

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

Atendimetrics

Software para auxiliar o gestor no
acompanhamento do nível do atendimento dos seus
estabelecimentos

Marco Antonio Borges Mateus

PROJETO FINAL DE GRADUAÇÃO

CENTRO TÉCNICO CIENTÍFICO – CTC

DEPARTAMENTO DE INFORMÁTICA

Curso de Graduação em Ciência da Computação

Rio de Janeiro, julho 2025



Marco Antonio Borges Mateus

Atendimetrics

**Software para auxiliar o gestor no acompanhamento do nível do atendimento
dos seus estabelecimentos**

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: Marcos Vianna Villas

Rio de Janeiro
julho de 2025

“Por ardua ad astra”

Agradecimentos

Agradeço, em primeiro lugar, aos meus pais, pelo apoio incondicional, por sempre acreditarem em mim e me incentivarem em todos os momentos da minha trajetória. São, para mim, uma fonte constante de inspiração e exemplo de dedicação, amor e perseverança. Sem a paciência, o carinho e a confiança que sempre depositaram em mim, este trabalho não seria possível.

Ao meu orientador, Marcos Villas, por estar sempre disponível para sanar dúvidas e contribuir com o desenvolvimento deste projeto, pela paciência, compreensão e por exercer a profissão com tanta dedicação.

À minha noiva, Giovanna, por sempre estar presente quando precisei, pela paciência durante os momentos mais estressantes e pelo apoio incondicional. Seu amor e compreensão foram essenciais para a conclusão deste trabalho.

Estendo minha gratidão a todos os professores e membros da faculdade, que contribuíram para minha formação acadêmica e pessoal, compartilhando conhecimentos, experiências e valores que foram fundamentais para a construção desta jornada.

Por fim, dedico um agradecimento especial ao meu filho, que, apesar de uma passagem breve por este mundo, deixou em meu coração um amor eterno e uma força imensurável para seguir em frente. Sua lembrança é e sempre será uma inspiração para cada conquista da minha vida, e guardo a certeza de que, um dia, nossos caminhos voltarão a se encontrar. Até.

Resumo

Mateus, Marco Antonio Borges. Marcos Vianna Villas. *Atendimetrics*, Projeto Final II (INF1951). Rio de Janeiro, 2025, 47 páginas. Relatório de Projeto Final II – Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

Este trabalho, consiste no desenvolvimento do software *Atendimetrics*, o qual visa ajudar na gestão do atendimento, torná-la mais dinâmica e personalizável. O software é responsável por transcrever os áudios, os quais contêm os atendimentos e a partir dessas transcrições ele gera algumas métricas para auxiliar o gestor a avaliar o nível do atendimento em seu estabelecimento.

Palavras-chave

Atendimento, gestão, métricas, transcrição de áudio, software

Abstract

Mateus, Marco Antonio Borges. Marcos Vianna Villas. *Atendimetrics*, Undergraduate Thesis II (INF1951). Rio de Janeiro, 2025, 47 pages. Undergraduate Thesis II – Computer Department. Pontifícia Universidade Católica do Rio de Janeiro.

This work consists of the development of the *Atendimetrics* software, which aims to assist in customer service management, making it more dynamic and customizable. The software is responsible for transcribing audio recordings that contain customer service interactions, and based on these transcriptions, it generates metrics to help the manager assess the level of service provided at their establishment.

Keywords

Customer service, management, metrics, audio transcription, software

Sumário

1.	Introdução.....	1
2.	Situação Atual.....	1
3.	Objetivos.....	3
4.	Metodologia.....	4
4.1	Etapas.....	4
4.2	Cronograma.....	4
4.2.1	Cronograma Planejado.....	4
4.2.2	Cronograma Realizado.....	7
5.	Estudos Iniciais.....	8
5.1	Obtenção dos Áudios.....	9
5.2	Processamento de áudio para texto.....	9
5.3	Processamento dos áudios: atendentes e diálogo.....	11
5.4	Geração de Métricas.....	12
5.5	Utilização das Métricas.....	13
5.6	Cenários hipotéticos de uso do software.....	13
6.	Requisitos.....	14
6.1	Requisitos funcionais.....	14
6.2	Requisitos não funcionais.....	15
6.3	Conformidade com a Lei Geral de Proteção de Dados (LGPD).....	15
7.	Casos de uso.....	16
8.	Modelagem de dados.....	26
8.1	UML.....	26
8.2	Modelo relacional de dados.....	26
9.	Desenvolvimento do software.....	28
10.	Interface do sistema.....	31
11.	Testes e avaliações.....	35
11.1	Feedback.....	36
12.	Considerações finais.....	37
13.	Referências bibliográficas.....	38

Lista de Figuras

1. Figura 1 – Custo histórico de memória e armazenamento.....	2
2. Figura 2 – Performance GPU por dólar gasto.....	3
3. Figura 3 – Diagrama de casos de uso.....	25
4. Figura 4 – Diagrama UML.....	26
5. Figura 5 – Fluxo do software.....	31
6. Figura 6 – Tela de login.....	32
7. Figura 7 – Tela de cadastro de usuário.....	32
8. Figura 8 – Tela após login bem-sucedido.....	33
9. Figura 9 – Tela de cadastro de nova palavra-chave.....	34
10. Figura 10 – Tela de palavras chaves cadastradas.....	34
11. Figura 11 – Tela do dashboard, onde é possível visualizar as métricas.....	35

Lista de Abreviaturas e Siglas

- ERP - *Enterprise Resource Planning*
- PDV - Ponto de Venda
- SAC - Serviço de Atendimento ao Consumidor

1. Introdução

O panorama do varejo está a evoluir rapidamente devido ao aumento da concorrência, ao aumento das fusões e aquisições, clientes mais informados e com expectativas mais elevadas e o mundo digital. Neste contexto, os varejistas precisam se destacar, satisfazendo as necessidades dos seus clientes de forma mais eficaz do que os seus concorrentes. [1]

Além disso, na era do mundo digital em que vivemos, um cliente insatisfeito tem o poder de passar essa insatisfação para milhares de outros clientes em questão de minutos, podendo trazer para as empresas grandes perdas, tanto financeiras como reputacionais, o que reforça a necessidade de a empresa prevenir e ser ágil na resposta. Enfim, é vital para as empresas procurarem avaliar e aprimorar o atendimento ao cliente.

2. Situação Atual

No segmento de lojas de conveniência, uma das formas mais conhecidas e praticadas para a avaliação do atendimento é a contratação de uma empresa terceirizada para fazer o que chamamos de cliente misterioso ou cliente oculto, na qual sua equipe vai até o estabelecimento e faz uma avaliação do atendimento, verificando os pontos-chaves que a contratante deseja avaliar.

Porém esse método possui alguns pontos negativos, dentre eles os principais:

- Alto custo (\$) ¹
- Demora para se ter o feedback, ou seja, se você deseja saber hoje como está seu atendimento você precisa esperar pelo processo todo para ter essa resposta. (Contratação da empresa -> Empresa contratada fazer a avaliação -> Empresa contratada informar e enviar o resultado da avaliação). Isso acarreta morosidade na resposta para correções de problemas que venham a existir.
- Para se ter uma avaliação constante e fidedigna de como está o nível do atendimento é necessário ter uma alta frequência de avaliações, o que reforça o elevado custo.

Uma outra forma é a empresa possuir um canal direto de comunicação com o cliente, como por exemplo um SAC, porém a empresa fica dependente do cliente ter a boa vontade de comunicar e lhe dar *feedbacks*. Ademais, possui alguns pontos negativos:

- Demora para se ter o *feedback*, ou seja, se o gestor deseja saber hoje como está o atendimento, ele depende de quando o cliente dará o retorno.
- O feedback pode estar enviesado e muito provavelmente não reflete a totalidade da situação.

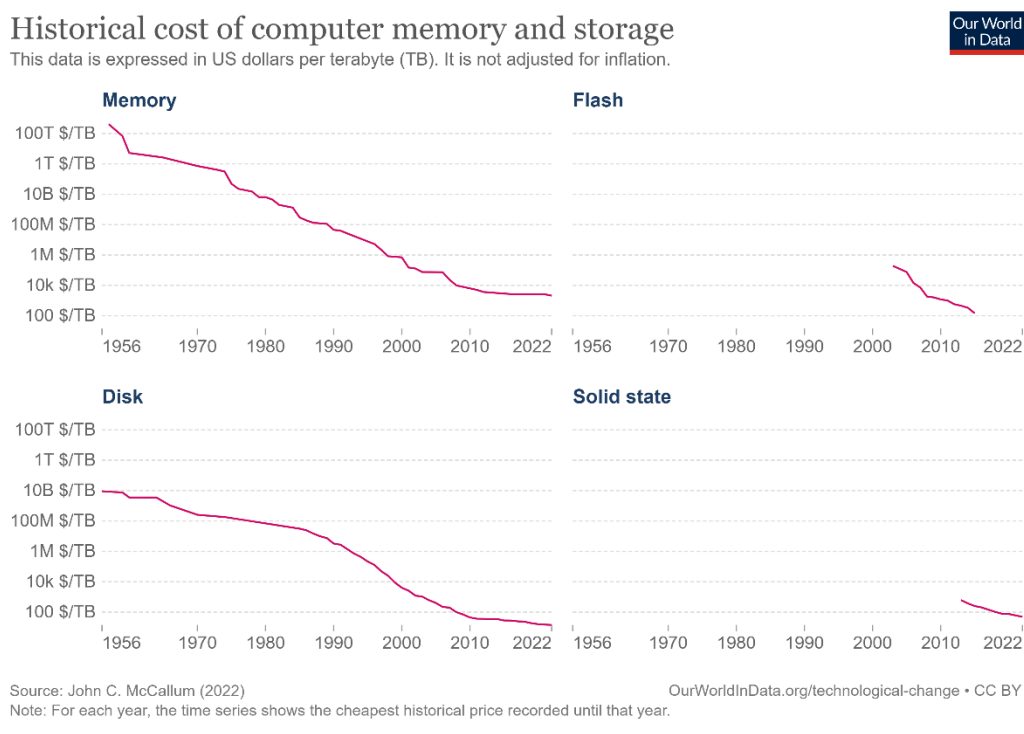
¹ Valor de R\$1.500,00 por estabelecimento, para uma visita mensal. (Baseado em proposta recebida de 2021)

Ambas as formas possuem pontos negativos e, sendo em comum, um *delay* significativo entre os *feedbacks* e os acontecimentos que geraram os mesmos, fazendo que o gestor tome medidas reativas com alto tempo de resposta para os problemas identificados. Portanto, fica evidente o valor de uma ferramenta que forneça uma avaliação rápida do atendimento.

Com o avanço exponencial da tecnologia, tanto no que diz respeito ao *hardware* como ao *software*, tornou-se viável fazer o acompanhamento dos atendimentos de forma rápida, através da captura direta das informações e do processamento delas, de forma que o gestor consiga tomar medidas mais rápidas, podendo assim evitar os problemas antes que eles escalonem.

As imagens abaixo, figura 1 e 2, representam alguns desses avanços tecnológicos que hoje permitem que essa nova solução seja viável.

Figura 1 – Custo histórico de memória e armazenamento

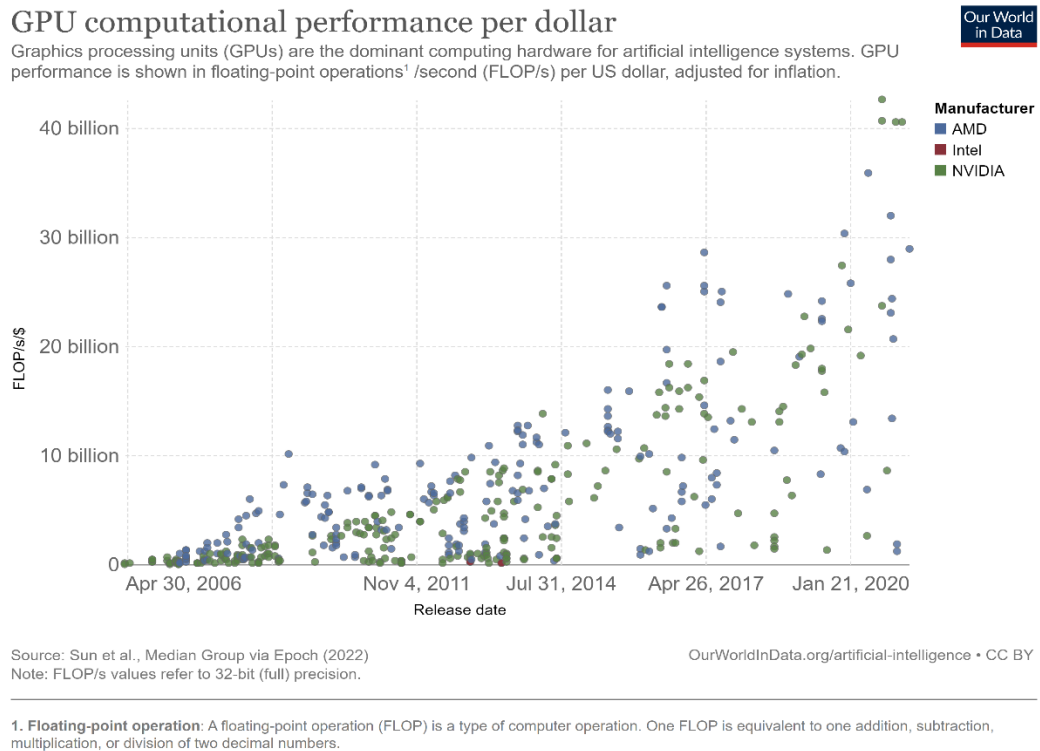


Fonte: Our World in Data. Disponível em:

<https://ourworldindata.org/grapher/historical-cost-of-computer-memory-and-storage>

Acesso em: 18 setembro de 2023.

Figura 2 – Performance da GPU por dólar gasto



Fonte: Our World in Data. Disponível em:

<https://ourworldindata.org/grapher/gpu-price-performance>

Acesso em: 18 setembro de 2023.

3. Objetivos

O objetivo do presente projeto é desenvolver um software que consiga fornecer ao gestor de comércio varejista, com foco em mercearias/lojas de conveniência/supermercados, uma avaliação do atendimento ao cliente de forma rápida, sem viés do atendente ou do cliente (baseado em métricas pré-estabelecidas) e de baixo custo. Isso será feito a partir da captura do áudio em tempo real através de um microfone e/ou webcam, a(o) qual ficará estabelecida no caixa, onde se concentra a maioria das interações dos clientes com os atendentes, e a partir do mesmo será feita uma análise a qual irá gerar as métricas que auxiliarão o gestor a tomar melhores decisões baseadas em dados.

4. Metodologia

Essa seção trata sobre as etapas realizadas anteriormente e durante o desenvolvimento do software e como elas foram distribuídas no período.

4.1. Etapas

1. Estudo de como os áudios poderiam ser obtidos e gravados.
2. Estudo de como os áudios serão convertidos para texto.
3. Estudo de como os áudios serão processados com a finalidade de identificar os atendentes e os diálogos com os clientes.
4. Estudo de quais métricas serão utilizadas e como elas serão geradas.
5. Estudo sobre quais são as possíveis necessidades e exigências do público-alvo do software e a partir desse estudo fazer um relatório dos requisitos.
6. Relatório sobre os casos de uso do software.
7. Modelagem dos dados.
8. Estudo sobre as bibliotecas a serem utilizadas e modelagem de interface.
9. Desenvolvimento do software.
10. Teste piloto por uma semana e ao término um terceiro irá dar um feedback sobre o software.

4.2. Cronograma

4.2.1. Cronograma Planejado

Projeto Final – Parte 1	Outubro 2023				Novembro 2023				Dezembro 2023			
1. Estudo de como os áudios serão obtidos e gravados.												
2. Estudo de como os áudios serão convertidos para texto												

3. Estudo de como os áudios serão processados com a finalidade de identificar os atendentes e os diálogos com os clientes.												
4. Estudo de quais métricas serão utilizadas e como elas serão geradas.												
5. Requisitos												

Projeto Final – Parte 2	Março 2024				Abril 2024				Maio 2024				Junho 2024				Julho 2024			
7. Casos de Uso																				
8. Modelagem de dados																				
9. Modelagem da Interface																				
10. Desenvolvimento do Software																				
11. Implementação dos testes																				
12. Teste piloto																				

4.2.2. Cronograma Realizado

Projeto Final – Parte 1	Outubro 2023				Novembro 2023				Dezembro 2023			
1. Estudo de como os áudios serão obtidos e gravados.												
2. Estudo de como os áudios serão convertidos para texto												
3. Estudo de como os áudios serão processados com a finalidade de identificar os atendentes e os diálogos com os clientes.												
4. Estudo de quais métricas serão utilizadas e como elas serão geradas.												
5. Requisitos												

Projeto Final – Parte 2	Março 2025				Abril 2025				Maio 2025				Junho 2025				Julho 2025			
7. Casos de Uso																				
8. Modelagem de dados																				
9. Modelagem da Interface																				
10. Desenvolvimento do Software																				
11. Implementação dos testes																				
12. Teste piloto																				

5. Estudos Iniciais

Essa seção trata sobre os estudos iniciais realizados para identificar quais seriam as melhores formas para se conseguir obter os áudios, transformar o áudio em texto, processar os diálogos e as gerações das métricas de forma a auxiliar o melhor andamento na hora do desenvolvimento do software.

5.1. Obtenção dos áudios

Essa etapa do projeto consistiu em analisar as diferentes interações dos clientes em uma loja de conveniência, as quais foram categorizadas em quatro cenários distintos. Além disso, teve-se como objetivo entender como os clientes se comportam durante suas visitas, desde a entrada na loja até a possível interação com os atendentes e tentar identificar padrões de comportamento do cliente durante o processo de atendimento.

Considerando as possibilidades de interação do cliente com a loja, as quais podem ser descritas da seguinte forma:

- Cenário 1: O cliente entra na loja olha os produtos, não encontra o que deseja e vai embora.
- Cenário 2: O cliente entra na loja olha os produtos, não encontra o que deseja e fala com o atendente.
- Cenário 3: O cliente entra na loja e olha os produtos, ele pega e leva para o atendente e paga.
- Cenário 4: O cliente entra na loja olha os produtos, o que ele deseja precisa ser pedido, então ele vai até o atendente faz o pedido e paga.

O software, que foi desenvolvido, captura o áudio através de um microfone ou webcam instalado no ponto de venda, que é onde ocorre, na maioria das vezes, a troca de palavras do cliente com o atendente no ponto de venda. Cada gravação de áudio, que cobre um dia inteiro de interações será processada no dia seguinte no horário da madrugada

5.2. Processamento de áudio para texto

Essa etapa do projeto consistiu em buscar um algoritmo que seja open source e eficiente para transformar um áudio português (Brasil) em texto, a partir do qual iremos fazer todas as análises necessárias.

Durante a pesquisa por qual algoritmo utilizar, foi identificada uma competição de algoritmos de machine learning com o foco em transformar áudio em texto, feita na plataforma huggingface2 em 2022, a qual tinha entre os participantes empresas como a Google e a Microsoft. A competição avaliou os algoritmos seguindo as métricas WER e CER.

A WER (*Word Error Rate*) é derivada da distância de Levenshtein³, trabalhando no nível da palavra em vez do nível do fonema.

A taxa de erro de palavra pode então ser calculada como:

$$WER = (S + D + I) / N = (S + D + I) / (S + D + C)$$

² <https://huggingface.co/>

³ A distância de Levenshtein é uma métrica de string para medir a diferença entre duas sequências. Normalmente, a distância Levenshtein entre duas palavras é o número mínimo de edições de um caractere (inserção, exclusão, substituição) necessárias para transformar uma palavra na outra.[2]

Onde S é o número de substituições, D é o número de deleções, I é o número de inserções, C é o número de palavras corretas, N é o número de palavras na referência ($N=S+D+C$).

Este valor indica o número médio de erros por palavra de referência. Quanto menor o valor, melhor o desempenho do sistema ASR (*Automatic Speech Recognition*), sendo um WER de 0 sendo uma pontuação perfeita.

O CER (*Character Error Rate*) é semelhante à taxa de erro de palavra (WER), mas opera em caracteres em vez de palavras.

A taxa de erro de caractere pode ser calculada como:

$$\text{CER} = (S + D + I) / N = (S + D + I) / (S + D + C)$$

Onde S é o número de substituições, D é o número de deleções, I é o número de inserções, C é o número de caracteres corretos, N é o número de caracteres na referência ($N=S+D+C$).

Este valor é frequentemente associado à porcentagem de caracteres que foram incorretamente previstos. Quanto menor o valor, melhor o desempenho do sistema ASR, sendo um CER de 0 sendo uma pontuação perfeita.

O algoritmo vencedor da competição para a língua portuguesa foi do participante Jonatas Grosman [3], o algoritmo utiliza o modelo de rede neural Wav2Vec2⁴. Wave2vec2 é um modelo de rede neural convolucional, ele usa um método de treinamento chamado aprendizado de representação contrastiva (*Contrastive Representation Learning*, CRL).

O CRL é um método de aprendizado profundo que usa um processo de reforço para treinar um modelo a gerar representações de dados que são discriminativas. No caso de Wave2vec2, o modelo é treinado para gerar representações de áudio que são discriminativas entre diferentes sons e palavras.

Na última versão disponível do algoritmo [4], ele apresenta as seguintes performances:

Test WER on Common Voice 8⁵: 8,7%

Test CER on Common Voice 8: 2,55%

A escolha do modelo desenvolvido por Grosman não se baseia apenas em seu desempenho superior nas métricas WER e CER, mas por todo conteúdo demonstrado em sua tese '*Assessing the Robustness of Large Pre-trained Models in the Speech Recognition*' (2022) [5]. Grosman investigou como modelos pré-treinados de grande escala mantêm sua performance em condições adversas, incluindo ambientes ruidosos e falas com diferentes sotaques e velocidades, características típicas do ambiente de uma loja de conveniência. Sua pesquisa demonstrou que o modelo Wav2Vec2, quando adequadamente ajustado para o

⁴ https://huggingface.co/docs/transformers/model_doc/wav2vec2

⁵ Common Voice 8 é a versão 8 da base de dados de fala crowdsourced mantido pela Mozilla.

português brasileiro, apresenta perda mínima de performance mesmo em cenários com ruído de fundo significativo, tornando-o ideal para a aplicação onde os áudios são capturados em um ambiente comercial real, com interferências de diversos ruídos.

5.3. Processamento dos áudios: atendentes e diálogos

Essa etapa do projeto consistiu em pesquisar e analisar as possibilidades de como processar os áudios e os textos gerados, com o objetivo de ao final ter um arquivo formatado e que através dele seja possível extrair de forma fácil e objetiva todas as informações que serão necessárias para conseguir gerar as métricas necessárias.

O áudio capturado tem duração de 24 horas, abrangendo todas as interações realizadas durante um dia de vendas. Esse áudio é processado no dia seguinte, preferencialmente durante a madrugada, período de menor movimento na loja. Durante o processamento, é realizada uma análise das vozes com o objetivo de identificar os atendentes. Para isso, os trechos de fala são comparados com amostras previamente armazenadas no banco de dados, onde estarão armazenadas as amostras das vozes dos atendentes, sendo atribuída ao atendente cuja voz apresentar o maior coeficiente de similaridade. Após a identificação, o áudio é convertido em texto por meio do algoritmo de transcrição selecionado.

O objetivo é ter após todas as manipulações feitas no arquivo de áudio uma saída no seguinte formato:

12:38:53 Atendente 1: "Falas do atendente"

Para conseguir chegar a esse objetivo existem duas possíveis abordagens, sendo elas:

A primeira abordagem é quebrar o problema em 3 partes:

- Identificação das vozes: A cada voz diferente que aparecer no áudio será feito uma comparação com o sample das vozes dos atendentes que estará salvo em um banco de dados, caso a comparação tenha um coeficiente de similaridade superior a um valor estipulado consideraremos como a voz sendo do respectivo atendente, caso contrário será considerado como ruído de terceiros.

Assim teremos a seguinte saída:

12:38:53 Atendente 1

12:44:59 Atendente 2

...

12:47:57 Atendente 3

- Transcrição dos áudios: Através do algoritmo de áudio-para-texto iremos converter todo o arquivo áudio em texto, temos a seguinte saída:

12:38:53 texto

12:44:59 texto

- Junção das informações: O *timestamp* será utilizado como ponto de interseção entre os dados de identificação de voz e os dados de transcrição. Com base nele, será possível combinar as duas saídas e gerar o resultado no formato desejado.

12:38:53 Atendente 1: “Falas do atendente”

A segunda abordagem consiste para cada amostra de voz dos atendentes no banco de dados, faz-se uma quebra do arquivo áudio filtrando-o pela amostra, por exemplo:

Amostra do atendente 1, faz uma comparação com todo o arquivo de áudio e todo trecho que tiver uma correspondência positiva é salva em um novo arquivo, no final teremos um arquivo de áudio para cada atendente que foi identificado no áudio original e a partir desses novos áudios faz-se a transcrição já sabendo quem falou.

5.4. Geração de Métricas

Essa etapa consistiu na pesquisa por métricas que seriam consideradas como essenciais para a avaliação de um atendimento e que poderiam ser avaliadas considerando o escopo do projeto. Para isso, além da pesquisa normal também foi conversado com um gestor de lojas de conveniência para ter uma maior noção do que é avaliado por eles quando falamos de atendimento.

A partir dos arquivos textos gerados, eles serão percorridos através de um algoritmo que irá comparar as palavras do arquivo com as palavras chaves. Inicialmente, o software terá algumas métricas já pré-definidas, porém ele permitirá a customização e criação de novas métricas.

As métricas já pré-definidas serão:

- Métrica de Educação:

Será feito uma comparação com a quantidade de vezes que as palavras chaves aparecem versus quantas vendas foram realizadas no dia, essa informação será dada por um software terceiro, por exemplo: tivemos 300 ocorrências de “bom dia” no texto e temos a informação que houve 600 vendas, assim temos um índice de 50% de educação.

- Métrica de Oferecimento:

Será feito uma comparação com a quantidade de vezes a promoção aparece versus quantas vendas foram realizadas no dia, essa informação será dada por um software terceiro,

por exemplo: tivemos 300 aparições de “promoção” no texto e temos a informação que houve 600 vendas, assim temos um índice de 50% de oferecimento.

5.5. Exemplos de uso

A partir das informações salvas no banco de dados, teremos os possíveis cenários de uso:

- Envio automático de um relatório para algum e-mail cadastrado de forma diária, semanal, quinzenal, mensal, trimestral e anual.
 - Diário: Mostra como foram as métricas no dia e faz um comparativo com o dia anterior.
 - Semanal: Mostra como foram as métricas da semana e faz um comparativo com a semana anterior.
 - Quinzenal: Mostra como foram as métricas da quinzena e faz um comparativo com a quinzena anterior.
 - Mensal: Mostra como foram as métricas do mês e faz um comparativo com o mês anterior.
 - Trimestral: Mostra como foram as métricas do trimestre e faz um comparativo com o trimestre anterior.
 - Anual: Mostra como foram as métricas do ano e faz um comparativo com o ano anterior.

Esse relatório, independentemente da versão dele, pode apresentar as informações de uma forma mais macro, a qual mostra a performance do atendimento geral da loja ou de uma forma mais micro, a qual mostra a performance individual de cada atendente da loja. Considerando uma hierarquia diretor -> gerente -> atendente, o acompanhamento macro seria voltado para os diretores para uma gestão mais holística enquanto o acompanhamento micro seria mais voltado para os gerentes fazerem o acompanhamento individual da performance da sua equipe.

Além disso, através do dashboard interativo, será possível acompanhar essas métricas filtrando pelo período desejado, sendo possível fazer comparações entre períodos e verificar sua evolução além de filtrar pelo funcionário, podendo assim acompanhar também a performance individual para cada atendente.

5.6. Cenários hipotéticos de uso do software

A empresa deseja implementar uma nova oferta na loja e precisa que essa nova oferta seja oferecida em pelo menos 90% das vendas. Com o software o(a) diretor(a) pode

selecionar o período desejado e verificar qual foi esse percentual, e assim poder cobrar ou parabenizar o gerente pela performance.

No caso do gerente, ele consegue ter uma informação mais granular, verificar como está a performance de cada atendente e a partir dela poder parabenizar ou cobrar um melhor atendimento de algum atendente que esteja abaixo da meta.

A empresa deseja garantir que todos seus clientes sejam bem atendidos, ou seja, que recebam um “bom dia”, “boa tarde”, “boa noite”, “volte sempre” em 100% das vendas. Com o software o(a) diretor(a) pode selecionar o período desejado e verificar qual foi esse percentual, e assim poder cobrar ou parabenizar o gerente pela performance.

No caso do gerente, ele consegue ter uma informação mais granular, verificar como está a performance de cada atendente e a partir dela poder parabenizar ou cobrar um melhor atendimento de algum atendente que esteja abaixo da meta.

6. Requisitos

6.1. Requisitos funcionais

Áudio	
RF1	O sistema ser capaz de capturar o áudio via microfone.
RF2	O sistema deve ser capaz de armazenar os áudios.
RF3	O sistema deve ser capaz de identificar as falas dos atendentes no áudio e convertê-las em texto.
RF4	O sistema deve ser capaz de armazenar “assinaturas” de voz dos atendentes
RF5	O sistema deve ser capaz de identificar atendentes

Login	
RF6	O sistema deve ser capaz de criar usuários.
RF7	O sistema deve ser capaz de registrar tentativas de login, tanto as bem-sucedidas quanto as fracassadas, para fins de auditoria e segurança.

Texto	
RF8	O sistema deve ser capaz de processar e analisar os textos em busca das palavras-chaves.

Métricas	
RF9	O sistema deve permitir o cadastro de novas palavras-chaves.
RF10	O sistema deve ser capaz de gerar as métricas cadastradas.

RF11	O sistema deve ser capaz de gerar relatórios gerenciais com as métricas em pdf.
------	---

Interface	
RF12	O sistema deve ter uma tela contendo um dashboard onde o usuário consegue verificar as métricas.

Extras	
RF13	O sistema deve permitir o cadastro de e-mails para o recebimento dos relatórios.

6.2. Requisitos não funcionais

Disponibilidade	
RNF 1	O sistema deve estar disponível 99% do tempo.

Segurança e Privacidade	
RNF 2	O sistema deve estar de acordo com a LGPD.

Desempenho	
RNF 3	O sistema deve ser capaz de processar todos os áudios e textos dentro do período da madrugada (02:00 - 05:00)

Usabilidade	
RNF 4	O sistema deve ser de fácil de uso, incluindo uma interface intuitiva, documentação clara.

6.3. Conformidade com a Lei Geral de Proteção de Dados (LGPD)

O sistema realiza o tratamento de dados pessoais através da captura e processamento de gravações de voz, configurando-se como atividade sujeita à Lei nº 13.709/2018 (LGPD)[6]. A voz humana é considerada dado pessoal por permitir a identificação de pessoa natural, e sua gravação sistemática exige adequação aos princípios da finalidade, adequação, necessidade e transparência estabelecidos no Art. 6º da lei. Para os atendentes, a base legal para o tratamento fundamenta-se no Art. 7º, inciso V (execução de contrato de trabalho). Para os clientes, aplica-se o legítimo interesse do controlador (Art. 7º, inciso IX), desde que respeitados os direitos e liberdades fundamentais do titular, sendo imprescindível a afixação de avisos claros e ostensivos no estabelecimento informando sobre a gravação em curso.

O sistema deve implementar medidas de segurança administrativas e técnicas conforme Art. 46 da LGPD, incluindo: (i) criptografia dos dados sensíveis armazenados, especialmente as assinaturas de voz e transcrições; (ii) controle rigoroso de acesso com diferentes níveis de permissão, garantindo que apenas usuários autorizados acessem dados pessoais; (iii) registro de logs de todas as operações realizadas com dados pessoais para fins de auditoria.

7. Casos de uso

Descrição de Casos de Uso - UC01– Capturar áudio via microfone [RF 1]

Nome:	Capturar áudio via microfone
Objetivo:	Permitir que o sistema capture áudio através do microfone.
Atores:	Microfone
Pré-condição:	O sistema deve ter acesso a um microfone operacional.
Pós-condição (cenário de sucesso):	O sistema captura o áudio e o armazena temporariamente.
Pós-condição (cenário de insucesso):	O sistema informa ao usuário sobre a falha na captura do áudio.
Trigger:	O sistema é inicializado.
Fluxo Principal:	1. O sistema é inicializado. 2. O sistema acessa o microfone e começa a capturar o áudio. 3. O sistema armazena o áudio capturado.
Fluxos Alternativos:	
Fluxos de Exceção	
Pontos de Extensão:	
Regras de Negócio	O sistema deve capturar áudio em formato .wav

Descrição de Casos de Uso - UC02 – Armazenar áudios [RF 2]

Nome:	Armazenar áudios
Objetivo:	Armazenar áudios capturados para posterior processamento.
Atores:	Timer.
Pré-condição:	O áudio deve ter sido capturado.
Pós-condição (cenário de sucesso):	O áudio é armazenado no banco de dados.
Pós-condição (cenário de insucesso):	O sistema informa sobre a falha no armazenamento do áudio.
Trigger:	Áudio capturado pronto para armazenamento.
Fluxo Principal:	1. O sistema recebe o áudio capturado.

	2. O sistema armazena o áudio no banco de dados.
Fluxos Alternativos:	
Fluxos de Exceção	
Pontos de Extensão:	
Regras de Negócio	Os áudios devem ser armazenados de maneira segura e acessível.

Descrição de Casos de Uso - UC03 – Converter áudio em texto [RF 3]

Nome:	Converter áudio em texto
Objetivo:	Identificar as falas dos atendentes e convertê-las em texto.
Atores:	Gerente, Usuário Master.
Pré-condição:	Áudio capturado e armazenado.
Pós-condição (cenário de sucesso):	As falas são identificadas e convertidas em texto.
Pós-condição (cenário de insucesso):	O sistema informa sobre a falha na conversão do áudio em texto.
Trigger:	Áudio armazenado pronto para processamento.
Fluxo Principal:	1. O sistema acessa o áudio armazenado. 2. O sistema processa o áudio para identificar as falas. 3. O sistema converte as falas em texto. 4. O sistema salva as informações relevantes no banco de dados.
Fluxos Alternativos:	
Fluxos de Exceção	
Pontos de Extensão:	
Regras de Negócio	

Descrição de Casos de Uso - UC04 – Armazenar assinaturas de voz [RF 4]

Nome:	Armazenar assinaturas de voz
Objetivo:	Armazenar as assinaturas de voz dos atendentes.
Atores:	Gerente, Usuário Master.
Pré-condição:	Assinatura de voz capturada.
Pós-condição (cenário de sucesso):	A assinatura de voz é armazenada no banco de dados.
Pós-condição (cenário de insucesso):	O sistema informa sobre a falha no armazenamento da assinatura de voz.
Trigger:	Assinatura de voz capturada pronta para armazenamento.
Fluxo Principal:	1. O sistema recebe a assinatura de voz capturada.

	2. O sistema armazena a assinatura de voz no banco de dados.
Fluxos Alternativos:	
Fluxos de Exceção	
Pontos de Extensão:	
Regras de Negócio	As assinaturas de voz devem ser armazenadas de maneira segura e acessível.

Descrição de Casos de Uso - UC05 – Identificar atendentes [RF 5]

Nome:	Identificar atendentes
Objetivo:	Identificar atendentes a partir das assinaturas de voz.
Atores:	Gerente, Usuário Master.
Pré-condição:	Assinatura de voz capturada e armazenada.
Pós-condição (cenário de sucesso):	O atendente é identificado corretamente.
Pós-condição (cenário de insucesso):	O sistema informa sobre a falha na identificação do atendente.
Trigger:	Áudio pronto para análise de identificação.
Fluxo Principal:	1. O sistema acessa a assinatura de voz armazenada. 2. O sistema processa o áudio para identificar o atendente.
Fluxos Alternativos:	
Fluxos de Exceção	
Pontos de Extensão:	
Regras de Negócio	

Descrição de Casos de Uso - UC06 – Criar Usuário [RF 6]

Nome:	Criar Usuários
Objetivo:	Permitir que o Usuário Master crie novos usuários no sistema.
Atores:	Usuário Master
Pré-condição:	O Usuário Master deve estar autenticado.
Pós-condição (cenário de sucesso):	Um novo usuário é criado e armazenado no sistema com as permissões adequadas.
Pós-condição (cenário de insucesso):	O sistema informa ao Usuário Master sobre o erro na criação do usuário (por exemplo, informações faltando ou inválidas) e não cria o novo usuário.
Trigger:	O Usuário Master seleciona a opção de criar um novo usuário no sistema.
Fluxo Principal:	1. O sistema exibe a página para a criação de novos usuários. 2. O Usuário Master preenche as informações necessárias 3. O Usuário Master submete as informações. 4. O sistema valida as informações.

	5. O sistema cria o novo usuário com as permissões especificadas. 6. O sistema informa ao Usuário Master que o usuário foi criado com sucesso.
Fluxos Alternativos:	Usuário já existe: 1. Se o sistema identificar que o nome de usuário já está em uso, ele informa ao Usuário Master. 2. O Usuário Master pode alterar as informações e tentar criar o usuário novamente.
Fluxos de Exceção	Informações inválidas: 1. Se as informações submetidas não passarem na validação do sistema (por exemplo, senha muito fraca, usuário já existente...), o sistema informa ao Usuário Master sobre os erros específicos. 2. O Usuário Master corrige as informações e submete o formulário novamente.
Pontos de Extensão:	Editar permissões do usuário: 1. Após a criação do usuário, pode-se estender para um caso de uso onde o Usuário Master edita as permissões específicas para o novo usuário.
Regras de Negócio	R1: O e-mail do novo usuário deve ser único no sistema. R2: A senha do novo usuário deve atender aos critérios de segurança estabelecidos (por exemplo, tamanho, caracteres especiais). R3: O Usuário Master deve escolher pelo menos um tipo de permissão para o novo usuário (por exemplo, atendente, administrador, gerente). R4: Somente Usuários Master podem alterar o nível de permissão de um usuário.

Descrição de Casos de Uso - UC06.1 – Editar Usuário Existente [Extensão]

Nome:	Editar Usuário Existente
Objetivo:	Modificar informações de usuários já cadastrados.
Atores:	Usuário Master
Pré-condição:	O Usuário Master deve estar autenticado.
Pós-condição (cenário de sucesso):	Informações do usuário são atualizadas no sistema com sucesso.
Pós-condição (cenário de insucesso):	Sistema informa ao Usuário Master o motivo específico da falha. Sistema mantém informações originais do usuário sem alterações.
Trigger:	Usuário Master seleciona "Editar" em usuário existente.
Fluxo Principal:	1. Sistema exibe lista de usuários. 2. Usuário Master seleciona usuário para editar. 3. Sistema carrega dados atuais. 4. Usuário Master modifica informações necessárias. 5. Sistema valida alterações.

	6. Sistema atualiza registro. 7. Sistema confirma atualização.
Fluxos Alternativos:	
Fluxos de Exceção	Email já em uso: 1. Sistema informa que email já está cadastrado Último Master: 2. Sistema impede remoção de permissão Master se for o último
Pontos de Extensão:	
Regras de Negócio	

Descrição de Casos de Uso - UC07 – Registrar tentativas de login [RF 7]

Nome:	Registrar tentativas de login
Objetivo:	Registrar todas as tentativas de login para auditoria e segurança.
Atores:	Usuários.
Pré-condição:	Tentativa de login iniciada.
Pós-condição (cenário de sucesso):	A tentativa de login é registrada no log.
Pós-condição (cenário de insucesso):	O sistema informa sobre a falha no registro da tentativa de login.
Trigger:	Tentativa de login realizada.
Fluxo Principal:	1. O usuário tenta fazer login. 2. O sistema verifica as credenciais. 3. O sistema registra a tentativa de login no log.
Fluxos Alternativos:	
Fluxos de Exceção	
Pontos de Extensão:	
Regras de Negócio	Todas as tentativas de login devem ser registradas com detalhes.

Descrição de Casos de Uso - UC08 – Processar e analisar textos [RF 8]

Nome:	Processar e analisar textos
Objetivo:	Processar e analisar textos em busca de palavras-chave.
Atores:	Timer.
Pré-condição:	Texto disponível para análise.
Pós-condição (cenário de sucesso):	As palavras-chave são identificadas e registradas.
Pós-condição (cenário de insucesso):	O sistema informa sobre a falha na análise do texto.
Trigger:	Texto pronto para análise.
Fluxo Principal:	1. O sistema acessa o texto.

	2. O sistema processa e analisa o texto. 3. O sistema identifica e registra as palavras-chave e as informações relevantes em relação a elas.
Fluxos Alternativos:	
Fluxos de Exceção	
Pontos de Extensão:	
Regras de Negócio	A análise deve ser precisa e eficiente.

Descrição de Casos de Uso - UC09 – Cadastro de palavras-chave [RF 9]

Nome:	Cadastro de palavras-chave
Objetivo:	Permitir o cadastro de novas palavras-chave.
Atores:	Usuário Master.
Pré-condição:	Usuário Master autenticado no sistema.
Pós-condição (cenário de sucesso):	A nova palavra-chave é cadastrada.
Pós-condição (cenário de insucesso):	O sistema informa sobre a falha no cadastro da palavra-chave.
Trigger:	Administrador acessa a função de cadastro de palavras-chave.
Fluxo Principal:	1. O administrador acessa a função de cadastro de palavras-chave. 2. O administrador insere a nova palavra-chave. 3. O sistema cadastra a nova palavra-chave.
Fluxos Alternativos:	
Fluxos de Exceção	
Pontos de Extensão:	
Regras de Negócio	As palavras-chave devem ser únicas e relevantes.

Descrição de Casos de Uso – UC10 – Gerar métricas cadastradas [RF 10]

Nome:	Gerar métricas cadastradas
Objetivo:	Gerar as métricas cadastradas.
Atores:	Usuários.
Pré-condição:	Métricas cadastradas no sistema.
Pós-condição (cenário de sucesso):	As métricas são geradas.
Pós-condição (cenário de insucesso):	O sistema informa sobre a falha na geração das métricas.
Trigger:	Usuário faz alguma requisição que necessita das métricas.
Fluxo Principal:	1. O usuário faz alguma requisição que necessita das métricas.

	2. O sistema gera as métricas.
Fluxos Alternativos:	
Fluxos de Exceção	
Pontos de Extensão:	
Regras de Negócio	As métricas devem ser precisas e baseadas em dados atualizados.

Descrição de Casos de Uso – UC11 – Gerar relatórios gerenciais em PDF [RF 11]

Nome:	Gerar relatórios gerenciais em PDF
Objetivo:	Gerar relatórios gerenciais com as métricas em formato PDF.
Atores:	Usuários.
Pré-condição:	Métricas geradas.
Pós-condição (cenário de sucesso):	O relatório é gerado e disponibilizado em formato PDF.
Pós-condição (cenário de insucesso):	O sistema informa sobre a falha na geração do relatório.
Trigger:	Usuário solicita a geração de um relatório gerencial em PDF.
Fluxo Principal:	1. O usuário acessa a função de geração de relatórios gerenciais. 2. O usuário seleciona as métricas para incluir no relatório. 3. O sistema gera o relatório em formato PDF. 4. O sistema disponibiliza o relatório ao usuário.
Fluxos Alternativos:	
Fluxos de Exceção	
Pontos de Extensão:	
Regras de Negócio	Os relatórios devem ser gerados em conformidade com as métricas selecionadas e em formato legível.

Descrição de Casos de Uso – UC12 – Exibir dashboard [RF 12]

Nome:	Exibir dashboard
Objetivo:	Exibir um dashboard onde o usuário pode verificar as métricas.
Atores:	Usuário Master, Gerente.
Pré-condição:	Métricas geradas e atualizadas.
Pós-condição (cenário de sucesso):	O dashboard é exibido com as métricas atualizadas.
Pós-condição (cenário de insucesso):	O sistema informa sobre a falha na exibição das métricas.
Trigger:	Usuário acessa a tela de dashboard.
Fluxo Principal:	1. O usuário acessa a tela de dashboard. 2. O sistema recupera as métricas atualizadas. 3. O sistema exibe as métricas no dashboard.

Fluxos Alternativos:	
Fluxos de Exceção	
Pontos de Extensão:	
Regras de Negócio	O dashboard deve ser atualizado em tempo real ou em intervalos regulares.

Descrição de Casos de Uso – UC13 – Agendar envio de relatórios automáticos [RF 13]

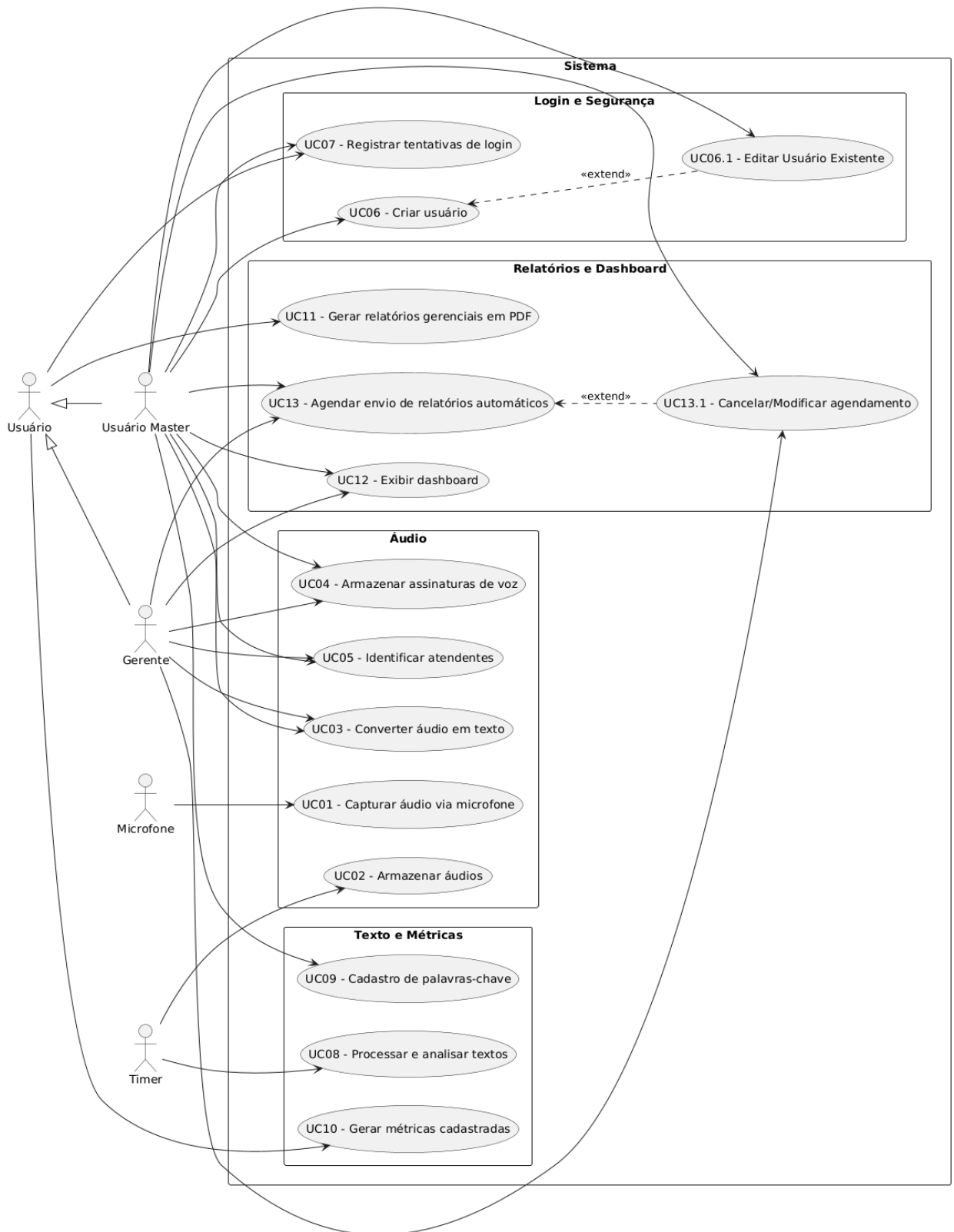
Nome:	Agendar envio de relatórios automáticos
Objetivo:	Permitir que os usuários Master e Gerente configurem e agendem o envio automático de relatórios para si mesmos ou para outros usuários.
Atores:	Usuário Master, Gerente.
Pré-condição:	O usuário (Master ou Gerente) deve estar autenticado no sistema com permissões adequadas para configurar e agendar relatórios.
Pós-condição (cenário de sucesso):	O sistema agenda o envio automático do relatório conforme as configurações definidas pelo usuário, e o relatório é enviado automaticamente na data e hora especificadas.
Pós-condição (cenário de insucesso):	O sistema informa ao usuário sobre a falha na configuração do agendamento (por exemplo, informações faltando ou inválidas, problemas técnicos).
Trigger:	O usuário acessa a seção de relatórios do sistema e seleciona a opção para agendar um novo envio de relatório.
Fluxo Principal:	1. O sistema exibe a tela de agendamento de relatório. 2. O usuário seleciona o relatório que deseja enviar, define os destinatários, a frequência do envio (por exemplo, diário, semanal, mensal) e a data/hora específica para o envio. 3. O usuário confirma as configurações de agendamento. 4. O sistema valida as informações fornecidas. 5. O sistema agenda o envio automático do relatório conforme especificado. 6. O sistema exibe uma mensagem de confirmação do agendamento para o usuário.
Fluxos Alternativos:	Modificar Agendamento Existente: 1. O usuário seleciona um agendamento existente para modificar. 2. O usuário ajusta as configurações de agendamento conforme necessário e confirma as alterações. 3. O sistema valida e aplica as alterações no agendamento.
Fluxos de Exceção	Informações Inválidas: 1. Se o usuário fornece informações inválidas ou incompletas durante o agendamento, o sistema exibe uma mensagem de erro especificando o problema. 2. O usuário ajusta as informações conforme necessário e tenta novamente.
Pontos de Extensão:	Cancelar Agendamento: Permite ao usuário cancelar um agendamento de envio de relatório existente.

Regras de Negócio	R1: Todos os relatórios agendados devem ter pelo menos um destinatário válido. R2: A data e hora do agendamento devem ser futuras no momento da configuração. R3: Usuários devem ter permissões adequadas para agendar o envio de relatórios específicos.

Descrição de Casos de Uso – UC13.1 – Cancelar Agendamento de Relatório [Extensão]

Nome:	Cancelar agendamento de relatório.
Objetivo:	Permitir que usuários cancelem agendamentos de relatórios automáticos previamente configurados.
Atores:	Usuário Master, Gerente.
Pré-condição:	O usuário (Master ou Gerente) deve estar autenticado no sistema com permissões adequadas para configurar e agendar relatórios. Deve existir pelo menos um agendamento ativo criado pelo usuário.
Pós-condição (cenário de sucesso):	Agendamento cancelado e removido do sistema.
Pós-condição (cenário de insucesso):	Sistema mantém o agendamento ativo e informa o erro.
Trigger:	Usuário seleciona opção "Cancelar Agendamento Existente".
Fluxo Principal:	1. O usuário acessa a lista de agendamentos ativos. 2. O sistema exibe todos os agendamentos configurados pelo usuário. 3. O usuário seleciona o agendamento que deseja cancelar. 4. O sistema solicita confirmação do cancelamento. 5. O usuário confirma a ação. 6. O sistema remove o agendamento e exibe mensagem de sucesso.
Fluxos Alternativos:	
Fluxos de Exceção	
Pontos de Extensão:	
Regras de Negócio	

Figura 3 - Diagrama de casos de uso

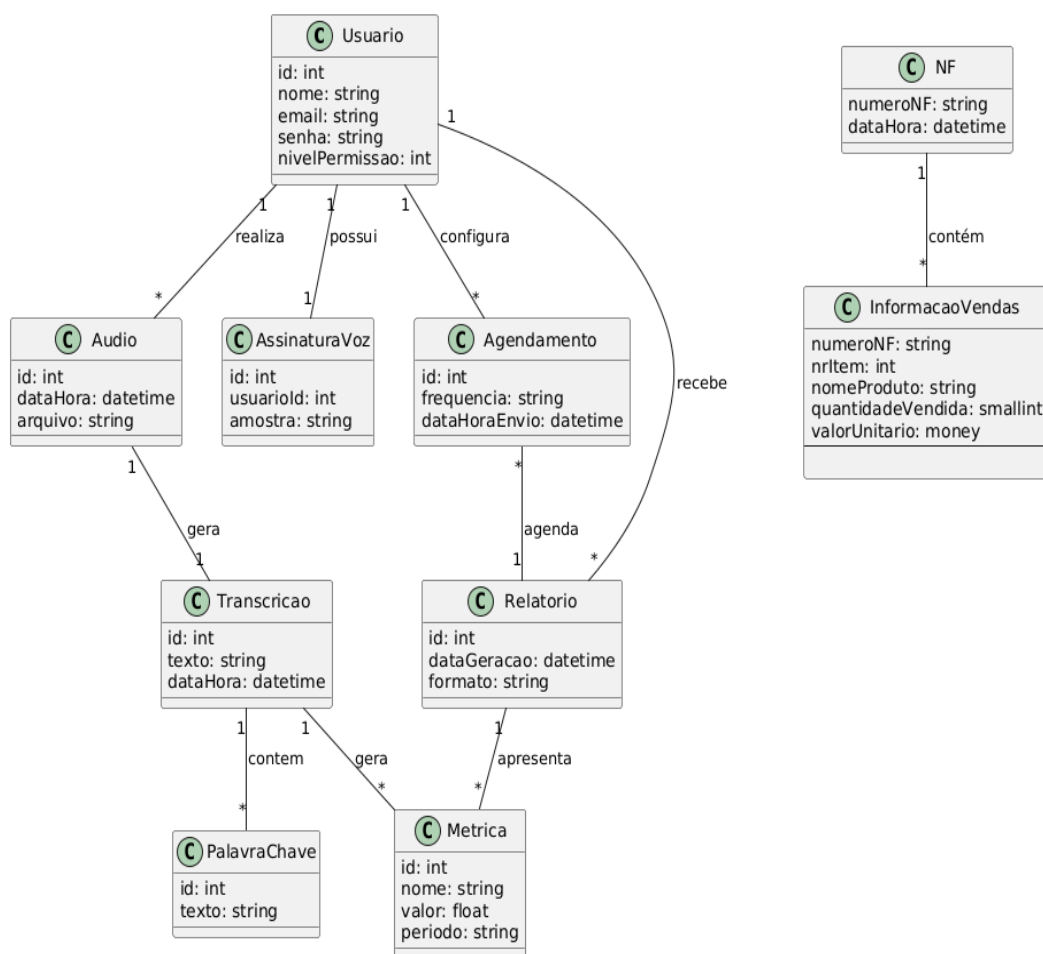


8. Modelagem de dados

Essa seção trata sobre a forma de representar e organizar as informações do sistema, utilizando técnicas de modelagem de dados. São apresentadas as representações em UML, que descrevem visualmente os elementos e suas interações, bem como o modelo relacional de dados, que estrutura as entidades e seus relacionamentos.

8.1. UML

Figura 4 – Diagrama UML



8.2. Modelo relacional de dados

Tabela agendamento			
column_name	data_type	key_type	references_to
id	integer	PK	
usuario_id	integer	FK	usuario.id
relatorio_id	integer	FK	relatorio.id
frequencia	character varying		
data_hora_envio	timestamp without time zone		

Tabela assinatura_voz			
<i>column_name</i>	<i>data_type</i>	<i>key_type</i>	<i>references_to</i>
id	integer	PK	
usuario_id	integer	FK	usuario.id
amostra	bytea		

Tabela audio			
<i>column_name</i>	<i>data_type</i>	<i>key_type</i>	<i>references_to</i>
id	integer	PK	
usuario_id	integer	FK	usuario.id
data_hora	timestamp without time zone		
caminho_arquivo	text		
formato	character varying		

Tabela metrica			
<i>column_name</i>	<i>data_type</i>	<i>key_type</i>	<i>references_to</i>
id	integer	PK	
tipo	character varying		
periodo_inicio	date		
periodo_fim	date		
valor	numeric		
gerada_em	timestamp without time zone		

Tabela palavra_chave			
<i>column_name</i>	<i>data_type</i>	<i>key_type</i>	<i>references_to</i>
id	integer	PK	
transcricao_id	integer	FK	transcricao.id
palavra	character varying		

Tabela relatorio			
<i>column_name</i>	<i>data_type</i>	<i>key_type</i>	<i>references_to</i>
id	integer	PK	
usuario_id	integer	FK	usuario.id
data_geracao	timestamp without time zone		
formato	character varying		

Tabela transcricao			
<i>column_name</i>	<i>data_type</i>	<i>key_type</i>	<i>references_to</i>
id	integer	PK	
audio_id	integer	FK	audio.id
texto	text		
data_hora	timestamp without time zone		

Tabela usuario			
<i>column_name</i>	<i>data_type</i>	<i>key_type</i>	<i>references_to</i>
id	integer	PK	
nome	character varying		
email	character varying		
senha	character varying		

Tabela informacao_vendas			
<i>column_name</i>	<i>data_type</i>	<i>key_type</i>	<i>references_to</i>
numero_nf	character varying	PK	nf.numero_nf
nr_item	integer	PK	
nome_produto	character varying		
quantidade_vendida	smallint		
valor_unitario	money		

Tabela nf			
<i>column_name</i>	<i>data_type</i>	<i>key_type</i>	<i>references_to</i>
numero_nf	character varying	PK	
data_hora	timestamp without time zone		

9. Desenvolvimento do software

O módulo `audio.py` começa com a implementação da funcionalidade de gravação de áudio, onde configura os parâmetros de captação, como taxa de amostragem e formato, para criar arquivos de áudio em 16 kHz usando a biblioteca `pyaudio`. Após capturar o áudio, o programa permite transcrever o mesmo para texto por meio de um modelo pré-treinado da biblioteca `huggingsound`, especializado em reconhecimento de fala. Adicionalmente, o programa realiza a diarização, utilizando o modelo `pyannote`, para identificar os falantes em diferentes trechos do áudio, associando esses trechos a rótulos como "Speaker 1"/"Speaker 2".

Com os resultados da diarização, o programa corta o áudio original em segmentos individuais correspondentes a cada falante, armazenando-os como arquivos separados. Em seguida, os arquivos segmentados são processados para gerar uma transcrição ordenada, em que cada segmento é transcrito e agrupado em um arquivo de texto, indicando qual falante está associado a cada trecho do conteúdo transcrito.

Por outro lado, o programa `reconhecimento_audio.py` utiliza um modelo de aprendizado de máquina para identificar locutores com base em características acústicas. Ele começa carregando áudios de treinamento previamente classificados (por exemplo, áudios de "Marco" e "Ana") e extrai características acústicas utilizando o `librosa`. A principal técnica usada é a análise de MFCCs, que são indicadores importantes para a identificação de padrões em sinais de voz.

Após a extração das características, o programa treina um modelo de K-Vizinhos (K-Nearest Neighbors) usando os dados extraídos e salva o modelo resultante e os rótulos dos locutores em arquivos. Quando novos áudios precisam ser analisados, o programa carrega o modelo salvo, extrai as características do áudio de teste e compara essas informações com os dados previamente armazenados. Caso a distância calculada entre o áudio de teste e os vizinhos mais próximos seja aceitável (abaixo do limiar pré-definido), o locutor é identificado(atendente); caso contrário, o locutor é considerado como sendo um cliente.

Essas funcionalidades de processamento de áudio são integradas ao sistema interativo para os usuários, que começa com uma tela de login centralizada. Na interface inicial, o usuário pode se autenticar inserindo login e senha ou cadastrar uma nova conta. Após o login bem-sucedido, a tela principal exibe opções baseadas no nível de acesso do usuário. Os usuários de nível 3 ou superior também podem cadastrar, visualizar palavras-chave e visualizar as métricas.

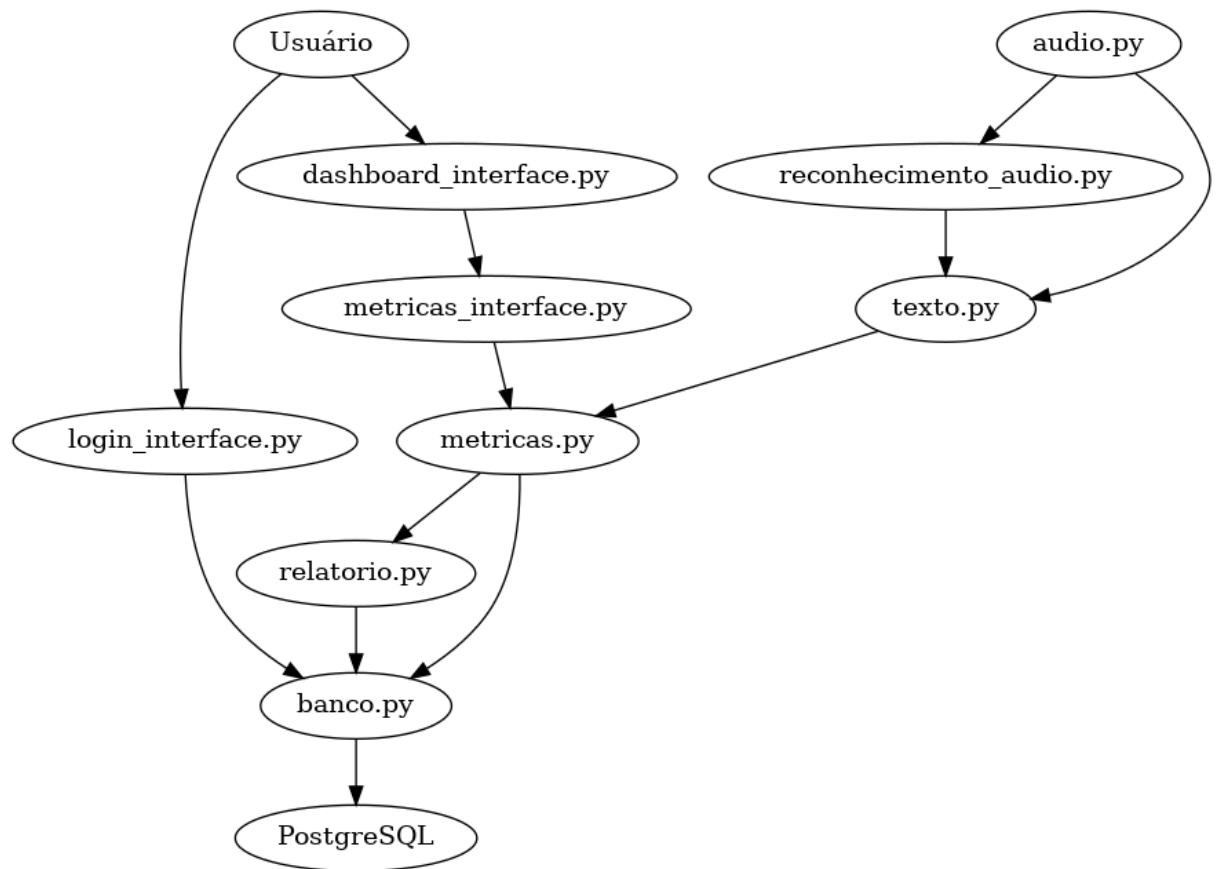
A tela de métricas permite filtrar dados por período, horário e atendente, exibindo gráficos comparativos de palavras-chave mencionadas e NFEs distintas, com percentuais de relação calculados automaticamente. Há também uma opção para visualizar métricas segmentadas dia a dia. A análise utiliza gráficos gerados com o Matplotlib e apresentados de forma interativa na interface gráfica construída com Tkinter. A integração com tkcalendar facilita a seleção de datas nos filtros.

O sistema utiliza um banco de dados PostgreSQL para gerenciar informações. Ele armazena dados de usuários, como login, senha criptografada (usando bcrypt), e-mail e nível de acesso. Dados de palavras-chave, NFEs e informações derivadas de áudios também são registrados.

Biblioteca	Finalidade	Módulo onde é utilizada
tkinter	Criar e gerenciar a interface gráfica.	main.py
tkcalendar	Adicionar um componente de calendário interativo para seleção de datas.	main.py
psycopg2	Conectar-se ao PostgreSQL, executar consultas e manipular dados no banco.	main.py
pandas	Manipulação de dados tabulares.	main.py

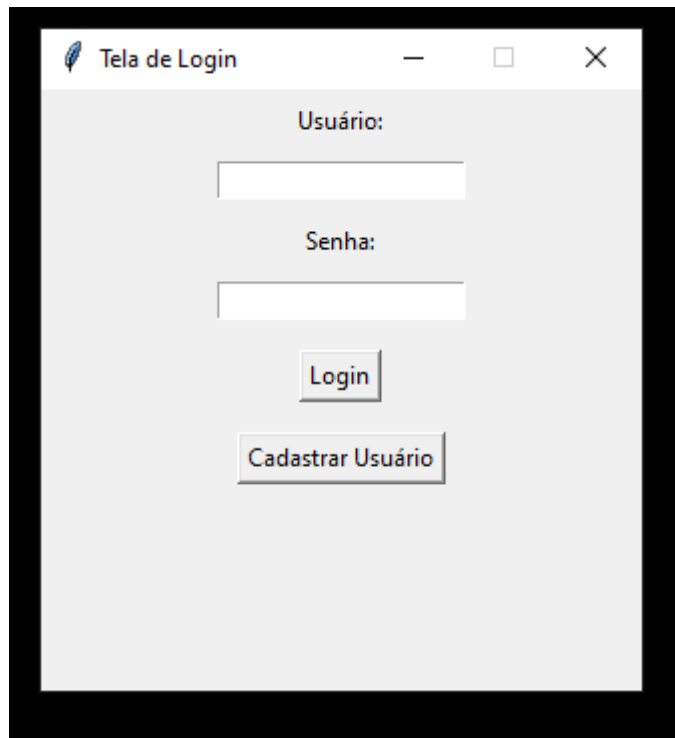
matplotlib	Gerar gráficos para exibir as métricas na interface.	main.py
bcrypt	Gerenciar criptografia de senhas para segurança no sistema.	main.py
pydub	Manipulação de arquivos de áudio, como cortar e exportar segmentos.	audio.py
pyannote.audio	Realizar diarização de falantes, identificando quem está falando em quais trechos de um áudio.	audio.py
huggingsound	Utilizar modelos pré-treinados para transcrição de fala para texto.	audio.py
pyaudio	Gravação de áudio em tempo real através do microfone.	audio.py
wave	Manipulação de arquivos de áudio no formato WAV para leitura e escrita.	audio.py
os	Manipular diretórios e arquivos do sistema.	audio.py reconhecimento_a udio.py
re	Utilizar expressões regulares para extrair informações específicas de strings.	audio.py
librosa	Análise e extração de características de arquivos de áudio.	audio.py reconhecimento_a udio.py
numpy	Manipulação de arrays numéricos, usado em conjunto com a biblioteca librosa.	audio.py reconhecimento_a udio.py
sklearn.neighbors.KNeighborsClassifier	Classificação baseada no algoritmo KNN para identificar padrões.	audio.py reconhecimento_a udio.py
sklearn.preprocessing.LabelEncoder	Codificação de rótulos para uso no modelo de machine learning.	audio.py reconhecimento_a udio.py
joblib	Salvar e carregar modelos treinados e objetos em disco.	reconhecimento_a udio.py

Figura 5 - Fluxo do programa



10. Interface do sistema

O sistema começa com uma tela de login centralizada, onde o usuário pode inserir seu login e senha. Há duas opções disponíveis: fazer login ou cadastrar um novo usuário. Caso o usuário escolha fazer login, o sistema verifica as credenciais no banco de dados. Se as informações forem corretas, ele é direcionado para a tela principal. Caso contrário, uma mensagem de erro é exibida. Se o usuário optar por cadastrar, uma nova tela é aberta para que ele insira as informações de login, senha e e-mail. O novo nível do usuário é configurado automaticamente como 1.



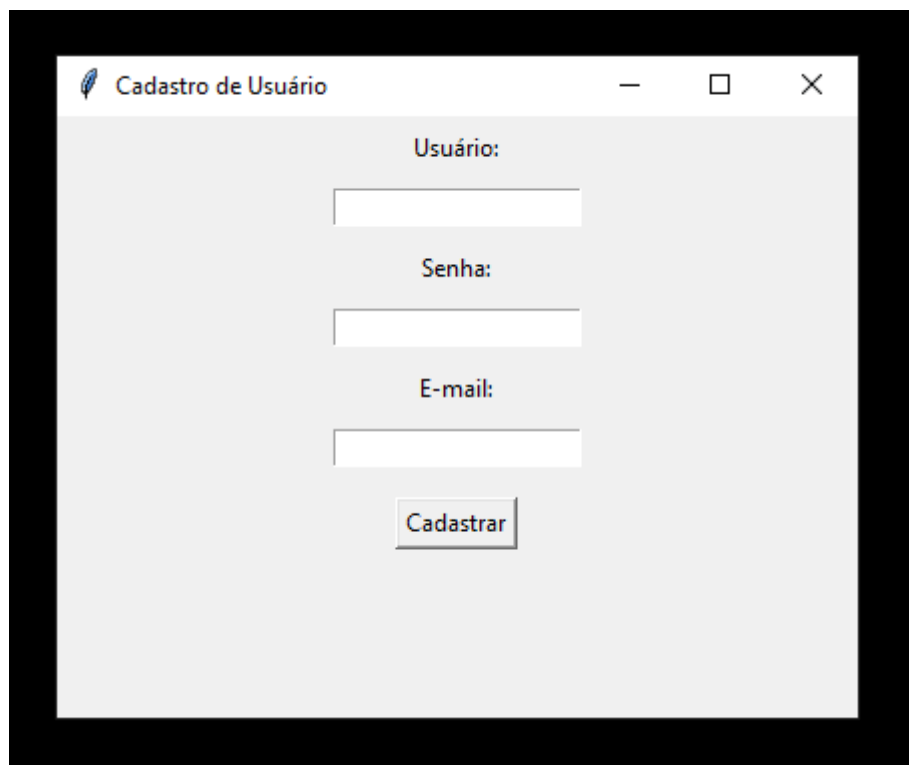
Usuário:

Senha:

Login

Cadastrar Usuário

Figura 6 - Tela de login



Usuário:

Senha:

E-mail:

Cadastrar

Figura 7 - Tela de cadastro de usuário

Após um login bem-sucedido, o usuário acessa a tela principal, que exibe uma mensagem de boas-vindas e oferece diferentes opções dependendo do nível de acesso. Os usuários com nível 3 ou superior têm acesso à opção de visualizar métricas e podem cadastrar palavras-chave e visualizar palavras-chave já cadastradas. Existe ainda a opção de sair, que encerra o programa.

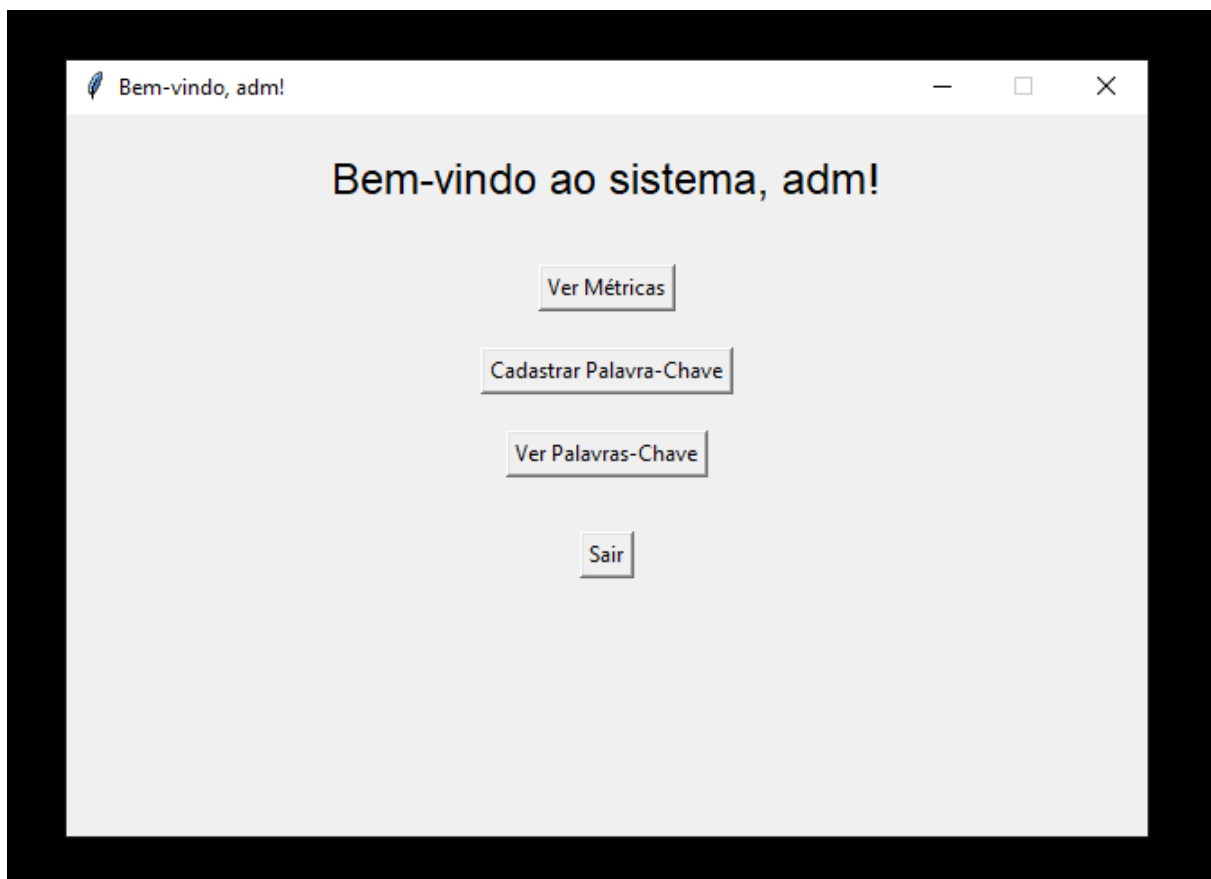


Figura 8 - Tela após login bem-sucedido

A tela de métricas permite ao usuário configurar filtros, como data inicial, data final, horário inicial, horário final e atendente. O usuário pode clicar em "Atualizar Métricas" para calcular e exibir as métricas com base nos filtros selecionados. Além disso, há a opção "Visualizar Métricas Dia a Dia", que gera gráficos segmentados por dia dentro do período selecionado. Os gráficos exibidos comparam o total de palavras-chave mencionadas com o total de NFEs distintas no mesmo período, incluindo o percentual de relação entre esses valores. Cada gráfico é gerado para categorias específicas, como "Bom Dia/Boa Tarde/Boa Noite", "Tudo Bem", "Volte Sempre", "Obrigado" e "Promoção".

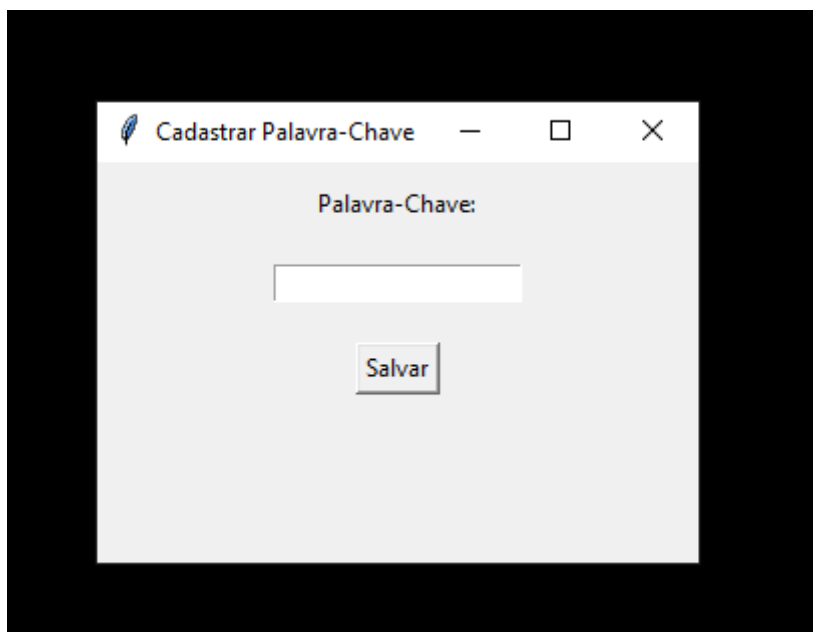


Figura 9 - Tela de cadastro de nova palavra-chave

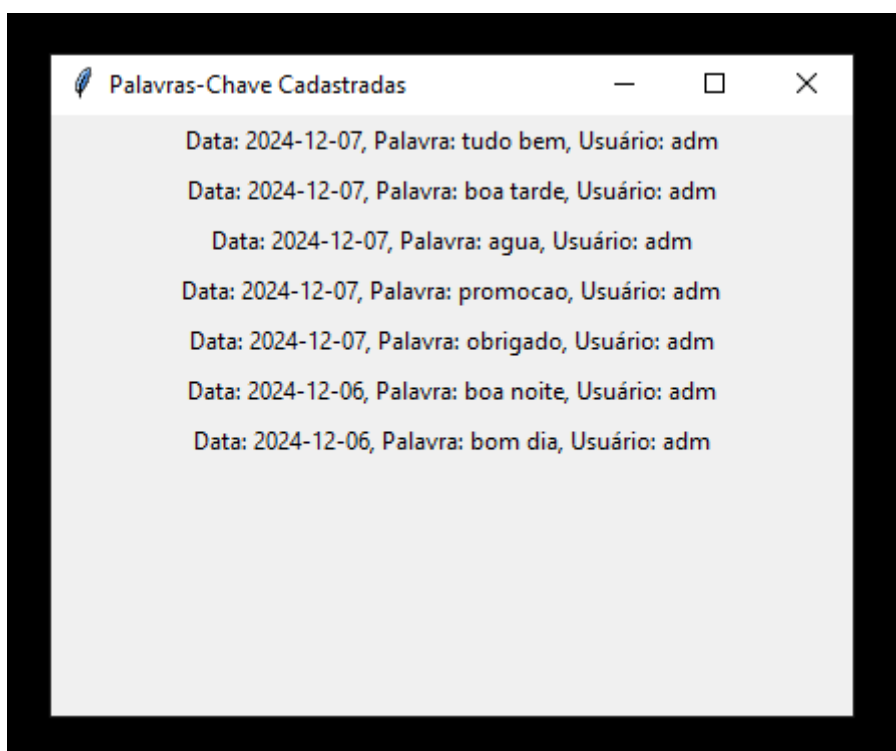


Figura 10 - Tela de palavras chaves cadastradas

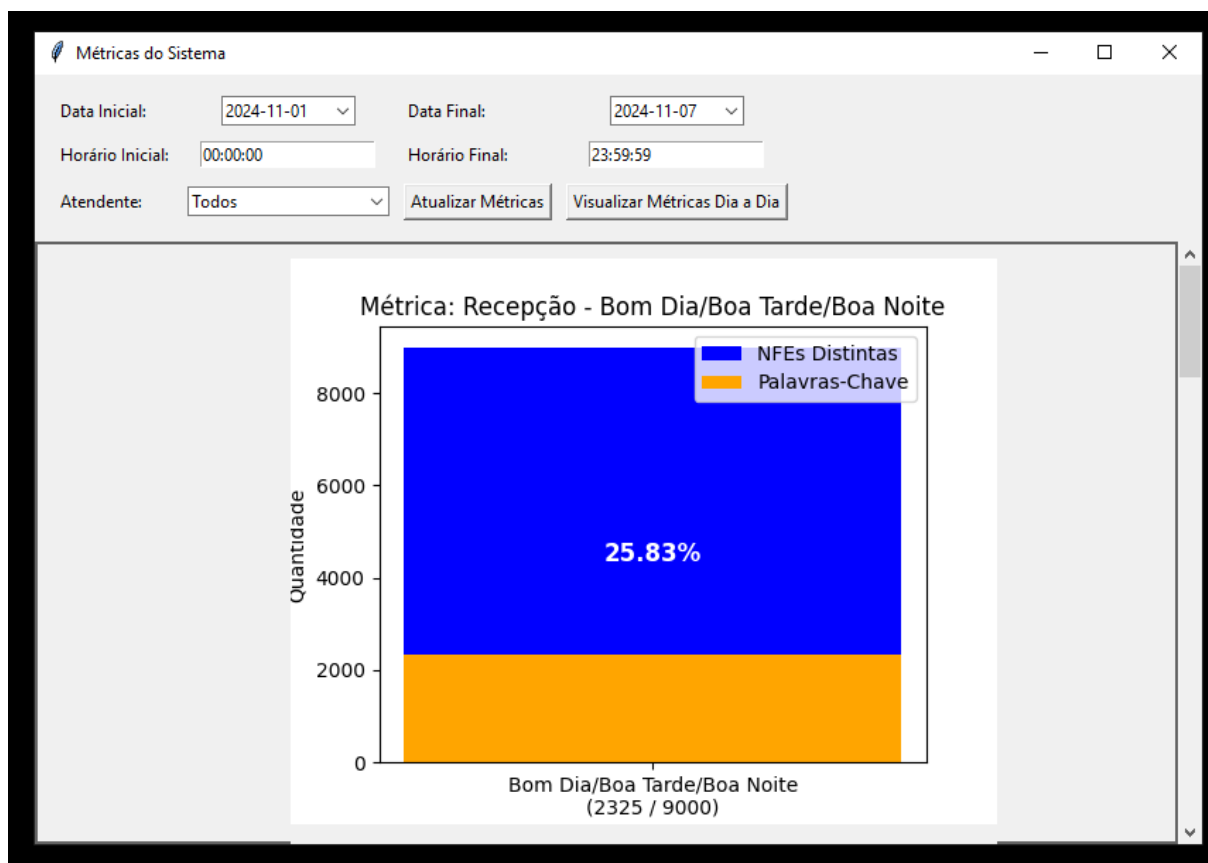


Figura 11 - Tela do dashboard, onde é possível visualizar as métricas

11. Testes e avaliações

Nesta seção, são apresentados os testes realizados no sistema desenvolvido, bem como as avaliações feitas para verificar o seu correto funcionamento.

A funcionalidade de transcrição do áudio foi testada de forma incremental, utilizando diferentes tipos e durações de gravações para avaliar a precisão do reconhecimento de fala.

Inicialmente, foram realizados testes com áudios curtos gravados por mim, com duração média de 1 a 2 minutos, contendo frases simples e em ambiente silencioso. O objetivo foi verificar se o sistema era capaz de capturar e transcrever corretamente o conteúdo falado em condições ideais. Nessa etapa, a transcrição apresentou alto nível de precisão, com apenas pequenas falhas em palavras com pronúncia menos clara ou em palavras as quais a dicção não estava boa.

Em seguida, foi realizado um teste com um áudio mais longo, de aproximadamente 30 minutos, referente a uma simulação real de atendimento. Esse teste tinha como objetivo avaliar o desempenho do sistema em transcrições prolongadas, verificando tanto a

estabilidade do processo quanto a coerência do texto gerado. Nesse cenário, as transcrições apresentaram muito mais falhas, principalmente em trechos com sobreposição de falas ou ruídos de fundo. Ainda assim, o nível de acerto permaneceu satisfatório, permitindo compreender o contexto geral do atendimento.

Para avaliar a etapa da diarização, foram utilizados áudios curtos e gravados em ambiente controlado, com falas bem delimitadas entre os participantes e mínima presença de ruídos.

Mesmo nessas condições ideais, o sistema apresentou muitas falhas na segmentação e identificação dos interlocutores, frequentemente confundindo as falas de diferentes pessoas, gerando mais segmentos do que realmente existiam ou até mesmo não reconhecendo ninguém.

Quando testada em áudios mais longos e realistas, com sobreposição de falas e ruídos típicos de um ambiente de atendimento, a taxa de erro foi ainda maior, tornando os resultados pouco confiáveis para uso prático no contexto do sistema.

Após algumas tentativas distintas de solucionar o problema sem êxito, decidiu-se postergar a implementação plena da diarização para trabalhos futuros, priorizando as demais etapas do sistema que apresentaram melhores resultados.

Além disso, teve uma limitação de hardware onde o software é utilizado. Ao rodar o software no PDV da loja, o PDV não aguentava rodar os sistemas de venda, o ERP e o software em questão ao mesmo tempo, chegando em um gargalo que, inicialmente, foi contornado deixando apenas a função de gravar o áudio ativa e fazer todo o restante do processamento do áudio e visualização das métricas em outro computador.

11.1. Feedback

O sistema é muito interessante e vai atender uma grande necessidade da empresa. Na avaliação da apresentação observo algumas sugestões tanto na apresentação das telas como nas métricas disponíveis.

Visualmente as telas são bem simples e monocromáticas, poderiam ser mais agradáveis e estimulantes para o usuário. As telas são intuitivas e fáceis de navegar para qualquer tipo de usuário, embora talvez alguns esclarecimentos possam ser dados na tela de cadastramento das palavras-chave, como explicar que não faz diferença o uso de maiúsculas ou minúsculas e a necessidade da entrada de cada uma das expressões individualmente.

Quanto as métricas, existe um potencial bem interessante a ser explorado, o que tornará o programa ainda mais relevante para empresa na gestão da qualidade do atendimento.

Em termo de métricas são várias as sugestões:

1. Apurar o % dos atendimentos que são iniciados com o cumprimento inicial e também finalizados com “Obrigado e Volte Sempre”.
2. Considerando os atendimentos tanto com cumprimento inicial como também a despedida, gerar estatística do tempo (duração) de atendimento.
3. Medição do tempo total dos intervalos com atendimento por dia.
4. Medição do tempo total dos intervalos sem nenhum atendimento por dia.
5. Medição do nível de tom de voz do atendente, buscando classificar como baixo, confortável ou alto.

12. Considerações finais

O trabalho tinha como objetivo construir um software que facilitasse a gestão do nível de atendimento de uma loja de conveniência, utilizando de recursos gratuitos e, preferencialmente, *open source*. O objetivo foi atingido parcialmente: o software é capaz de capturar o áudio das interações, transcrever o conteúdo gravado, gerir usuários e controlar suas respectivas permissões de acesso. Entretanto, não foi possível implementar com sucesso a identificação confiável dos interlocutores por meio da diarização, e a funcionalidade de envio automático dos relatórios ainda não está operacional. Assim, embora o sistema atenda a parte significativa dos requisitos propostos, algumas funcionalidades permanecem como oportunidades para trabalhos futuros.

Após a finalização do trabalho vejo algumas melhorias para trabalhos futuros, sendo elas:

1. Desenvolvimento e implementação dos testes unitários/automatizados.
2. Implementação da API do *speech-service* do Azure, para conseguir, com precisão, fazer a distinção dos interlocutores, o qual não foi implementado pois o software já estava com o desenvolvimento avançado e por se tratar de uma ferramenta paga.
3. Implementar novas métricas para o acompanhamento do nível de atendimento.
4. Melhoria da interface gráfica do software.
5. Aumentar o leque de visualizações possíveis das métricas.
6. Implementar novos gráficos.

13. Referências bibliográficas

[1] DABHOLKAR, Pratibha A.; THORPE, Dayle I.; RENTZ, Joseph O. A measure of service quality for retail stores: scale development and validation. *Journal of the Academy of marketing Science*, v. 24, p. 3-16, 1996.

[2] DISTÂNCIA de Levenshtein. In: Wiki - Augusto Baffa, 2025. Disponível em: https://www.augustobaffa.pro.br/wiki/Dist%C3%A2ncia_de_Levenshtein. Acesso em: 05 de maio de 2025.

[3] Jonatas Grosman, Doutor pelo DI, cria modelo que lidera speech-to-text para o inglês, com mais de 50 milhões de downloads no último mês. Departamento de Informática PUC-Rio, 2023. Disponível em: <https://www.inf.puc-rio.br/blog/noticia/noticia/jonatas-grosman-doutor-pelo-di-cria-modelo-que-lidera-speech-to-text-para-o-ingles-com-mais-de-50-milhoes-de-downloads-no-ultimo-mes>. Acesso em: 21 de outubro de 2023.

[4] GROSMAN, Jonatas. Hugging Face, 2023. Fine-tuned XLS-R 1B model for speech recognition in Portuguese. Disponível em: <https://huggingface.co/jonatasgrosman/wav2vec2-xls-r-1b-portuguese#fine-tuned-xls-r-1b-model-for-speech-recognition-in-portuguese>. Acesso em: 21 de outubro de 2023.

[5] GROSMAN, Jonatas Dos Santos. [en] ASSESSMENT OF FINE-TUNING ON END-TO-END SPEECH RECOGNITION MODELS. 2022.

[6] BRASIL. Lei nº 13.709, de 14 de agosto de 2018. Dispõe sobre a proteção de dados pessoais e altera a Lei nº 12.965, de 23 de abril de 2014 (Marco Civil da Internet). *Diário Oficial da União*: seção 1, Brasília, DF, n. 157, p. 59, 15 ago. 2018. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm. Acesso em: 25 de julho. 2025.