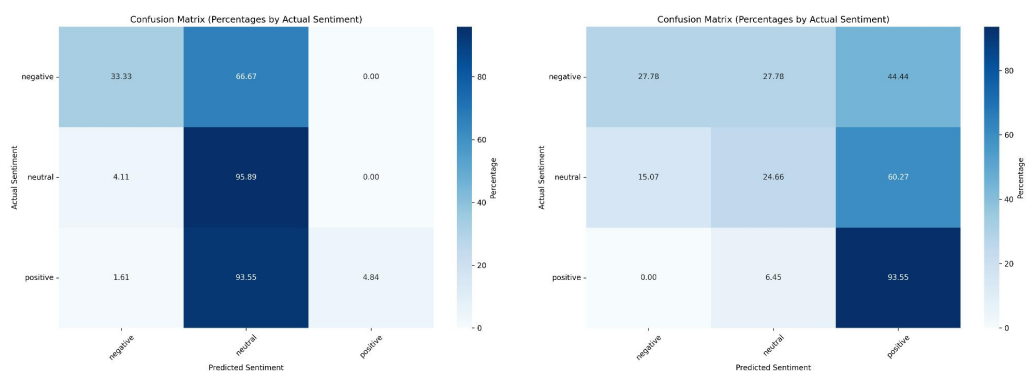


Figura 4: Matrizes de Confusão dos modelos *tbluhm/clf-sentimentos-cmts* e *ggrazzioli/cls-sentimento-sebrae*, respectivamente.

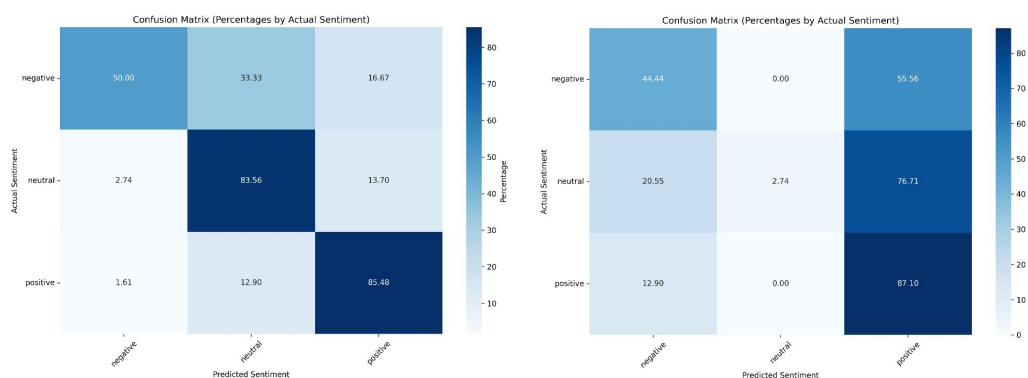


Figura 5: Matrizes de Confusão dos modelos *cardiffnlp/twitter-xlm-roberta-base-sentiment* e *lxyuan/distilbert-base-multilingual-cased-sentiments-student*, respectivamente..

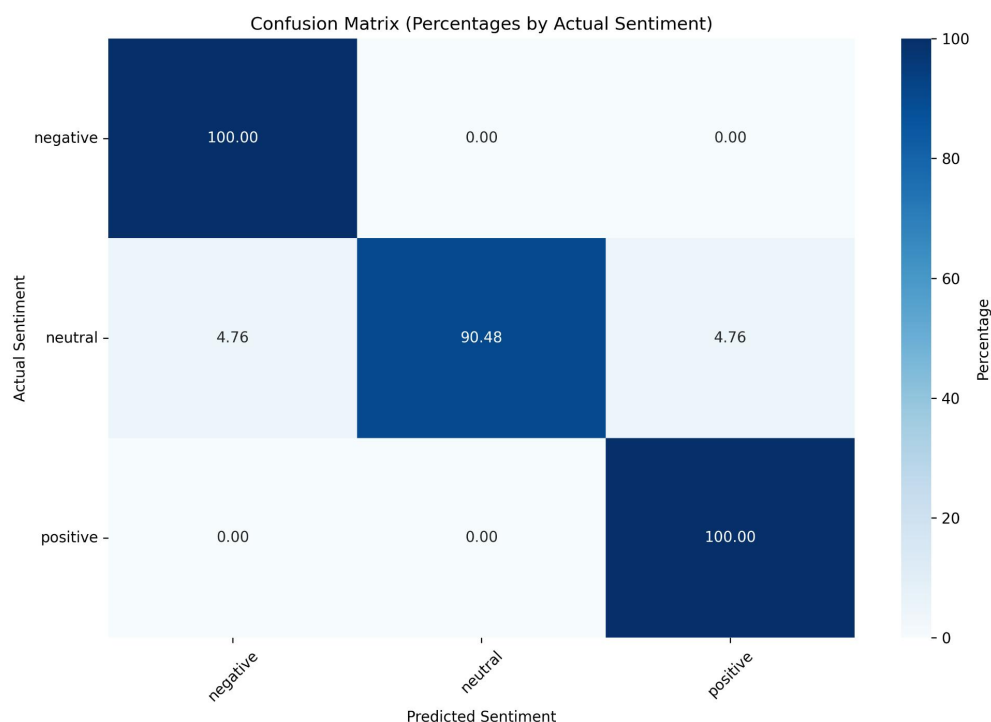


Figura 6: A Matriz de Confusão do *XLM-RoBERTa* após filtragem por confidence

experimentar com os parâmetros, foi feita a escolha de somente aceitar respostas acima de 75% de confidence, rendendo o seguinte resultado visto na Figura 6.

Acerca dos modelos de resumo, além de uma avaliação mais subjetiva da qualidade dos resumos, foram geradas 20 conversas com informações relevantes enterradas em mensagens de menos relevância, e conferido quantas das informações relevantes o modelo de resumo conseguiu extrair. Inicialmente, nem o uso de modelos de resumos especializados, como o SimCLS treinado na base CNNDM [18], nem LLMs, como o ChatGPT turbo, performaram bem, conseguindo extrair as informações relevantes 6 de 20 e 11 de 20 vezes respectivamente. No entanto, usando a LLMs foi possível explorar possibilidades quanto à prompt-engineering. Investigando essas técnicas, foi descoberto um prompt, com instruções focadas em explicar no que consiste uma informação relevante para o contexto e como ela deve ser apresentada, que aumentou a performance do modelo 3.5 turbo para 75% nos testes. No entanto, ainda havia uma dificuldade clara desses modelos em conseguir achar informações relevantes quando ocultas atrás de páginas de texto irrelevante.

Para amenizar essa questão, foi utilizado o *LlamaIndex*, especialmente o *QueryEngine* [19] oferecidos por eles. Essa ferramenta organiza dados não estruturados, como históricos de conversas e organiza eles em índices otimizados para a realização tarefas de processamento de linguagem natural, utilizando técnicas como *chunking* e a criação de *embeddings* para recuperação de trechos relevantes às tarefas dadas. Através da criação de um módulo de Python, responsável por processar os chats recebidos em Documentos *LlamaIndex*, é possível dar um acesso mais direto dos LLMs escolhidos aos dados que importam. Essa implementação parece ter miti-

gado as dificuldades encontradas nas tarefas de resumo previamente, com o GPT 3.5 conseguindo extrair as informações relevantes das 20 conversas com o uso do *LlamaIndex*.

Ademais, o QueryEngine do LlamaIndex também nos possibilita gerar insights qualitativos em linguagem natural. Isso permite a utilizar os dados do projeto, sejam esses os chats ou dados da API, para geração de feedback mais humanizado do que apenas visualizações e gráficos, que alinha com nossas propostas de transparência e usabilidade.

4.6 Investigação sobre Hospedagem

Aqui, será explicada a pesquisa feita sobre a hospedagem, a estrutura inicial proposta para a arquitetura dessa, bem como a escolha de não continuar investindo tempo de desenvolvimento nesse aspecto até o presente momento.

Para hospedar todos esses componentes do projeto, foi feito também uma pesquisa sobre possíveis arquiteturas de cloud para tal. Apesar de alguns possíveis provedores serem identificados, a escolha foi feita pelo uso do AWS (Amazon Web Service). No entanto, identificou-se a importância de ser feito uma análise comparativa de custo entre provedores em um momento futuro, que foi uma das razões que motivou a escolha pelo paradigma de IaC (Infrastructure as Code), utilizando a ferramenta Terraform.

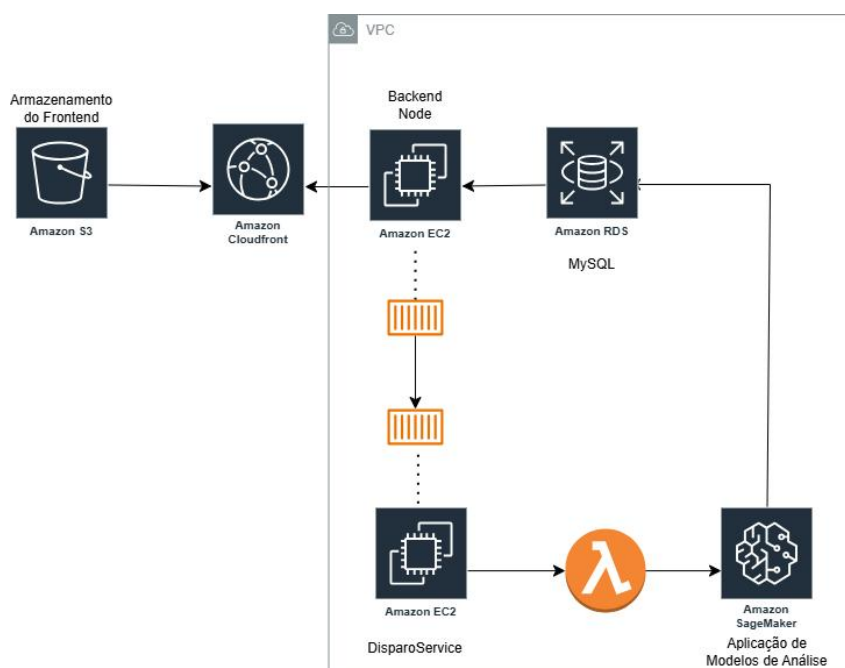


Figura 7: Arquitetura Proposta para Hospedagem AWS

Essa abordagem permite definir-se toda uma infraestrutura de cloud em forma de script, não só elevando a portabilidade dos serviços para outro ambientes cloud, mas permitindo também o versionamento e controle mais centralizado de tal. Nessa abordagem, definiu-se inicialmente um arquitetura para os componentes a serem

hospedados, visível na Figura 7.

No entanto, devido a limitações de tempo, os limites dos recursos gratuitos da AWS e a hospedagem dos componentes não ser essencial na validação do projeto, foi feita a escolha por não priorizar esse aspecto do desenvolvimento por enquanto. Então, essa arquitetura proposta não chegou a ser implementada por completo, apesar de testes preliminares em alguns componentes específicos, como executar o WhatsAppWebService em instâncias do EC2, serem realizados com sucesso.

4.7 Cronogramas de Atividades

Nessa parte, será apresentado o cronograma feito ao início da matéria de Projeto Final II, o cronograma real de execução do projeto e as explicações do atraso e abandono de algumas tarefas.

Devido a uma troca de tema de projeto feita entre a primeira e segunda parte dessa disciplina, não é possível comparar entre o cronograma proposto em Projeto Final 1 e o cronograma real de execução. No entanto, é possível fazer essa análise entre o cronograma planejado e o real da matéria de Projeto Final 2, visível na Tabela 3.

Etapa	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
Ultimos Views do Front End											
Atualizar Estrutura DB											
API para Tabelas Novas											
Entrevistas Iniciais e Análise											
Escrita Fase 1											
Servidor Node no DisparoService											
Detalhes Hospedagem DisparoService											
API DisparoService											
Escrita Fase 2											
Troca Dinâmica de Sessão/Chip											
Processamento de Mensagens com Modelos											
Entrevistas Avaliativas											
Hopedagem do Site na AWS											
Escrita Final											
Periodo pra Revisão											

Tabela 3: Cronograma Planejado, feito no começo da disciplina de Projeto Final II

É possível observar na Tabela 4 que algumas das tarefas originalmente propostas foram eliminadas, especialmente aquelas que tratam da hospedagem, pois estes foram decididos por ser fora do escopo atual do projeto. Acerca dos atrasos nas outras tarefas, esses foram principalmente devidos a dificuldades de agendamento das entrevistas com os usuários.

Etapa	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
Ultimos Views do Front End											
Atualizar Estrutura DB											
API para Tabelas Novas											
Entrevistas Iniciais e Análise											
Escrita Fase 1											
Servidor Node no DisparoService											
Detalhes Hospedagem DisparoService											
API DisparoService											
Escrita Fase 2											
Troca Dinâmica de Sessão/Chip											
Processamento de Mensagens com Modelos											
Entrevistas Avaliativas											
Hospedagem do Site na AWS											
Escrita Final											
Periodo pra Revisão											

Tabela 4: Cronograma real da execução do projeto, com atrasos em vermelho

5 Projeto e especificação do sistema

5.1 Estrutura do Sistema

Aqui, será descrito mais extensivamente a estrutura daquilo que foi desenvolvido, bem como o processo de escolha atrás de algumas tecnologias e frameworks escolhidos para o projeto.

O sistema é dividido em 3 principais partes, o container responsável pela interação com Whatsapp (que chamaremos de *WhatsAppWebService*), os scripts de análise das mensagens recebidas (que chamaremos de *MessageAnalysisService*) e o site que servirá de interface para o sistema.

Sobre o *WhatsAppWebService*, sua principal função é utilizar do *whatsapp-web.js* e, futuramente, o *WhatsApp Business Cloud API*, para realizar o envio e recebimento das mensagens. Para tal, deve estruturar uma fila de envios e realizá-los com formato e conteúdo corretos e com informações personalizadas para cada

contato. Para tal, foi construído um módulo mais orientado aos objeto de contatos, que provém um wrapper para manipulação de contatos com qualquer número de tags e propriedades, incluindo a customização de uma mensagem ao redor dos dados deste, bem como a criação de múltiplos contatos tanto a partir de arquivos .csv ou requests web. Essa abordagem mais modular e generalizada, permite que não seja necessário mudanças na mecânica de envio, mesmo com alterações nos dados utilizados para cada contato.

Mas, pelos requerimentos descritos nas atividades realizadas, também foi necessário construir um pequeno servidor node para comunicação com os outros processos. Esse servidor possui uma estrutura RESTful separada em dois recursos: mensagens, representando tanto mensagens recebidas e enviadas, e instâncias, representando instâncias do Whatsapp Web sob controle. No entanto, para conseguir-se implementar o recebimento de mensagens real-time e evitar que o site tenha que constantemente fazer requests para checar se há conteúdo novo, foi implementado um webhook para que o site possa apenas ser notificado quando houver mensagens. Ademais, apesar de não caber tão precisamente na lógica de eventos no qual webhooks são tipicamente usados, um também foi estabelecido para a inicialização das instâncias, pois esta pode demorar um mais de 60 segundos dependendo da disponibilidade de recursos. A estrutura dessa API pode ser vista abaixo, na Figura 8.

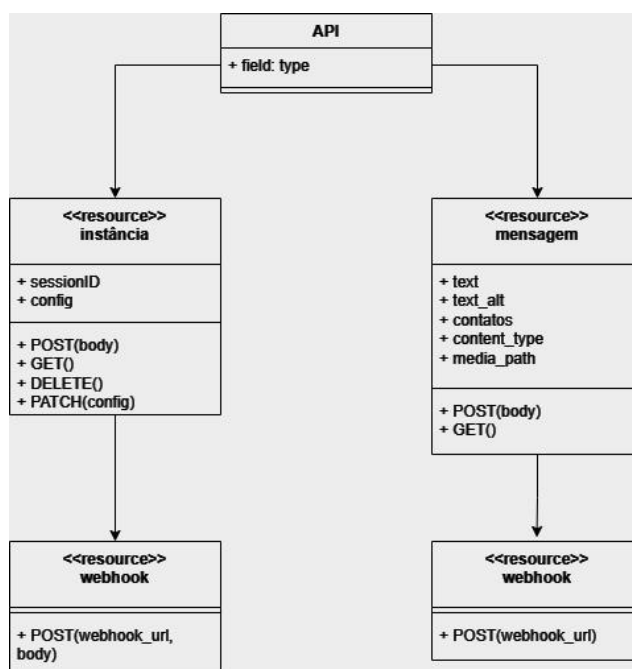


Figura 8: Estrutura do API do WhatsAppWebService

Acerca do MessageAnalysisService, esses foram implementados em scripts relativamente simples de Python, que aplicam seus respectivos modelos a lotes de dados, e enviam um request para o API do backend, afim das base de dados ser atualizada. No entanto, esses scripts são executados manualmente por enquanto, apesar de haverem planos para hospedá-los em tarefas de rotina no Amazon SageMaker, como descrito na Figura 7.

Após esses dados serem inseridos na base, são utilizados dentro das views relacionadas à chats, destacando chats com mensagens não lidas negativas e, colorindo mensagens dentro desses, para prover um feedback visual instantâneo. Além disso, essas avaliações são utilizadas para monitorar a performance da operação em diferentes segmentos de contatos, cuja importância foi destacada nas entrevistas, e para gráficos e visualizações para guiar estratégia.

Sobre o site, como mencionado previamente, foi feita a escolha por utilizar um stack composto de MySQL Server 8.0, Node + Express e React. Primeiramente, tratando da base de dados, essa foi construída seguindo a forma normal Boyce-Codd, para eliminar redundâncias e eliminar anomalias quando realizando operações. A parte da estrutura dessa base relacionada aos chats pode ser observada na Figura 9 e a estrutura completa no Apêndice A.

Para o Backend, foi desenvolvido um API RESTful em Node, com o total de 59 endpoints abrangendo o modelo de dados completo da base de dados. Foi seguido ao máximo o princípio da orientação à recursos do REST, com as poucas exceções sendo endpoints para aumentar performance em tarefas específicas repetidas frequentemente. Um exemplo disso é que, na chegada de uma mensagem nova, é necessário buscar, pelo número telefônico do remetente, se existe nos chips do usuário algum chat que seja dele e, caso não haja, criar o chat antes de adicionar a mensagem à base. Ao criar, então, um endpoint POST `/v0/mensagens/:telefone/telefone`, foram evitadas várias chamadas da API e possibilitou-se a realização queries muito mais otimizadas na base de dados. Um diagrama da API está disponível no Apêndice B.

Por fim, o front end React pelo projeto necessitar de uma alta interatividade, ter funções real-time e ter um gerenciamento de estado relativamente complexo, foi feita a escolha pelo modelo de SPA (Single-Page Application). Utilizando o `react-router-dom` e dashboards interativos para navegar entre diferentes views, eliminou-se a necessidade de um reload completo e criou-se uma experiência mais parecida com aplicativo desktop nativo. Adicionalmente, foi feito uso extensivo do Material UI, uma biblioteca de componentes para implementação do Material Design [15], o framework de design e UX da Google, a fim de implementar de forma eficiente e limpa uma interface condizente com as heurísticas de usabilidade do Nielsen.

5.2 Contribuições

Nessa seção, é discutido brevemente as contribuições principais do projeto e os avanços que eles configuram.

Considera-se que essa interface e as escolhas de UX feitas nelas contribuem positivamente ao estado da usabilidade de sistemas como estes, empregando técnicas para reduzir significativamente a carga de usuários não-técnicos ao realizar operações relativamente complexas.

A implementação do sentiment analysis configurou também uma contribuição até maior, desde os testes em diversos modelos com os dados desse subdomínio, até sua integração em aspectos visuais do chat e seu uso em painéis de dados.

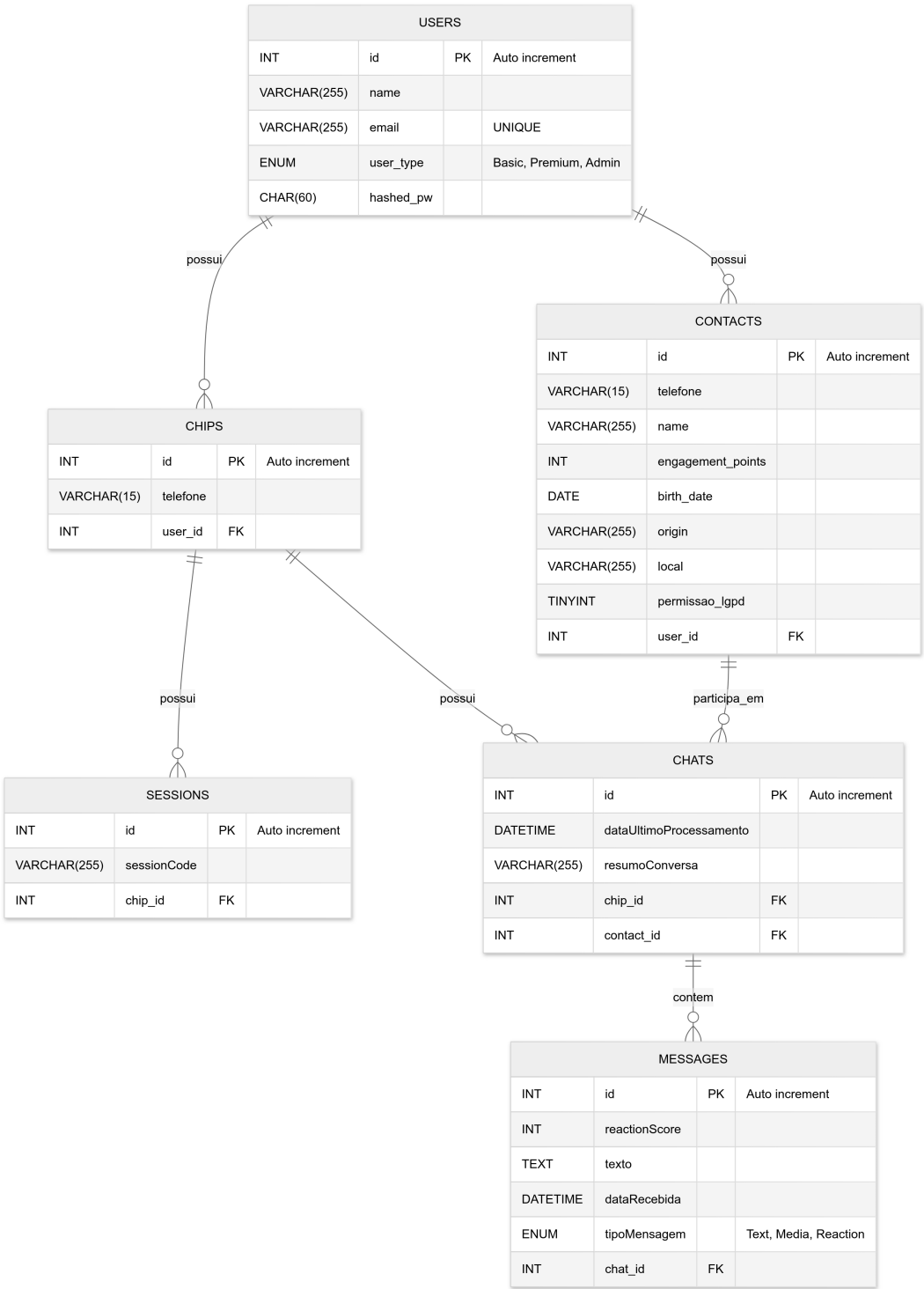


Figura 9: Diagrama ER da estrutura da parte base de dados rela

Similarmente, o uso do LlamaIndex, aplicado ao subdomínios de geração de resumos e insights a partir dos dados gerados configura um avanço em relação a qualidade dos resultados obtidos.

6 Implementação e avaliação

6.1 Testes Funcionais

Aqui, será descrito brevemente a estrutura e abrangência dos testes funcionais construídos para validação técnica do sistema. Apesar de alguns dos aspectos mais abstratos do projeto não qualificarem para testes funcionais, todos os componentes do sistema foram submetidos a um processo de teste. Utilizou-se amplamente o Postman para realizar avaliações do *WhatsAppWebService* e do Backend do projeto. Através dessa ferramenta, foi possível criar tanto testes individuais de endpoints, quanto testes de integração mais amplos. Esses foram cruciais para orientar e validar o desenvolvimento ao longo do projeto e tinham uma taxa de 100% de sucesso ao final do desenvolvimento.

Foi feito também o uso de Jest, apesar do escopo desses testes faltarem abrangência, por não terem sido criados ao longo dos ciclos de desenvolvimento do projeto. No entanto, alguns testes básicos para validação, como a presença de *divs* fundamentais e interações chaves foram implementadas.

6.2 Testes com usuários

Essa seção discute a realização do teste de usabilidade do sistema, bem como a condução de uma entrevista pós-teste e o processo de análise de resultados.

Os testes de usabilidade foram o principal teste não funcional feitos, realizados com os dois participantes, referidos como participantes 1 e 2 (P1 e P2), das entrevistas iniciais. Esses participantes assinaram um termo de consentimento livre e esclarecido (TCLE), cujo modelo está disponível no Apêndice G. Foram dadas 10 tarefas para realizar dentro do sistema, algumas com múltiplas etapas, que podem ser vistas com mais detalhe na Tabela 6. As sessões de teste foram conduzidas virtualmente através da funcionalidade de controle remoto de desktop do Zoom, onde os participantes conseguiam controlar o mouse e teclado para interagir com o software.

Os participantes foram incentivados a tentar realizar as tarefas sem auxílio e seguir o protocolo *think aloud*, que pede que eles expressem seus pensamentos, opiniões e motivos de escolhas em voz alta, para extrair mais informações viáveis dos testes. O roteiro e a transcrição desses pode ser encontrado no Apêndice D. A realização das tarefas por cada usuário pode ser observado na Tabela 5, que revela a facilidade dos usuários com a maioria das tarefas. Isso pode ser visto na taxa de 80% de sucesso sem orientação alguma e na taxa de 100% de sucesso com orientações mínimas.

No entanto, somente a partir desses testes, já foi possível descobrir uma series

Participante	T1	T2	T3	T4	T5	T5	T6	T7	T8	T9	T10
P1	TI	TA	TA	TI	TI	TI	TI	TI	TI	TI	TI
P2	TA	TA	TI	TI	TI	TI	TI	TI	TI	TI	TI

Tabela 5: Conclusão das tarefas pelos participantes do teste de usabilidade, com TI representando as tarefas realizadas independentemente, TA as tarefas realizadas com auxílios ou orientações básicas e NT as não terminadas.

Tarefa	Descrição
T1	Faça login no sistema e verifique o status do seu plano
T2	Importar contatos do arquivo X.csv no seu sistema, adicionando eles a um segmento novo chamado Y
T3	Adiciona o contato X ao segmento que acabou de criar
T4	Remova o contato Y do segmento
T5	Realize um envio novo para esse segmento que acabou de criar, com uma mensagem de boas vindas
T6	Você acabou de receber uma mensagem nova, veja se você consegue responder ela
T7	Tente agora analisar as conversas anteriores. Como está sendo a interação com eles? Consegue identificar algum problema ou questão?
T8	O contato Z disse que era engano, exclua ele da base.
T9	Tenta identificar como tem sido os resultados e progresso em geral da operação desse usuário.
T10	Faça logout do sistema

Tabela 6: Descrição das tarefas dadas aos participantes do teste de usabilidade

de melhorias a serem feitas a partir das dificuldades dos participantes. Os comentários feitos pelos participante devido ao protocolo *think aloud* também foram vitais para melhor identificar as intenções e concepções dos participantes. Algumas das mudanças principais a serem feitas que foram identificadas são:

- Na tela de importação de contatos, presente na tarefa T2, o participante 1 acabou clicando em um botão do primeiro passo do fluxo de importação enquanto estava no segundo passo. Podemos, pra prevenir esse tipo de erro, desabilitar ou esconder esse botão após o termino da etapa relevante.
- Também na tela importação de contatos, ambos os participantes não perceberam as instruções escritas e o botão de continuar da tela. Podemos, para prevenir isso, aumentar o destaque visual desses elementos na tela e reduzir a quantidade total de informação na tela através da técnica mencionada no item anterior.

- Em algumas tarefas, como a T2 e T5, os participantes verbalizaram uma informação incorreta sobre o estado do sistema, por exemplo dizendo "*Acredito que tenha ido (...)*" sobre a importação de contatos antes que tivessem apertado a confirmação para tal. Isso indica a necessidade de incluir indicações mais claras de qual etapa de qual processo está sendo realizada nesses fluxos com múltiplas etapas.
- Em alguns momentos, como no *popup* da de seleção de contatos, quando adicionando contatos a um segmento na tarefa T3, o participante 1 parece não ter entendido o texto dos botões presentes, verbalizando uma intenção diferente da ação correspondente do sistema. É possível retificar isso usando nomenclatura e descrições mais claras como, nesse exemplo do *popup*, trocando "Importar Contatos Novos" para "Importar Contatos de um Arquivo".
- Ambas as participantes, em alguns momentos, fizeram uma ação de "hover" com o mouse sobre elementos que não possuíam tooltips, buscando informações sobre a ação antes de executá-la. Em resposta a isso, pode-se expandir o uso de tooltips, utilizando-os em qualquer lugar que tenha uma ação não trivial a ser realizada.
- No Chat, presente na tarefa T6, o participante 1 teve dificuldade de identificar qual era o chat da mensagem que tinha chegado mais recentemente. Para apoiar o usuário nisso, pode-se exibir o horário da última mensagem já na seleção de chat e criar ferramentas de ordenação dos chats mais claras.
- Também na aba de chat, quando se recebe uma mensagem que contém uma das palavras ou frases do dicionário de sentiment analysis, ela é marcada como "flagged" e o chat que contém ela é destacado com um ponto de exclamação. No entanto, a participante 1 não conseguiu identificar o significado desse detalhe. Para resolver isso, pode-se utilizar uma combinação dos tooltips mencionados previamente e navegar diretamente a mensagem em questão quando o usuário clicar no ícone de exclamação na seleção de chats, associando o ícone com o conteúdo da mensagem e criando carga cognitiva germânica.
- Novamente no Chat, ambos os participantes aparentam ter ficado confusos sobre se suas respostas tinham sido enviadas ou não, pelo tempo de latência da comunicação dos componentes e a ausência de um feedback imediato sobre o status da mensagem enviada. Pode-se endereçar isso provendo esse feedback diretamente, seja através do sistema de notificações *snackbar* ou mostrando uma mensagem carregando visualmente no chat até ser confirmada como enviada.
- Na aba de painéis de dados, presente na tarefa T9, as visualizações são divididas em duas categorias, insights escritos, gerados automaticamente e gráficos mais tradicionais dos dados de performance da operação. Os participantes

não parecem ter conseguido extrair muitas informações úteis dos gráficos, permanecendo muito mais tempo nas áreas com insights escritos e com o participante 2 constatando plenamente que não sabia interpretar um dos gráficos. Isso indica que essa abordagem de insights provavelmente é mais efetiva para esse tipo de usuário e que deve-se explorar técnicas novas para geração de tais.

Em seguida, foram realizadas entrevistas pós-teste com cada um dos participantes, com perguntas orientadas a aspectos diferentes da usabilidade e utilidade do software desenvolvido. A partir das transcrições dessas entrevistas, disponíveis em completo no Apêndice E, foi feita uma análise temática. A análise temática é uma técnica que permite identificar padrões dentro de um conjunto de dados qualitativos [24]. A análise revelou três principais temas recorrentes nas respostas obtidas:

- UX e Interface: tratam de comentários sobre o processo de uso e navegação sobre a interface, tanto aqueles que falavam sobre aspectos que geraram facilidade ou dificuldade de uso. Foi identificado que, enquanto os comentários positivos eram mais gerais, os comentários sobre as dificuldades de uso tratavam das mesmas 2 telas específicas: a tela de importação de contatos e a tela de chat. Capturas dessas telas estão disponíveis nas Figuras 10 e 11 respectivamente. Alguns trechos da entrevista que refletem as tendências comentadas são:

"Na verdade, é mais difícil, assim (...) inserir novos contatos." [P1]

"No início, um pouco (de dificuldade). Por exemplo, você pediu para eu criar um novo segmento (na importação de contatos)." [P2]

"Eu achei tudo bem didático mesmo, bem direto, bem claro e objetivo." [P1]

"Eu tive um pouco de dificuldade na parte das mensagens (...)" [P1]

"Ele (o sistema) é bem instintivo mesmo" [P1]

"E eu achei fácil, intuitivo. (...) uma usabilidade bem... bem tranquila, bem fácil." [P2]

"Acho que qualquer pessoa que tem um conhecimento básico de um funcionamento de um computador consegue manusear" [P1]

- Features: acerca de comentários sobre features específicas, estes foram divididos aqueles sobre as features implementadas e aqueles sobre features desejadas. Diante das implementadas, a mais mencionada foi o uso de sentiment analysis, e das desejadas, um pedido unânime foi a inclusão de uma funcionalidade para gerenciar e repassar as demandas recebidas pelo WhatsApp. Além dessa funcionalidade das demandas a única outra desejada era uma menor latência no chat. Esses fatores são observáveis nos seguintes segmentos das entrevistas:

"Se foi uma mensagem positiva ou negativa. Tudo isso a gente tem que fazer manualmente. E a gente estando dentro de uma ferramenta inteligente o suficiente para entender toda aquela movimentação de mensagens. E já... Identificar, né? (...) Enfim. Vai economizar, assim, um tempo valiosíssimo." [P1]

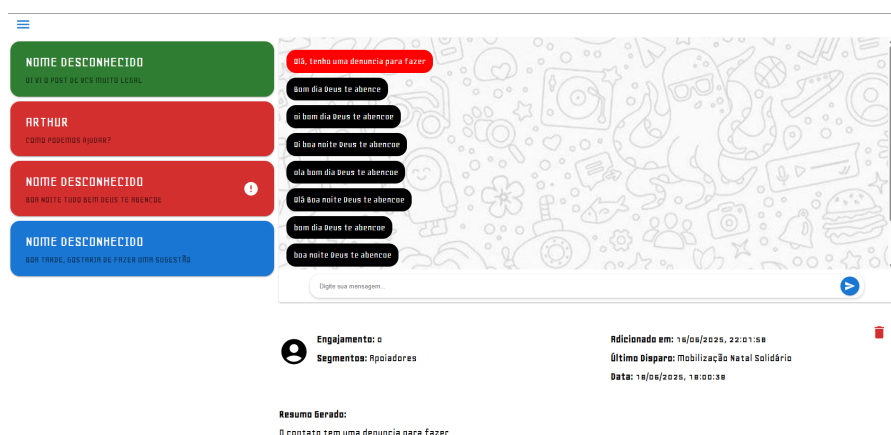


Figura 10: Captura da tela de chat

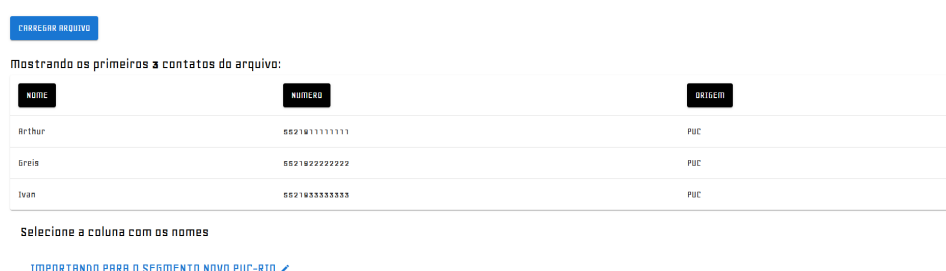


Figura 11: Captura da tela de importação de contatos

"Eu acho que poderia incluir uma opção (...) de demandas em que a gente consiga inserir uma pessoa que esteja mandando um convite, (...). Enfim, as demandas do cotidiano mesmo" [P1]

"Quantas demandas surgiram e foram encaminhadas (...) Eu acho isso importante." [P2]

"Eu senti falta de um local onde a gente insira as demandas, né? Porque acabou que é gerado um outro local para que a gente insira essas demandas com status de se foi resolvido, se não foi, ou a nível de informação mesmo para ficar registrado que determinado contato fez alguma solicitação, algum convite ou fez alguma denúncia." [P1]

"A nossa planilha de relatoria, as nossas planilhas de contatos, o status do contato, se ele é um contato positivo, se é um contato negativo, como nós já temos todas as nossas necessidades e todo o nosso fluxo de trabalho dentro de uma ferramenta só" [P1]

- Transparência: acerca de comentários sobre a transparência do sistema, apesar desses serem variados, podem ser vistas duas tendências principais: aqueles sobre a facilidade de acesso e compreensão de informações sobre os resultados da operação e de identificação do consentimento ou não, dos contatos. Os da primeira categoria tratavam tanto da disponibilidade de informação necessária

para compreensão e apoio das ações sendo tomadas, quanto entendimento do feedback recebido. A segunda categoria tratava principalmente da identificação de quais contatos queriam ou não receber mensagem.

"Conseguiria. Na verdade, tenho a opção da relatoria já, que vai automático. Eu consegui identificar na parte inferior da tela do disparo realizado, que ali já vão as informações mensais para a relatoria" [P1]

"Porque rapidamente dá para visualizar se o disparo foi bom ou ruim, se está indo mal ou bem. (...) Eu acho que isso sustenta muito a questão do parâmetro que eu falei." [P2]

"Ali tem todas as informações que a gente já utiliza no nosso dia-a-dia e que são extremamente necessárias, são as informações que a gente realmente ter para entender se está funcionando, como a gente está atingindo, engajando." [P1]

"Com as palavras em aleta, da sinalização com cores, facilita muito" [P2]

"Sim, isso bem claro. Os contatos que queriam receber mensagens estava já identificado na cor verde e os contrários a isso, com mensagens negativas, já apareceram em vermelho. [P1]"

Participante		P1	P2
UX e Interface	Facilidade de Uso	9	5
	Dificuldades de Uso	5	4
Funcionalidades	Utilidade de Features	4	7
	Falta de Features	3	3
Transparência		4	3

Tabela 7: Ocorrência de temas elencados nas entrevistas de cada participante.

Através de observar a correlação e ocorrência conjunta de temas, pode-se perceber algumas conexões:

- Uma das dificuldades de uso mencionadas recorrentemente foi a leitura dos gráficos. Essa dificuldade é frequentemente mencionada nas entrevistas em contraste com os insights escritos, que são descritos como um dos principais apoios à visualização de resultados.
- Como identificado anteriormente, houve dificuldades dos usuários na tela de Chats, gerado principalmente pela latência entre os componentes do projeto e a falta de feedback imediato sobre o status da mensagem.
- Dois dos temas identificados como mais fortemente relacionados foram a transparência do consentimento dos contatos a conduzirem a comunicação e o *sentiment*

analysis integrado visualmente através aos chats. Isso reforça a proposta de valor da aplicação desenvolvida da tecnologia.

É possível observar essas relações na Figura 12, com a composição de temas representada pelas setas e as correlação de temas distintos pelas linhas pontilhadas.

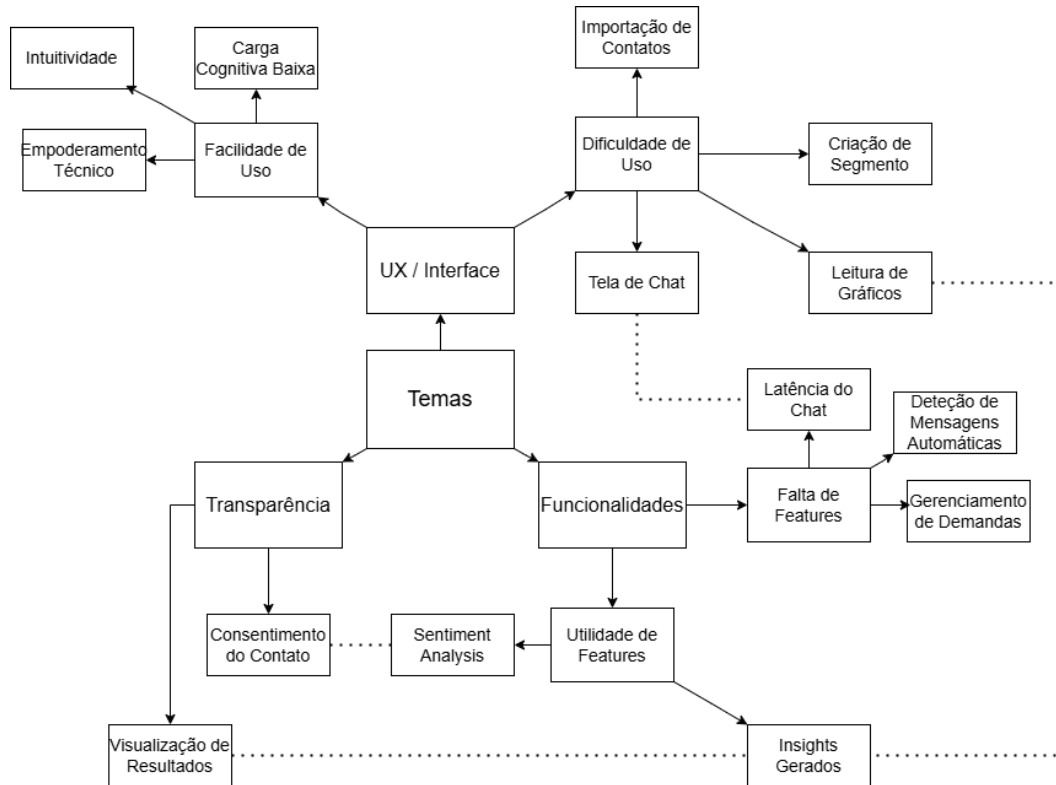


Figura 12: Relação dos temas encontrados na análise temática

6.3 Comentários sobre a implementação

Essa parte apresenta algumas limitações atuais do projeto e porque não foram, nesse momento, contornadas, bem como exemplos de barreiras técnicas e limitações que foram contornadas.

A implementação do projeto, como está atualmente, não representa um produto pronto, mas sim uma validação básica das principais contribuições e proposta de valor de um futuro produto. Sendo assim, há inúmeros refinamentos e melhorias pontuais a serem implementadas. Alguns dos principais mudanças identificadas são:

- As melhorias de UX e interface extraídas na análise do teste de usabilidade e entrevista posterior, que ajudarão a consolidar esse projeto como uma ferramenta acessível para usuários não técnicos
- O pedido de ambas os entrevistados por incluir um sistema para apoio às demandas que recebem pelo WhatsApp e tem que repassar para outros membros da equipe. A participante número 2 sugeriu a implementação de funcionalidades parecidas com Kanban [2], no qual pudessem atribuir a responsabilidade da demanda e ter lembretes das pendências a serem resolvidas.

- Apesar de, como discutido anteriormente, a arquitetura AWS para hospedagem do projeto ser planejada, essa não pode ser testada nem implementada em sua integridade. No entanto, já foram feitas explorações iniciais onde alguns componentes, como o *WhatsAppWebService* foram hospedados com sucesso.
- Por consequência também da incerteza associada a arquitetura cloud a ser utilizada, alguns dos passos de segurança necessários para um serviço web que são providos por serviços da AWS como o Virtual Private Cloud não foram implementados. Independente disso, apesar de não serem cruciais nos escopos da validação desse projeto, precisam ser implementadas medidas de segurança mais robustas antes que distribua amplamente o software.

Todas esses pontos deixaram de ser implementados, também, por falta de tempo e a escala limitada do que pode ser feito na matéria de Projeto Final II, que levou a priorização os aspectos essenciais desse projeto. No entanto, acredita-se que o único passo com barreiras técnicas significativas é a hospedagem, com as outras melhorias precisando apenas de mais alguns ciclos de desenvolvimento.

Apesar desses itens terem ficados para futuras iterações, houveram muitas dificuldades que foram resolvidos ao longo do projeto. Além daquelas mencionadas na seção de *Atividades Desenvolvidas*, algumas das barreiras técnicas principais técnicas resolvidas foram:

- Criação de um servidor node no *WhatsAppWebService*: um passos técnicos mais corriqueiros do projeto que precisou de mais estudo foi a implementação de um servidor node no módulo dockerizado responsável por de fato enviar e receber as mensagens dos usuários. A grande dificuldade disso era conciliar a estrutura event-driven do node com algumas funções *blocking* do *whatsapp-web.js*. Após descobrir os trechos responsáveis pela interferência, estes foram reimplementados e foi criado um fork privado da biblioteca, onde alguns parâmetros da sua interação com o *puppeteer* foram mudados.
- LlamaIndex e a aplicação de LLMs/modelos de resumo: Anteriormente até aos testes básicos aplicados aos modelos de resumo e LLMs, foram elencados uma diversidade de modelos muito maior. No entanto, nenhum desses conseguiu o performance desejado nem se sobressaiu como a escolha óbvia para o projeto. Apesar de acontecer tarde no desenvolvimento do projeto, com a descoberta do LlamaIndex e as funcionalidades que oferece em termo de organização especializada de dados não estruturados para tarefas NLP, não somente houve uma melhoria de performance dos modelos previamente investigados, mas também foram abertas várias possibilidades de interação direta do usuário com esses dados e geração de insights qualitativos.
- Importação de contatos a partir de um arquivo arbitrariamente formatado: Ao contrário dos contatos já presentes na base MySQL, os arquivos e planilhas

de contatos que o software precisa ser capaz de importar vem sem uma formatação pré-definida tanto na tabulação, quanto na estrutura interna dos dados. A grande maioria dos softwares similares mencionados na seção de situação atual resolvem isso por demandar um formato fixo para os contatos. No entanto, isso cria uma barreira técnica significativa para aqueles que não sabem realizar a formatação necessária para tal. Então, para contornar essa barreira, foi necessário passar por diferentes iterações de formas de importar esses contatos. Acabou-se conseguindo desenvolver um sistema que aceita qualquer formatação dos dados, com os usuários sendo pedidos, após carregarem o arquivo, a visualmente selecionar as colunas com os dados mínimos para construção de um contato na base, um elemento por vez. Isso permite que até usuários com muito pouco conhecimento técnico realizem esse processo importante de uma forma fácil e acessível. Apesar de haverem melhorias a serem feitas nessa tela, como indicadas na subseção *Outros Testes*, o que está implementado atualmente já permitiu que usuários realizassem a tarefa com orientação mínima.

Esperamos, diante de tudo que já foi resolvido, continuar o desenvolvimento após o término da disciplina e cada vez produzir iterações melhores do software.

7 Considerações finais

Aqui, é apresentado como os objetivos do projeto foram cumpridos e o que acredita-se ser nossas contribuições principais.

Em relação aos objetivos propostos, além de implementação um projeto full-stack e um componente dockerizado para controle do WhatsApp (*WhatsAppWeb-Service*), acredita-se que foram cumpridos os objetivos propostos da seguinte forma:

- UX e Acessibilidade Técnica: Através da implementação de um projeto de UX guiado pelas heurísticas de Nielsen e a Teoria de Carga Cognitiva, criou-se um sistema de fácil uso para quadros não-técnicos, possibilitando que uma diversidade de usuários novos consigam usar o Hermes. Isso é claramente visível em alguns trechos das entrevistas realizadas, como:

"Acho que qualquer pessoa que tem um conhecimento básico de um funcionamento de um computador consegue manusear"

"Acho que dá uma horizontalidade na equipe. Porque a gente não fica na dependência (de alguém operar o software pelo entrevistado)."

- Tecnologia em Prol da Humanização da Comunicação: Utilizando modelos de sentiment analysis testados e escolhidos para o subdomínio relevante e módulos desenvolvidos para o melhor processamento de chats por LLMs através do LlamaIndex, consegui-se criar mecanismos para aumentar a capacidade do usuário lidar com escalas grandes de contatos, sem perder a autenticidade da conexão humana. Tanto as indicações de sentimento mostradas visualmente, quanto as produção de insights qualitativos e resumos de conversa foram efetivamente validados pelos usuários como algo que contribuiria mensuravelmente para melhorar seu fluxo de trabalho atual.

Diante disso, pode-se afirmar a distinção do Hermes das soluções existentes e a efetividade daquilo que foi implementado.

Assim, acredita-se que, nossa principal contribuição seja ao usuário final, que é o principal beneficiado pelo desenvolvimento que expande as capacidades de produtos semelhantes, automatiza aspectos repetitivos da rotina e provém dados e insights sobre a operação. Espera-se também que o projeto de UX torne softwares como esse acessíveis a uma quantidade muito maior de usuários não técnicos, que compõe a maioria daqueles no público alvo.

No entanto, crê-se também que esse projeto reverta a benefícios para o contato final, com qual o usuário se comunica, por não somente dar mais condições aos usuários responderem de forma mais completa e eficiente, mas ilustrar claramente quem quer receber mensagens ou não.

Espera-se também que o projeto contribua para a comunidade técnica, no que consta ao desenvolvimento de software para comunicação humanizada e transparente. Acredita-se que essa contribuição é visível especialmente na integração de técnicas estado da arte de sentiment analysis e resumo ao projeto de UX acessível

adotado, bem como a os processos nos quais testou-se e aplicou-se tais ferramentas no contexto do subdomínio.

7.1 Aprendizados

Nessa seção, será discutido brevemente os principais aprendizados adquiridos ao longo do projeto, bem como o que seria modificado caso houvesse a oportunidade de começa-lo novamente.

Durante esse projeto, tive a oportunidade de aprofundar meu conhecimento sobre diversas tecnologias e frameworks específicos: React, Node, MySQL, Docker, hospedagem cloud, modelos de sentimento e resumo. Mas, pela escala de tudo que precisava ser implementado, também pude crescer como desenvolvedor, me tornando um profissional mais organizado, metódico e completo.

Se pudesse começar novamente o projeto com os conhecimentos que tenho agora, mudaria 5 principais aspectos da implementação do projeto:

1. Acelerar ciclos de desenvolvimento: Apesar de haverem muitas iterações de desenvolvimento anteriormente a adoção desse projeto para a matéria de Projeto Final II, o projeto foi limitado em sua capacidade de realizar múltiplos ciclos de construção e avaliação. Caso fosse possível, reestruturaria o cronograma de atividades para ter ciclos mais curtos, com mais oportunidades de interação com o usuário final.
2. Organização de código: Devido principalmente a uma falta de experiência com React e Node, foi necessário reorganizar e refatorar diversas vezes o código para que ficasse minimamente limpo, manutenível e reutilizável. Tendo a visão que eu tenho hoje sobre a estruturação de projetos maiores com essas tecnologias, conseguia poupar precioso tempo de desenvolvimento.
3. Refinamento de pesquisa: Podendo reiniciar o projeto, teria adotado uma atitude mais metódica em sua execução, poupando facilitando e acelerando as escolhas do projeto. Especialmente na escolha dos modelos de resumo e sentiment analysis usados, a grande diversidade de modelos para as mesmas tarefas, poderia ter sido feito de forma melhor com uma abordagem mais sistemática.
4. Testes mais amplos: Apesar do Postman ter sido adotado para testes individuais desde o início do projeto, caso tivesse a oportunidade, teria implementado testes mais abrangentes desde cedo, poupando tempo de debugging e permitindo uma abordagem mais sistemática. Isso é especialmente verdade para o caso do front end React do projeto, e seus testes com Jest.
5. Perspectiva Cloud: Apesar de não ser diretamente ligado a proposta de valor do projeto, adotar uma perspectiva de desenvolvimento para cloud desde o início do projeto talvez teria possibilitado a andar mais um passo em direção a ter um produto completo.

7.2 Futuros Trabalhos

Por último, nessa parte será descrito o desenvolvimento futuro planejado para esse projeto e possíveis temas para outros alunos conduzirem seus trabalhos.

Primeiramente, planeja-se continuar a trabalhar em iterações futuras iterações desse software, utilizando do conhecimento e feedback obtido para criar uma ferramenta mais útil, fácil de usar e que agrega mais valor. As principais mudanças que serão implementadas estão descritas na seção de "Comentários sobre a implementação" mas, com outros ciclos de interação e feedback com o usuário, será possível elencar ainda mais melhorias.

No entanto, dos elementos presentes, acredito que o melhor tema um potencial trabalho futuro trataria da hospedagem otimizada de todos os elementos desse projeto. A criação de uma arquitetura flexível e confiável, com aspectos únicos dos conteúdos hospedados, como framebuffers virtuais, certamente teria a escala e complexidade suficiente para um projeto dessa disciplina.

Mas, em geral, dado um mundo onde é cada vez mais comum ver-se a automatização do contato, humanizar esse contato torna-se cada vez mais necessário. Acredito que a investigação de como usar as técnicas do nosso campo de pesquisa para tal é valiosíssima, e que deveria ser explorada também por outros alunos.

References

- AI, Tabularis. **Multilingual Sentiment Analysis Model**. [S.l.: s.n.]. Hugging Face. Acesso em: 23-02-2025, Disponível em: <<https://huggingface.co/tabularisai/multilingual-sentiment-analysis>>.
- ANDERSON, David J.; CARMICHAEL, Andy. **Essential Kanban Condensed**. [S.l.]: Blue Hole Press, 2016. ISBN 0984521429.
- BARBIERI, Francesco; ESPINOSA ANKE, Luis; CAMACHO-COLLADOS, Jose. XLM-T: Multilingual Language Models in Twitter for Sentiment Analysis and Beyond. In: PROCEEDINGS of the Thirteenth Language Resources and Evaluation Conference. [S.l.: s.n.], 2022. P. 258–266. Disponível em: <<https://aclanthology.org/2022.lrec-1.27/>>. Acesso em: 23 fev. 2025.
- BASELAB. **BaseLab**. [S.l.: s.n.]. Acesso em: 23-02-2025, Disponível em: <<https://baselab.cc>>.
- BLUHM, Thiago D. Faria. **CLF Sentimento Model**. [S.l.: s.n.]. Hugging Face. Acesso em: 23-02-2025, Disponível em: <<https://huggingface.co/tbluhm/clf-sentimentos-cmts>>.
- BYTEBIO. **ByteBio ChatGPT integration for WhatsApp**. [S.l.: s.n.]. Acesso em: 23-02-2025, Disponível em: <<https://www.bytebio.com/whatsapp-chatgpt>>.
- CONECTAGABINETE. **ConectaGabinete**. [S.l.: s.n.]. Acesso em: 23-02-2025, Disponível em: <<https://conectagabinete.com.br>>.
- CRMGABINETE. **CRMGabinete**. [S.l.: s.n.]. Acesso em: 23-02-2025, Disponível em: <<https://www.crmgabinete.com.br>>.
- DEVELOPERS, Google. **Google Apps Script**. [S.l.: s.n.]. Acesso em: 23-02-2025, Disponível em: <<https://developers.google.com/apps-script?hl=pt-br>>.
- DEVELOPERS, Meta for. **WhatsApp Solution Providers**. [S.l.: s.n.]. Acesso em: 23-02-2025, Disponível em: <<https://developers.facebook.com/docs/whatsapp/solution-providers/>>.
- DOCKER. **Docker**. [S.l.: s.n.]. Acesso em: 23-02-2025, Disponível em: <<https://www.docker.com>>.
- ELEGIS. **Elegis**. [S.l.: s.n.]. Acesso em: 23-02-2025, Disponível em: <<https://elegis.com.br>>.
- EVANGELISTA, Rafael; BRUNO, Fernanda. WhatsApp and political instability in Brazil: targeted messages and political radicalisation. **Internet Policy Review**, v. 8, n. 4, p. 1–23, 2019. Disponível em: <<https://hdl.handle.net/10419/214094>>. Acesso em: 23 fev. 2025.
- FLEX. **Flex**. [S.l.: s.n.]. Acesso em: 23-02-2025, Disponível em: <<https://flexag.com.br>>.
- GOOGLE. **Material Design**. [S.l.: s.n.]. Acesso em: 23-02-2025, Disponível em: <<https://material.io>>.

GRAZZIOLI, G. **CLS Sentimento Sebrae Model**. [S.l.: s.n.]. Hugging Face.

Acesso em: 23-02-2025, Disponível em:

<https://huggingface.co/ggrazzioli/cls_sentimento_sebrae>.

INOGIC. **WhatsApp4Dynamics Documentation**. [S.l.: s.n.]. Acesso em:

23-02-2025, Disponível em: <<https://docs.inogic.com/whatsapp4dynamics>>.

LIU, Yixin; LIU, Pengfei. SimCLS: A Simple Framework for Contrastive Learning of Abstractive Summarization. **CoRR**, abs/2106.01890, 2021. Disponível em:

<<https://arxiv.org/abs/2106.01890>>. Acesso em: 23 fev. 2025.

LLAMAINDEX. **LlamaIndex Query Engine**. [S.l.: s.n.]. Disponível em:

<https://docs.llamaindex.ai/en/stable/module_guides/deploying/query_engine/>

Acesso em : 23 fev. 2025..

MAO, Yanying; LIU, Qun; ZHANG, Yu. Sentiment analysis methods, applications, and challenges: A systematic literature review. **Journal of King Saud**

University - Computer and Information Sciences, v. 36, n. 4, p. 102048,

2024. Disponível em:

<<https://www.sciencedirect.com/science/article/pii/S131915782400137X>>. Acesso em: 23 fev. 2025.

MUI. **MUI: The React component library you always wanted**. [S.l.: s.n.].

Acesso em: 23-02-2025, Disponível em: <<https://mui.com>>.

NIELSEN, Jakob. **10 Usability Heuristics for User Interface Design**.

[S.l.: s.n.]. Nielsen Norman Group. Acesso em: 23-02-2025, Disponível em:

<<https://www.nngroup.com/articles/ten-usability-heuristics/>>.

OLIVEIRA, Angela Camari de. **WhatsApp Is The Main Tech Platform For Brazilian Smallest Businesses**. [S.l.: s.n.]. Forbes. Acesso em: 23-02-2025,

Disponível em:

<<https://www.forbes.com/sites/angelicamarideoliveira/2024/02/28/whatsapp-is-the-main-tech-platform-for-brazilian-smallest-businesses/>>.

PINATTI DE CARVALHO, Apécido Fabiano. **Thematic Analysis for**

Interactive Systems Design: A Practical Exercise. [S.l.]: European Society for Socially Embedded Technologies (EUSSET), 2021. Proceedings of 19th

European Conference on Computer-Supported Cooperative Work. DOI:

10.18420/ecscw2021_wsmc06.

PUPPETEER. **Puppeteer**. [S.l.: s.n.]. Acesso em: 23-02-2025, Disponível em:

<<https://pptr.dev>>.

PYNPUT. **pynput Package**. [S.l.: s.n.]. Python Package Index. Acesso em:

23-02-2025, Disponível em: <<https://pypi.org/project/pynput/>>.

SWELLER, John. Cognitive load theory, learning difficulty, and instructional design. **Learning and Instruction**, v. 4, n. 4, p. 295–312, 1994. ISSN 0959-4752. DOI: [https://doi.org/10.1016/0959-4752\(94\)90003-5](https://doi.org/10.1016/0959-4752(94)90003-5). Available from:

<[https:](https://www.sciencedirect.com/science/article/pii/0959475294900035)

[//www.sciencedirect.com/science/article/pii/0959475294900035](https://www.sciencedirect.com/science/article/pii/0959475294900035)>.

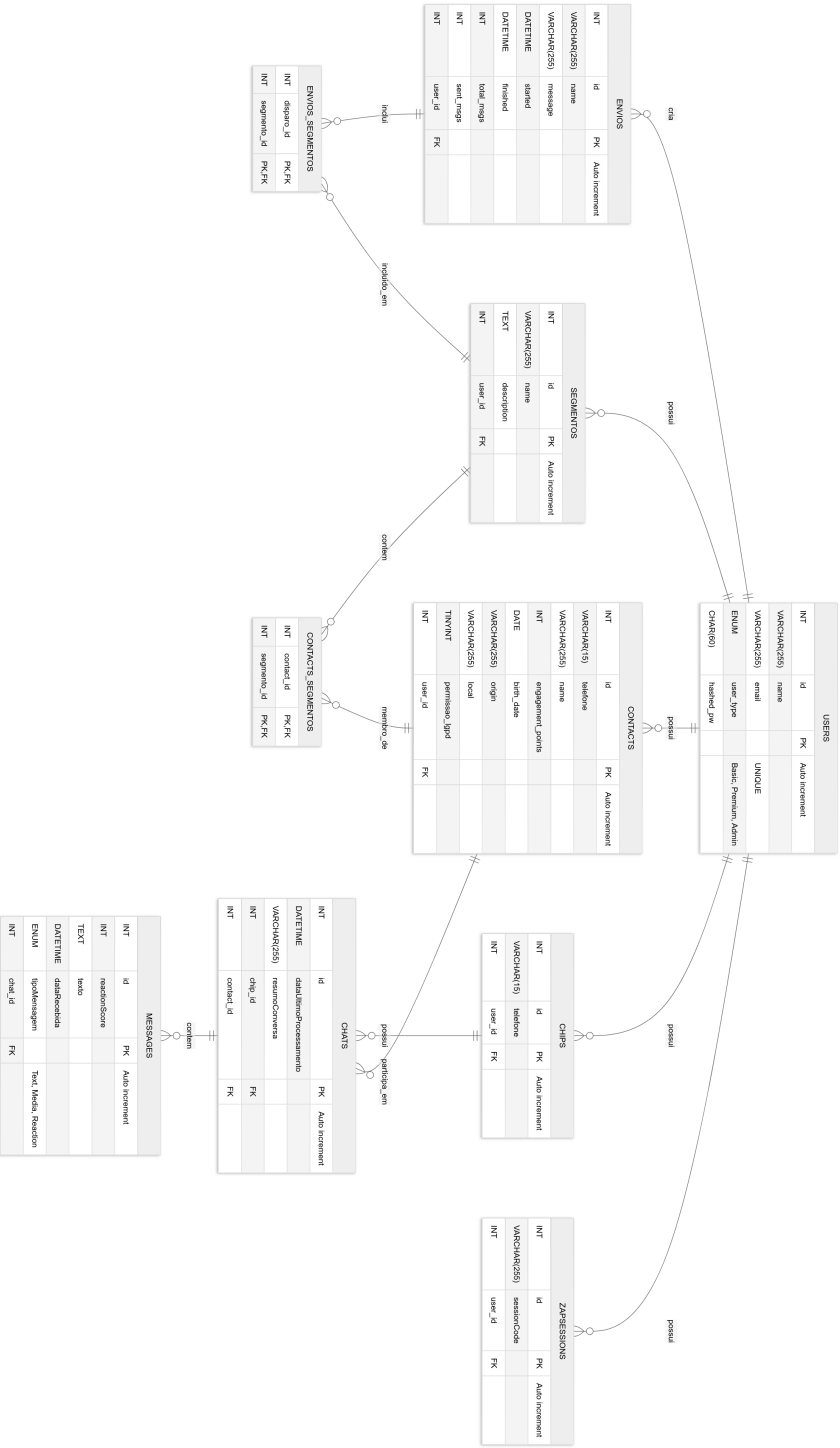
TIME, Mobile. **Panorama Mensageria no Brasil**. Acesso em: 19-02-2025, Disponível em: <<https://www.mobiletime.com.br/pesquisas/mensageria-no-brasil-marco-de-2024/>>.

TRACKMOB. **TrackMob CRM**. [S.l.: s.n.]. Acesso em: 23-02-2025, Disponível em: <<https://trackmob.com.br/produtos/crm/>>.

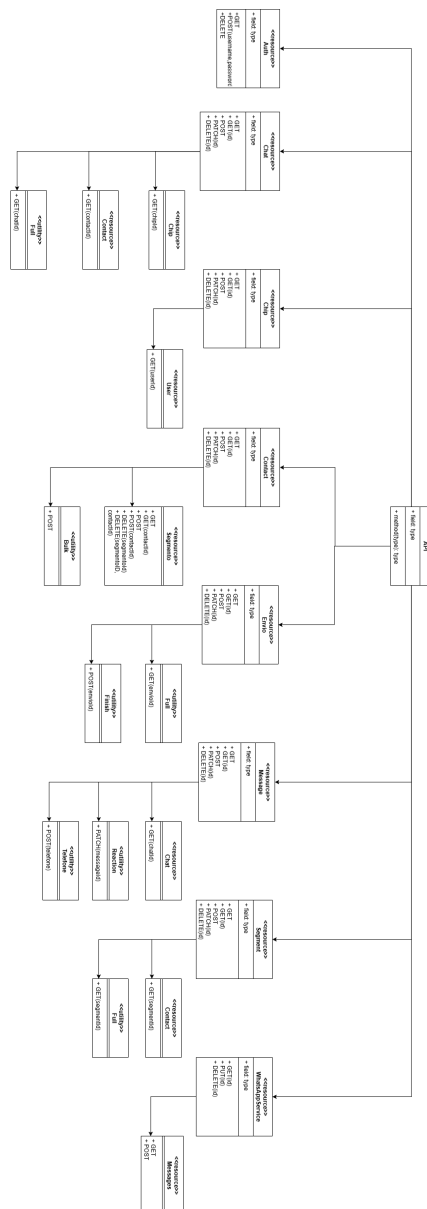
TWILIO. **AI Sentiment Analysis Chatbot for WhatsApp/SMS with OpenAI**. [S.l.: s.n.]. Acesso em: 23-02-2025, Disponível em: <<https://www.twilio.com/en-us/blog/ai-sentiment-analysis-chatbot-whatsapp-sms-openai>>.

ZOHO. **Zoho CRM para Organizações sem Fins Lucrativos**. [S.l.: s.n.]. Acesso em: 23-02-2025, Disponível em: <<https://www.zoho.com/pt-br/crm/non-profit/>>.

Apêndice A Diagrama ER da Base de Dados



Apêndice B Diagrama estrutura API



Apêndice C Entrevista Preliminar

1. Qual é sua ocupação atual?
 - (a) Do que consiste o dia a dia?
 - (b) Há quanto tempo você realiza essa função?
 - (c) Como é organizada a sua equipe?
 - (d) Quais aspectos do seu trabalho mais chamam sua atenção, sejam positivos ou negativos?
2. Como costuma ser o processo de envio de mensagens na sua equipe?
 - (a) Qual é o principal objetivo dos envios de mensagem?
 - (b) Você poderia mencionar o número aproximado de mensagens que costumam enviar por mês?
 - (c) Para aproximadamente quantas pessoas?
 - (d) O que poderia mudar sobre o processo de envio atual da sua equipe?
 - (e) Você sente falta de alguma informação no processo de envio de mensagens?
3. Como costuma ser o processo para lidar com as mensagens recebidas na sua equipe?
 - (a) Você poderia mencionar o número aproximado de mensagens que costumam receber por mês?
 - (b) De aproximadamente quantas pessoas?
 - (c) Quanto tempo demora, em média, para responder?
 - i. Como você costuma dividir a carga do trabalho de responder?
 - ii. Como você decide qual responder primeiro?
 - iii. Há alguma diferença de tratamento das mensagens dependendo do tempo da resposta?
 - (d) Você costuma reconhecer quem é cada contato à medida que responde?
 - i. Que informações vocês consultam sobre os contatos, à medida que vão respondendo?
 - ii. Que informações que vocês não consultam atualmente ajudariam na sua resposta?
4. Como vocês organizam a base de contatos na sua equipe?
 - (a) Como é feita a conexão entre a base de contatos e o software de envio de mensagem?
 - (b) Como sua equipe mede o nível de engajamento dos contatos?
 - (c) Como vocês subdividem ou estratificam os contatos da sua base?

5. Dentro da sua equipe, você já foi pessoalmente responsável por operar algum software de CRM ou disparador?
 - (a) Caso sim:
 - i. Como foi sua experiência com o produto?
 - ii. Qual o nome do software?
 - iii. Qual foi o processo de escolha desse produto?
 - iv. Por quanto tempo foi utilizado?
 - v. Qual sua percepção sobre o software utilizado, em termos do que facilita e do que dificulta o trabalho?
 - vi. O produto utilizado possui alguma funcionalidade para capturar, visualizar ou responder mensagens? Como você usava essas funcionalidades?
 - vii. O produto utilizado possui alguma funcionalidade para facilitar a priorização de mensagens urgentes ou respostas atrasadas? Se sim, como você acha que isso afetava seu trabalho?
 - viii. O produto utilizado possui alguma funcionalidade para visualizar a performance geral da equipe? Se sim, como você acha que isso afetava seu trabalho?
 - (b) Caso não:
 - i. Há algum impeditivo para você operar um software desse tipo?
 - ii. Você acha que conseguiria operar, dado as ferramentas certas? Se sim, quais são essas ferramentas?
 - iii. Você já esteve em alguma equipe que utilizava um software desse tipo? Se sim, entrevistador conduz seção (5.a)
6. Como é feita a avaliação de performance do envio e responsivo?
 - (a) Qual o processo utilizado pela sua equipe para identificar se um envio específico foi bem sucedido?
 - i. Quais dados vocês coletam?
 - ii. Qual é o processo utilizado para coletar esses dados?
 - (b) Qual o processo utilizado pela sua equipe para identificar se estão progredindo na conversão dos contatos?
 - (c) Tem algum outro indicador de performance que você considera que poderia ser utilizado?
7. Tem alguma etapa do processo de envio e recepção de mensagens que não abordamos ainda que considera essencial?
 - (a) Tem alguma funcionalidade ou requisito que você considera essencial para uma ferramenta de envio/resposta de mensagens?

- (b) E alguma funcionalidade que não considera essencial, mas facilitaria seu fluxo de trabalho?

Apêndice D Teste de Usabilidade

D.1 Roteiro do Teste

Introdução:

Bem vindo, nesse teste queremos avaliar a interface do sistema e como as pessoas interagem com ela. Iremos te dar acesso a um protótipo do sistema e te pediremos para realizar uma série de tarefas. Inicialmente, peço que tente realizar estas sem ajuda, só explorando a interface do sistema sozinho/sozinha. Peço também que tente pensar em voz alta, ou seja, falar em voz alta o seu processo de pensamento atrás das escolhas que faz, as suas reações e opiniões sobre o que aparece no sistema e etc. Ressalto que você não está sendo avaliado aqui e qualquer reação é válida. Lembro que tanto a tela do computador esse teste está sendo gravado.

Tarefas:

1. Faça login no sistema e verifique o status do seu plano.
2. Importar contatos do arquivo X no seu sistema, adicionando eles a um segmento novo chamado Y
3. Adiciona o contato X ao segmento que acabou de criar
4. Remova o contato Y do segmento
5. Realize um envio novo para esse segmento que acabou de criar, com uma mensagem de boas vindas
6. Você acabou de receber uma mensagem nova, veja se você consegue responder ela
7. Tente agora analisar as conversas anteriores. Como está sendo a interação com eles? Consegue identificar algum problema ou questão?
8. O contato Z disse que era engano, exclua ele da base.
9. Tenta identificar como tem sido os resultados e progresso da operação desse usuário.
10. Faça logout do sistema

D.2 Transcrição dos Testes Conduzidos

Participante 1: Disponível em <https://docs.google.com/document/d/153ufkQv8QRLnZo-40jvoBLpXALdF4QIt0LopqYa4M_s/edit?usp=sharing>

Participante 2: Disponível em <<https://docs.google.com/document/d/1ud0toSZh18g6LtQiHyImbno7gtZOQSQ1-ezPvBtR7k4/edit?usp=sharing>>

Apêndice E Entrevista Pós-Teste

E.1 Roteiro da Entrevista

1. Observações Gerais

- (a) Quais são suas observações e pensamentos gerais sobre o software que usou?
- (b) Quais tarefas das que foram dadas te ressaltaram como as mais difíceis?
- (c) Você acha que o software apresenta alguma melhoria em relação ao seu fluxo de trabalho atual?
- (d) Você acha que o software apresenta algo que atrapalharia seu fluxo de trabalho atual?
- (e) Havia algo que você esperava fazer e não conseguiu?
- (f) Houve alguma funcionalidade que você sentiu falta?

2. UX

- (a) Você se sentiu confusa em algum momento sobre como atingir seus objetivos?
- (b) Houve alguma tela que apresentou informações demais de uma vez?
- (c) Você sentiu que houve algo que poderia ser mais direto ou eficiente nos fluxos do sistema?

3. Humanização e Transparência

- (a) Você sentiu que conseguiu identificar quais contatos queriam ou não queriam receber mensagem? Como você identificou isso?
- (b) Você acha que conseguiria identificar se um certo envio foi mais bem sucedido que outro?
- (c) Você percebeu indicações de sentimento (cores/ícones) nas mensagens? Como isso afetou suas respostas?
- (d) No painel de dados, quais informações foram mais e menos úteis? Alguma métrica faltou?

4. Informações Extras

- (a) Você sente que as informações e dados gerados pelo sistema agregaram em algo?
- (b) Quais foram os dados mais valiosos? E os menos?
- (c) Você sente que há algum dado e informação que faltou no sistema?
- (d) Há mais alguma coisa que você gostaria de adicionar ou algum feedback que não foi abordado nas minhas perguntas?

E.2 Transcrições das Entrevistas

Participante 1: Disponível em <<https://docs.google.com/document/d/1hdL-83CTgkVXJADA1-02NyfmCCmTZFgRpQXiAYZKEVA/edit?usp=sharing>>

Participante 2: Disponível em <<https://docs.google.com/document/d/1ySR0BzD0yqiGE7GDSNbDJvVZoSYbB1Le4QH0952cEHQ/edit?usp=sharing>>

Apêndice F Wireframes

Disponível em: <https://drive.google.com/file/d/1LkoON0pJbuwUgWBVDhkziq2ZbpBvNFq_/view?usp=sharing>

Apêndice G Termo de Consentimento Livre e Esclarecido (TCLE)

Disponível em: <https://docs.google.com/document/d/1s_8028DKZ4nuvBLPd_1jHoWsBsf0q0wpF5K-hPqN-Kg/edit?usp=sharing>