



masterizAI

Felipe Meneses RM: 98020

Luiz Henrique Xavier RM:99532

Maria Alice Ferreiros RM:550831

Maria Clara F Souza RM: 99776

Sergio Augusto RM: 551331

Challenge – Entrega Final FAQe

São Paulo – SP

2024

**PROPOSTA DE SOLUÇÃO**

A proposta de solução envolve os seguintes passos: processamento de áudios, transcrição para texto e análises feitas por IA dentro desses textos para gerar insights sobre os clientes. Dessa forma, busca-se atingir o objetivo de aumentar o número de clientes promotores e a pontuação no NPS além de reduzir o número de clientes detratores e neutros.

## **PUBLICO ALVO**

O projeto é voltado para empresas com ouvidorias ou departamentos de satisfação do cliente que precisam gerenciar e analisar as ligações realizadas diariamente. O objetivo é fornecer um Dashboard que permita a análise e a extração de insights valiosos a partir das transcrições das chamadas, visando melhorar a retenção de clientes e otimizar a experiência do cliente.

## **APLICAÇÃO DO PROJETO**

Os áudios são enviados de um repositório para um ambiente Python, onde são processados com o intuito de melhorar a qualidade, removendo ruídos e realçando as vozes do entrevistador da TOTVS e do entrevistado. Em seguida, os áudios passam por um modelo de transcrição, e o texto resultante é processado por IA para extrair informações valiosas. Essas informações são então armazenadas e enviadas para uma plataforma de visualização, permitindo que a área de negócios tome decisões com base nos insights disponibilizados na plataforma.

## **PROCESSO DE CRIAÇÃO**

Para a execução do projeto, foram disponibilizados 80 áudios pela TOTVS, que foram carregados no Drive para serem utilizados. Os áudios são transferidos para o ambiente Python, onde são processados para melhorar drasticamente a qualidade da transcrição. Após o processamento, são transcritos pelo modelo "Medium" da biblioteca Whisper, criada pela OpenAI. As transcrições, juntamente com diversas instruções prévias, são submetidas à API do ChatGPT, que gera respostas de acordo com essas instruções. As respostas são

organizadas em um dicionário Python, que é transformado em um dataframe. Esse dataframe passa por processos de normalização de dados e, por fim, é convertido em um arquivo CSV, que é enviado de volta ao Drive. Esse CSV alimenta um dashboard no PowerBI, que exibe as informações extraídas dos áudios.

Atualmente, o processo se baseia nos 80 áudios fixos disponibilizados. No entanto, para atender a novas demandas, como o recebimento constante de áudios, medidas mais sofisticadas podem ser facilmente implementadas, como a automação de processos por meio da migração do código para a nuvem e/ou o armazenamento das informações em um banco de dados conectado ao PowerBI, em vez da criação de um CSV.

### **2.1.3 Funcionamento do Projeto**

O processo começa com a importação das bibliotecas que serão utilizadas no ambiente Python. Neste caso, a ferramenta utilizada foi o Google Colab, onde algumas bibliotecas já vêm instaladas por padrão, o que elimina a necessidade de instalar todas elas manualmente. No entanto, outras ferramentas também poderiam ser utilizadas sem problemas, com a única mudança sendo a instalação das bibliotecas por meio de um requirements.txt .

```
1 !pip install Pydub openai-whisper openai noisereduce -q

Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing metadata (pyproject.toml) ... done
----- 798.6/798.6 kB 11.4 MB/s eta 0:00:00
----- 375.6/375.6 kB 27.5 MB/s eta 0:00:00
----- 76.4/76.4 kB 7.0 MB/s eta 0:00:00
----- 77.9/77.9 kB 6.9 MB/s eta 0:00:00
----- 318.9/318.9 kB 22.5 MB/s eta 0:00:00
----- 168.1/168.1 MB 6.8 MB/s eta 0:00:00
----- 1.1/1.1 MB 48.7 MB/s eta 0:00:00
----- 58.3/58.3 kB 4.2 MB/s eta 0:00:00
Building wheel for openai-whisper (pyproject.toml) ... done

[ ] 1 !pip install --upgrade --no-deps --force-reinstall git+https://github.com/openai/whisper.git -q

Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing metadata (pyproject.toml) ... done
Building wheel for openai-whisper (pyproject.toml) ... done

[ ] 1 from pydub import AudioSegment
2 from pydub.silence import split_on_silence
3 from pydub.effects import low_pass_filter, high_pass_filter, compress_dynamic_range
4
5 import whisper
6 from openai import OpenAI
7
8 import librosa
9 import scipy.signal as signal
10 from scipy.signal import butter, filtfilt
11
12 import numpy as np
13 import soundfile as sf
14
15 import tempfile
16
17 from google.colab import drive
18 import os
19
20 import re
21 import pandas as pd
22
23 import noisereduce as nr
```

Agora começa a importação dos arquivos do Drive: primeiro os áudios, depois as instruções para o ChatGPT e, por último, a chave da API, por questões de segurança. Em seguida, o modelo de transcrição é carregado e as listas de armazenamento são inicializadas.

```

# montando o drive e pegando o diretorio com os áudios
drive.mount('/content/drive')
diretorio = '/content/drive/MyDrive/Challenge TOTVS Amostra de Dados v2/Amostra de Dados'

arquivos = os.listdir(diretorio)

# pegando as instruções do chat got por meio de um txt no drive, deixando o
# código mais limpo (as instruções estão na documentação)
with open('/content/drive/MyDrive/Challenge TOTVS Amostra de Dados v2/instrucoes.txt', 'r') as file:
    instrucoes_gpt = file.read()

# pegando a api por um txt no drive por questões de segurança
with open('/content/drive/MyDrive/Challenge TOTVS Amostra de Dados v2/api_key.txt', 'r') as file:
    api_key = file.read()

# baixando o modelo medium do whisper para transcrição ed inicializando as listas
model = whisper.load_model('medium')
dfs, respostas = [], []

```

Aqui é criada a função de processamento de áudio, que recebe o caminho do áudio como parâmetro. A função começa carregando o áudio, capturando uma amostra do ruído e, em seguida, utilizando-a para remover todo o ruído do arquivo. Após esse processo, é criado um arquivo temporário do áudio com o ruído reduzido, que é equalizado por meio de filtros nas frequências altas e baixas. Por fim, o volume do áudio é aumentado em dez decibéis, e a forma final é retornada.

```

# criando a função de processamento de áudios, que recebe o áudio do drive como parâmetro
def processar_audio(caminho_arquivo):

    # carregando os áudios e retirando o ruído
    y, sr = librosa.load(caminho_arquivo, sr=None)
    noise_sample = y[:int(sr)] # amostra do ruído
    y_reduced_noise = nr.reduce_noise(y=y, sr=sr, y_noise=noise_sample, prop_decrease=0.8)

    # criando um arquivo temporário para continuar os processos
    with tempfile.NamedTemporaryFile(suffix='.wav', delete=False) as temp_wav:
        sf.write(temp_wav.name, y_reduced_noise, sr)
        temp_wav_path = temp_wav.name

    # carregando o arquivo temporário e passando filtros nas frequências do áudio para melhorá-lo
    audio = AudioSegment.from_wav(temp_wav_path)
    audio_low = low_pass_filter(audio, cutoff=4000)
    audio_high = high_pass_filter(audio_low, cutoff=80)

    # aumentando o volume do áudio
    audio_amplificado = audio_high.apply_gain(7.0)

    #retornando o áudio final
    return audio_amplificado

```

Para garantir que as respostas fornecidas pela API do ChatGPT sigam um padrão, é criada uma função para normalizar essas respostas. A função remove quebras de linha e caracteres especiais desnecessários, retornando a resposta devidamente normalizada.

```
# definindo a função que normaliza as respostas do GPT
def limpar_resposta(resposta):
    resposta = re.sub(r'^[-\*\s]+', '', resposta, flags=re.MULTILINE).strip()
    return re.sub(r'\n\s*\n', '\n', resposta).split('\n')
```

Depois de normalizar a resposta da IA, essas respostas são separadas por “:” para os tópicos e “-” para as notas e justificativas e essas são adicionadas em suas listas correspondentes, que por fim são transformadas em DataFrame e retornadas

```
# definindo a função que transforma as respostas do GPT em dataframe
def processar_resposta(linhas, arquivo):

    # inicializando as listas que virarão colunas
    topicos, notas, justificativas = [], [], []

    # fazendo a separação das respostas por linhas, separando tópicos, notas e justificativas por : e -
    for linha in linhas:
        topico, detalhes = linha.split(':', 1) #separando os tópicos
        if '-' in detalhes:
            nota, justificativa = detalhes.split('- ', 1) #separando as notas das justificativas
        else:
            nota, justificativa = detalhes, ''
        topicos.append(topico)
        notas.append(nota)
        justificativas.append(justificativa)

    # retornando o DF utilizando as listas e também o nome do arquivo como ID
    return pd.DataFrame({
        'Tópico': topicos,
        'Nota': notas,
        'Justificativa': justificativas,
        'ID': arquivo
    })
```

Após a criação das funções, começa um loop que passar por todos os arquivos de áudio no diretório e esses arquivos passam pelo seguinte processo: Processamento de áudio, criação de um arquivo temporário de áudio para a realização da transcrição e por fim, a remoção deste arquivo temporário

```

# utilizando um loop dentro da pasta de áudios
for arquivo in arquivos:
    if arquivo.endswith('.wav'):
        caminho_arquivo = os.path.join(diretorio, arquivo)

        # realizando o processamento do áudio
        try:
            audio_amplificado = processar_audio(caminho_arquivo)

            # criando um arquivo temporário para que a transcrição seja feita
            with tempfile.NamedTemporaryFile(suffix='.wav', delete=False) as temp_final_wav:
                audio_amplificado.export(temp_final_wav.name, format='wav')
                temp_final_wav_path = temp_final_wav.name

            result = model.transcribe(temp_final_wav_path)

        # desfazendo o arquivo temporário
        finally:
            os.unlink(temp_final_wav_path)

```

Continuando o processamento, as instruções são unidas com a transcrição e então são enviadas para a API para que a resposta seja feita. Depois de respondido, o processo de normalização é ativado e por último o DataFrame é criado e colocado na lista de DFs

```

# criando o texto do gpt que pega as instruções junto do resultado da transcrição
texto_gpt = instrucoes_gpt + result['text'].replace('\n', ' ')

# acionando a API
client = OpenAI(api_key)

chat_completion = client.chat.completions.create(
    messages=[{"role": "user", "content": texto_gpt}],
    model="gpt-4o-2024-08-06",
)

# limpando a resposta do gpt, criando o DF e adicionando na lista
resposta = chat_completion.choices[0].message.content
linhas = limpar_resposta(resposta)
df = processar_resposta(linhas, arquivo)
dfs.append(df)

```

O processo de extração de informações termina com a concatenação dos DFs, um tratamento rápido e a transformação do DF final em CSV

Para contextualização, as instruções da API junto das respostas logo em sequência:

Por favor se limite a respostas objetivas e não adicione tópicos extras, você receberá um texto de um questionário feito com um cliente, responda os tópicos utilizando as respostas do cliente e o classifique no final utilizando a métrica explicada.

Responda as perguntas abaixo usando as notas que o cliente der, assim: nota - justificativa do cliente, caso não haja justificativa, apenas coloque 'Sem justificativa'

Se o cliente disser que não tem contato, não tem acesso ou qualquer coisa do tipo, coloque o que o cliente disse, por exemplo: "Não tenho contato", "Não tenho acesso" ou qualquer coisa do tipo, dependendo do que foi respondido e coloque " - Sem justificativa".

Aqui estão os tópicos:

Probabilidade de recomendar a TOTVS a um amigo ou colega:

Agilidade nos processos do suporte técnico:

Atendimento do suporte técnico:

Atendimento do executivo de vendas:

Satisfação com o custo dos produtos contratados:

Satisfação com a implantação do software:

Entrega do benefício esperado pelo departamento: (Entrega total, parcial e não entrega)

Experiência com a última atualização do software:

Avaliação da unidade TOTVS:

Pontos adicionais: (caso não haja nenhum, apenas escreva "Não")

Média das notas: (não há necessidade de mostrar todo o cálculo, caso o resultado seja um número decimal, desconsidere a casa decimal)

Classificação do Cliente: Após responder os tópicos, calcule a média das notas (ignore casas decimais). Classifique o cliente conforme a tabela abaixo:

Detrator: Média 0 a 6

Neutro: Média 7 a 8

Promotor: Média 9 a 10

Adicione um novo tópico chamado "Sugestões da IA" e escreva em poucas palavras o que a empresa poderia melhorar baseado nas notas e justificativas do cliente

Não adicione nenhum \* ou - ou # ou qualquer outro caractere além dos que já são especificados anteriormente.

Caso o texto recebido esteja fora dos padrões e/ou não aparente ter sentido, preencha todos os tópicos com "Áudio obstruído", como por exemplo: (Tópico: Áudio obstruído - Áudio obstruído). Isso também se aplica aos tópicos de média, classificação e sugestão.

Aqui está o texto:



Uma das respostas feitas pelo GPT-4o:

Probabilidade de recomendar a TOTVS a um amigo ou colega: 6 - Sem justificativa

Agilidade nos processos do suporte técnico: 7 - Sem justificativa

Atendimento do suporte técnico: 8 - Sem justificativa

Atendimento do executivo de vendas: Não tenho contato - Sem justificativa

Satisfação com o custo dos produtos contratados: Não sei dizer - Sem justificativa

Satisfação com a implantação do software: 9 - Sem justificativa

Entrega do benefício esperado pelo departamento: Parcial - Reclamação em relação ao Fiscal Flex com falhas e erros

Experiência com a última atualização do software: 6 - Atualizações geram muitos erros

Avaliação da unidade TOTVS: 7 - Prefere a TOTVS em comparação a outros sistemas trabalhados na hotelaria

Pontos adicionais: Não

Média das notas: 7

Classificação do Cliente: Neutro

Sugestões da IA: A TOTVS poderia melhorar o desenvolvimento e suporte do módulo Fiscal Flex e garantir atualizações de software que não gerem erros frequentes.

Print do data frame gerado. Vale ressaltar que a repetição de dados, como ID, tópicos e sugestões de IA, foi mantida para facilitar a integração com o PowerBI. No entanto, uma normalização dos dados, eliminando a repetição e transformando as linhas da coluna "tópico" em colunas separadas, poderia ser realizada caso o código fosse implementado em um banco de dados para um processo mais sofisticado.

	Tópico	Nota	Justificativa	ID	Sugestões da IA
6	Entrega do benefício esperado pelo departamento	Parcial	Reclamação em relação ao Fiscal Flex com falha...	2967631.wav	A TOTVS poderia melhorar o desenvolvimento e s...
7	Experiência com a última atualização do software	6	Atualizações geram muitos erros	2967631.wav	A TOTVS poderia melhorar o desenvolvimento e s...
8	Avaliação da unidade TOTVS	7	Prefere a TOTVS em comparação a outros sistema...	2967631.wav	A TOTVS poderia melhorar o desenvolvimento e s...
12	Probabilidade de recomendar a TOTVS a um amigo...	Áudio obstruído	Áudio obstruído	2967602.wav	Melhorar a entrega completa dos benefícios esp...
13	Agilidade nos processos do suporte técnico	10	Sem justificativa	2967602.wav	Melhorar a entrega completa dos benefícios esp...
...	...	...	...	...	...
941	Satisfação com a implantação do software	8	Foi muito bem	2874774.wav	Melhorar a entrega dos benefícios completos es...
943	Experiência com a última atualização do software	9	Tudo certo	2874774.wav	Melhorar a entrega dos benefícios completos es...
944	Avaliação da unidade TOTVS	10	Atendimento é 10, em especial, o analista Marc...	2874774.wav	Melhorar a entrega dos benefícios completos es...
948	Probabilidade de recomendar a TOTVS a um amigo...	10	O sistema é muito completo, muito bom.	2874830.wav	Melhorar a comunicação e eficiência do suporte...
950	Atendimento do suporte técnico	6	É difícil nosso contato.	2874830.wav	Melhorar a comunicação e eficiência do suporte...

```
# concatenando todos os DFs e transformando em csv
df_final = pd.concat(dfs, ignore_index=True)

# Dicionário de mapeamento de números por extenso para algarismos
numeros_extenso_para_algarismos = {
    'Um': 1,
    'Dois': 2,
    'Três': 3,
    'Quatro': 4,
    'Cinco': 5,
    'Seis': 6,
    'Sete': 7,
    'Oito': 8,
    'Nove': 9,
    'Dez': 10
}

# Substituindo os números por extenso pelos algarismos na coluna 'Nota'
df_final['Nota'] = df_final['Nota'].replace(numeros_extenso_para_algarismos)

# retirando os NaN's da coluna justificativa na parte de "Notas",
# "pontos adicionais(quando respondidos com 'não') e "classificação do cliente"
df_final['Justificativa'] = df_final['Justificativa'].fillna('Sem justificativa')

# transformando em csv e armazenando no Drive
df_final.to_csv('/content/drive/MyDrive/Challenge TOTVS Amostra de Dados v2s/respostas_gpt_4.csv', index=False)
```

Por fim, um link com exemplos dos áudios processados, transcrições, o csv final, algumas respostas da API e as instruções do ChatGPT: [exemplos](#)

## 2.1.4 Funcionamento do Projeto: Visualização

## DESCREVENDO OS DADOS

Os dados utilizados no projeto consistem nas transcrições das ligações de controle de satisfação do cliente da empresa Totvs. A partir dessas transcrições, estruturamos os dados em tabelas específicas para facilitar a criação do painel. Esse processo de estruturação visa organizar as informações de maneira a oferecer uma visão clara e útil para análise e geração de insights no dashboard.

### Tabela 1: Respostas dos Clientes

Esta tabela é a base de dados principal e armazena informações gerais sobre o que ocorre durante as chamadas de controle de satisfação do cliente. Os principais campos desta tabela são:

- **Tópico:** Cada pergunta realizada pelo atendente.
- **Nota/Texto:** Nota atribuída a cada tópico. Inicialmente, esta coluna pode conter notas numéricas ou frases curtas relacionadas à satisfação do cliente. Para uniformizar os dados, criamos um mapeamento para converter frases em notas numéricas, resultando em uma coluna de notas puramente numérica.
- **Justificativa:** Comentários adicionais fornecidos pelo cliente sobre o tópico perguntado ou explicações sobre a nota dada.
- **Nota:** Nota atribuída a cada tópico após a normalização dos dados.
- **ID chamada:** Código identificador que cada ligação

### Tabela 2: ClassificacaoCliente

Esta tabela é derivada da tabela original e tem o objetivo de agrupar os dados por cliente, permitindo uma classificação de cada cliente. Os campos desta tabela são:

- **ID Chamada:** Código identificador único de cada ligação.

- **Média Nota Cliente:** Média das notas atribuídas pelo cliente em todas as interações registradas. Esta métrica proporciona uma visão geral do nível de satisfação do cliente.
- **Classificação Cliente:** Classificação do cliente com base nas notas que ele forneceu. Os clientes são categorizados como:
  - **Detrator:** Clientes insatisfeitos, que deram notas abaixo de 7.
  - **Neutro:** Clientes que deram notas intermediárias, não muito positivas nem negativas notas entre 7 e 8.
  - **Promotor:** Clientes satisfeitos, que deram notas altas entre 9 e 10.

#### Tabela 3: Tópicos&Médias

Esta tabela é derivada da tabela original e visa consolidar os dados por tópico, permitindo uma análise detalhada da satisfação em relação a cada pergunta realizada. Os principais campos desta tabela são:

- **Tópico:** Cada pergunta realizada pelo atendente durante a chamada.
- **Média Nota Serviço:** Média das notas atribuídas para cada tópico. Esta métrica fornece uma visão geral do nível de satisfação dos clientes em relação a cada aspecto do serviço questionado.

#### Tabela 2: sugestãoIA

Esta tabela é derivada da tabela original e tem o objetivo agora de possibilitar a visualização da sugestão que a IA deu para cliente, permitindo assim que o público do Dashboard tenha não acesso apenas ao problema, mas também a uma possível solução. Os campos desta tabela são:

- **ID Chamada:** Código identificador único de cada ligação.
- **ID:** Código agora tratado, sem o .wav.
- **Sugestão IA:** Texto com o que foi sugerido para atender a dor de cada cliente

#### Tabela de mapeamento de Notas:

Descrição: Esta tabela é utilizada para tratar e transformar a coluna original "Nota/Texto" das transcrições das chamadas. O objetivo principal é normalizar as notas, convertendo frases e textos em valores numéricos e tratando casos específicos, como áudios obstruídos ou respostas sem parâmetros claros.

**Campos e Processos:**

- **Texto:** Coluna com a nota inicial ou comentários textuais fornecidos pelos clientes.
- **Nota:** Valor numérico atribuído após a conversão das respostas textuais e tratamento de dados. Inclui:
  - **Valores Positivos:** Notas convertidas a partir de feedback positivo ou neutro.
  - **Valores Negativos:** Notas atribuídas a áudios obstruídos ou respostas sem parâmetros suficientes para avaliação. Esses valores são usados para filtrar os dados e garantir que não influenciem nas métricas finais.

**Observações:**

**Tratamento de Dados:** Clientes com áudios obstruídos ou respostas sem parâmetros claros recebem valores negativos para garantir que esses casos não distorçam as notas médias ou outras métricas.

**Exclusão do Modelo Final:** Esta tabela é usada apenas para o tratamento dos dados e não é carregada no modelo final. A tabela pode ser visualizada apenas no Power Query, onde os dados são preparados e limpos antes de serem usados no dashboard e análises finais.

	A <sub>1</sub> Texto	i <sub>2</sub> Nota
1	Não tenho contato	-1
2	Não tenho parâmetro	-1
3	Parcial	5
4	Parcialmente	5
5	Não tenho conhecimento	-1
6	Não tenho participação	-1
7	Não tive participação	-1
8	Total	10
9	Não	3
10	Totalmente	10
11	Não tenho relacionamento	-1
12	Não participei	-1
13	Entrega	9
14	Bela	9
15	Foi bem	8
16	Não tenho acesso	-1
17	Não tenho anotado	-1
18	Entrega totalmente	10
19	Entrega total	10
20	Entrega tudo	10
21	Copos Amazono	-1
22	Bem	8
23	Não vou avaliar agora, porque se realmente for verdade essa desconti...	-3
24	Não percebi	-1
25	Nossa	-1
26	Não tive mais contato	-1
27	Não tenho como avaliar	-1
28	Não entrega	1

**MÉTRICAS:**

### 1. Média de Nota Dada pelos Clientes:

Média geral das notas atribuídas por todos os clientes em todas as interações.

Fornecer uma visão geral do nível de satisfação geral dos clientes.

### 2. Média de Nota por Serviço:

Média das notas atribuídas para cada serviço específico durante as chamadas.

Avaliar o desempenho de cada serviço individualmente, permitindo identificar quais serviços estão atendendo bem ou precisam de melhorias.

### 3. Serviço Melhor Avaliado e Pior Avaliado:

Identificação do serviço com a maior e a menor média de notas, respectivamente.

Destacar quais serviços são mais e menos apreciados pelos clientes, ajudando a direcionar ações para melhorias.

### 4. Quantidade de Clientes:

Número total de clientes que participaram das avaliações.

Compreender a amplitude da amostra e a representatividade das avaliações.

### 5. Classificação dos Clientes:

Percentual de clientes classificados como Detratores, Neutros e Promotores.

Oferecer uma visão geral da distribuição dos clientes em relação à sua satisfação.

### 6. Quantidade de Clientes Detratores, Promotores e Neutros:

Número absoluto de clientes em cada categoria de classificação:

Medir a quantidade de clientes em cada grupo para avaliar a saúde geral da satisfação do cliente e identificar áreas de ação.

## **KPI**

## Taxa de NPS (Net Promoter Score)

O Net Promoter Score (NPS) mede a satisfação geral e a lealdade dos clientes em relação à empresa. O NPS avalia a probabilidade de os clientes recomendarem o serviço ou produto da empresa a outros. Através desse KPI, é possível entender melhor a percepção dos clientes e identificar áreas de melhoria para promover uma imagem positiva da empresa.

### **Cálculo do NPS:**

- **Porcentagem de Promotores:** Percentual de clientes que deram notas de 9 ou 10.
- **Porcentagem de Detratores:** Percentual de clientes que deram notas de 0 a 6.
- **Fórmula do NPS:**

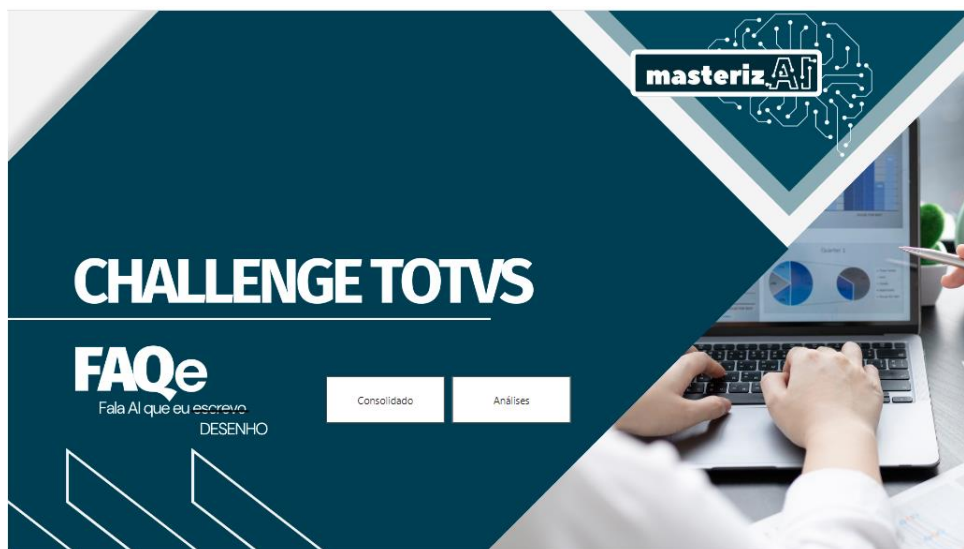
$$\text{NPS} = \% \text{ de Promotores} - \% \text{ de Detratores}$$

**Objetivo:** Atender à zona de excelência, alcançando um NPS de 75% ou mais para assegurar uma ótima imagem pública e fortalecer a lealdade dos clientes.

## **EXPLICANDO O DASHBOARD**

O propósito deste dashboard é fornecer à área de satisfação ao cliente informações e análises que facilitem a tomada de decisões estratégicas para aprimorar a experiência do cliente. Ele apresenta dados sobre a classificação dos clientes (Detratores, Neutros e Promotores), distribui clientes por essas categorias e avalia os tópicos tratados nas chamadas. Com a capacidade de aplicar filtros por tipo de cliente e tópicos, o dashboard permite uma visão detalhada e personalizada, ajudando a identificar áreas de melhoria e a otimizar as estratégias para aumentar a satisfação e lealdade dos clientes.

Capa do relatório



Painel 1: Visão Consolidada

Neste painel, são apresentados dados e insights rápidos sobre as dores do cliente, extraídos da transcrição e focados nos pontos críticos identificados.



Gráfico de Indicador de NP





Descrição: Mostra a taxa de NPS e o progresso em relação à meta de alcançar a zona de excelência (75% ou mais).

Objetivo: Visualizar a satisfação geral dos clientes e monitorar o desempenho em relação à meta

estabelecida.

Métricas: cálculo do NPS:  $NPS = \% \text{ de Promotores} - \% \text{ de Detratores}$

KPI: Melhoria do NPS

Filtros: Segmento do tópico tratado

### Cards de Quantidade de Cada Tipo de Cliente



Descrição: Exibe a quantidade total de clientes classificados em cada categoria: Detratores, Neutros e Promotores.

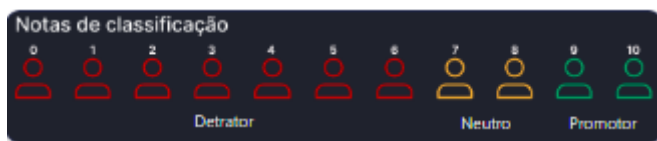
Objetivo: Fornecer uma visão rápida da distribuição dos clientes e identificar a proporção de cada grupo.

Métricas: Quantidade de cada classificação de clientes

KPI: Comparação entre quantidade de clientes por classificação

Filtros: Segmento do tópico tratado

### Descrição de notas e filtro de Classificação do Cliente:



Descrição: Permite filtrar os dados com base na classificação dos clientes

(Detratores, Neutros, Promotores).

Objetivo: Permitir uma análise detalhada das informações de acordo com a categoria de satisfação dos clientes.

Métricas: Clientes detratores, neutros e promotores

### Card de Quantidade de Clientes:



Descrição: Apresenta a quantidade de clientes que foram analisados.

Objetivo: Permitir observar o tamanho da população analisada

Métricas: Quantidade de clientes

### % áudios obstruídos:



Descrição: Apresenta a porcentagem de áudios obstruídos (não foi possível realizar a transcrição).

Objetivo: Permitir uma análise sobre a porcentagem de aproveitamento da transição.

Métricas: quantidade de áudios obstruídos por áudios gerais

### Ponto de Atenção:

PONTO CRÍTICO
Satisfação com o custo dos produtos contratados
Comentário recorrente
" É um produto caro."

Descrição: Cards com tópico com a menor média de notas.

Objetivo: Identificar a área que os clientes mais estão insatisfeitos e qual a maior reclamação.

Métricas: Tópico que possui menor nota e o comentário mais frequente sobre.

## Serviços a serem analisados:

SERVIÇOS ANALISADOS
Satisfação com o custo dos produtos contratados
Satisfação com a implantação do software
Probabilidade de recomendar a TOTVS a um amigo ou colega
Experiência com a última atualização do software
Entrega do benefício esperado pelo departamento
Avaliação da unidade TOTVS
Atendimento do suporte técnico
Atendimento do executivo de vendas
Agilidade nos processos do suporte técnico

Descrição: Serviços os quais foram analisados.

Objetivo: Permitir a visualização e identificação sobre o que está sendo tratado no projeto.

Métricas: Tópicos

## Painel 2: Visão Analítica

Neste painel, são apresentadas análises em cima da transcrição e focados nos pontos levantados pelos clientes.



### Filtro de Tópicos:

**TÓPICOS**

Todos

Descrição: Permite filtrar os dados de acordo com diferentes tópicos abordados nas chamadas.

Objetivo: Facilitar a análise de dados específicos para tópicos selecionados e entender melhor a satisfação relacionada a cada área.

Métricas: Tópicos comentados durante a chamada

### Filtro de Chamada:

**ID chamada**

Todos

Descrição: Permite filtrar os dados de acordo com cada ID da chamada.

Objetivo: Facilitar a análise de dados específicos por cliente selecionado e entender melhor a satisfação relacionada a cada cliente.

Métricas: Chamadas realizadas.

### Cards de Quantidade de Cada Tipo de Cliente:



Descrição: Exibe a quantidade total de clientes classificados em cada categoria: Detratores, Neutros e Promotores.

Objetivo: Fornecer uma visão rápida da distribuição dos clientes e identificar a proporção de cada grupo.

Métricas: Quantidade de cada classificação de clientes

KPI: Comparação entre quantidade de clientes por classificação

Filtros: Segmento do tópico tratado ou por ID da chamada.

### Tabela de Justificativas:

JUSTIFICATIVAS
"Acho que a Fósforo deveria ter dado mais prazo."
"Agora está funcionando perenamente."
"dois"
"é uma coisa muito tranquila"
"Ela atendia bem, mas não respondeu mais não."
"Eu fui contratado para finalizar a implantação."
"no outro dia"
"Os donos aqui reclamam bastante desse valor que está alto."
"Parcialmente"
"Pelo valor que a empresa paga, eu considero bom que eu suba ele aqui."
"tranquilo"
A atualização de versão, na prática, a gente não percebe muito.
A atualização tem um pouco mais de dificuldade.
A entrega não traz o que se espera.
A nossa implantação é mais complexa.
A quantidade de telas que tem no TOTVS acaba sendo algo que não é tão prático.
A Toca tem um péssimo relacionamento com o cliente.
A última atualização foi tranquila, sem surpresas.

Descrição: Exibe a justifica de cada cliente por tópico ou algum ponto que o mesmo quis apresentar.

Objetivo: Fornecer a visualização do que o cliente deu tal nota.

Métricas: Justificativas por tópico

KPI: Comparação entre tópicos

Filtros: Segmento do tópico tratado ou por ID de chamada.

Matriz com Tópicos e Suas Médias de Notas:

TÓPICOS	Nota Média
Agilidade nos processos do suporte técnico	7,26
Atendimento do executivo de vendas	7,69
Atendimento do suporte técnico	7,61
Avaliação da unidade TOTVS	7,80
Entrega do benefício esperado pelo departamento	7,60
Experiência com a última atualização do software	7,32
Probabilidade de recomendar a TOTVS a um amigo ou colega	7,60
Satisfação com a implantação do software	7,72
Satisfação com o custo dos produtos contratados	6,67

Descrição: Mostra os tópicos discutidos durante as chamadas e suas médias de notas atribuídas.

Objetivo: Avaliar o desempenho de cada tópico específico e identificar quais áreas estão sendo bem avaliadas ou precisam de atenção.

Métricas: Cada tópico tratado, e sua respectiva média de nota

Tabela de Sugestões:

Sugestão IA para cada cliente	
ID	Sugestões
2874774	Melhorar a entrega dos benefícios completos esperados pela empresa.
2874830	Melhorar a comunicação e eficiência do suporte técnico.
2961972	Melhorar a comunicação e atenção do atendimento de vendas e oferecer suporte adicional para personalizações em novas atualizações.
2962046	Melhorar a comunicação sobre atualizações do software com os usuários finais e considerar notificações diretas por e-mail.
2962074	Melhorar o suporte técnico e relacionamento com o cliente. Aumentar a qualificação dos técnicos para evitar erros graves.
2962184	Simplificar sistema de contato e reavaliar manutenção de licença inativa.
2962219	Melhorar a adequação do sistema às necessidades básicas dos clientes e revisar custos de atualização.

Descrição: Exibir uma possível solução em cima da dor de cada cliente.

Objetivo: Fornecer não só a visualização do projeto, mas também uma possível solução gerada pela IA.

Métricas: Sugestões geradas pela IA

Filtros: Segmento do tópico tratado ou por ID de chamada.

Link do GitHub para acessar o código, documentação e o dashboard funcional:

[GitHub](#)