

**FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA**  
**DATA SCIENCE**

RM 98119 – CESAR OLIVEIRA GOES

RM 97885 – FIAMA DOS SANTOS TRAJANO

RM 550759 – GABRIEL SILVA DE NEGREIROS LEAL DA ROCHA

RM 551770 – KARINA MACIEL PALMEIRA

**SPRINT 04: CHALLENGE TOTVS**



# SUMÁRIO

<b>INTRODUÇÃO .....</b>	<b>3</b>
<b>1. ARQUITETURA DA SOLUÇÃO FINAL .....</b>	<b>4</b>
<b>2. APACHE AIRFLOW: ORQUESTRAÇÃO .....</b>	<b>8</b>
<b>3. WEB APP USANDO O TERRAFORM NO OCI.....</b>	<b>14</b>
<b>3.1 INTERFACE GRÁFICA DE LOGIN .....</b>	<b>20</b>
<b>4. INTEGRAÇÃO DE DADOS WEB APP COM O STORAGE NO GCP.....</b>	<b>22</b>
<b>5. ARMAZENAMENTO DOS ÁUDIOS .....</b>	<b>23</b>
<b>6. ARMAZENAMENTO NO BANCO TRANSACIONAL MONGODB.....</b>	<b>24</b>
<b>7. INSTALAÇÃO DO MONGO DB VIA LINUX.....</b>	<b>28</b>
<b>7.1. INTEGRAÇÃO DE DADOS DO MONGO DB COM O GOOGLE COLAB .....</b>	<b>29</b>
<b>7.2. BACKUP DO MONGO DB .....</b>	<b>30</b>
<b>8. TRATAMENTO DE ÁUDIO: AUDACITY .....</b>	<b>31</b>
<b>9. SCRIPTS PYTHON DE TRANSCRIÇÃO DE ÁUDIO, ANÁLISE DE SENTIMENTO E DATAFRAME .....</b>	<b>32</b>
<b>10. DATA WAREHOUSE .....</b>	<b>47</b>
<b>10.1 INSTALAÇÃO DO MYSQL .....</b>	<b>50</b>
<b>10.2 SCRIPT SQL PARA CRIAÇÃO DAS TABELAS.....</b>	<b>52</b>
<b>10.3 POPULANDO O DATA WAREHOUSE .....</b>	<b>57</b>
<b>10.4 BACKUP DO WAREHOUSE MYSQL .....</b>	<b>61</b>
<b>11. AMBIENTE DE BUSINESS INTELLIGENCE .....</b>	<b>62</b>
<b>11.1 CÓDIGO N-GRAMS .....</b>	<b>76</b>
<b>12. GOVERNANÇA DE DADOS E LGDP .....</b>	<b>82</b>
<b>LINKS PARA PITCH COMERCIAL E VÍDEO TÉCNICO.....</b>	<b>83</b>
<b>CONCLUSÃO.....</b>	<b>84</b>
<b>FONTES .....</b>	<b>85</b>

# INTRODUÇÃO

Esta documentação tem como objetivo apresentar uma visão técnica do nosso protótipo final para o Challenge FIAP Desafio TOTVS, que consiste em desenvolver uma solução para melhorar o sistema de NPS (Net Promoter Score).

A TOTVS, ao fornecer seus produtos e consultoria aos seus clientes, realiza continuamente um processo de avaliação de satisfação por meio de ligações. Essas ligações têm por objetivo medir a satisfação do cliente por meio do NPS de seus produtos e serviços de outsourcing, classificando-os dentro de suas expectativas. Essas classificações são: Detratores, Neutros e Promotores.

Os áudios das ligações são transcritos para serem analisados por meio de técnicas de NLP (Natural Language Processing) e reconhecimento de voz usando Python. Contudo, essas transcrições possuem baixa qualidade devido aos ruídos no áudio, às gírias, dialetos e sotaques regionais, além de poucas ferramentas de suporte para a língua portuguesa. Assim, a baixa qualidade nas transcrições impacta diretamente na tomada de decisão por parte da liderança da TOTVS, pois, com dados de baixa qualidade, os insights consequentemente não serão tão eficazes.

Dado este contexto, nesta documentação será abordado, primeiramente, como encontrar soluções para esse problema em um trabalho que envolve as mais diversas áreas de profissionais de dados e tecnologia.

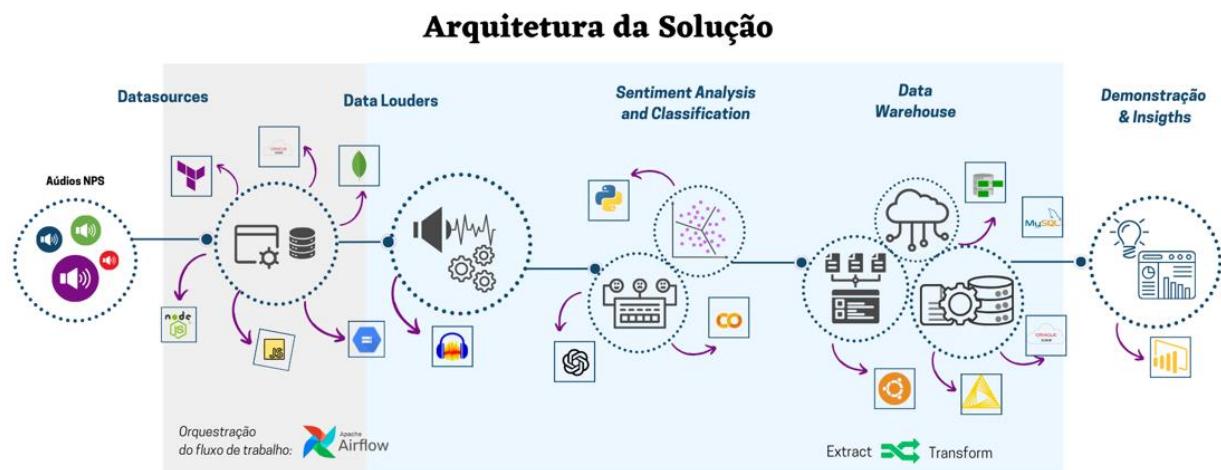
Esta documentação apresentará detalhadamente as mudanças que fizemos ao longo das sprints, desde nossa arquitetura de solução inicial até o que entregaremos na solução final, demonstrando que em projetos de longa duração há diversos obstáculos e que, dentro da metodologia ágil, mudamos algumas ferramentas que julgamos melhores para o que queríamos entregar.

Em seguida, apresentaremos um detalhamento de cada fase do pipeline e de suas ferramentas e códigos aplicados, seguindo nossa arquitetura de solução final de forma lógica.

Por fim, pode-se dizer que esta documentação é a visão do nosso protótipo final explicado com detalhes, no qual esperamos ter alcançado uma excelente solução para o problema apresentado pela TOTVS.

# 1. ARQUITETURA DA SOLUÇÃO FINAL

A solução **InoVoice** foi desenvolvida para fornecer uma abordagem completa e escalável no tratamento de áudios de interações entre clientes e atendentes da TOTVS, visando melhorar a análise de satisfação via **NPS (Net Promoter Score)**. A arquitetura foi projetada para otimizar o processamento de dados, garantindo precisão nas análises e proporcionando insights valiosos para a tomada de decisões estratégicas. Abaixo, descrevemos cada etapa da arquitetura, conforme ilustrado no diagrama.



## Datasources:

**Node.js e WebApp:** O primeiro passo da solução envolve um WebApp criado em **Node.js**, que permite aos usuários inserir áudios diretamente na interface gráfica. Esses áudios, fornecidos pela TOTVS, são enviados para o banco de dados NoSQL via WebApp. A escolha pelo **Node.js** foi feita devido à sua execução em **JavaScript**, facilitando a criação de interfaces dinâmicas e interativas, conforme ilustrado no diagrama.

**MongoDB:** O MongoDB está hospedado em uma máquina virtual dentro do **Oracle Cloud Infrastructure (OCI)**, aproveitando a escalabilidade e o custo-benefício oferecidos pela plataforma de nuvem da Oracle.

**Oracle Cloud Infrastructure (OCI):** O OCI foi escolhido para hospedar a infraestrutura, fornecendo um ambiente seguro e escalável para o MongoDB e outros componentes da solução. O uso de **Terraform** para provisionar a infraestrutura permite automação de recursos em várias plataformas de nuvem, tornando a gestão e o escalonamento mais eficientes.

**Google Cloud Platform (GCP):** O GCP armazenará em seu storage free tier, os áudios que foram colocados no webapp. Desse storage, os áudios irão para o MongoDB.

### **Datalouders:**

O Audacity é um software de código aberto e gratuito, amplamente utilizado para a edição e tratamento de áudios. No contexto de gravações de ligações entre atendentes e clientes para análise de NPS (Net Promoter Score), o Audacity pode desempenhar um papel importante em diversas etapas do processamento de áudio, como na limpeza de áudio e na conversão de formatos de áudio.

O Audacity pode ser utilizado para preparar as gravações para análise, otimizando a qualidade do áudio, removendo ruídos, editando partes irrelevantes e convertendo os arquivos para formatos adequados.

### **Sentiment Analysis and Classification:**

Nesta etapa, após a limpeza dos áudios e seu armazenamento no banco de dados, extraímos os arquivos para um ambiente de desenvolvimento (IDE). Optamos por utilizar o Google Colab devido à sua praticidade em processos de manipulação de dados, suporte abrangente a bibliotecas e alta escalabilidade.

No ambiente Python, os áudios são transcritos e suas transcrições são salvas em formato .txt. A seguir, utilizamos essas transcrições para treinar modelos de **Machine Learning** e **Deep Learning** para realizar a análise de sentimentos. Essa análise tem o objetivo de identificar a classificação NPS (Net Promoter Score) associada a cada transcrição, ajudando a identificar clientes Detratores, Neutros ou Promotores.

Os dados resultantes são organizados em um dataset que contém:

- As notas atribuídas a cada produto e serviço mencionados na transcrição;
- A média das notas do NPS;
- A classificação final do NPS.

Esse dataset, por sua vez, será utilizado para alimentar o **Data Warehouse**, sendo posteriormente integrado ao ambiente de **Business Intelligence (BI)**, onde análises mais aprofundadas e visualizações podem ser realizadas para suportar a tomada de decisões.

As principais ferramentas utilizadas nesta etapa incluem o **Google Colab** como IDE, a linguagem de programação **Python**, e a biblioteca **Whisper** da OpenAI, que é usada para transcrição de áudio. A API do Whisper é gratuita para uso em IDEs e se mostrou adequada para nossas necessidades de processamento de áudio.

### **Data Warehouse:**

**Oracle Data Modeler & MySQL:** O **dataset** resultante da análise de sentimentos é carregado em um **Data Warehouse** para consultas e análises posteriores. Utilizamos o **Oracle Data Modeler** para modelar o Data Warehouse com base na metodologia **Star Schema** de Ralph Kimball, ideal para consultas rápidas e eficientes.

O **MySQL**, hospedado no OCI, é utilizado como banco de dados relacional para armazenar o dataset de notas NPS. Sua robustez e eficiência o tornam uma escolha ideal para atender às necessidades de performance e escalabilidade da solução.

**Backup e Recovery com mysqldump:** Para garantir a segurança e a integridade dos dados, implementamos políticas de backup usando o **mysqldump**. Esse comando permite realizar backups lógicos dos bancos de dados, gerando arquivos em formato **SQL**, **CSV** ou **XML** para fácil recuperação em caso de falhas.

### **Demonstração e insights:**

Power BI: Utilizaremos o Power BI como ferramenta de Business Intelligence, pois, possui uma interface de fácil usabilidade, capacidade de processar grandes volumes de dados com velocidade, integração com diversas fontes de dados e visualizações que nos permite demonstrar os resultados obtidos em nossa solução final de forma clara e objetiva.

### **Orquestração:**

Apache Airflow: Desempenha um papel essencial na orquestração do fluxo de trabalho, coordenando as diferentes etapas do pipeline de processamento de dados. O Apache Airflow atua como o "cérebro" por trás de toda a orquestração do fluxo de trabalho, garantindo que as tarefas sejam executadas na ordem correta e que todas as dependências sejam satisfeitas. Ele automatiza cada etapa, desde a coleta e preparação dos áudios, passando pela transcrição e análise de sentimentos, até o carregamento dos dados processados no Data Warehouse e integração com ferramentas de BI.

A estimativa de custo de processamento prevê uma **redução de 27%** quando comparada a outras arquiteturas mais robustas e caras oferecidas por grandes players do mercado. O cliente **TOTVS** economiza significativamente com o **InoVoice**, que oferece uma arquitetura otimizada e de menor custo. Essa economia é alcançada graças ao uso de ferramentas como o **Apache Airflow** para orquestração, o **KNIME** para ETL e a **API Whisper** integrada diretamente no **Google Colab**, eliminando a necessidade de bibliotecas caras e oferecendo uma abordagem mais flexível e eficiente para o processamento de dados. Isso permite manter altos níveis de desempenho e escalabilidade sem incorrer em custos elevados, focando nas fases que costumam gerar mais despesas nos concorrentes.

A arquitetura de solução do InoVoice segue os seguintes princípios:

- **Eficiência nas operações:** Agiliza coleta, processamento e análise, otimizando os fluxos de trabalho, consequentemente reduzindo os custos do parceiro.
- **Precisão nas Análises:** Garante insights detalhados e altamente precisos sobre as percepções dos clientes.
- **Escalabilidade e Confiabilidade dos Dados:** Gerencia grandes volumes de dados de forma eficaz, mantendo desempenho consistente.

- **Flexibilidade e Customização:** Permitindo-se a todos momentos as adaptações específicas às necessidades da TOTVS, desde ajustes nos algoritmos até personalização dos painéis de visualização.
- **Insights de valor:** Auxiliar a TOTVS a tomar decisões estratégicas fundamentadas, impulsionando a excelência operacional e aprimorando a experiência do cliente com base nas pesquisas NPS, de forma mais assertiva.

Em resumo, nossa arquitetura foi projetada para fornecer uma abordagem completa e escalável para lidar com os dados de interações dos clientes das pesquisas NPS (*Net Promoter Score*) do parceiro TOTVS. Cada elemento foi cuidadosamente selecionado com base em sua capacidade de oferecer eficiência, precisão e insights relevantes, garantindo assim uma solução robusta e orientada para resultados.

## 2. APACHE AIRFLOW: ORQUESTRAÇÃO

Nesta parte da documentação, mostraremos como instalar, inicializar e executar o Apache Airflow para orquestração de dados de nossa solução

### 1: Atualização das dependências do sistema:

```
[opc@inovoice-rr-airflow ~]$ sudo yum update -y
```

```
Last metadata expiration check: 3:09:07 ago on Fri 20 Sep 2024 07:22:03 PM GMT.  
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
kernel-uek	x86_64	5.15.0-210.163.7.el8uek	ol8_UEKR7	2.5 M
kernel-uek-core	x86_64	5.15.0-210.163.7.el8uek	ol8_UEKR7	61 M
kernel-uek-devel	x86_64	5.15.0-210.163.7.el8uek	ol8_UEKR7	21 M
kernel-uek-modules	x86_64	5.15.0-210.163.7.el8uek	ol8_UEKR7	69 M
Upgrading:				
audit	x86_64	3.1.2-1.0.1.el8	ol8_baseos_latest	264 k
auditlibs	x86_64	3.1.2-1.0.1.el8	ol8_baseos_latest	124 k
bind-export-libs	x86_64	32:9.11.36-16.el8_10.2	ol8_baseos_latest	1.1 M
bind-libs	x86_64	32:9.11.36-16.el8_10.2	ol8_appstream	176 k
bind-lites	x86_64	32:9.11.36-16.el8_10.2	ol8_appstream	1.2 M
bind-license	noarch	32:9.11.36-16.el8_10.2	ol8_appstream	104 k
bind-utils	x86_64	32:9.11.36-16.el8_10.2	ol8_appstream	453 k
bpftrace	x86_64	5.15.0-210.163.7.el8uek	ol8_UEKR7	3.2 M
bubblewrap	x86_64	0.4.0-2.el8_10	ol8_baseos_latest	50 k
ca-certificates	noarch	2024.2.69-v8.0.303-80.0.el8_10	ol8_baseos_latest	981 k
cloud-init	noarch	23.4-7.0.1.el8_10.7	ol8_appstream	1.3 M
curl	x86_64	7.61.1-34.el8_10.2	ol8_baseos_latest	352 k
firewalld	noarch	0.9.11-8.0.1.el8_10	ol8_baseos_latest	509 k
firewalld-filesystem	noarch	0.9.11-8.0.1.el8_10	ol8_baseos_latest	78 k
glIBC	x86_64	2.28-251.0.2.el8_10.4	ol8_baseos_latest	2.2 M
glIBC-common	x86_64	2.28-251.0.2.el8_10.4	ol8_baseos_latest	1.0 M
glIBC-devel	x86_64	2.28-251.0.2.el8_10.4	ol8_baseos_latest	90 k
glIBC-gconv-extra	x86_64	2.28-251.0.2.el8_10.4	ol8_baseos_latest	1.6 M
glIBC-headers	x86_64	2.28-251.0.2.el8_10.4	ol8_baseos_latest	495 k
glIBC-langpack-en	x86_64	2.28-251.0.2.el8_10.4	ol8_baseos_latest	834 k
initscripts	x86_64	10.00.18-1.0.2.el8	ol8_baseos_latest	339 k
iproute	x86_64	6.8.0-2.el8_10	ol8_UEKR7	873 k
iproute-tc	x86_64	6.8.0-2.el8_10	ol8_UEKR7	451 k

```
[opc@inovoice-rr-airflow ~]$ iproute-6.8.0-2.el8_10.x86_64  
iproute-tc-6.8.0-2.el8_10.x86_64  
krb5-devel-1.18.2-29.0.1.el8_10.x86_64  
libcurl-7.61.1-34.el8_10.2.x86_64  
libipa_hbac-2.9.4-4.0.1.el8_10.x86_64  
libkadm5-1.18.2-29.0.1.el8_10.x86_64  
libss- certmap-2.9.4-4.0.1.el8_10.x86_64  
libsss_nss_idmap-2.9.4-4.0.1.el8_10.x86_64  
mdadm-4.2-14.0.4.el8_10.x86_64  
nss-3.101.0-7.el8_8.x86_64  
nss-softokn-freebl-3.101.0-7.el8_8.x86_64  
nss-util-3.101.0-7.el8_8.x86_64  
openssl-clients-8.0p1-25.0.1.el8_10.x86_64  
pcp-5.3.7-22.0.1.el8_10.x86_64  
pcp-doc-5.3.7-22.0.1.el8_10.noarch  
pcp-pmda-dm-5.3.7-22.0.1.el8_10.x86_64  
pcp-pmda-openmetrics-5.3.7-22.0.1.el8_10.x86_64  
pcp-system-tools-5.3.7-22.0.1.el8_10.x86_64  
platform-python-setuptools-39.2.0-8.el8_10.noarch  
python3-bind-32:9.11.36-16.el8_10.noarch  
python3-firewall-0.9.11-8.0.1.el8_10.noarch  
python3-libdnf-0.63.0-20.0.1.el8_10.x86_64  
python3-setuptools-39.2.0-8.el8_10.noarch  
python3-sssdconfig-2.9.4-4.0.1.el8_10.noarch  
python36-oci-sdk-2.134.0-1.el8_10.x86_64  
source-highlight-3.1.8-18.el8_10.x86_64  
sssd-ad-2.9.4-4.0.1.el8_10.x86_64  
sssd-common-2.9.4-4.0.1.el8_10.x86_64  
sssd-ipa-2.9.4-4.0.1.el8_10.x86_64  
sssd-ldap-2.9.4-4.0.1.el8_10.x86_64  
sssd-proxy-2.9.4-4.0.1.el8_10.x86_64  
  
Installed:  
kernel-uek-5.15.0-210.163.7.el8uek.x86_64  
kernel-uek-devel-5.15.0-210.163.7.el8uek.x86_64  
  
Complete!  
[opc@inovoice-rr-airflow ~]$
```

## 2: Instalação das dependências necessárias do Airflow:

```
[opc@inovoice-rr-airflow ~]$ sudo yum install -y gcc python3 python3-devel python3-pip libpq-devel git
Last metadata expiration check: 3:14:25 ago on Fri 20 Sep 2024 07:22:03 PM GMT.
Package gcc-8.5.0-22.0.1.el8_10.x86_64 is already installed.
Package python36-3.6.8-39.module+el8.10.0+90274+07ba55de.x86_64 is already installed.
Package python3-pip-9.0.3-24.el8.noarch is already installed.
Dependencies resolved.

=====
| Package           | Architecture | Version      | Repository | Size |
|=====             |=====         |=====        |=====       |===== |
| Installing:      |              |              |            |        |
|   git             | x86_64       | 2.43.5-1.el8_10 | ol8_appstream | 91 k |
|   libpq-devel    | x86_64       | 13.11-1.el8     | ol8_appstream | 98 k |
|   python36-devel | x86_64       | 3.6.8-39.module+el8.10.0+90274+07ba55de | ol8_appstream | 15 k |
| Installing dependencies: |          |              |            |        |
|   git-core        | x86_64       | 2.43.5-1.el8_10 | ol8_appstream | 11 M |
|   git-core-doc   | noarch       | 2.43.5-1.el8_10 | ol8_appstream | 3.1 M |
|   libpq           | x86_64       | 13.11-1.el8     | ol8_appstream | 198 k |
|   perl-Error     | noarch       | 1:0.17025-2.el8 | ol8_appstream | 46 k |
|   perl-Git        | noarch       | 2.43.5-1.el8_10 | ol8_appstream | 78 k |
|   perl-TermReadkey | x86_64       | 2.37-7.el8     | ol8_appstream | 40 k |
|   platform-python-devel | x86_64       | 3.6.8-62.0.1.el8_10 | ol8_appstream | 240 k |
|   python-rpm-macros | noarch       | 3-45.el8       | ol8_appstream | 16 k |
|   python-srpm-macros | noarch       | 3-45.el8       | ol8_appstream | 16 k |
|   python3-rpm-generators | noarch       | 5-8.el8        | ol8_appstream | 25 k |
|   python3-rpm-macros | noarch       | 3-45.el8       | ol8_appstream | 15 k |
| Transaction Summary |          |              |            |        |
|=====             |=====         |=====        |=====       |===== |
| Install 14 Packages |          |              |            |        |
| Total download size: 15 M |          |              |            |        |
| Installed size: 48 M |          |              |            |        |
| Downloading Packages: |          |              |            |        |
| (1/14): git-core-doc-2.43.5-1.el8_10.noarch.rpm | 19 MB/s | 3.1 MB | 00:00 |
| (2/14): git-2.43.5-1.el8_10.x86_64.rpm | 556 kB/s | 91 kB | 00:00 |
```

```
3.64.181.164.115 (opc)
=====
| Installing      : git-core-doc-2.43.5-1.el8_10.noarch | 4/14 |
| Installing      : python3-rpm-macros-3-45.el8.noarch | 5/14 |
| Installing      : python3-rpm-generators-5-8.el8.noarch | 6/14 |
| Installing      : platform-python-devel-3.6.8-62.0.1.el8_10.x86_64 | 7/14 |
| Installing      : perl-TermReadKey-2.37-7.el8.x86_64 | 8/14 |
| Installing      : perl-Error-1:0.17025-2.el8.noarch | 9/14 |
| Installing      : perl-Git-2.43.5-1.el8_10.noarch | 10/14 |
| Installing      : git-2.43.5-1.el8_10.x86_64 | 11/14 |
| Installing      : libpq-13.11-1.el8.x86_64 | 12/14 |
| Installing      : libpq-devel-13.11-1.el8.x86_64 | 13/14 |
| Installing      : python36-devel-3.6.8-39.module+el8.10.0+90274+07ba55de.x86_64 | 14/14 |
| Running scriptlet: python36-devel-3.6.8-39.module+el8.10.0+90274+07ba55de.x86_64 | 14/14 |
| Verifying       : git-2.43.5-1.el8_10.x86_64 | 1/14 |
| Verifying       : git-core-2.43.5-1.el8_10.x86_64 | 2/14 |
| Verifying       : git-core-doc-2.43.5-1.el8_10.noarch | 3/14 |
| Verifying       : libpq-13.11-1.el8.x86_64 | 4/14 |
| Verifying       : libpq-devel-13.11-1.el8.x86_64 | 5/14 |
| Verifying       : perl-Error-1:0.17025-2.el8.noarch | 6/14 |
| Verifying       : perl-Git-2.43.5-1.el8_10.noarch | 7/14 |
| Verifying       : perl-TermReadKey-2.37-7.el8.x86_64 | 8/14 |
| Verifying       : platform-python-devel-3.6.8-62.0.1.el8_10.x86_64 | 9/14 |
| Verifying       : python-rpm-macros-3-45.el8.noarch | 10/14 |
| Verifying       : python-srpm-macros-3-45.el8.noarch | 11/14 |
| Verifying       : python3-rpm-generators-5-8.el8.noarch | 12/14 |
| Verifying       : python3-rpm-macros-3-45.el8.noarch | 13/14 |
| Verifying       : python36-devel-3.6.8-39.module+el8.10.0+90274+07ba55de.x86_64 | 14/14 |
| Installed:      git-2.43.5-1.el8_10.x86_64 | git-core-2.43.5-1.el8_10.x86_64 |
|                  git-core-doc-2.43.5-1.el8_10.noarch | libpq-13.11-1.el8.x86_64 |
|                  libpq-devel-13.11-1.el8.x86_64 | perl-Error-1:0.17025-2.el8.noarch |
|                  perl-Git-2.43.5-1.el8_10.noarch | perl-TermReadKey-2.37-7.el8.x86_64 |
|                  platform-python-devel-3.6.8-62.0.1.el8_10.x86_64 | python-rpm-macros-3-45.el8.noarch |
|                  python-srpm-macros-3-45.el8.noarch | python3-rpm-generators-5-8.el8.noarch |
|                  python36-devel-3.6.8-39.module+el8.10.0+90274+07ba55de.x86_64 | python36-devel-3.6.8-39.module+el8.10.0+90274+07ba55de.x86_64 |
| Complete!
[opc@inovoice-rr-airflow ~]$
```

## 3: Configuração do ambiente Python e ativação:

```
[opc@inovoice-rr-airflow ~]$ mkdir ~/airflow && cd ~/airflow
[opc@inovoice-rr-airflow airflow]$ python3 -m venv venv
[opc@inovoice-rr-airflow airflow]$ source venv/bin/activate
```

## 4: Atualização do Python dentro do ambiente e do pip, para não haver erros na instalação, pois o Airflow depende do python para executar:

```
(venv) [opc@inovoice-rr-airflow airflow]$ pip install --upgrade pip
Collecting pip
  Downloading https://files.pythonhosted.org/packages/d4/55/90db48d85f7689ec6f81c0db0622d704306c5284850383c090e6c7195a5c/pip-24.2-py3-none-any.whl (1.8MB)
    |██████████| 1.8MB 22.2MB/s
Installing collected packages: pip
  Found existing installation: pip 19.3.1
  Uninstalling pip-19.3.1...
    Successfully uninstalled pip-19.3.1
Successfully installed pip-24.2
(venv) [opc@inovoice-rr-airflow airflow]$
```

## 5: Instalação do Airflow:

```
(venv) [opc@inovoice-rr-airflow airflow]$ pip install apache-airflow==2.2.5
Collecting apache-airflow==2.2.5
  Using cached apache_airflow-2.2.5-py3-none-any.whl.metadata (98 kB)
Collecting alembic<2.0,>=1.5.1 (from apache-airflow==2.2.5)
  Downloading alembic-1.13.2-py3-none-any.whl.metadata (7.4 kB)
Collecting argcomplete<3.0,>=1.10 (from apache-airflow==2.2.5)
  Downloading argcomplete-2.1.2-py3-none-any.whl.metadata (17 kB)
Collecting attrs<21.0,>=20.0 (from apache-airflow==2.2.5)
  Using cached attrs-20.3.0-py2.py3-none-any.whl.metadata (10 kB)
Collecting blinker (from apache-airflow==2.2.5)
  Downloading blinker-1.8.2-py3-none-any.whl.metadata (1.6 kB)
Collecting clickclick<1.2 (from apache-airflow==2.2.5)
  Using cached clickclick-20.10.2-py2.py3-none-any.whl.metadata (7.6 kB)
Collecting colorlog<7.0,>=4.0.2 (from apache-airflow==2.2.5)
  Downloading colorlog-6.8.2-py3-none-any.whl.metadata (10 kB)
Collecting connexion>=2.10.0 (from connexion[flask,swagger-ui]>=2.10.0->apache-airflow==2.2.5)
  Downloading connexion-3.1.0-py3-none-any.whl.metadata (12 kB)
Collecting croniter>=0.3.17 (from apache-airflow==2.2.5)
  Downloading croniter-3.0.3-py2.py3-none-any.whl.metadata (28 kB)
Collecting cryptography>=0.9.3 (from apache-airflow==2.2.5)
  Downloading cryptography-43.0.1-cp37abi3-manylinux_2_28_x86_64.whl.metadata (5.4 kB)
Collecting deprecated<1.2.13 (from apache-airflow==2.2.5)
  Downloading Deprecated-1.2.14-py2.py3-none-any.whl.metadata (5.4 kB)
Collecting dill<0.4,>=0.2.2 (from apache-airflow==2.2.5)
  Downloading dill-0.3.8-py3-none-any.whl.metadata (10 kB)
Collecting docutils<0.17 (from apache-airflow==2.2.5)
  Using cached docutils-0.16-py2.py3-none-any.whl.metadata (2.7 kB)
Collecting flask<2.0,>=1.1.0 (from apache-airflow==2.2.5)
  Using cached Flask-1.1.4-py2.py3-none-any.whl.metadata (4.6 kB)
Collecting flask-appbuilder==3.4.5 (from apache-airflow==2.2.5)
  Using cached Flask_AppBuilder-3.4.5-py3-none-any.whl.metadata (10 kB)
Collecting flask-caching<2.0.0,>=1.5.0 (from apache-airflow==2.2.5)
  Downloading Flask_Caching-1.11.1-py3-none-any.whl.metadata (2.2 kB)
```

```
(venv) [opc@inovoice-rr-airflow airflow]$ pip install apache-airflow
Requirement already satisfied: apache-airflow in ./venv/lib/python3.8/site-packages (2.2.5)
Requirement already satisfied: alembic<2.0,>=1.5.1 in ./venv/lib/python3.8/site-packages (from apache-airflow) (1.13.2)
Requirement already satisfied: argcomplete<3.0,>=1.10 in ./venv/lib/python3.8/site-packages (from apache-airflow) (2.1.2)
Requirement already satisfied: attrs<21.0,>=20.0 in ./venv/lib/python3.8/site-packages (from apache-airflow) (20.3.0)
Requirement already satisfied: blinker in ./venv/lib/python3.8/site-packages (from apache-airflow) (1.8.2)
Requirement already satisfied: clickclick<1.2 in ./venv/lib/python3.8/site-packages (from apache-airflow) (20.10.2)
Requirement already satisfied: colorlog<7.0,>=4.0.2 in ./venv/lib/python3.8/site-packages (from apache-airflow) (6.8.2)
Requirement already satisfied: connexion>=2.10.0 in ./venv/lib/python3.8/site-packages (from connexion[flask,swagger-ui]>=2.10.0->apache-airflow) (2.14.2)
Requirement already satisfied: croniter>=0.3.17 in ./venv/lib/python3.8/site-packages (from apache-airflow) (3.0.3)
Requirement already satisfied: cryptography>=0.9.3 in ./venv/lib/python3.8/site-packages (from apache-airflow) (43.0.1)
Requirement already satisfied: deprecated<1.2.13 in ./venv/lib/python3.8/site-packages (from apache-airflow) (1.2.14)
Requirement already satisfied: dill<0.4,>=0.2.2 in ./venv/lib/python3.8/site-packages (from apache-airflow) (0.3.8)
Requirement already satisfied: docutils<0.17 in ./venv/lib/python3.8/site-packages (from apache-airflow) (0.16)
Requirement already satisfied: flask<2.0,>=1.1.0 in ./venv/lib/python3.8/site-packages (from apache-airflow) (1.1.4)
Requirement already satisfied: flask-appbuilder==3.4.5 in ./venv/lib/python3.8/site-packages (from apache-airflow) (3.4.5)
Requirement already satisfied: flask-caching<2.0.0,>=1.5.0 in ./venv/lib/python3.8/site-packages (from apache-airflow) (1.11.1)
Requirement already satisfied: flask-login<0.5,>=0.3 in ./venv/lib/python3.8/site-packages (from apache-airflow) (0.4.1)
Requirement already satisfied: flask-session<=0.4.0,>=0.3.1 in ./venv/lib/python3.8/site-packages (from apache-airflow) (0.4.0)
Requirement already satisfied: flask-wtf<0.15,>=0.14.3 in ./venv/lib/python3.8/site-packages (from apache-airflow) (0.14.3)
Requirement already satisfied: graphviz>=0.12 in ./venv/lib/python3.8/site-packages (from apache-airflow) (0.20.3)
Requirement already satisfied: gunicorn>=20.1.0 in ./venv/lib/python3.8/site-packages (from apache-airflow) (23.0.0)
Requirement already satisfied: httpx in ./venv/lib/python3.8/site-packages (from apache-airflow) (0.27.2)
Requirement already satisfied: iso8601<=0.1.19 in ./venv/lib/python3.8/site-packages (from apache-airflow) (2.1.0)
Requirement already satisfied: itsdangerous<2.0,>=1.1.0 in ./venv/lib/python3.8/site-packages (from apache-airflow) (1.1.0)
Requirement already satisfied: jinja2<3.1,>=2.10.1 in ./venv/lib/python3.8/site-packages (from apache-airflow) (2.11.3)
Requirement already satisfied: jsonschema<=3.0 in ./venv/lib/python3.8/site-packages (from apache-airflow) (3.2.0)
Requirement already satisfied: lazy-object-proxy<=.venv/lib/python3.8/site-packages (from apache-airflow) (1.10.0)
Requirement already satisfied: lockfile<=0.12.2 in ./venv/lib/python3.8/site-packages (from apache-airflow) (0.12.2)
Requirement already satisfied: markdown<4.0,>=2.5.2 in ./venv/lib/python3.8/site-packages (from apache-airflow) (3.7)
Requirement already satisfied: markupsafe<2.1.0,>=1.1.1 in ./venv/lib/python3.8/site-packages (from apache-airflow) (2.0.1)
Requirement already satisfied: marshmallow-oneofschema>=2.0.1 in ./venv/lib/python3.8/site-packages (from apache-airflow) (3.1.1)
Requirement already satisfied: packaging>=14.0 in ./venv/lib/python3.8/site-packages (from apache-airflow) (24.1)
Requirement already satisfied: pendulum<=2.0 in ./venv/lib/python3.8/site-packages (from apache-airflow) (2.1.2)
Requirement already satisfied: psutil<6.0.0,>=4.2.0 in ./venv/lib/python3.8/site-packages (from apache-airflow) (5.9.8)
```

```
Requirement already satisfied: dnspython>=1.15.0 in ./venv/lib/python3.8/site-packages (from email-validator<2,>=1.0.5->flask-appbuilder==3.4.5->apache-airflow) (2.6.1)
Requirement already satisfied: Babel>=2.3 in ./venv/lib/python3.8/site-packages (from Flask-Babel<3,>=1->flask-appbuilder==3.4.5->apache-airflow) (2.16.0)
Requirement already satisfied: mdurl<=0.1 in ./venv/lib/python3.8/site-packages (from markdown-it-py>=2.2.0->rich>=9.2.0->apache-airflow) (0.1.2)
Requirement already satisfied: charset-normalizer<4,>=2 in ./venv/lib/python3.8/site-packages (from requests<3,>=2.9.1->connexion>=2.10.0->connexion[flask,swagger-ui]>=2.10.0->apache-airflow) (3.3.2)
Requirement already satisfied: urllib3<3,>=1.21.1 in ./venv/lib/python3.8/site-packages (from requests<3,>=2.9.1->connexion>=2.10.0->connexion[flask,swagger-ui]>=2.10.0->apache-airflow) (2.2.3)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in ./venv/lib/python3.8/site-packages (from aiohttp--apache-airflow-providers-http->apache-airflow) (2.4.0)
Requirement already satisfied: aiosignal<=1.1.2 in ./venv/lib/python3.8/site-packages (from aiohttp--apache-airflow-providers-http--apache-airflow) (1.3.1)
Requirement already satisfied: frozenlist<=1.1.1 in ./venv/lib/python3.8/site-packages (from aiohttp--apache-airflow-providers-http--apache-airflow) (1.4.1)
Requirement already satisfied: multidict<7.0,>=4.5 in ./venv/lib/python3.8/site-packages (from aiohttp--apache-airflow-providers-http--apache-airflow) (6.1.0)
Requirement already satisfied: yarl<2.0,>=1.0 in ./venv/lib/python3.8/site-packages (from aiohttp--apache-airflow-providers-http--apache-airflow) (1.11.1)
Requirement already satisfied: async-timeout<5.0,>=4.0 in ./venv/lib/python3.8/site-packages (from aiohttp--apache-airflow-providers-http--apache-airflow) (4.0.3)
Requirement already satisfied: exceptiongroup<=1.0.2 in ./venv/lib/python3.8/site-packages (from anyio--httpx--apache-airflow) (1.2.2)
(venv) [opc@inovoice-rr-airflow airflow]$ █
```

## 6: Inicialização do banco:

```
(venv) [opc@inovoice-rr-airflow airflow]$ airflow db init

[INFO] [airflow.models.dag] Creating ORM DAG for example_time_delta_sensor_async
[INFO] [airflow.models.dag] Setting next_dagrun for example_bash_operator to 2024-09-19 00:00:00+00:00
[INFO] [airflow.models.dag] Setting next_dagrun for example_branch_datetime_operator_2 to 2024-09-19 00:00:00+00:00
[INFO] [airflow.models.dag] Setting next_dagrun for example_branch_dop_operator_v3 to 2024-09-20T22:48:00+00:00
[INFO] [airflow.models.dag] Setting next_dagrun for example_branch_labels to 2024-09-19 00:00:00+00:00
[INFO] [airflow.models.dag] Setting next_dagrun for example_branch_operator to 2024-09-19 00:00:00+00:00
[INFO] [airflow.models.dag] Setting next_dagrun for example_complex to None
[INFO] [airflow.models.dag] Setting next_dagrun for example_dag_decorator to None
[INFO] [airflow.models.dag] Setting next_dagrun for example_external_task_marker_child to None
[INFO] [airflow.models.dag] Setting next_dagrun for example_external_task_marker_parent to None
[INFO] [airflow.models.dag] Setting next_dagrun for example_nested_branch_dag to 2024-09-19 00:00:00+00:00
[INFO] [airflow.models.dag] Setting next_dagrun for example_passing_params_via_test_command to 2024-09-20T22:48:00+00:00
[INFO] [airflow.models.dag] Setting next_dagrun for example_python_operator to None
[INFO] [airflow.models.dag] Setting next_dagrun for example_short_circuit_operator to 2024-09-19T22:49:44.976375+00:00
[INFO] [airflow.models.dag] Setting next_dagrun for example_skip_dag to 2024-09-19T22:49:44.976638+00:00
[INFO] [airflow.models.dag] Setting next_dagrun for example_sla_dag to 2024-09-20T22:46:00+00:00
[INFO] [airflow.models.dag] Setting next_dagrun for example_subdag_operator to 2024-09-18T00:00:00+00:00
[INFO] [airflow.models.dag] Setting next_dagrun for example_subdag_operator.section-1 to None
[INFO] [airflow.models.dag] Setting next_dagrun for example_subdag_operator.section-2 to None
[INFO] [airflow.models.dag] Setting next_dagrun for example_task_group to 2024-09-19T22:49:44.980027+00:00
[INFO] [airflow.models.dag] Setting next_dagrun for example_task_group_decorator to 2024-09-19T22:49:44.980350+00:00
[INFO] [airflow.models.dag] Setting next_dagrun for example_time_delta_sensor_async to None
[INFO] [airflow.models.dag] Setting next_dagrun for example_trigger_controller_dag to 2021-01-01T00:00:00+00:00
[INFO] [airflow.models.dag] Setting next_dagrun for example_trigger_target_dag to None
[INFO] [airflow.models.dag] Setting next_dagrun for example_weekday_branch_operator to 2024-09-19 00:00:00+00:00
[INFO] [airflow.models.dag] Setting next_dagrun for example_xcom to 2021-01-01T00:00:00+00:00
[INFO] [airflow.models.dag] Setting next_dagrun for example_xcom_args to None
[INFO] [airflow.models.dag] Setting next_dagrun for example_xcom_args_with_operators to None
[INFO] [airflow.models.dag] Setting next_dagrun for latest_only to 2024-09-20T18:49:44.983035+00:00
[INFO] [airflow.models.dag] Setting next_dagrun for latest_only_with_trigger to 2024-09-20T18:49:44.983303+00:00
[INFO] [airflow.models.dag] Setting next_dagrun for tutorial to 2024-09-19T22:49:44.983513+00:00
[INFO] [airflow.models.dag] Setting next_dagrun for tutorial_elt_dag to None
[INFO] [airflow.models.dag] Setting next_dagrun for tutorial_taskflow_api_elt to None
[INFO] [airflow.models.dag] Sync 2 DAGs
[INFO] [airflow.models.dag] Setting next_dagrun for example_subdag_operator.section-1 to None
[INFO] [airflow.models.dag] Setting next_dagrun for example_subdag_operator.section-2 to None
Initialization done
(venv) [opc@inovoice-rr-airflow airflow]$
```

## 7: Criação do usuário admin:

```
(venv) [opc@inovoice-rr-airflow airflow]$ airflow users create \
> --username admin \
> --firstname fiap \
> --lastname inovoice \
> --role Admin \
> --email fiama.santos84@gmail.com \
> --password Fiap24
/home/opc/airflow/venv/lib64/python3.8/site-packages/airflow/configuration.py:276: DeprecationWarning: distutils Version classes are deprecated. Use packaging.version instead.
  if StrictVersion(sqlite3.sqlite_version) < StrictVersion(min_sqlite_version):
[2024-09-20 22:53:32,805] {manager.py:779} WARNING - No user yet created, use flask fab command to do it.
[2024-09-20 22:53:32,934] {manager.py:496} INFO - Created Permission View: menu access on List Users
[2024-09-20 22:53:32,940] {manager.py:558} INFO - Added Permission menu access on List Users to role Admin
[2024-09-20 22:53:32,953] {manager.py:496} INFO - Created Permission View: menu access on Security
[2024-09-20 22:53:32,960] {manager.py:558} INFO - Added Permission menu access on Security to role Admin
[2024-09-20 22:53:32,989] {manager.py:496} INFO - Created Permission View: menu access on List Roles
[2024-09-20 22:53:32,996] {manager.py:558} INFO - Added Permission menu access on List Roles to role Admin
[2024-09-20 22:53:33,013] {manager.py:496} INFO - Created Permission View: can read on User Stats Chart
[2024-09-20 22:53:33,020] {manager.py:558} INFO - Added Permission can read on User Stats Chart to role Admin
[2024-09-20 22:53:33,034] {manager.py:496} INFO - Created Permission View: menu access on User's Statistics
[2024-09-20 22:53:33,040] {manager.py:558} INFO - Added Permission menu access on User's Statistics to role Admin
[2024-09-20 22:53:33,070] {manager.py:496} INFO - Created Permission View: menu access on Base Permissions
[2024-09-20 22:53:33,077] {manager.py:558} INFO - Added Permission menu access on Base Permissions to role Admin
[2024-09-20 22:53:33,104] {manager.py:496} INFO - Created Permission View: can read on View Menus
[2024-09-20 22:53:33,110] {manager.py:558} INFO - Added Permission can read on View Menus to role Admin
[2024-09-20 22:53:33,124] {manager.py:496} INFO - Created Permission View: menu access on Views/Menus
[2024-09-20 22:53:33,130] {manager.py:558} INFO - Added Permission menu access on Views/Menus to role Admin
```

Inicialização do scheduler que cuida da monitoração das DAGs (Directed Acyclic Graphs) e executa as tarefas agendadas no fluxo de trabalho.

DAGs é o termo em inglês que divide o trabalho em uma ou mais etapas em tarefas (tasks), todas essas tarefas juntas formam uma DAG. Nossa solução terá uma DAG que pega todas as etapas do fluxo de dados do storage até o carregamento dos dados no Power BI.

```
(venv) [opc@inovoice-rr-airflow airflow]$ airflow scheduler
/home/opc/airflow/venv/lib64/python3.8/site-packages/airflow/configuration.py:276: DeprecationWarning: distutils Version classes are deprecated. Use packaging.version instead.
  if StrictVersion(sqlite3.sqlite_version) < StrictVersion(min_sqlite_version):
____ /| \____ /____ /____ /____ /____ | /| /____
____ /| \____ /____ /____ /____ /____ | /| /____
____ /| \____ /____ /____ /____ /____ | /| /____
____ /| \____ /____ /____ /____ /____ | /| /____
____ /| \____ /____ /____ /____ /____ | /| /____
[2024-09-20 22:54:24,795] {scheduler_job.py:694} INFO - Starting the scheduler
[2024-09-20 22:54:24,795] {scheduler_job.py:699} INFO - Processing each file at most -1 times
[2024-09-20 22:54:24 +0000] [107417] [INFO] Starting unicorn 23.0.0
[2024-09-20 22:54:24 +0000] [107417] [INFO] Listening at: http://0.0.0.0:8793 (107417)
[2024-09-20 22:54:24 +0000] [107417] [INFO] Using worker: sync
```

Obs.: Antes foi necessário liberar as portas do firewall na 8080 tanto na máquina via OCI quanto via prompt:

Ingress Rules		Egress Rules (1)							
		Add Ingress Rules		Edit	Remove				
	Stateless ▾	Source	IP Protocol	Source Port Range	Destination Port Range	Type and Code	Allows	Description	⋮
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	22			TCP traffic for ports: 22 SSH Remote Login Protocol	⋮
<input type="checkbox"/>	No	0.0.0.0/0	ICMP			3, 4		ICMP traffic for: 3, 4 Destination Unreachable: Fragmentation Needed and Don't Fragment was Set	⋮
<input type="checkbox"/>	No	10.0.0.0/16	ICMP			3		ICMP traffic for: 3 Destination Unreachable	⋮
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	3306			TCP traffic for ports: 3306	⋮
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	1521			TCP traffic for ports: 1521 Oracle Port	⋮
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	80			TCP traffic for ports: 80	⋮
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	3000			TCP traffic for ports: 3000	⋮
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	8080			TCP traffic for ports: 8080 Airflow	⋮
0 selected									Showing 8 items < 1 of 1 >

```
[opc@inovoice-rr-airflow ~]$ sudo su
[root@inovoice-rr-airflow opc]# sudo firewall-cmd --state
running
[root@inovoice-rr-airflow opc]# sudo firewall-cmd --zone=public --add-port=8080/tcp --permanent
success
[root@inovoice-rr-airflow opc]# sudo firewall-cmd --reload
success
[root@inovoice-rr-airflow opc]# sudo firewall-cmd --list-ports
8080/tcp
```

## 7: Liberando a url para login do Airflow:

## 8: Login



# Airflow

64.181.164.115:8080/login/?next=http%3A%2F%2F64.181.164.115%3A8080%2Fhome

23:24 UTC 

### Sign In

Enter your login and password below:

**Username:**

**Password:**

**Sign In**

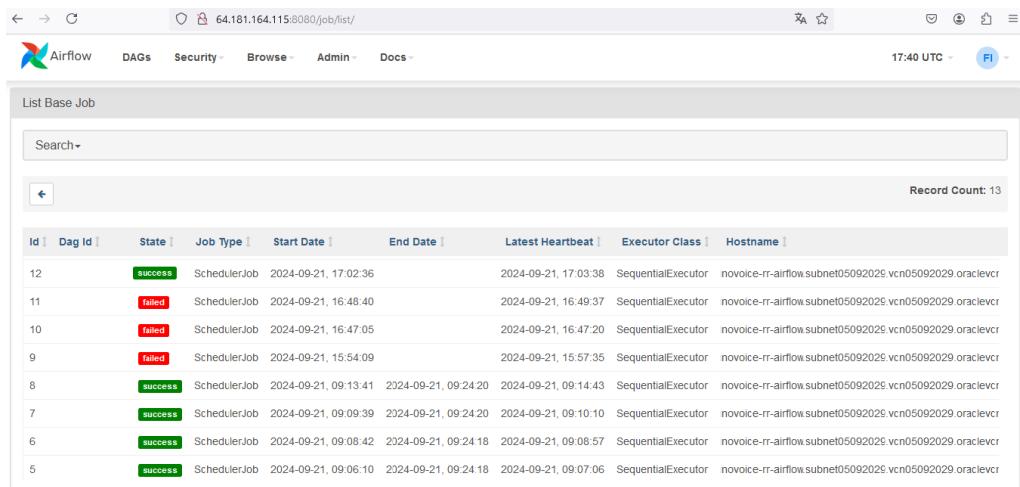
## 9: Edição do arquivo py, para aparecer na DAG da interface gráfica do Airflow:

```
[opc@inovoice-rr-airflow ~]$ nano airflow/dags/inovoice.py  
[opc@inovoice-rr-airflow ~]$ ll airflow/dags/  
total 16  
-rw-rw-r--. 1 opc airflow 4600 Sep 21 08:43 inovoice.py
```

## 10: Exemplificação teste do arquivo .py para cadastramento da DAG no Airflow:

```
GNU nano 2.9.8                                         airflow/dags/  
  
from airflow import DAG  
from airflow.operators.python_operator import PythonOperator  
from airflow.utils.dates import days_ago  
import datetime  
import os  
import audacity  
import sentiment_analysis_library as sentiment # Biblioteca de análise de sentimentos  
import mysql.connector  
import power_bi_integration  
  
# Função para processar áudio com Audacity  
def process_audio(**kwargs):  
    audio_file = kwargs['audio_file']  
    # Simulação de processamento de áudio no Audacity (substitua pela lógica real)  
    processed_audio = audacity.process(audio_file)  
    return processed_audio  
  
# Função para carregar dados no Data Lake (exemplo fictício usando MongoDB)  
def load_to_data_lake(**kwargs):  
    processed_audio = kwargs['ti'].xcom_pull(task_ids='process_audio')  
    # Exemplo de armazenamento de dados no MongoDB (substitua pela sua integração)  
    import pymongo  
    client = pymongo.MongoClient("mongodb://localhost:27017/")  
    db = client["data_lake"]  
    collection = db["audio_files"]  
    collection.insert_one({"processed_audio": processed_audio, "timestamp": datetime.datetime.now()})  
    return "Audio file loaded into data lake."  
  
# Função para realizar a análise de sentimentos  
def sentiment_analysis(**kwargs):  
    audio_data = kwargs['ti'].xcom_pull(task_ids='load_to_data_lake')  
    sentiment_result = sentiment.analyze(audio_data)  
    return sentiment_result  
  
# Função para classificar os dados (adicionar uma camada de machine learning)  
def classify_sentiments(**kwargs):  
    sentiment_data = kwargs['ti'].xcom_pull(task_ids='sentiment_analysis')  
    # Classificação com machine learning  
    classification_result = sentiment.classify(sentiment_data)  
    return classification_result  
  
# Função para armazenar no Data Warehouse  
def store_in_warehouse(**kwargs):  
    classification_result = kwargs['ti'].xcom_pull(task_ids='classify_sentiments')  
    db_conn = mysql.connector.connect(  
        host="your_mysql_host",  
        user="your_mysql_user",  
        password="your_mysql_password",  
        database="your_mysql_database")  
    cursor = db_conn.cursor()  
    cursor.execute("INSERT INTO your_table (classification_result) VALUES (%s)", (classification_result,))  
    db_conn.commit()  
    cursor.close()  
    db_conn.close()
```

## 11: Confirmação do fluxo cadastrado no Airflow para orquestração:



ID	Dag ID	State	Job Type	Start Date	End Date	Latest Heartbeat	Executor Class	Hostname
12		success	SchedulerJob	2024-09-21, 17:02:36		2024-09-21, 17:03:38	SequentialExecutor	inovoice-rr-airflow subnet05092029 vcn05092029 oraclevcr
11		failed	SchedulerJob	2024-09-21, 16:48:40		2024-09-21, 16:49:37	SequentialExecutor	inovoice-rr-airflow subnet05092029 vcn05092029 oraclevcr
10		failed	SchedulerJob	2024-09-21, 16:47:05		2024-09-21, 16:47:20	SequentialExecutor	inovoice-rr-airflow subnet05092029 vcn05092029 oraclevcr
9		failed	SchedulerJob	2024-09-21, 15:54:09		2024-09-21, 15:57:35	SequentialExecutor	inovoice-rr-airflow subnet05092029 vcn05092029 oraclevcr
8		success	SchedulerJob	2024-09-21, 09:13:41	2024-09-21, 09:24:20	2024-09-21, 09:14:43	SequentialExecutor	inovoice-rr-airflow subnet05092029 vcn05092029 oraclevcr
7		success	SchedulerJob	2024-09-21, 09:09:39	2024-09-21, 09:24:20	2024-09-21, 09:10:10	SequentialExecutor	inovoice-rr-airflow subnet05092029 vcn05092029 oraclevcr
6		success	SchedulerJob	2024-09-21, 09:08:42	2024-09-21, 09:24:18	2024-09-21, 09:08:57	SequentialExecutor	inovoice-rr-airflow subnet05092029 vcn05092029 oraclevcr
5		success	SchedulerJob	2024-09-21, 09:06:10	2024-09-21, 09:24:18	2024-09-21, 09:07:06	SequentialExecutor	inovoice-rr-airflow subnet05092029 vcn05092029 oraclevcr

Com isso, a orquestração precisa do Airflow do InoVoice demonstra ser uma ferramenta crucial para a otimização dos processos e a redução de custos para o nosso cliente TOTVS. Acreditamos que ao utilizar as instâncias do OCI no módulo Free Tier, conseguimos maximizar a utilização dos recursos computacionais disponíveis, minimizando gastos com infraestrutura. A flexibilidade e escalabilidade do Airflow permitem-nos adaptar os processos de forma dinâmica para as demandas, garantindo maior eficiência e agilidade na entrega de soluções.

### 3. WEB APP USANDO O TERRAFORM NO OCI

Para auxiliar as áreas técnicas da TOTVS na análise do NPS (Net Promoter Score) dos áudios de atendimento na plataforma InoVoice, foi desenvolvida uma solução de webapp na Oracle Cloud Infrastructure (OCI). Essa solução permite o envio de áudios, que são armazenados na nuvem e integrados a um banco de dados transacional, possibilitando a análise de sentimentos dos dados. A infraestrutura do webapp foi provisionada por meio do Terraform, garantindo a automação e escalabilidade dos recursos no ambiente OCI.

Confira abaixo o passo a passo do provisionamento:

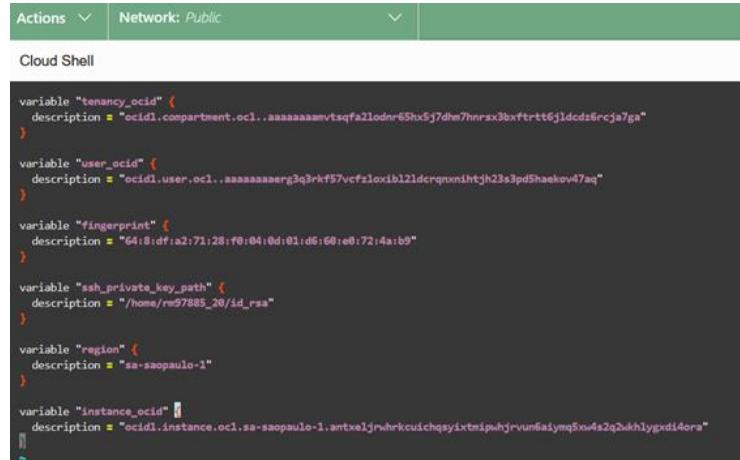
1: Acessando o cloud shell pelo OCI:

```
rm97885_20@cloudshell:~ (sa-saopaulo-1)$ pwd
/home/rm97885_20/fiap/oci/webapp-terraform
rm97885_20@cloudshell:~ (sa-saopaulo-1)$ ll
total 0
rm97885_20@cloudshell:~ (sa-saopaulo-1)$ cd
rm97885_20@cloudshell:~ (sa-saopaulo-1)$ ll
total 18656
drwxr-xr-x 2 rm97885_20 oci 23 Sep 6 22:26 bin
drwxr-xr-x 3 rm97885_20 oci 17 Sep 5 20:22 lib
-rw-r--r-- 1 rm97885_20 oci 495 Sep 5 21:56 id_rsa
-rw-r--r-- 1 rm97885_20 oci 495 Sep 5 21:56 id_rsa.pub
drwxr-xr-x 2 rm97885_20 oci 4096 Sep 7 02:17 inoice
drwxr-xr-x 3 rm97885_20 oci 25 Sep 7 01:56 my_project
-rw-r--r-- 1 rm97885_20 oci 1704 Sep 7 02:01 oc1_api_key.pem
-rw-r--r-- 1 rm97885_20 oci 19075160 Sep 6 22:23 terraform_3.0.6_linux_amd64.zip
drwxr-xr-x 2 rm97885_20 oci 343 Sep 5 21:18 variables.tf
drwxr-xr-x 2 rm97885_20 oci 4096 Sep 7 02:39 web_app
rm97885_20@cloudshell:~ (sa-saopaulo-1)$ cat id_rsa
-----BEGIN PRIVATE KEY-----
MIIEQzIBADANBgkqhkiG9W0BAQEFAASCBwggS5AgEAAoIBACQrcgutUNFRhLS
OZMjAgBq48CjYlF+eQn1DLnVvYD4w/rYK80L16C0MnGYz13jVNcgeYHmNdrfgatd
gfJpX0B6rpv0dTAnQWdQd4e37Sy49M7xvAhdQ+sBzQfL27R989ocFssC
P3v8lHfYv+a0u505TQJUgocd5t5h9p9y6nOvXobw/ujqNNK002/JOG1381
LNUfAz/Xu6tch26253JmVfhd+e7t93d9yGSt31XeXgQvLdavUsQ/d
OpT6R0aX1Zwhh6ygtA003dmDfr79c6tRwvpgpEcKt90k0cs7rEB6+23
2kYEc9Rqg#BAAECggEAlSopeTC0NlrhAx0Rugh011G17W9P+Y0W0icj0kCh
Yzaff1Jdu4c+nH6807675dfEPJuua1susu141R00fyWeB41BvLM42573j0z/+o
w8R+n1bs92+0400G1tpx113+k+j1V02M1001n-L72vvnQxf/7vPg2n9
Nw+fluryINRk8UNr7oxSc2021TSQ266mAv00+Msq=q981Vp33chfhwXq
Mh1c7xv9vysu1sfhdc00RKAHA113HGL+tQg+e8h19/GtVY0636p51k1
L091c0d+e7w0RfGt6u-MGJcfhB+1Hv+u2p0005ccvW4t1UJd
-----END PRIVATE KEY-----
```

Welcome to Oracle Cloud Shell.  
Upgrade Notification: Cloud Shell will be upgrading to Oracle Linux 8.  
Your Cloud Shell machine comes with 5GB of storage for your home directory. Your Cloud Shell (machine and home directory) are located in: Brazil East (Sao Paulo). You are using Cloud Shell in tenancy rm97885\_2024 as OCI local user rm97885\_2024@fiap.com.br  
Type 'help' for more info.  
rm97885\_20@cloudshell:~ (sa-saopaulo-1)\$ mkdir -p fiap/oci/webapp-terraform  
rm97885\_20@cloudshell:~ (sa-saopaulo-1)\$ pwd  
/home/rm97885\_20  
rm97885\_20@cloudshell:~ (sa-saopaulo-1)\$ vi variables.tf

## 2: Adicionando as informações no arquivo variables.tf :

Nessa etapa realizamos a definição de parâmetros que podem ser reutilizados ou alterados dinamicamente. Tornando a configuração mais flexível e escalável.



```
variable "tenancy_ocid" {
  description = "ocid1.compartment.oc1..aaaaaaaaamvtsqfa2lodnr65hx5j7dhw7hnrx3bx*trtt6j1dcds6rcja7ga"
}

variable "user_ocid" {
  description = "ocid1.user.oc1..aaaaaaaaaerg3q3rkf57vcfsloxibl2ldeqrqnxxnhtjh23s3pd5haekov47aq"
}

variable "fingerprint" {
  description = "64:8:idf:ia2:71:28:f9:04:0d:01:d6:60:e0:72:4a:b9"
}

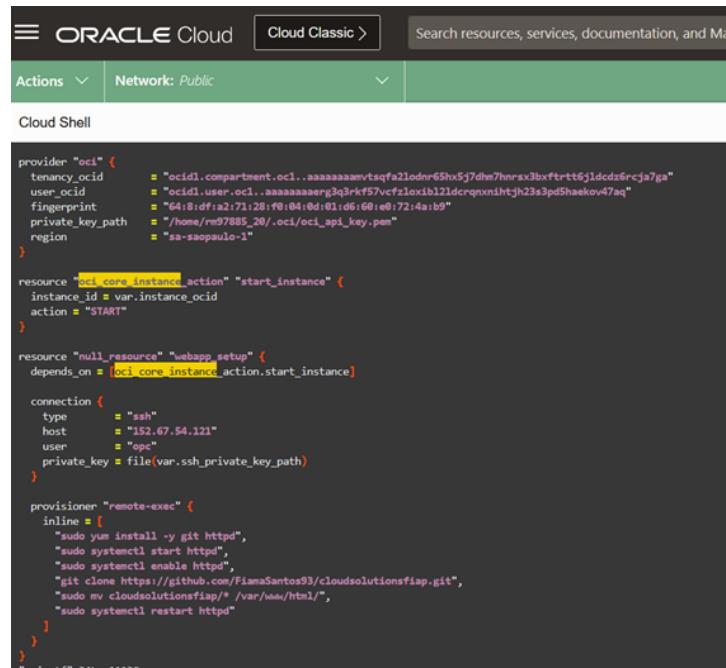
variable "ssh_private_key_path" {
  description = "/home/rm97885_20/id_rsa"
}

variable "region" {
  description = "sa-saopaulo-1"
}

variable "instancia_ocid" {
  description = "ocid1.instance.oc1.sa-saopaulo-1.amtxeljrvhrkcuichqsyixtmipwhjrvun6aiymq5xw4s2q2kh1ygd14ora"
}
```

## 3: Adicionando as informações na main.tf:

Nessa fase provisionamos a infraestrutura principal com todas as informações do OCI, GitHub e informações da instância chamando as variáveis. Aqui é onde definimos os recursos e como eles se inter-relacionam.



```
provider "oci" {
  tenancy_ocid      = "ocid1.compartment.oc1..aaaaaaaaamvtsqfa2lodnr65hx5j7dhw7hnrx3bx*trtt6j1dcds6rcja7ga"
  user_ocid         = "ocid1.user.oc1..aaaaaaaaaerg3q3rkf57vcfsloxibl2ldeqrqnxxnhtjh23s3pd5haekov47aq"
  fingerprint       = "64:8:idf:ia2:71:28:f9:04:0d:01:d6:60:e0:72:4a:b9"
  private_key_path  = "/home/rm97885_20/.oci/oci_api_key.pem"
  region            = "sa-saopaulo-1"
}

resource "oci_core_instance_action" "start_instance" {
  instance_id = var.instance_ocid
  action      = "START"
}

resource "null_resource" "webapp_setup" {
  depends_on = [oci_core_instance_action.start_instance]
}

connection {
  type     = "ssh"
  host    = "192.67.54.121"
  user    = "opc"
  private_key = file(var.ssh_private_key_path)
}

provisioner "remote-exec" {
  inline = [
    "sudo yum install -y git httpd",
    "sudo systemctl start httpd",
    "sudo systemctl enable httpd",
    "git clone https://github.com/FiammaSantos93/cloudsolutionsfiap.git",
    "sudo mv cloudsolutionsfiap/* /var/www/html/",
    "sudo systemctl restart httpd"
  ]
}
```

#### 4: Adicionando as informações do output.tf :

Neste arquivo, exportamos os dados importantes da infraestrutura provisionada, que podem ser usados em outras partes da configuração.

```
Actions Network: Public
Cloud Shell

```hcl
output "instance_public_ip" {
    value = "152.67.54.121"
}

output "webapp_url" {
    value = "http://152.67.54.121"
}
```

```

#### 5: Confirmação dos arquivos Terraform:

```
... (Output truncated)
```

6: Comando terraform init que prepara o Terraform para executar os comandos de provisionamento dos recursos.

```
Actions Network: Public
Cloud Shell

running this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
rw97885_20@cloudshell:web_app (sa-saopaulo-1)$ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/null from the dependency lock file
- Reusing previous version of hashicorp/oci from the dependency lock file
- Using previously-installed hashicorp/null v3.2.2
- Using previously-installed hashicorp/oci v6.8.0
Terraform has been successfully initialized!

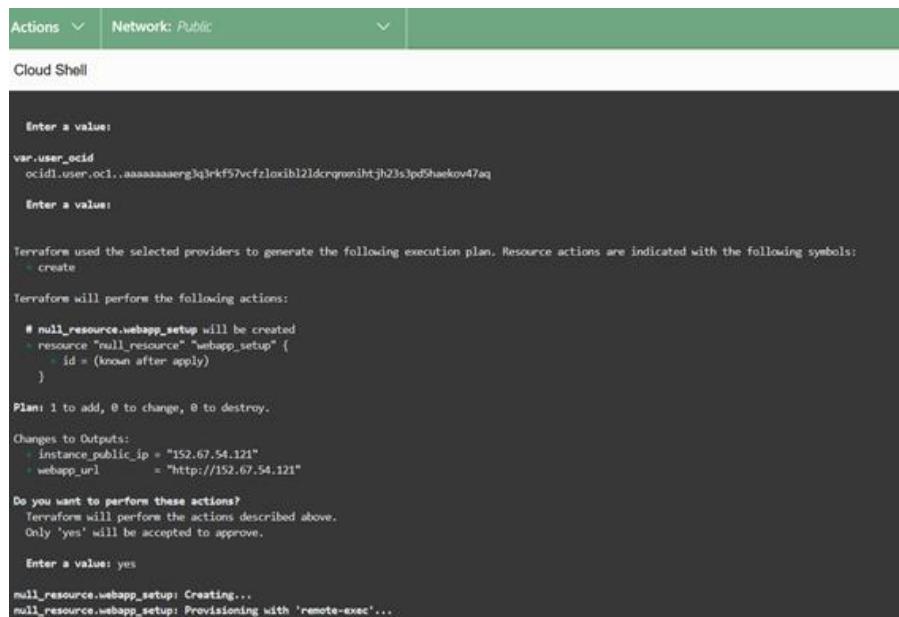
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
run this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
rw97885_20@cloudshell:web_app (sa-saopaulo-1)$
```

7: Comando terraform.plan que fará a análise dos arquivos do terraform .tf e mostra quais mudanças serão aplicadas à infraestrutura, como a criação, modificação (quando tem) ou exclusão de recursos. No caso do InoVoice é apenas criação.

```
rm97885_20@cloudshell:web_app (sa-saopaulo-1)$
rm97885_20@cloudshell:web_app (sa-saopaulo-1)$
rm97885_20@cloudshell:web_app (sa-saopaulo-1)$
rm97885_20@cloudshell:web_app (sa-saopaulo-1)$
rm97885_20@cloudshell:web_app (sa-saopaulo-1)$ terraform plan
var.fingerprint
64:8:df:a2:71:28:f0:04:0d:01:d6:60:e0:72:4a:b9

Enter a value:
```



The screenshot shows the Google Cloud Shell interface. At the top, there are tabs for 'Actions' and 'Network: Public'. Below that is a section labeled 'Cloud Shell' containing a terminal window. The terminal output is as follows:

```
Enter a value:

var.user_ocid
ocidi.user.oc1..aaaaaaaaerg3q3rkf57vcfzloxicbl2ldcrqnuinhtjh23s3pdShaeov47aq

Enter a value:

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# null_resource.webapp_setup will be created
+ resource "null_resource" "webapp_setup" {
  + id = (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.

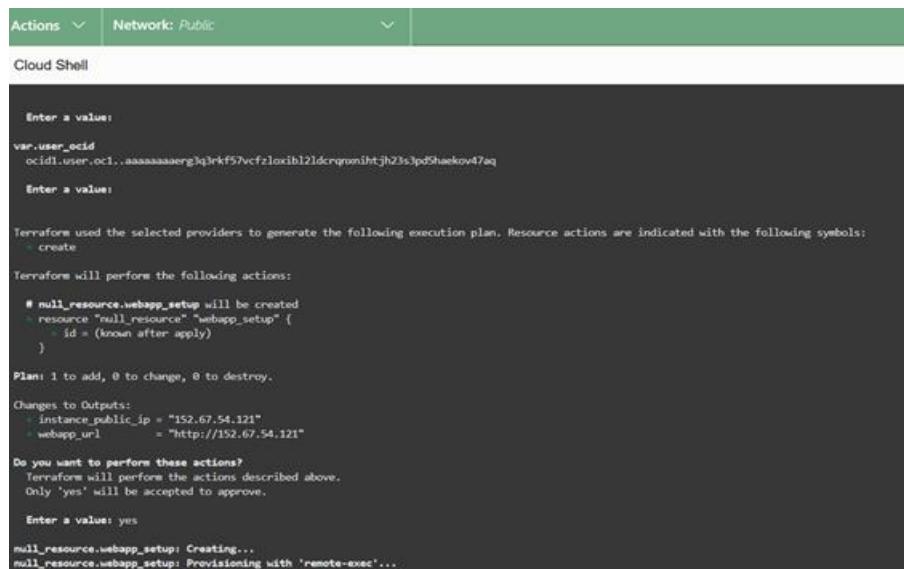
Changes to Outputs:
+ instance_public_ip = "152.67.54.121"
+ webapp_url       = "http://152.67.54.121"

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

null_resource.webapp_setup: Creating...
null_resource.webapp_setup: Provisioning with 'remote-exec'...
```

8: Precisamos dar enter para confirmar os recursos:



The screenshot shows the Google Cloud Shell interface again. The terminal output is identical to the previous one, but it ends with the user entering 'yes' to confirm the actions:

```
Enter a value:

var.user_ocid
ocidi.user.oc1..aaaaaaaaerg3q3rkf57vcfzloxicbl2ldcrqnuinhtjh23s3pdShaeov47aq

Enter a value:

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# null_resource.webapp_setup will be created
+ resource "null_resource" "webapp_setup" {
  + id = (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ instance_public_ip = "152.67.54.121"
+ webapp_url       = "http://152.67.54.121"

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

null_resource.webapp_setup: Creating...
null_resource.webapp_setup: Provisioning with 'remote-exec'...
```

9: Utilizamos o comando terraform apply que fará a aplicação das mudanças descritas no plano de execução gerado pelo comando terraform plan.

```
rw97885_20@cloudshell:web_app (sa-saopaulo-1)$ 
rw97885_20@cloudshell:web_app (sa-saopaulo-1)$ 
rw97885_20@cloudshell:web_app (sa-saopaulo-1)$ 
rw97885_20@cloudshell:web_app (sa-saopaulo-1)$ 
rw97885_20@cloudshell:web_app (sa-saopaulo-1)$ 
rw97885_20@cloudshell:web_app (sa-saopaulo-1)$ 
rw97885_20@cloudshell:web_app (sa-saopaulo-1)$ terraform apply
var.fingerprint
64:8:df:a2:71:28:f0:04:0d:01:d6:60:e0:72:4a:b9

Enter a value:

var.instance_ocid
ocid1.instance.oc1.sa-saopaulo-1.antxeljrwhrkcuichqsyixtmipwhjrvun6aiymq5ox4s2q2whlygxd4ora

Enter a value:

var.region
sa-saopaulo-1

Enter a value:

var.ssh_private_key_path
/hose/rw97885_20/id_rsa
```

```
☰ ORACLE Cloud Cloud Classic > Search resources, services, documentation, and Market

Actions Network: Public

Cloud Shell

null_resource.webapp_setup (remote-exec): Receiving objects: 98% (72/73)
null_resource.webapp_setup (remote-exec): Receiving objects: 100% (73/73)
null_resource.webapp_setup (remote-exec): Receiving objects: 100% (73/73), 20.00 KiB | 10.00 MiB/s, done.
null_resource.webapp_setup (remote-exec): Resolving deltas: 0% (0/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 4% (1/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 13% (3/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 18% (4/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 22% (5/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 27% (6/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 31% (7/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 36% (8/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 40% (9/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 45% (10/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 50% (11/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 54% (12/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 59% (13/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 63% (14/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 68% (15/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 72% (16/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 77% (17/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 81% (18/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 86% (19/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 90% (20/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 95% (21/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 100% (22/22)
null_resource.webapp_setup (remote-exec): Resolving deltas: 100% (22/22), done.
null_resource.webapp_setup: Creation complete after 13s [id=7530571807166206414]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:
instance_public_ip = "152.67.54.121"
webapp_url = "http://152.67.54.121"
rw97885_20@cloudshell:web_app (sa-saopaulo-1)$
```

Obs.: Foram realizadas as configurações de liberação firewall e diretórios do apache para liberação do app pelo <http://152.67.54.121>

10: Agora a áreas técnicas podem subir os arquivos:

The screenshot shows a web browser window with the URL 152.67.54.121. The main content is a blue header "Bem-vindo ao Inovoice!". Below it, a text instruction "Insira um áudio para analisar:" is displayed. Underneath this, there are three buttons: "Procurar...", "Nenhum arquivo selecionado.", and "Enviar Áudio".

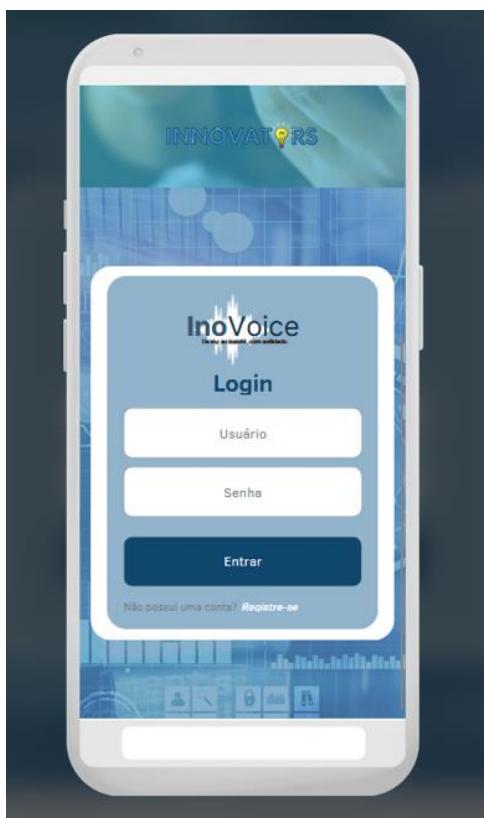
Graças à automação proporcionada pelo Terraform na Oracle Cloud Infrastructure (OCI), foi possível configurar e disponibilizar o webapp InoVoice, facilitando a análise de áudios de atendimento. Portanto, as áreas técnicas da TOTVS podem utilizar a plataforma para processar arquivos de áudio, armazená-los de forma segura na nuvem e realizar análises de sentimentos com eficiência. Esse ambiente integrado agiliza as operações e melhora a qualidade das avaliações de NPS, contribuindo para decisões mais assertivas e baseadas em dados.

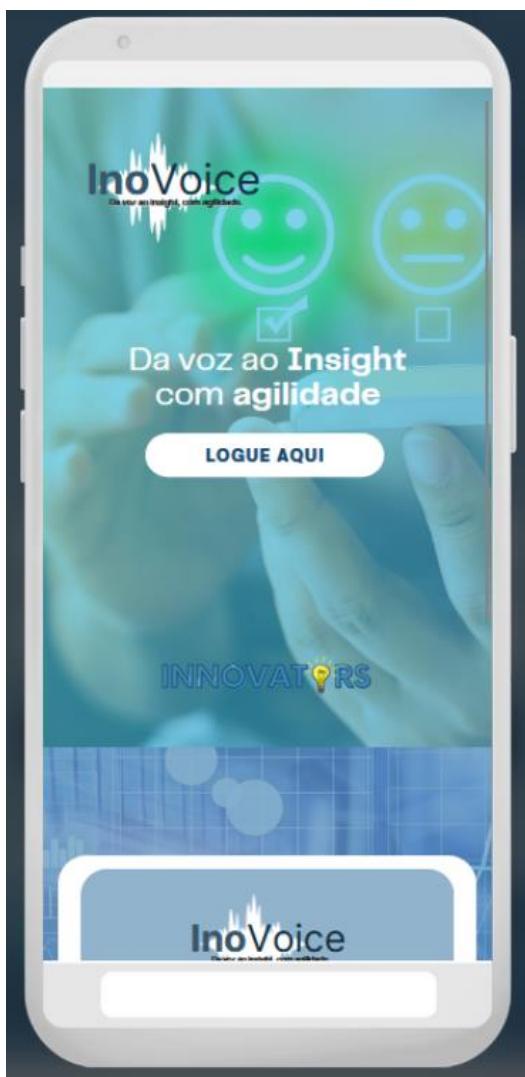
### 3.1 INTERFACE GRÁFICA LOGIN

Segue exemplo de tela de login em desktop para nossa aplicação



Segue imagens da tela de login em aplicativo mobile de nossa ferramentaa.

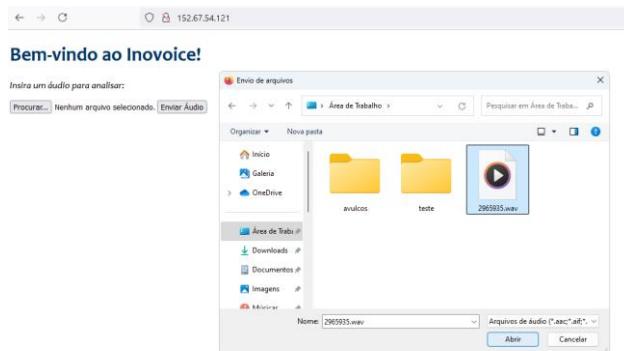




## 4. INTEGRAÇÃO DE DADOS WEB APP COM STORAGE NO GCP

Nessa fase, o InoVoice conta com uma integração de forma eficiente onde o usuário carrega o áudio, para que ele passe por um backend em node.js:

1: Áudio sendo carregado:



2: Áudio inserido:

Bem-vindo ao Inovoice!

Insira um áudio para analisar:

2965935.wav

3: Modelo de script que realizará a integração dos dados para armazenamento:

```
const express = require('express');
const { Storage } = require('@google-cloud/storage');
const multer = require('multer');

const app = express();

// Configuração do Google Cloud Storage
const storage = new Storage({
  keyFilename: 'gs://audiosinovoice.json',
});
const bucketName = 'audiosinovoice';

// Configuração do multer
const upload = multer({
  storage: multer.memoryStorage(),
});

// Rota para upload de áudio
app.post('/upload', upload.single('audio'), async (req, res) => {
  try {
    const audioFile = req.file;
    if (!audioFile) {
      return res.status(400).send('Nenhum arquivo enviado.');
    }

    const blob = storage.bucket(bucketName).file(audioFile.originalname);
    const blobStream = blob.createWriteStream({
      resumable: false,
    });
  
```

```
    blobStream.on('error', (err) => {
      console.error(err);
      return res.status(500).send('Erro ao armazenar o arquivo.');
    });

    blobStream.on('finish', () => {
      res.status(200).send(`Arquivo ${audioFile.originalname} armazenado com sucesso.`);
    });

    blobStream.end(audioFile.buffer);
  } catch (error) {
    console.error(error);
    res.status(500).send('Erro no upload.');
  }
});
```

## 5. ARMAZENAMENTO DOS ÁUDIOS

Esta etapa de importação dos áudios do storage via GCP para o SGBD (Sistema para banco de dados) MongoDB.

Para que os áudios recebam a carga diária, optamos pelo armazenamento no GCP, pois, por se tratar de arquivos de áudio e considerando que o free tier do OCI não comportaria 80 áudios diários devido à menor quantidade de GBs disponível, utilizaremos os recursos do free tier do GCP para esse armazenamento inicial.

<https://cloud.google.com/free/docs/free-cloud-features?hl=pt-br#storage>

| Cloud Storage | <ul style="list-style-type: none"><li>• 5 GB por mês de armazenamento regional (somente regiões dos EUA)</li><li>• 5.000 operações de Classe A por mês</li><li>• 50.000 operações de Classe B por mês</li><li>• 1 GB de saída de rede da América do Norte para todos os destinos regionais por mês (exceto China e Austrália)</li></ul> |
|---------------|---|
|               | <p>O Nível gratuito está disponível apenas <a href="#">nas regiões us-east1, us-west1 e us-central1</a>. Os cálculos de uso são combinados entre essas regiões.</p> <p><a href="#">Saiba mais</a></p>   |

Pensando num fluxo de 80 áudios por dia com uma rotina de exclusão do storage, o InoVoice comportaria muito bem, pois como o storage é apenas um recurso facilitador para armazenamento temporário dos dados, não será necessário deixar acumulando no recurso. Portanto, nesse caso também rodará uma rotina em python excluindo os dados do storage.

Pensando em uma média de 2.300Kb de áudios, o cliente TOTVS pode armazenar 2057 áudios no total com folga de 0.5Gb e a conta pensada nisso, foi com base nos áudios fornecidos no teste para o MVP.

Se formos calcular com sobra de armazenamento, a conta utilizada será:

### Conversão de 4.5 GB para KB:

- $4.5 \text{ GB} = 4.5 \times 1.024 \text{ MB} = 4608 \text{ MB}$
- $4608 \text{ MB} = 4608 \times 1.024 \text{ KB} = 4,718,592 \text{ KB}$

### Calculo dos áudios:

- Número de áudios = Total em KB / Tamanho de cada áudio
- Número de áudios =  $4.718,592 \text{ KB} / 2.300 \text{ KB} = \mathbf{2.057}$  áudios

## 6. ARMAZENAMENTO NO BANCO TRANSACIONAL MONGODB

Configuração e instalação do GCP SDK

1: Atualização dos pacotes:

```
root@inovoice:/home/ubuntu# sudo apt update
Ign:1 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 InRelease
Hit:2 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 Release
Hit:3 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:5 http://sa-saopaulo-1-ad-1.clouds.archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Hit:6 http://sa-saopaulo-1-ad-1.clouds.archive.ubuntu.com/ubuntu focal-updates InRelease
Get:7 http://sa-saopaulo-1-ad-1.clouds.archive.ubuntu.com/ubuntu focal-backports InRelease [128 kB]
Fetched 393 kB in 2s (226 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
27 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@inovoice:/home/ubuntu#
```

2: Instalação do GCP SDK

```
root@inovoice:/home/ubuntu# sudo snap install google-cloud-sdk --classic
Fetch and check assertions for snap "core20" (2379)
```

Cada pacote foi instalado item por item como o exemplo abaixo:

```
Download snap "google-cloud-sdk" (487) from channel "stable" 97% 14.8MB/s 907ms
```

3: Confirmação da instalação:

```
root@inovoice:/home/ubuntu# sudo snap install google-cloud-sdk --classic
google-cloud-sdk 493.0.0 from Cloud SDK (google-cloud-sdk✓) installed
```

4: Inicialização do SDK :

```
root@inovoice:/home/ubuntu# gcloud init
Welcome! This command will take you through the configuration of gcloud.

Your current configuration has been set to: [default]

You can skip diagnostics next time by using the following flag:
  gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...:■
```

5: Autenticação da conta:

```
root@inovoice:/home/ubuntu# gcloud auth login
Go to the following link in your browser, and complete the sign-in prompts:
https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com&redirect_uri=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-identity&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fppengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fscript+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.readonly&state=t0idcQgs2IRpzb5oyadFoydzxp1fP&prompt=consent&token_usage=remote
Access_type=offline&code_challenge=a2l03dawfdqntwRce_BdHtaLax89q7c-NKsoCDeW6Cdc_code_challenge_method=S256
Once finished, enter the verification code provided in your browser: 4/QAQLEd8wxdUjPBFXLwdefA809tLo6vq5xfib_FGqRApztGD33YxkWDLZxMe2WPLn0A3wkQ
```

## 5: Atrelando com o ID do projeto no GCP:

```
Your current project is [None]. You can change this setting by running:  
$ gcloud config set project PROJECT_ID  
root@inovoice:/home/ubuntu# gcloud config set project project-259863  
Updated property [core/project].  
root@inovoice:/home/ubuntu#
```

## 6: Criação da pasta para inserção dos áudios na pasta temporária:

```
root@inovoice:/home/ubuntu# mkdir -p ~/audios/
```

## 7: Criação da pasta e copiando os áudios:

```
root@inovoice:/home/ubuntu# gsutil cp gs://audiosinovoice/*.wav ~/audios/  
Copying gs://audiosinovoice/2874774.wav...  
Copying gs://audiosinovoice/2874830.wav...  
Copying gs://audiosinovoice/2961972.wav...  
Copying gs://audiosinovoice/2962046.wav...  
- [4 files][ 21.3 MiB/ 21.3 MiB]  
==> NOTE: You are performing a sequence of gsutil operations that may  
run significantly faster if you instead use gsutil -m cp ... Please  
see the -m section under "gsutil help options" for further information  
about when gsutil -m can be advantageous.  
  
Copying gs://audiosinovoice/2962074.wav...  
■ [4 files][ 21.3 MiB/ 31.4 MiB]
```

## 8: Para subir os arquivos de áudio no MongoDB, será preciso instalar o python:

```
inovoice> exit  
ubuntu@inovoice-rt:~$ python --version  
Command 'python' not found, did you mean:  
  command 'python3' from deb python3  
  command 'python' from deb python-is-python3  
  
ubuntu@inovoice-rt:~$ sudo apt update  
Ign:1 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 InRelease  
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease  
Get:3 http://sa-saopaulo-1-ad-1.clouds.archive.ubuntu.com/ubuntu focal InRelease [265 kB]  
Hit:4 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 Release  
Get:5 http://sa-saopaulo-1-ad-1.clouds.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]  
Hit:7 http://sa-saopaulo-1-ad-1.clouds.archive.ubuntu.com/ubuntu focal-backports InRelease  
Get:8 http://sa-saopaulo-1-ad-1.clouds.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [3598 kB]  
Get:9 http://sa-saopaulo-1-ad-1.clouds.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [553 kB]  
Get:10 http://sa-saopaulo-1-ad-1.clouds.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [17.7 kB]  
Get:11 http://sa-saopaulo-1-ad-1.clouds.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [3279 kB]  
Get:12 http://sa-saopaulo-1-ad-1.clouds.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [459 kB]  
Get:13 http://sa-saopaulo-1-ad-1.clouds.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1231 kB]  
Get:14 http://sa-saopaulo-1-ad-1.clouds.archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [295 kB]  
Get:15 http://sa-saopaulo-1-ad-1.clouds.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [28.3 kB]  
Fetched 9853 KB in 4s (2497 kB/s)  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
55 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

## 9: Também foi necessária a instalação do pymongo para chamarmos a rotina de inserção do áudio:

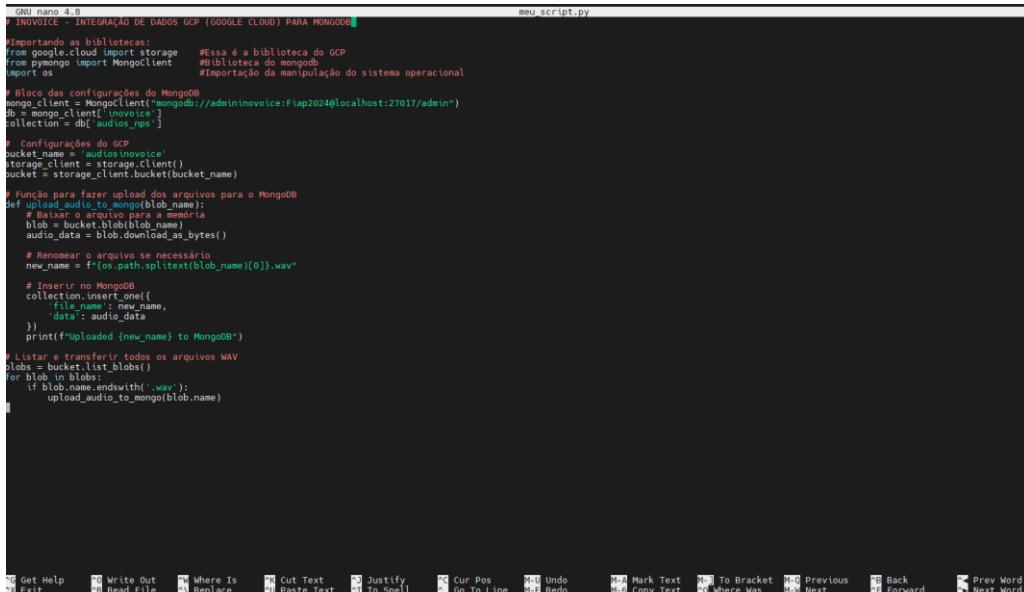
```
root@inovoice:/home/ubuntu# pip3 install pymongo  
Collecting pymongo  
  Downloading pymongo-4.9.1-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (926 kB)
    |██████████| 926 kB 7.0 MB/s  
Collecting dnspython<3.0.0,>=1.16.0  
  Downloading dnspython-2.6.1-py3-none-any.whl (307 kB)
    |██████████| 307 kB 93.9 MB/s  
Installing collected packages: dnspython, pymongo  
Successfully installed dnspython-2.6.1 pymongo-4.9.1  
root@inovoice:/home/ubuntu#
```

## 10: Instalação das bibliotecas do GCP no pymongo:

```
root@inovoice:/home/ubuntu# pip3 install google-cloud-storage pymongo
Collecting google-cloud-storage
  Downloading google-cloud-storage-2.18.2-py2.py3-none-any.whl (130 kB)
Requirement already satisfied: pymongo<3.0,!=3.0.0,!=3.0.1,>=2.8.0 from /usr/local/lib/python3.8/dist-packages (4.9.1)
  Downloading google-cloud-core-2.4.1-py2.py3-none-any.whl (29 kB)
Requirement already satisfied: requests<3.0.0dev,>=2.18.0 in /usr/lib/python3/dist-packages (from google-cloud-storage) (2.22.0)
Collecting google-auth<2.0dev,>=1.26.0
  Downloading google-auth-2.35.0-py2.py3-none-any.whl (208 kB)
Requirement already satisfied: google-auth-oauthlib<1.0,!=1.0.0,!=1.0.1,>=0.4.1 from /usr/local/lib/python3.8/dist-packages (from google-auth) (0.4.1)
  Downloading google-auth-oauthlib-0.4.1-py2.py3-none-any.whl (20 kB)
Collecting google-api-core<3.0.0dev,>=2.15.0
  Downloading google-api-core-2.20.0-py2.py3-none-any.whl (142 kB)
Requirement already satisfied: google-crc32c<2.0dev,>=1.0.0 from /usr/local/lib/python3.8/dist-packages (4.9.1)
  Downloading google-crc32c-1.5.0-cp38-cp38-manylinux_2_17_x86_64_manylinux2014_x86_64.whl (32 kB)
Collecting google-resumable-media<2.7.2-py2.py3-none-any.whl (81 kB)
  Downloading google-resumable-media-2.7.2-py2.py3-none-any.whl (81 kB)
Requirement already satisfied: dns<python>=3.0.0,>=1.16.0 in /usr/lib/python3/dist-packages (from pymongo) (2.6.1)
Collecting cachetools<3.0.0,>=2.0.0
  Downloading cachetools-3.5.3-py3-none-any.whl (9.5 kB)
Collecting rsa<5,>=3.1.4
  Downloading rsa-4.9-py3-none-any.whl (34 kB)
Collecting pyasn1-modules<>0.2.1
  Downloading pyasn1_modules-0.4.1-py3-none-any.whl (181 kB)
Collecting protobuf<=2.20.0,>=3.20.1,!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<=6.0.0.dev0,>=3.19.5
  Downloading protobuf-5.28.2-cp38-ab3-manylinux2014_x86_64.whl (316 kB)
Collecting proto-plus<2.0.0dev,>=1.22.3
  Downloading proto_plus-1.24.0-py3-none-any.whl (50 kB)
Collecting googleapis-common-protos<2.0.0dev,>=1.65.0,!=1.65.0-py3-none-any.whl (220 kB)
  Downloading googleapis_common_protos-1.65.0-py3-none-any.whl (220 kB)
Collecting pyasn1<>0.1.3
  Downloading pyasn1-0.6.1-py3-none-any.whl (83 kB)
Installing collected packages: protobuf, proto-plus, googleapis-common-protos, cachetools, pyasn1, rsa, pyasn1-modules, google-auth, google-api-core, google-cloud-core, google-crc32c, google-resumable-media, google-cloud-storage

```

## 11: Criação do arquivo python no diretório para execução, usando o comando: nano upload\_audios.py no nano.



```
#!/usr/bin/env python3
# INOVOICE - INTEGRAÇÃO DE DADOS GCP (GOOGLE CLOUD) PARA MONGODB
# Importando as bibliotecas
from google.cloud import storage #Essa é a biblioteca do GCP
from google import MongoClient #Biblioteca do mongoDB
import os

# Acesso das configurações do MongoDB
mongo_client = MongoClient("mongodb://admin:inovoice:Fiap2024@localhost:27017/admin")
db = mongo_client['inovoice']
collection = db['audios_npis']

# Configurações do GCP
bucket_name = 'audiosinovoice'
storage_client = storage.Client()
bucket = storage_client.bucket(bucket_name)

# Função para fazer upload dos arquivos para o MongoDB
def upload_audio_to_mongo(blob_name):
    # Baixar o arquivo para a memória
    blob = bucket.blob(blob_name)
    audio_data = blob.download_as_bytes()

    # Renomear o arquivo se necessário
    new_name = f'{os.path.splitext(blob_name)[0]}.wav'

    # Inserir no MongoDB
    collection.insert_one({
        'file_name': new_name,
        'data': audio_data
    })
    print(f"Uploaded {new_name} to MongoDB")

# Listar e transferir todos os arquivos WAV
blobs = bucket.list_blobs()
for blob in blobs:
    if blob.name.endswith('.wav'):
        upload_audio_to_mongo(blob.name)
```

## 12: Modelo do código da integração:

```
# INVOICE - INTEGRAÇÃO DE DADOS GCP (GOOGLE CLOUD) PARA MONGODB
|
#Importando as bibliotecas:
from google.cloud import storage      #Essa é a biblioteca do GCP
from pymongo import MongoClient        #Biblioteca do mongodb
import os                            #Importação da manipulação do sistema operacional

# Bloco das configurações do MongoDB
mongo_client = MongoClient("mongodb://admininovoice:Fiap2024@localhost:27017/admin")
db = mongo_client['inovoice']
collection = db['audios_nps']

# Configurações do GCP
bucket_name = 'audiosinovoice'
storage_client = storage.Client()
bucket = storage_client.bucket(bucket_name)

# Função para fazer upload dos arquivos para o MongoDB
def upload_audio_to_mongo(blob_name):
    # Baixar o arquivo para a memória
    blob = bucket.blob(blob_name)
    audio_data = blob.download_as_bytes()

    # Renomear o arquivo se necessário
    new_name = f"{os.path.splitext(blob_name)[0]}.wav"

    # Inserir no MongoDB
    collection.insert_one({
        'file_name': new_name,
        'data': audio_data
    })

    print(f"Uploaded {new_name} to MongoDB")

# Listar e transferir todos os arquivos WAV
blobs = bucket.list_blobs()
for blob in blobs:
    if blob.name.endswith('.wav'):
        upload_audio_to_mongo(blob.name)
```

## 7. INSTALAÇÃO DO MONGODB VIA LINUX

1: Realizando a instalação do MongoDB, no ubuntu para essa versão:

```
ubuntu@inovoice-rt:~$ wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add -
ubuntu@inovoice-rt:~$ echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list
Ign:1 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 InRelease
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:3 http://sa-saopaulo-1-ad-1.clouds.archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:4 http://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 Release [300 B]
Get:5 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0/bionic [1866 B]
Get:6 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0/multiverse arm64 Packages [70.2 kB]
Get:7 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0/multiverse amd64 Packages [73.9 kB]
Hit:8 http://sa-saopaulo-1-ad-1.clouds.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:9 http://repo.mongodb.org/apt/ubuntu focal-backports InRelease
Fetched 413 KB in 26 (257 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
47 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

```
ubuntu@inovoice-rt:~$ sudo apt update
Get:1 http://sa-saopaulo-1-ad-1.clouds.archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Ign:3 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 InRelease
Hit:4 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 Release
Get:6 http://sa-saopaulo-1-ad-1.clouds.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Get:7 http://sa-saopaulo-1-ad-1.clouds.archive.ubuntu.com/ubuntu focal-backports InRelease [128 kB]
Fetched 521 kB in 2s (297 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
47 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

2: Edição para incluir o vm.max\_map\_count=262144 no nano, pois na instalação ele apontou uns possíveis problemas devido a alocação de memória.

```
ubuntu@inovoice-rt:~$ sudo nano /etc/sysctl.conf
ubuntu@inovoice-rt:~$ mongosh
Current Mongosh Log ID: 66ee6844fc8e8771964032
Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.1
Using MongoDB: 6.0.17
Using Mongosh: 2.3.1
For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/
-----
The server generated these startup warnings when booting
2024-09-18T02:58:22.253+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2024-09-18T02:58:22.422+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2024-09-18T02:58:22.424+00:00: vm.max_map_count is too low
-----
```

3: Nessa etapa, estamos retornando com a execução do script criado anteriormente para subida dos dados:

```
root@inovoice:/home/ubuntu# python3 upload_audios.py
```

4: Criação do usuário:

```
root@inovoice:/home/ubuntu# mongosh -u admininovoice -p Fiap2024 --authenticationDatabase admin
Current Mongosh Log ID: 66ee6844fc8e8771964032
Connecting to: mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin&appName=mongosh+2.3.1
Using MongoDB: 6.0.17
Using Mongosh: 2.3.1
For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/
test> use invoice
switched to db invoice
invoice> db.createUser({
...   user: "admininovoice",
...   pwd: "#Fiap2024",
...   roles: [{ role: "readWrite", db: "invoice" }]
... })
{ ok: 1 }
invoice> db.getUsers()
{
  users: [
    {
      _id: 'invoice.admininovoice',
      userId: UUID('20ce1690-3d29-40f3-a556-5e63f7ee9f09'),
      user: "admininovoice",
      db: "invoice",
      roles: [ { role: 'readWrite', db: 'invoice' } ],
      mechanisms: [ 'SCRAM-SHA-1', 'SCRAM-SHA-256' ]
    }
  ],
  ok: 1
}
invoice> 
```

5: Inserção dos dados na tabela no MongoDB:

```
root@invoice-r-1:~# mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("c2acf592-856d-4938-b8bf-bb72f8032542") }
MongoDB server version: 6.0.17
WARNING: shell and server versions do not match
      - startup warnings
{"t":1,"s":1,"date":2024-09-18T02:58:22.422+00:00,"s":1,"c":"STORAGE","id":22297,"ctx":"initandlisten","msg":"Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/pronotes-filesystem","tags":["startupWarnings"]}
{"t":1,"s":1,"date":2024-09-18T02:58:22.422+00:00,"s":1,"c":"CONTROL","id":22120,"ctx":"initandlisten","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}
{"t":1,"s":1,"date":2024-09-18T02:58:22.422+00:00,"s":1,"c":"CONTROL","id":5123300,"ctx":"initandlisten","msg":"vm.max_map_count is too low","attr":{"currentValue":65530,"recommendedMinimum":10240,"maxConns":51200}, "tags":["startupWarnings"]}

> use invoice
switched to db invoice
> db.audios.chunks
> db.audios.files
> db['audios.files'].find().pretty()
{
  "_id" : "20274774",
  "filename" : "20274774.wav",
  "chunksize" : 261120,
  "length" : NumberLong(4806812),
  "uploadDate" : ISODate("2024-09-18T06:31:59.034Z")
}

{
  "_id" : "20265935",
  "filename" : "20265935.wav",
  "chunksize" : 261120,
  "length" : NumberLong(26276712),
  "uploadDate" : ISODate("2024-09-18T06:31:59.048Z")
}

{
  "_id" : "20267671",
  "filename" : "20267671.wav",
  "chunksize" : 261120,
  "length" : NumberLong(3798952),
  "uploadDate" : ISODate("2024-09-18T06:31:59.062Z")
}

{
  "_id" : "2964360",
  "filename" : "2964360.wav",
  "chunksize" : 261120,
  "length" : NumberLong(6461372),
  "uploadDate" : ISODate("2024-09-18T06:31:59.096Z")
}
```

A ideia do **InoVoice** sempre é trazer as melhores soluções não só de análise e insights, mas também de processamento, então pensando nesse custo com muita segurança, será utilizado uma rotina através do comando “crontab -e e o 0 0 \* \* \* find ~/audios/ -type f -name “\*.wav” -mtime +7 -exec rm {} \;” na qual todos os arquivos serão excluídos da máquina a cada 7 dias, uma vez que já teremos os mesmos no banco de dados MongoDB.

## 7.1 INTEGRAÇÃO DE DADOS DO MONGODB PARA O GOOGLE COLAB

Nessa fase terá uma integração onde o dado que está armazenado no banco de dados MongoDB será transferido para o Google Colab na análise de sentimentos. Fizemos o teste com 2 áudios, mas a ideia é utilizar a biblioteca scheduler que faz a programação do armazenamento dos dados.

Exemplo da integração do teste de integração via Google Colab:

```
[20] client = MongoClient("mongodb://admininovoice:Fiap2024@localhost:27017/inovoice")
    db = client.inovoice

[25] import pymongo
    import gridfs

def transfer_audios():
    # Conectar ao MongoDB
    mongo_client = pymongo.MongoClient("mongodb://admininovoice:Fiap2024@127.0.0.1:27017/inovoice")
    db = mongo_client["inovoice"]
    fs = gridfs.GridFS(db)

    # Recuperar os dois primeiros áudios armazenados no MongoDB
    audios = fs.find().limit(2)

    # Salvar os áudios em arquivos locais
    for audio in audios:
        filename = audio.filename
        with open(filename, 'wb') as f:
            # Ler o conteúdo do áudio como binário
            f.write(audio.read())

    print("Dois áudios foram transferidos para o local.")

# Executar a função de transferência agora
transfer_audios()
```

Modelo de script com a integração programada para todos os dias às 09h00:

```
def transfer_audios():
    # Conectar ao MongoDB
    mongo_client = pymongo.MongoClient("mongodb://admininovoice:Fiap2024@127.0.0.1:27017/inovoice")
    db = mongo_client["inovoice"]
    fs = gridfs.GridFS(db)

    # Recuperar os dois primeiros áudios armazenados no MongoDB
    audios = fs.find().limit(2)

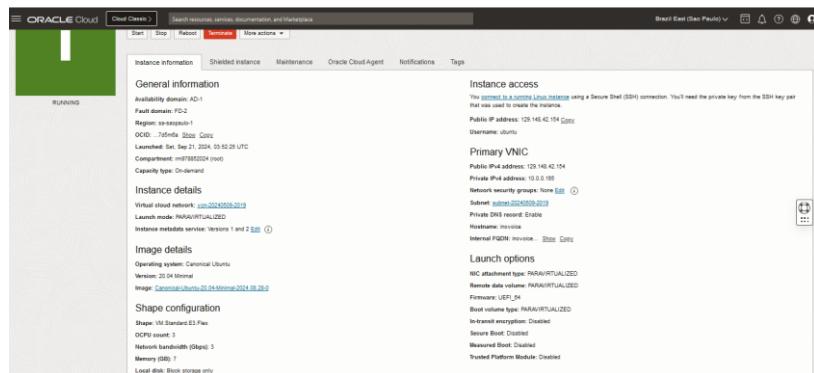
    # Salvar os áudios em arquivos locais
    for audio in audios:
        filename = audio.filename
        with open(filename, 'wb') as f:
            f.write(audio.read())

    print("Dois áudios foram transferidos para o local.")

# Agendar a execução da função todos os dias às 09:00
schedule.every().day.at("09:00").do(transfer_audios)

print("Agendado para execução todos os dias às 09:00.")

# Manter o script em execução
while True:
    schedule.run_pending()
    time.sleep(1)
```



## 7.2 BACKUP DO MONGODB

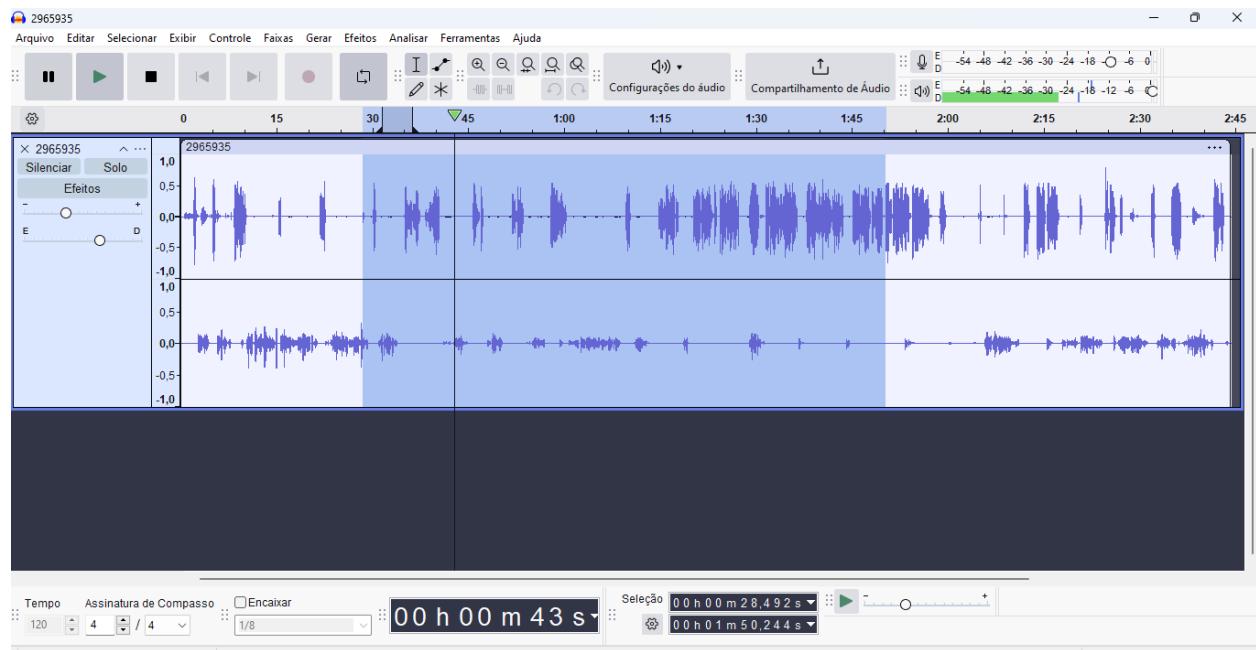
O MongoDB oferece diversas estratégias para garantir a segurança e a disponibilidade dos dados. Uma das ferramentas mais comuns é o mongodump, que cria cópias de segurança em formato BSON.

No caso do InoVoice, os backups foram definidos com base na criticidade dos dados, pois envolve LGPD e nos requisitos de negócio para não perder os dados dos áudios NPS e por isso, segundo a MongoDB recomenda-se a realização de backups diários, tanto para os dados históricos, quanto backups semanais quando necessário. Também é utilizado a boa prática de retenção dos backups, pois acreditamos que é muito importante para o cliente TOTVS devido aos aspectos como compliance e economia no custo de armazenamento.

Em caso de desastres, além dos backups, o InoVoice utiliza-se da replicação para alta disponibilidade, garantindo que os dados sejam replicados em tempo real para outros servidores.

## 8. TRATAMENTO DE ÁUDIO: AUDACITY

Nesta etapa do tratamento do áudio, vamos utilizar o audacity, que antecede a realização das análises de sentimentos, pois garante a qualidade e a precisão dos resultados, como: Ruídos, eco, variações de volume e outros artefatos presentes em gravações podem interferir significativamente na capacidade dos algoritmos de reconhecimento de fala e análise de sentimento de identificar corretamente as emoções e intenções expressas na fala. Ao eliminar esses ruídos como o exemplo acima, normalizamos o áudio, aumenta-se a precisão da transcrição e, consequentemente, da análise de sentimento.



Além disso, um ponto muito importante sobre essa etapa é que o tratamento de áudio permite adaptar o áudio às especificidades dos algoritmos de análise de sentimento. Diferentes algoritmos podem ter requisitos distintos em relação à qualidade do áudio. Por exemplo, alguns algoritmos podem ser mais sensíveis a ruídos de fundo, enquanto outros podem exigir um nível de volume específico. Por isso, no InoVoice ao preparar o áudio no Audacity, é possível otimizá-lo para cada algoritmo, maximizando a precisão dos resultados dos áudios NPS e realizando essas etapas de pré-processamento, os dados a serem demonstrados passam a ter mais confiança e permitindo a extração de insights mais confiáveis auxiliando na tomada de decisão.

## **9. SCRIPT PYTHON DE TRANSCRIÇÃO DE ÁUDIO, ANÁLISE DE SENTIMENTO E DATAFRAME**

A seguir, apresentamos nosso código de transcrição, análise de sentimentos e classificação NPS. Dividimos em três etapas para melhor entendimento do processo e também por performance. O exemplo a seguir foi executado em um ambiente do Google Colab, porém note que também pode ser utilizado em uma IDE local como PyCharm ou VSCode, ou ainda em uma máquina virtual instanciada em alguma nuvem onde a escalabilidade é maior.

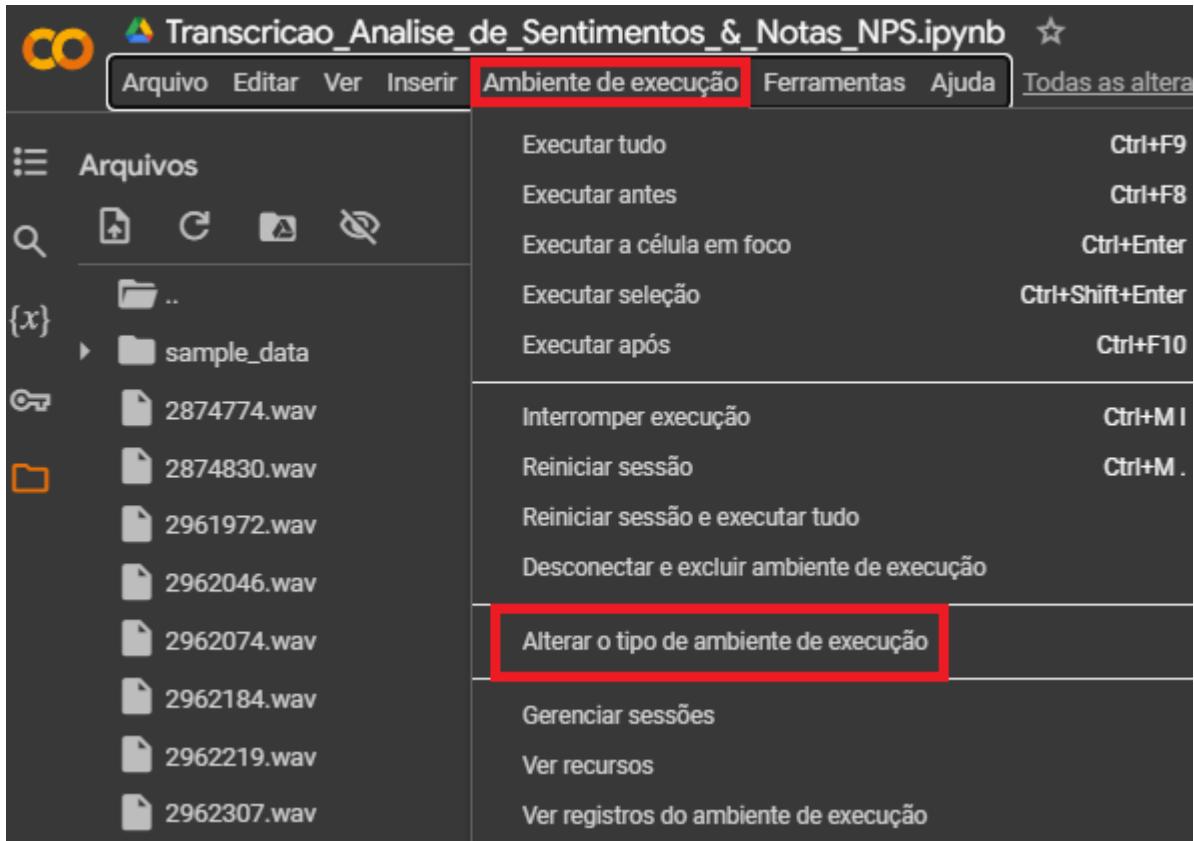
A primeira parte do código realiza a transcrição dos áudios, que são as conversas entre atendentes e clientes da TOTVS, nas quais uma série de perguntas referentes a notas de produtos e NPS são feitas. Essa primeira etapa irá realizar a transcrição e salvar o resultado em formato .txt.

Na segunda parte, realizamos a análise de sentimento para prever a classificação NPS do cliente. Utilizamos um dataset já preparado e modelado para análise de sentimentos com três classificadores: neutral, positive e negative, que equivalem a neutros, promotores e detratores. Esse dataset consiste em uma série de postagens do Twitter em português brasileiro e sua respectiva classificação, ou seja, é um dataset que utiliza linguagem natural, perfeito para treinar modelos de NLP (Natural Language Processing).

A terceira parte consiste em varrer as transcrições salvas em .txt e, por meio do código, selecionar as notas mencionadas pelo entrevistado, para assim armazenar essas notas em um dataset contendo as notas de cada categoria e a média NPS final. Em outras palavras, isso é uma forma de validar se a classificação feita na análise de sentimento anteriormente está condizente com as notas dadas de fato. Além disso, esses dados serão úteis para um ambiente de Business Intelligence posteriormente.

A seguir, segue o passo a passo do código.

1: Alterar o ambiente de execução para considerar a GPU, no exemplo do Google Colab, vá no menu Ambiente de Execução > Alterar o tipo de ambiente de execução > Seleciona T4 GPU. Isso permitirá que o processamento do áudio seja muito mais rápido e eficiente



The screenshot shows the Google Colab interface with the 'Alterar o tipo de ambiente de execução' (Change execution environment type) dialog box open. The 'T4 GPU' option is selected, indicated by a blue circle. The dialog box also includes a note about premium GPU access and buttons for 'Cancelar' (Cancel) and 'Salvar' (Save).

```

pip install git+https://github.com/openai/whisper.git
pip install torch torchaudio --index-url https://download.pytorch.org/whl/cu118
# Importar as bibliotecas necessárias
import whisper # Biblioteca para transcrição de áudio
import torch # É uma biblioteca de machine learning

# Collecting git+https://github.com/openai/whisper.git
Cloning https://github.com/openai/whisper.git to /tmp/pip-req-build-80n4fzvc
Running command git clone https://github.com/openai/whisper.git /tmp/pip-req-build-80n4fzvc
Submodule https://github.com/openai/whisper.git
Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: filelock>=3.0.0 in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: torch in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: torchaudio in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: torchaudio<0.10.0 in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: triton in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: ujson in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: networks in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: jinja2 in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: fsspec in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: certifi>=2021.4.17 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: numpy<1.18.0,!=1.17.0 in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: libedit>=2.1.3 in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: certifi>=2021.4.17 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: MarkupSafe>0.0.1 in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: numpy<1.18.0,!=1.17.0 in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: triton-3.0.0-cp310-cp310-manylinux_2_17_x86_64.whl (295.4 MB)
Building wheels for collected packages: openai-whisper (pyproject.toml)
Building wheel for openai-whisper (pyproject.toml) ... done
Created wheel for openai-whisper: filename=openai_whisper-20231117-py3-none-any.whl size=802819 sha256=2a735d7ee581cd06fb0030973fed466bf0960cc274bf13a1fe7a5144ef0a217b

```

2: Instalar as bibliotecas necessárias para a transcrição dos áudios, no caso utilizamos Whisper e Torch explicação.

**Whisper:** É um modelo de transcrição de áudio desenvolvido pela OpenAI. Ele é utilizado para converter áudio em texto, suportando múltiplos idiomas e podendo ser aplicado em tarefas de reconhecimento de fala (speech-to-text).

**PyTorch:** É uma biblioteca de machine learning de código aberto amplamente utilizada para desenvolvimento e treinamento de modelos de inteligência artificial. Ela é conhecida pela sua facilidade de uso e flexibilidade, permitindo a criação de redes neurais complexas de forma eficiente.

**CUDA (Compute Unified Device Architecture):** É uma plataforma de computação paralela e uma interface de programação de aplicativos (API) criada pela NVIDIA. Ela permite que desenvolvedores utilizem o poder de processamento de GPUs (unidades de processamento gráfico) para realizar cálculos e tarefas computacionais que não estão restritas ao processamento gráfico tradicional.

Essas bibliotecas funcionam de forma integrada, e o PyTorch com suporte a CUDA permite o uso de GPUs para acelerar o treinamento e a inferência de modelos, o que é especialmente útil em tarefas que envolvem grandes quantidades de dados, como transcrição de áudio com Whisper.

```
[ ] 1 # Instalar o Whisper e o PyTorch com suporte a CUDA
2 !pip install git+https://github.com/openai/whisper.git
3 !pip install torch torchvision torchaudio --index-url https://download.pytorch.org/wheel/cu118
4
5 # Importar as bibliotecas necessárias
6 import whisper # Biblioteca para transcrição de áudio
7 import torch # É uma biblioteca de machine learning

Collecting git+https://github.com/openai/whisper.git
  Cloning https://github.com/openai/whisper.git to /tmp/pip-req-build-2h4t3v2r
    Running command git clone --filter-blob:none --quiet https://github.com/openai/whisper.git /tmp/pip-req-build-2h4t3v2r
  Resolved https://github.com/openai/whisper.git to commit 27913e3107392276dc509148da1f41fb532c7e
  Installing build dependencies ... done
    Getting requirements to build wheel ... done
      Preparing metadata (pyproject.toml) ... done
  Requirement already satisfied: numba in /usr/local/lib/python3.10/dist-packages (from openai-whisper==20231117) (0.60.0)
  Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from openai-whisper==20231117) (1.26.4)
  Requirement already satisfied: torch in /usr/local/lib/python3.10/dist-packages (from openai-whisper==20231117) (2.4.1+cu121)
  Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from openai-whisper==20231117) (4.66.5)
  Requirement already satisfied: more-itertools in /usr/local/lib/python3.10/dist-packages (from openai-whisper==20231117) (10.5.0)
  Collecting tiktoken (from openai-whisper==20231117)
    Downloading tiktoken-0.7.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (6.6 kB)
  Collecting triton=2.0.0 (from openai-whisper==20231117)
    Downloading triton-3.0.0-e1-cp310-cp310-manylinux2014_x86_64.manylinux_2_17_x86_64.whl.metadata (1.3 kB)
  Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from triton==2.0.0->openai-whisper==20231117) (3.16.1)
  Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/local/lib/python3.10/dist-packages (from numba>openai-whisper==20231117) (0.43.0)
  Requirement already satisfied: regex<2022.1.18 in /usr/local/lib/python3.10/dist-packages (from tiktoken>openai-whisper==20231117) (2024.9.11)
  Requirement already satisfied: requests<2.26.0 in /usr/local/lib/python3.10/dist-packages (from tiktoken>openai-whisper==20231117) (2.32.3)
  Requirement already satisfied: typing-extensions<4.8.0 in /usr/local/lib/python3.10/dist-packages (from torch>openai-whisper==20231117) (4.12.2)
  Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>openai-whisper==20231117) (1.13.2)
  Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>openai-whisper==20231117) (3.3)
  Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>openai-whisper==20231117) (3.1.4)
  Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch>openai-whisper==20231117) (2024.6.1)
  Requirement already satisfied: charset-normalizer<4,>=2.26.0 in /usr/local/lib/python3.10/dist-packages (from tiktoken>openai-whisper==20231117) (3.3.2)
  Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>2.26.0->tiktoken>openai-whisper==20231117) (3.18)
  Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>2.26.0->tiktoken>openai-whisper==20231117) (2.0.7)
  Requirement already satisfied: certifi=>2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>2.26.0->tiktoken>openai-whisper==20231117) (2024.8.30)
  Requirement already satisfied: MarkupSafe=>2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2>torch>openai-whisper==20231117) (2.1.5)
  Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy>torch>openai-whisper==20231117) (1.3.0)
  Downloading triton-3.0.0-e1-cp310-cp310-manylinux2014_x86_64.manylinux_2_17_x86_64.whl (289.4 MB)
    289.4/289.4 MB 4.3 MB/s eta 0:00:00
  Downloading tiktoken-0.7.0-e1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.1 MB)
    1.1/1.1 MB 36.2 MB/s eta 0:00:00
Building wheels for collected packages: openai-whisper
  Building wheel for openai-whisper (pyproject.toml) ... done
  Created wheel for openai-whisper: filename=openai_whisper-20231117-py3-none-any.whl size=802819 sha256=f11fb20cffea304508243813b93c90a37e5481deccbd420e5c57f5ac0a7d4b8
  Stored in directory: /tmp/pip-ephem-wheel-cache-0ipods4/wheels/bb/6c/d0/622666868c179f156cf595c8b6f06f88bc5d80c4b31dcca03
Successfully built openai-whisper
Installing collected packages: triton, tiktoken, openai-whisper
  Traceback (most recent call last):
    File "/usr/local/lib/python3.10/dist-packages/pip/_internal/cli/base_command.py", line 179, in exc_logging_wrapper
      status = run_func(*args)
    File "/usr/local/lib/python3.10/dist-packages/pip/_internal/cli/req_command.py", line 67, in wrapper
      return func(self, options, args)
    File "/usr/local/lib/python3.10/dist-packages/pip/_internal/commands/install.py", line 455, in run
      installed = install_given_reqs(
    File "/usr/local/lib/python3.10/dist-packages/pip/_internal/req/_init_.py", line 70, in install_given_reqs
      requirement.install()
    File "/usr/local/lib/python3.10/dist-packages/pip/_internal/req/req_install.py", line 851, in install
      install_wheel(
    File "/usr/local/lib/python3.10/dist-packages/pip/_internal/operations/install/wheel.py", line 726, in install_wheel
      _install_wheel(
    File "/usr/local/lib/python3.10/dist-packages/pip/_internal/operations/install/wheel.py", line 584, in _install_wheel
      file.save()
    File "/usr/local/lib/python3.10/dist-packages/pip/_internal/operations/install/wheel.py", line 382, in save
```

3: Após instalação, verificar se o ambiente está de fato verificando a GPU, se sim, o código seguirá para função de transcrição via Whisper.

Note que utilizamos o modelo “medium” pois além de ser rápido, já faz uma excelente transcrição do áudio desejado.

No prompt, será solicitado o input do nome completo do arquivo e sua extensão (.wav por exemplo), após digitar, pressione enter e a transcrição será realizada e printada no prompt e salva como um arquivo .txt, caso deseje outra transcrição, responda “sim” no prompt e repita o ciclo, caso não queira, basta digitar “não” quando perguntado.

Um dos pontos principais desse código, é a velocidade de processamento para transcrição dos áudios, note que abaixo da imagem destacamos que dois áudios foram processados em 03:06 minutos, ou seja, ao selecionar a GPU via código Torch e Cuda, aumentamos consideravelmente a velocidade de transcrição dos áudios, o que era uma das dores da proposta.

```
# Verificar se o ambiente está utilizando GPU
device = "cuda" if torch.cuda.is_available() else "cpu"
print(f"Dispositivo em uso: {device}")

#Função para carregar um ou mais arquivo de áudio e realizar a transcrição
# Carregar o modelo 'medium' do Whisper
model = whisper.load_model('medium', device=device)

while True:
    # Solicitar o nome do arquivo de áudio
    audio_file = input("Digite o nome do arquivo de áudio (com extensão): ")

    # Transcrever o arquivo de áudio com as configurações otimizadas
    result = model.transcribe(audio_file, fp16=torch.cuda.is_available())

    # Exibir o texto transcritos
    print("\nTexto transcritos:")
    print(result['text'])

    # Salvar a transcrição em um arquivo .txt
    txt_file = audio_file.rsplit('.', 1)[0] + '.txt' # Substitui a extensão por .txt
    with open(txt_file, 'w', encoding='utf-8') as f:
        f.write(result['text'])
    print(f"\nTranscrição salva em: {txt_file}")

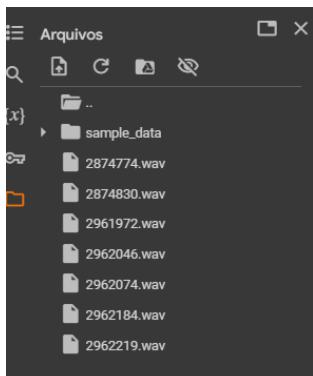
    # Perguntar se o usuário deseja transcrever outro arquivo
    continuar = input("\nDeseja transcrever outro arquivo? (sim/não): ").strip().lower()
    if continuar != 'sim':
        print("Encerrando o programa.")
        break

Dispositivo em uso: cuda
/usr/local/lib/python3.8/dist-packages/whisper/_init__.py:146: FutureWarning: You are using `torch.load` with `weights_only=False` (the current default value), which uses the default pickle module implicitly. It is possible
checkpoint = torch.load(fp, map_location=device)
Digite o nome do arquivo de áudio (com extensão): 2874830.wav
Texto transcritos:
Ol, bom dia, falo com o Sr. Pedro? Bom dia, sou eu mesmo. Sr. Pedro, me chamo Beatriz, falo da TOTS, tudo bem? Da TOTS? Isso. Certo. Tudo bem, Sr. Pedro? Tudo bom, é? Que bom. O Sr. seria o responsável ainda pelo DP e util
Transcrição salva em: 2874830.txt
Deseja transcrever outro arquivo? (sim/não): sim
Digite o nome do arquivo de áudio (com extensão): 2962046.wav
Texto transcritos:
Olí, Cláudio. Bom dia. Lilia da TOTS, tudo bem? Olí, tudo bem. Que ótimo. Sou do Departamento Voz do Cliente. A gente entrou em contato com o senhor no passado para fazer o NPS, uma avaliação geral. Estou ligando novamente
Transcrição salva em: 2962046.txt
Deseja transcrever outro arquivo? (sim/não): não
Encerrando o programa.
```

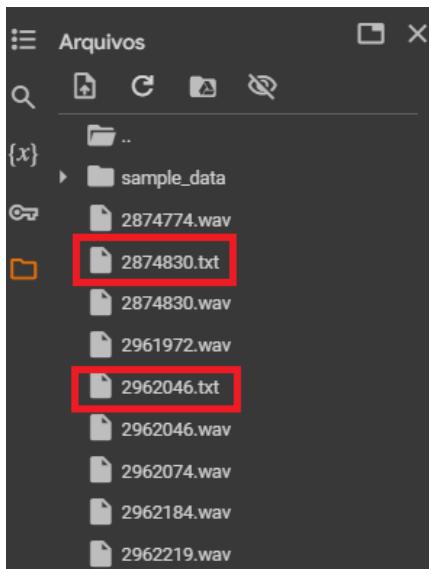
✓ Análise de Sentimento para Classificação

3m6s conclusão: 20:25

Exemplo de carregados no Google Colab para utilização



Observe os arquivos .txt gerados e salvos com as transcrições



3.1 Note que colocamos uma alternativa a esse código onde extraí de repositório do Github os áudios, fizemos isso caso estejam com problemas para anexar ou baixar o áudio e inseri-lo no Google Colab, basta puxar do repositório do Github que funcionará da mesma forma.

```
[2] 1 # Instalar as bibliotecas necessárias
2 !pip install git+https://github.com/openai/whisper.git
3 !pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118
4 !pip install requests
5
6 # Importar as bibliotecas necessárias
7 import whisper
8 import torch
9 import requests
10 import os
```

```
[1] 1 # Verificar se o ambiente está utilizando GPU
2 device = "cuda" if torch.cuda.is_available() else "cpu"
3 print(f"Dispositivo em uso: {device}")
4
5 # Carregar o modelo 'medium' no dispositivo selecionado
6 model = whisper.load_model('medium', device=device)
7
8 # Função para obter a lista de arquivos de áudio do repositório GitHub
9 def get_audio_files_from_github(repo_owner, repo_name, folder_path=''):
10     url = f'https://api.github.com/repos/{repo_owner}/{repo_name}/contents/{folder_path}'
11     response = requests.get(url)
12     if response.status_code == 200:
13         contents = response.json()
14         audio_files = [file['name'] for file in contents if file['type'] == 'file' and file['name'].lower().endswith('.mp3', '.wav', '.m4a', '.flac', '.aac', '.ogg')]
15         return audio_files
16     else:
17         print(f"Erro ao acessar o repositório GitHub: {response.status_code}")
18         return []
19
20 # Função para baixar o arquivo de áudio do GitHub
21 def download_file_from_github(repo_owner, repo_name, file_path, local_filename):
22     url = f'https://raw.githubusercontent.com/{repo_owner}/{repo_name}/main/{file_path}'
23     response = requests.get(url)
24     if response.status_code == 200:
25         with open(local_filename, 'wb') as f:
26             f.write(response.content)
27         return True
28     else:
29         print(f"Erro ao baixar o arquivo: {response.status_code}")
30         return False
31
32 # Informações do repositório GitHub
33 repo_owner = 'cesaroliveiragoes'
34 repo_name = 'Audios_Sprint_03'
```

```

36 while True:
37     # Listar os arquivos de áudio disponíveis no GitHub
38     print("\nArquivos de áudio disponíveis no GitHub:")
39     audio_files = get_audio_files_from_github(repo_owner, repo_name)
40     if not audio_files:
41         print("Nenhum arquivo de áudio encontrado.")
42         break
43     for filename in audio_files:
44         print(filename)
45
46     # Solicitar o nome do arquivo de áudio
47     audio_file = input("Digite o nome do arquivo de áudio para transcrever: ")
48
49     if audio_file not in audio_files:
50         print("Arquivo não encontrado no repositório GitHub.")
51         continue
52
53     try:
54         # Baixar o arquivo de áudio do GitHub
55         if download_file_from_github(repo_owner, repo_name, audio_file, audio_file):
56             # Transcrever o arquivo de áudio
57             result = model.transcribe(audio_file, fp16=torch.cuda.is_available())
58
59             # Exibir o texto transscrito
60             print("\nTexto transscrito:")
61             print(result['text'])
62
63             # Salvar a transcrição em um arquivo .txt
64             txt_file = audio_file.rsplit('.', 1)[0] + '.txt'
65             with open(txt_file, 'w', encoding='utf-8') as f:
66                 f.write(result['text'])
67             print(f"\nTranscrição salva em: {txt_file}")
68
69             # Remover o arquivo de áudio temporário
70             os.remove(audio_file)
71
72     except Exception as e:
73         print(f"Ocorreu um erro: {e}")
74
75     # Perguntar se o usuário deseja transcrever outro arquivo
76     continuar = input("\nDeseja transcrever outro arquivo? (sim/não): ").strip().lower()
77     if continuar != 'sim':
78         print("Encerrando o programa.")
79         break

```

## Exemplo do prompt com a alternativa do Github

```

Dispositivo em uso: cpu
100%|██████████| 1.42G/1.42G [00:15:00:00, 97.8MB/s]
/usr/local/lib/python3.10/dist-packages/whisper/_init_.py:146: FutureWarning: You are using `torch.load` with `weights_only=False` (the current default value), which uses the default pickle module implicitly. It is possible that your code will break in the future.
checkpoint = torch.load(fp, map_location=device)

Arquivos de áudio disponíveis no GitHub:
2874830.wav
2962046.wav
2962047.wav
2962219.wav
2962417.wav
2962972.wav
2963930.wav
2964126.wav
2964909.wav
2965023.wav
2967070.wav
Digite o nome do arquivo de áudio para transcrever: 2874830.wav

Texto transscrito:
Olá, bom dia, falo com o Sr. Pedro? Bom dia, sou eu mesmo. Sr. Pedro, me chamo Beatriz, falo da TOTS, tudo bem? Da TOTS? Isso. Certo. Tudo bem, Sr. Pedro? Tudo bom, é? Que bom. O Sr. seria o responsável ainda pelo DP e uti
Transcrição salva em: 2874830.txt

Deseja transcrever outro arquivo? (sim/não): não
Encerrando o programa.

```

4: A partir desta parte do código, iniciamos a Análise de Sentimentos, primeiramente importe as bibliotecas necessárias para aplicar os modelos de Machine Learning, Rede Neural e NLP conforme abaixo.

- **nltk (Natural Language Toolkit):** É uma biblioteca em Python para processamento de linguagem natural (PLN). Fornece ferramentas para tarefas como tokenização, stemming, lematização, análise sintática e semântica
- **from sklearn.model\_selection import train\_test\_split:** Importa a função train\_test\_split da biblioteca scikit-learn. Usada para dividir um conjunto de dados em subconjuntos de treinamento e teste de forma aleatória.
- **sklearn.naive\_bayes import GaussianNB, MultinomialNB, BernoulliNB:** Importa três classificadores Naive Bayes da scikit-learn.
  - GaussianNB: Adequado para dados contínuos e assume que as características seguem uma distribuição normal (gaussiana).
  - MultinomialNB: Ideal para dados discretos, como contagens de palavras em textos. Comumente usado em classificação de documentos.
  - BernoulliNB: Adequado para dados binários (0 ou 1), como indicadores de presença ou ausência de uma característica.
- **sklearn.metrics import accuracy\_score:** Importa a função accuracy\_score da scikit-learn. Utilizada para calcular a acurácia de um modelo de classificação, ou seja, a proporção de previsões corretas em relação ao total de previsões realizadas.
- **pickle:** Módulo padrão do Python para serialização e deserialização de objetos. Permite salvar objetos Python (como modelos treinados) em arquivos e carregá-los posteriormente, facilitando o armazenamento e a reutilização.
- **nltk.corpus import stopwords:** Importa a lista de stopwords (palavras irrelevantes) do corpus do NLTK. Stopwords são palavras comuns em um idioma (como "o", "a", "e") que geralmente são removidas em tarefas de processamento de texto, pois não agregam muito significado.
- **sklearn.feature\_extraction.text import CountVectorizer:** Importa o CountVectorizer da scikit-learn. Transforma uma coleção de documentos de texto em uma matriz de contagem de tokens (bag-of-words). Converte texto em dados numéricos, essencial para treinar modelos de aprendizado de máquina em tarefas de NLP.
- **re:** Módulo padrão do Python para trabalhar com expressões regulares. Permite realizar correspondência de padrões, substituições e outras manipulações avançadas de strings. Muito útil para limpar e pré-processar texto, removendo ou substituindo caracteres indesejados.

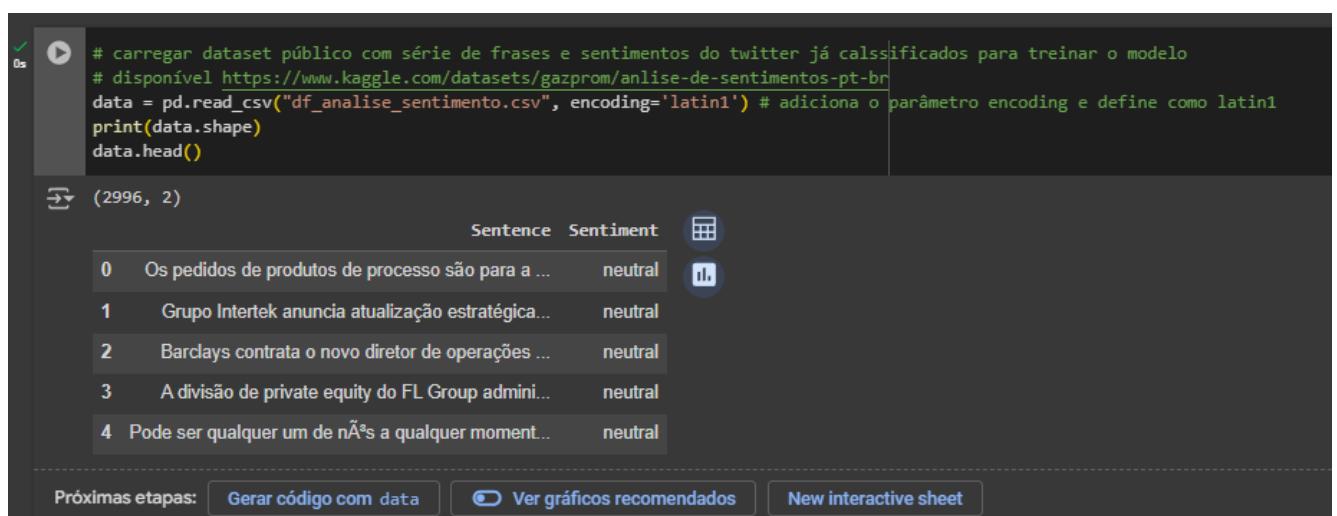
#### ▼ Análise de Sentimento para Classificação

```
[15] import nltk # Importar a biblioteca NLTK (Natural Language Toolkit) em Python.
      from sklearn.model_selection import train_test_split # Essa função é usada para dividir um conjunto de dados em subconjuntos de treino e teste de forma aleatória
      from sklearn.naive_bayes import GaussianNB, MultinomialNB, BernoulliNB # classificadores
      from sklearn.metrics import accuracy_score # calcula a precisão (acurácia) de um modelo de classificação
      import pickle # serializar e salvar objetos Python em arquivos e também para carregar esses objetos de volta para a memória
      from nltk.corpus import stopwords # Da biblioteca nltk.corpus, ela contém uma lista de palavras irrelevantes (como "o", "a", "e") que são frequentemente removidas
      from nltk.stem import SnowballStemmer # Da biblioteca nltk, o stemmer é usado para reduzir palavras ao seu radical, removendo sufixos ou prefixos
      from sklearn.feature_extraction.text import CountVectorizer # Essa classe converte um conjunto de documentos em uma matriz de bag-of-words, essencial para transformar texto em dados numéricos para ML
      import pandas as pd
      import numpy as np
      import re
```

5: O Código a seguir, carrega o dataset público com série de frases e sentimentos do twitter em português brasileiro, já classificados e treinados para aplicarmos no modelo.

Uma das maiores fontes de texto e expressões de sentimentos atualmente, são as redes sociais, e o Twitter com sua versatilidade e agilidade, e sempre apresentou uma enorme variedade de opções e dados, que, podem ser extremamente úteis para Machine Learning.

Sendo assim, minha ideia de usar um dataset de análise de sentimentos do Twitter, se deu pois, como queremos analisar o NPS de áudios transcritos em outras palavras, queremos entender o sentimento do cliente em relação ao produto, nada melhor do que usar uma base do Twitter onde milhares de opiniões sobre os mais diversos assuntos são publicadas diariamente, então, ensinar um modelo o que é positivo (promotor), negativo (detrator) e neutro através dos comentários do Twitter, facilmente disponibilizados em websites como kaggle ou hugging face, é uma forma extremamente eficiente de analisar NPS de clientes por meio de áudios transcritos.



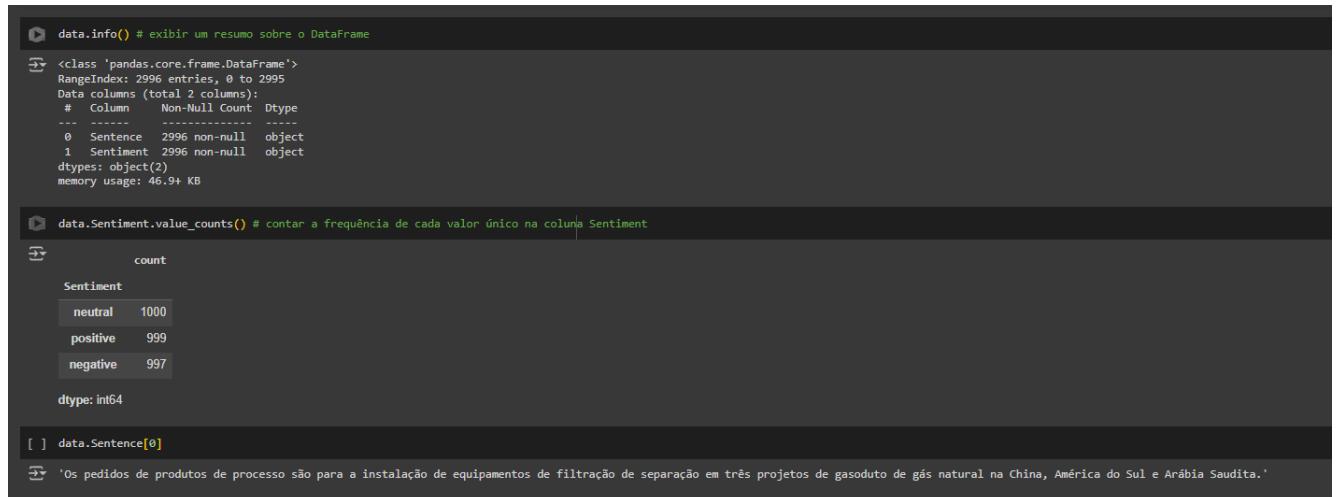
```
# carregar dataset público com série de frases e sentimentos do twitter já calssificados para treinar o modelo
# disponível https://www.kaggle.com/datasets/gazprom/anlise-de-sentimentos-pt-br
data = pd.read_csv("df_analise_sentimento.csv", encoding='latin1') # adiciona o parâmetro encoding e define como latin1
print(data.shape)
data.head()
```

(2996, 2)

|   | Sentence  | Sentiment |
|---|---|-----------|
| 0 | Os pedidos de produtos de processo são para a ... | neutral   |
| 1 | Grupo Intertek anuncia atualização estratégica... | neutral   |
| 2 | Barclays contrata o novo diretor de operações ... | neutral   |
| 3 | A divisão de private equity do FL Group admini... | neutral   |
| 4 | Pode ser qualquer um de nÃ³s a qualquer moment... | neutral   |

Próximas etapas: [Gerar código com data](#) [Ver gráficos recomendados](#) [New interactive sheet](#)

## 6: Verificando as características do dataframe



```
data.info() # exibir um resumo sobre o DataFrame
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2996 entries, 0 to 2995
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype  
--- 
 0   Sentence  2996 non-null   object 
 1   Sentiment  2996 non-null   object 
dtypes: object(2)
memory usage: 46.9+ KB
```

```
data.Sentiment.value_counts() # contar a frequência de cada valor único na coluna Sentiment
```

| Sentiment | count |
|-----------|-------|
| neutral   | 1000  |
| positive  | 999   |
| negative  | 997   |

```
[ ] data.Sentence[0]
```

```
'Os pedidos de produtos de processo são para a instalação de equipamentos de filtração de separação em três projetos de gasoduto de gás natural na China, América do Sul e Arábia Saudita.'
```

7: Aplicando as funções para pré-processamento do texto, removendo partes e palavras indesejadas, removendo caracteres especiais e convertendo tudo para minúsculo.

```
✓ [16] # Função será usada para limpar o texto fornecido, removendo partes indesejadas
def limpar(texto):
    limpo = re.compile(r'<.*?>') # Criar um padrão de correspondência. O padrão r'<.*?>' corresponde a qualquer texto que esteja dentro de tags HTML,
    #ou seja, algo que comece com <, tenha qualquer conteúdo entre os símbolos e termine com >.
    return re.sub(limpo,'',texto) # Retornar o texto limpo de qualquer um desses símbolos de HTML

data.Sentence = data.Sentence.apply(limpar)
data.Sentence[0]

→ 'Os pedidos de produtos de processo são para a instalação de equipamentos de filtração de separação em três projetos de gasoduto de gás natural na China América do Sul e Arábia Saudita'

✓ [17] # Função para remover caracteres especiais
def especial(texto):
    rem = '' # variável rem como uma string vazia
    for i in texto: # loop for que percorre cada caractere individual (i) no texto
        if i.isalnum(): # Retorna True se o caractere for uma letra ou um número
            rem = rem + i # Se o caractere for alfanumérico (letra ou número), ele é adicionado à string rem
        else:
            rem = rem + ' '
    return rem

data.Sentence = data.Sentence.apply(especial)
data.Sentence[0]

→ 'Os pedidos de produtos de processo são para a instalação de equipamentos de filtração de separação em três projetos de gasoduto de gás natural na China América do Sul e Arábia Saudita'

✓ [18] # Converter pra minúsculas
def minusculo(texto):
    return texto.lower()

data.Sentence = data.Sentence.apply(lambda x: x.lower())
data.Sentence[0]

→ 'os pedidos de produtos de processo são para a instalação de equipamentos de filtração de separação em três projetos de gasoduto de gás natural na china américa do sul e arábia saudita'
```

8: Baixando funções do nlkt para mais tratamento de texto, para remoção de palavras desnecessárias ao modelo

```
✓ [1] # Limpeza do dataset usando o nltk
nltk.download('stopwords') # Faz o download da lista de palavras de stopwords do NLTK, que são palavras comuns como 'de', 'a', 'o' que geralmente não agregam valor
nltk.download('punkt') # Faz o download de dados para a função de tokenização de palavras, que divide frases em palavras individuais
from nltk.tokenize import word_tokenize # Importa a função de tokenização que quebra o texto em palavras individuais (tokens)

def rem_stopwords(texto):
    stop_words = set(stopwords.words('portuguese')) # Obtém a lista de stopwords em português e as armazena em um conjunto
    words = word_tokenize(texto) # Usa o word_tokenize para quebrar o texto em uma lista de palavras (tokens)
    return [w for w in words if w not in stop_words] # Retorna uma lista de palavras filtrada, removendo as stopwords.

data.Sentence = data.Sentence.apply(rem_stopwords)
data.Sentence[0]

→ [nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]  Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]  Unzipping tokenizers/punkt.zip.
['pedidos',
 'produtos',
 'processo',
 'instalação',
 'equipamentos',
 'filtração',
 'separação',
 'três',
 'projetos',
 'gasoduto',
 'gás',
 'natural',
 'china',
 'américa',
 'sul',
 'arábia',
 'saudita']
```

9: Da biblioteca nltk, o stemmer é usado para reduzir palavras ao seu radical, removendo sufixos ou prefixos. Após isso, dividir o dataset em treino e teste para treinarmos o modelo de classificação.

```

✓ [20] def stem_txt(texto):
    ss = SnowballStemmer('portuguese') # Cria um objeto stemmer para a língua portuguesa usando o algoritmo Snowball Stemmer
    return ' '.join([ss.stem(palavra) for palavra in texto]) # Aplica o stemming em cada palavra do texto e une-as de volta em uma string

data.Sentence = data.Sentence.apply(stem_txt)
data.Sentence[0]

⇒ 'ped produt process instal equip filtraçā separ três projet gasodut gás natural chin amér sul aráb saudit'

✓ [21] X = np.array(data.iloc[:,0].values) # Extrai todos os valores da primeira coluna do DataFrame data usando o .iloc
y = np.array(data.Sentiment.values) # Extrai os valores da coluna Sentiment do DataFrame data, que contém os rótulos que tentaremos prever
cv = CountVectorizer(max_features=1000) # Ferramenta do sklearn usada para converter uma coleção de documentos de texto em uma matriz de Bag of Words
X = cv.fit_transform(data.Sentence).toarray() # Aplica a transformação CountVectorizer na coluna Sentence do DataFrame
print('X.shape = ', X.shape)
print('y.shape = ', y.shape)

⇒ X.shape = (2996, 1000)
y.shape = (2996,)

✓ [22] print(X)

⇒ [[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]

✓ [23] # Treinar test split
# Dividir os dados em conjuntos de treino e teste
trainx, textx, trainy, testy = train_test_split(X, y, test_size=0.2, random_state=9)
print('Train shapes : X = {}, y = {}'.format(trainx.shape, trainy.shape)) # Exibe as dimensões do conjunto de treino
print('Test shapes : X = {}, y = {}'.format(textx.shape, testy.shape)) # Exibe as dimensões do conjunto de teste

⇒ Train shapes : X = (2396, 1000), y = (2396,)
Test shapes : X = (600, 1000), y = (600,)

```

10: Escolhendo o melhor modelo para aplicar, no caso o que apresentou melhor acurácia considerando o dataset utilizado foi Bernoulli

### Criando o Modelo - Bag of Words

```

✓ [1] # Definir os modelos e treiná-los
# Modelo Naive Bayes Gaussiano, adequado para dados contínuos e assume que as características seguem uma distribuição normal.
# Modelo Naive Bayes Multinomial, adequado para dados categóricos e texto
# Modelo Naive Bayes Bernoulli, adequado para dados binários ou booleanos;
gnb, mnb, bnb = GaussianNB(), MultinomialNB(alpha=1.0, fit_prior=True), BernoulliNB(alpha=1.0, fit_prior=True)
gnb.fit(trainx, trainy) # Treina o modelo GaussianNB com o conjunto de treino (features e rótulos)
mnb.fit(trainx, trainy) # Treina o modelo MultinomialNB com o conjunto de treino
bnb.fit(trainx, trainy) # Treina o modelo BernoulliNB com o conjunto de treino

⇒ ▾ BernoulliNB
BernoulliNB()

✓ [25] # Predição e acurácia para escolher o melhor modelo
# Realiza previsões no conjunto de teste usando cada um dos modelos treinados
ypg = gnb.predict(textx)
ypm = mnb.predict(textx)
ypb = bnb.predict(textx)

✓ [26] # Mostrar qual possui maior acurácia, quanto mais próximo de 1 melhor
print('Gaussian = ', accuracy_score(testy, ypg))
print('Multinomial = ', accuracy_score(testy, ypm))
print('Bernoulli = ', accuracy_score(testy, ypb))

⇒ Gaussian = 0.55
Multinomial = 0.6433333333333333
Bernoulli = 0.65

✓ [27] pickle.dump(bnb, open('model.pkl', 'wb'))

```

11: Função para ler o arquivo .txt resultante das transcrições de áudio para classificá-lo como neutral, positive ou negative.

Note que como este é um modelo já treinado, equivalem para a análise NPS, onde neutral seria cliente neutro, positive promotor e negative detrator.

```
 6 def ler_arquivo():
 7     nome_arquivo = input("Digite o nome do arquivo .txt: ")
 8     rev = "" # Inicializa a variável rev como uma string vazia
 9
10    try:
11        with open(nome_arquivo, 'r') as arquivo:
12            rev = arquivo.read()
13    except FileNotFoundError:
14        print(f'O arquivo "{nome_arquivo}" não foi encontrado.')
15    except Exception as e:
16        print(f'OCorreu um erro ao ler o arquivo: {e}')
17
18    return rev
19
20 # Chamando a função e utilizando o valor retornado
21 conteudo_do_arquivo = ler_arquivo()
22 print("O conteúdo do arquivo é:", conteudo_do_arquivo)
```

Digitte o nome do arquivo .txt: 2874774.txt  
O conteúdo do arquivo é: Scandinavia Natalia, bom dia. Bom dia, Natalia Beatriz da Tato Falano. Gostaria de falar com a Sra. Silvana, do TI. Seria ela ainda a responsável pelo sistema? Só um momento. Obrigada. Silvana. Bo

12: Aplicação das funções para obter a análise de sentimento da transcrição

```
✓ [30] # Aplicar as funções de pré-processamento no conteúdo do arquivo
      f1 = limpar(conteudo_do_arquivo) # Remove as tags HTML do texto usando a função 'limpar' criada anteriormente
      f2 = especial(f1) # Remove caracteres especiais, mantendo apenas caracteres alfanuméricos e espaços.
      f3 = minusculo(f2) # Converte o texto para minúsculas
      f4 = rem_stopwords(f3) # Remove as stopwords (palavras comuns que não agregam muito significado) do texto
      f5 = stem_txt(f4) # Aplica stemming, reduzindo palavras às suas raízes

      # Criar a bolsa de palavras (bag of words) a partir do texto pré-processado
      bow, words = [], word_tokenize(f5) # Tokeniza o texto
      for word in words:
          bow.append(words.count(word)) # Conta a frequência de cada palavra e adiciona ao 'bow'.

      # Salvar o dicionário de palavras (vocabulario) em um arquivo usando pickle
      word_dict = cv.vocabulary_ # Obtém o dicionário de palavras
      pickle.dump(word_dict, open('bow.pk1', 'wb')) # Serializa o dicionário de palavras e salva em 'bow.pk1'

      inp = [] # Preparar o vetor de entrada para o modelo
      for i in word_dict:
          inp.append(f5.count(i[0])) # Conta a ocorrência de cada palavra do dicionário no texto pré-processado

      # Fazer a previsão com o modelo treinado
      y_pred = bnb.predict(np.array(inp).reshape(1, -1)) # Usa o modelo BernoulliNB para prever, pois tinha melhor acurácia
      # Use reshape(1, -1) para calcular automaticamente o número de colunas baseadas no array inputado
      # Printar o resultado final da previsão de sentimentos
      print("O arquivo escolhido para análise de sentimento apresenta um sentimento:", y_pred)
```

Entrada: O arquivo escolhido para análise de sentimento apresenta um sentimento: ['neutral']

13: Nessa parte final, extraímos as notas dadas para cada serviço e produto pelo cliente na ligação, pois dessa forma, ao extraí-las conseguimos de fato verificar se o modelo de análise de sentimentos previu corretamente a classificação NPS do cliente.

Além disso, conseguimos armazenar essas notas em um dataset para ser armazenado posteriormente em um Data Warehouse e utilizá-lo em um ambiente de Business Intelligence.

A primeira parte do código é a instalação e importação da biblioteca transformers, que nos fornece acesso a modelos de deep learning para NLP.

#### ▼ Extração das Notas para um Dataset

```
[31] # Instalar as bibliotecas necessárias
!pip install transformers

Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-packages (4.44.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from transformers) (3.16.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.23.2 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.24.7)
Requirement already satisfied: numpy<1.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (1.26.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (24.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (6.0.2)
Requirement already satisfied: regex<=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (2024.9.11)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformers) (2.32.3)
Requirement already satisfied: safetensors>=0.4.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.4.5)
Requirement already satisfied: tokenizers<0.20,>=0.19 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.19.1)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers) (4.66.5)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.23.2->transformers) (2024.6.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.23.2->transformers) (4.12.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2024.8.30)
```

A biblioteca Transformers é uma poderosa e popular ferramenta de código aberto desenvolvida pela empresa Hugging Face. Ela fornece acesso a modelos de aprendizado profundo de última geração baseados na arquitetura Transformer, que são amplamente usados em várias tarefas de processamento de linguagem natural (NLP).

```
[32] # Importar a biblioteca
from transformers import pipeline
```

14: Abaixo, segue o Código baseado em Regular Expressions (Regex ou re) para realizar todo o tratamento textual, eliminando palavras, sentenças, caracteres indesejados, permitindo realizar correspondências, detecção de padrões, substituições e outras manipulações avançadas de strings.

Todo esse processamento é essencial para a extração das notas em uma lista para depois ser inserida em um dataset analítico.

```
[24] # Função ajustada para extrair notas via Regular Expressions
def extrair_notas_apos_interrogacao_ajustada(transcricao):
    """
    Função para extrair a primeira nota (número entre 0 e 10) que aparece após uma interrogacão "?",
    ou números seguidos de um ponto final "." que aparecem isolados após uma frase,
    ou que seguem o termo "nota".
    Ignora números seguidos diretamente de uma interrogacão "?",
    bem como números irrelevantes como CNPJ, expressões "0 a 10", "0 a 10?", "0 a 10," e números
    no formato "01", "02", ..., "09", ou números maiores que 10 como 27.
    Também ignora números que aparecem antes da palavra "pergunta" ou "perguntas", e ignora "6 meses" ou qualquer número seguido de "meses".
    Considera apenas um número se ele aparecer repetido consecutivamente, como em "9. 9."
    """

    # Remover expressões "0 a 10", "0 a 10?", "0 a 10," e "CNPJ" e números irrelevantes (números longos com mais de 2 dígitos)
    transcricao_limpa = re.sub(r"\0\s*\a\s*\*10|\?|,\s]", "", transcricao)
    transcricao_limpa = re.sub(r"CNPJ|[0-9]{3,}", "", transcricao_limpa)

    # Remover números seguidos por "?" (ex.: "9?") e remover números maiores que 10 (preservando o "10")
    transcricao_limpa = re.sub(r"\b\d+\?\b", "", transcricao_limpa)
    transcricao_limpa = re.sub(r"\b(?!\0\b)\d{2,}\b", "", transcricao_limpa) # Remove números maiores que 10, mas preserva 10

    # Remover números no formato "01", "02", ..., "09"
    transcricao_limpa = re.sub(r"\b0[1-9]\b", "", transcricao_limpa)

    # Remover números que aparecem antes das palavras "pergunta" ou "perguntas"
    transcricao_limpa = re.sub(r"\b\d+\s*(pergunta|perguntas)\b", "", transcricao_limpa, flags=re.IGNORECASE)

    # Remover números seguidos da palavra "meses" (ex.: "6 meses")
    transcricao_limpa = re.sub(r"\b\d+\s*meses\b", "", transcricao_limpa, flags=re.IGNORECASE)

    # Capturar números seguidos por ponto final após uma frase (ex.: "O atendimento deles. 9. 9.")
    notas_com_ponto_final = re.findall(r"\.\s*(\d+)\.", transcricao_limpa)

    # Capturar números que vêm após o termo "nota" (ex.: "nota 7")
    notas_apos_termo_nota = re.findall(r"notas*(\d+)", transcricao_limpa)

    # Incluir números seguidos de ponto (ex.: "7.") e tratá-los como notas válidas
    transcricao_limpa = re.sub(r"(\b\d+)\.", r"\1", transcricao_limpa)

    # Dividir a transcrição em partes com base na presença "?"
    partes = re.split(r"\?", transcricao_limpa)

    notas = []

    # Iterar sobre as partes após cada interrogacão
    for parte in partes[1:]: # Ignorar o que vem antes da primeira "?"
        # Procurar o primeiro número após a interrogacão
        match = re.search(r"\b\d+\b", parte)
        if match:
            nota = int(match.group())
            if 0 <= nota <= 10 and (not notas or notas[-1] != nota): # Apenas números entre 0 e 10, sem duplicatas consecutivas
                notas.append(nota)

    # Adicionar as notas capturadas no padrão "O atendimento deles. 9. 9." e remover duplicatas consecutivas
    for i, nota in enumerate(notas_com_ponto_final):
        if i == 0 or int(nota) != int(notas_com_ponto_final[i - 1]): # Evitar duplicatas consecutivas
            nota = int(nota)
            if 0 <= nota <= 10:
                notas.append(nota)

    # Adicionar as notas capturadas após o termo "nota"
    for nota in notas_apos_termo_nota:
        nota = int(nota)
        if 0 <= nota <= 10 and (not notas or notas[-1] != nota): # Evitar duplicatas consecutivas
            notas.append(nota)

    return notas
```

```

# Inicializar uma lista para armazenar todas as listas de notas extraídas
todas_as_listas_de_notas = []

while True:
    # Pedir o nome do arquivo .txt como input do usuário
    nome_arquivo = input("Digite o nome do arquivo .txt (com extensão): ")

    # Ler o conteúdo do arquivo fornecido pelo usuário
    try:
        with open(nome_arquivo, 'r', encoding='utf-8') as f:
            nova_transcricao = f.read()

        # Processamento da transcrição com a função ajustada
        notas_ajustadas_nova_transcricao = extrair_notas_apos_interrogacao_ajustada(nova_transcricao)

        # Adicionar as notas extraídas à lista de todas as listas de notas
        todas_as_listas_de_notas.append(notas_ajustadas_nova_transcricao)

    # Calcular a média das notas para este arquivo
    if notas_ajustadas_nova_transcricao:
        media_nps = np.mean(notas_ajustadas_nova_transcricao)
    else:
        media_nps = np.nan # Ou você pode definir como 0 ou outro valor padrão

    # Imprimir as notas e a média para este arquivo
    print("\nlista_notas = {notas_ajustadas_nova_transcricao}")
    print(f"media_nps = {media_nps}\n")

    # Perguntar se o usuário deseja processar outro arquivo
    continuar = input("Deseja processar outro arquivo? (sim/não): ").strip().lower()
    if continuar != 'sim':
        break

    except FileNotFoundError:
        print(f"O arquivo '{nome_arquivo}' não foi encontrado. Verifique o nome e tente novamente.")

# Encontrar o número máximo de notas entre todas as listas
max_len = max(len(lista) for lista in todas_as_listas_de_notas) if todas_as_listas_de_notas else 0

if max_len > 0:
    # Preencher as listas menores com valores NaN para que todas as listas tenham o mesmo comprimento
    listas_alinhadas = [lista + [np.nan] * (max_len - len(lista)) for lista in todas_as_listas_de_notas]

    # Criar um DataFrame onde cada coluna será uma "nota 1", "nota 2", etc.
    df = pd.DataFrame(listas_alinhadas, columns=[f"nota_{i+1}" for i in range(max_len)])

    # Calcular a média das notas para cada linha, ignorando NaN
    df["media_nps"] = df.mean(axis=1, skipna=True)

    # Exibir o DataFrame resultante
    print("DataFrame com todas as notas e médias:")
    print(df)
else:
    print("Nenhuma nota foi extraída dos arquivos fornecidos.")

Digitando o nome do arquivo .txt (com extensão): 2874830
O arquivo '2874830' não foi encontrado. Verifique o nome e tente novamente.
Digitando o nome do arquivo .txt (com extensão): 2874830.txt

lista_notas = [10, 6, 9, 7]
media_nps = 8.0

Deseja processar outro arquivo? (sim/não): sim
Digitando o nome do arquivo .txt (com extensão): 2962046.txt

lista_notas = [8, 6, 7, 6, 8, 7, 6]
media_nps = 6.857142857142857

Deseja processar outro arquivo? (sim/não): não
DataFrame com todas as notas e médias:
  nota 1  nota 2  nota 3  nota 4  nota 5  nota 6  nota 7  media_nps
0      10       6       9       7     NaN     NaN     NaN   8.000000
1       8       6       7       6      8.0     7.0     6.0   6.857143

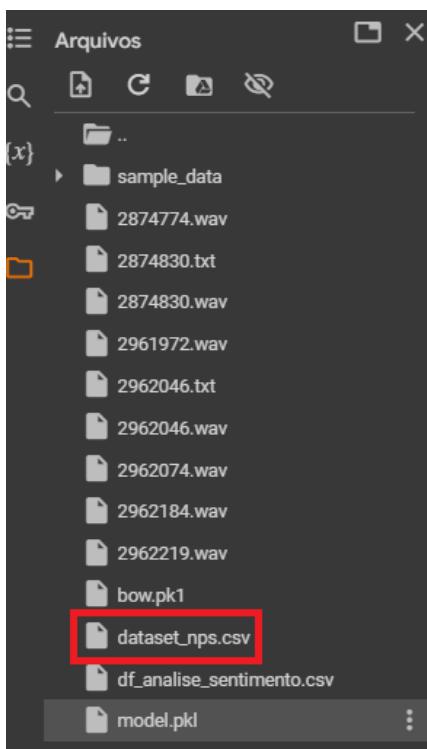
```

Note no output exemplificado, as listas com as notas dentro dela e o data frame com as notas inseridas em colunas. Além disso, a análise de sentimento do arquivo 2874830.txt foi considerada neutra, e analisando as notas, de fato é um cliente neutro, isso mostra a eficiência do modelo desenvolvido

15: Por fim, salvar o data frame como .csv para popular um DW e utilizá-lo em um ambiente de Business Intelligence.

```
[ ] 1 # Salvar o DataFrame como 'dataset_nps.csv'  
2 df.to_csv('dataset_nps.csv', index=False)  
3 print("\nO DataFrame foi salvo como 'dataset_nps.csv'.")  
  
→ O DataFrame foi salvo como 'dataset_nps.csv'.
```

Observe o dataset salvo

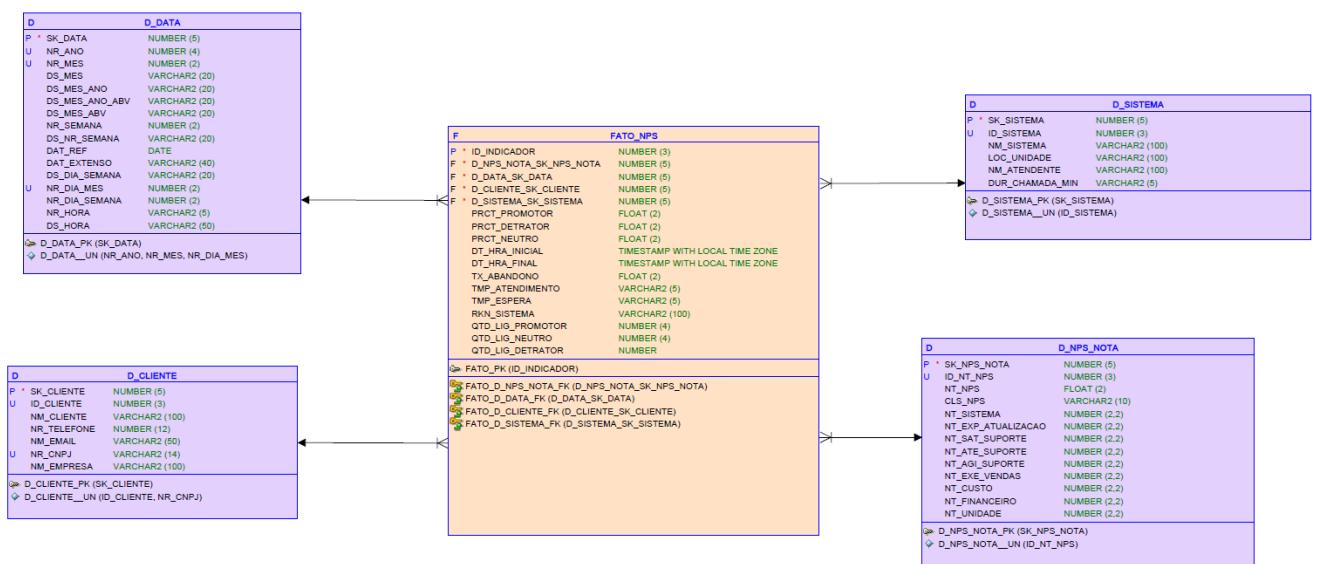


## 10. DATA WAREHOUSE

Nesta parte da documentação, iremos mostrar nossa modelagem do Data Warehousing e porque pensamos nela dessa forma, e por fim, os scripts de criação.

Fizemos essa modelagem baseada nas transcrições dos áudios e os dados que poderíamos extrair deles, como por exemplo, notas, produtos avaliados pelos clientes, nomes dos clientes, unidade TOTVS que os atendem entre outras. Em outras palavras, podemos dizer que o DW foi modelado baseado nas transcrições dos áudios das entrevistas NPS, com dados relevantes extraídos dessas transcrições para que gerem insights valiosos para a TOTVS em uma visão de Clientes, Sistema e Operacional conforme mostraremos adiante.

Abaixo segue imagem de nosso Modelo Star Schema de nosso Data Warehousing, contendo 4 dimensões: D\_DATA, D\_CLIENTE, D\_SISTEMA e D\_NPS\_NOTA e a fato FATO\_NPS.



**D\_DATA:** A dimensão data, como em qualquer data Warehouse, foi a primeira a ser criada, ela contém informações sobre as datas associadas aos fatos de um sistema de dados, servindo como uma referência temporal para análises. Ela é fundamental em sistemas de Business Intelligence (BI), pois permite que dados sejam analisados ao longo do tempo, facilitando a agregação e filtragem de informações em diferentes intervalos.

| D_DATA         |               |
|----------------|---------------|
| P * SK_DATA    | NUMBER (5)    |
| U NR_ANO       | NUMBER (4)    |
| U NR_MES       | NUMBER (2)    |
| DS_MES         | VARCHAR2 (20) |
| DS_MES_ANO     | VARCHAR2 (20) |
| DS_MES_ANO_ABV | VARCHAR2 (20) |
| DS_MES_ABV     | VARCHAR2 (20) |
| NR_SEMANA      | NUMBER (2)    |
| DS_NR_SEMANA   | VARCHAR2 (20) |
| DAT_REF        | DATE          |
| DAT_EXTENSO    | VARCHAR2 (40) |
| DS_DIA_SEMANA  | VARCHAR2 (20) |
| U NR_DIA_MES   | NUMBER (2)    |
| NR_DIA_SEMANA  | NUMBER (2)    |
| NR_HORA        | VARCHAR2 (5)  |
| DS_HORA        | VARCHAR2 (50) |

**D\_DATA\_PK (SK\_DATA)**  
**D\_DATA\_UN (NR\_ANO, NR\_MES, NR\_DIA\_MES)**

**D\_CLIENTE:** A dimensão cliente, em nosso Data Warehouse, é uma tabela que contém dados coletados sobre os clientes entrevistados para o NPS, como por exemplo, CNPJ da empresa que ele representa, nome da empresa, email, telefone, nome do próprio cliente e seu ID. Esses atributos são essenciais para entender quem são os clientes avaliados pelo NPS e eventualmente, esses dados poderão ser analisados em um ambiente de BI para descobrir insights valiosos e auxiliar na tomada de decisões e melhorias.

| D_CLIENTE                          |                           |
|------------------------------------|---------------------------|
| P *                                | SK_CLIENTE NUMBER (5)     |
| U                                  | ID_CLIENTE NUMBER (3)     |
|                                    | NM_CLIENTE VARCHAR2 (100) |
|                                    | NR_TELEFONE NUMBER (12)   |
|                                    | NM_EMAIL VARCHAR2 (50)    |
| U                                  | NR_CNPJ VARCHAR2 (14)     |
|                                    | NM_EMPRESA VARCHAR2 (100) |
| D_CLIENTE_PK (SK_CLIENTE)          |                           |
| D_CLIENTE_UN (ID_CLIENTE, NR_CNPJ) |                           |

**D\_NPS\_NOTA:** Essa dimensão, em nosso Data Warehouse, contém todas as notas dadas pelo cliente, durante a entrevista NPS, para cada produto e serviço perguntado pelos atendentes, essas notas são base para o cálculo NPS e classificar esse cliente como promotor, detrator ou neutro, o que, é o objetivo final de toda entrevista NPS.

| D_NPS_NOTA                  |                                 |
|-----------------------------|---------------------------------|
| P *                         | SK_NPS_NOTA NUMBER (5)          |
| U                           | ID_NT_NPS NUMBER (3)            |
|                             | NT_NPS FLOAT (2)                |
|                             | CLS_NPS VARCHAR2 (10)           |
|                             | NT_SISTEMA NUMBER (2,2)         |
|                             | NT_EXP_ATUALIZACAO NUMBER (2,2) |
|                             | NT_SAT_SUPORTE NUMBER (2,2)     |
|                             | NT_ATE_SUPORTE NUMBER (2,2)     |
|                             | NT_AGI_SUPORTE NUMBER (2,2)     |
|                             | NT_EXE_VENDAS NUMBER (2,2)      |
|                             | NT_CUSTO NUMBER (2,2)           |
|                             | NT_FINANCEIRO NUMBER (2,2)      |
|                             | NT_UNIDADE NUMBER (2,2)         |
| D_NPS_NOTA_PK (SK_NPS_NOTA) |                                 |
| D_NPS_NOTA_UN (ID_NT_NPS)   |                                 |

**D\_SISTEMA:** Essa dimensão, contém informações dos sistemas mencionados durante a entrevista NPS, pois, cada cliente possui uma plataforma TOTVS instalada em sua empresa, sendo assim, entender se esse sistema está ou não atendendo as demandas dos clientes é essencial para uma empresa de tecnologia como a TOTVS, portanto, dentro dessa dimensão, os atributos são indicadores dos sistemas.

| D   | D_SISTEMA                    |
|-----|------------------------------|
| P * | SK_SISTEMA NUMBER (5)        |
| U   | ID_SISTEMA NUMBER (3)        |
|     | NM_SISTEMA VARCHAR2 (100)    |
|     | LOC_UNIDADE VARCHAR2 (100)   |
|     | NM_ATENDENTE VARCHAR2 (100)  |
|     | DUR_CHAMADA_MIN VARCHAR2 (5) |
|     | ↳ D_SISTEMA_PK (SK_SISTEMA)  |
|     | ↳ D_SISTEMA__UN (ID_SISTEMA) |

**FATO\_NPS:** A tabela fato, é a nossa principal tabela que contém os dados numéricos ou quantitativos sobre o processo de negócio que está sendo analisado. Ela armazena os **fatos**, ou seja, os eventos medidos, que representam transações ou interações importantes. Dentro do contexto de NPS abordado pela TOTVS entendemos que atributos como percentual de promotores, detratores e neutros, data e hora inicial e final, tempo de espera, ranking de sistemas mais usados, quantidade total de detratores, promotores e neutros são essenciais para os insights que a TOTVS deseja no final.

| F   | FATO_NPS                                      |
|-----|---|
| P * | ID_INDICADOR NUMBER (3)                       |
| F * | D_NPS_NOTA_SK_NPS_NOTA NUMBER (5)             |
| F * | D_DATA_SK_DATA NUMBER (5)                     |
| F * | D_CLIENTE_SK_CLIENTE NUMBER (5)               |
| F * | D_SISTEMA_SK_SISTEMA NUMBER (5)               |
|     | PRCT_PROMOTOR FLOAT (2)                       |
|     | PRCT_DETRETOR FLOAT (2)                       |
|     | PRCT_NEUTRO FLOAT (2)                         |
|     | DT_HRA_INICIAL TIMESTAMP WITH LOCAL TIME ZONE |
|     | DT_HRA_FINAL TIMESTAMP WITH LOCAL TIME ZONE   |
|     | TX_ABANDONO FLOAT (2)                         |
|     | TMP_ATENDIMENTO VARCHAR2 (5)                  |
|     | TMP_ESPERA VARCHAR2 (5)                       |
|     | RKN_SISTEMA VARCHAR2 (100)                    |
|     | QTD_LIG_PROMOTOR NUMBER (4)                   |
|     | QTD_LIG_NEUTRO NUMBER (4)                     |
|     | QTD_LIG_DETRETOR NUMBER                       |
|     | ↳ FATO_PK (ID_INDICADOR)                      |
|     | ↳ FATO_D_NPS_NOTA_FK (D_NPS_NOTA_SK_NPS_NOTA) |
|     | ↳ FATO_D_DATA_FK (D_DATA_SK_DATA)             |
|     | ↳ FATO_D_CLIENTE_FK (D_CLIENTE_SK_CLIENTE)    |
|     | ↳ FATO_D_SISTEMA_FK (D_SISTEMA_SK_SISTEMA)    |

## 10.1 INSTALAÇÃO DO MYSQL

A seguir a instalação do MySQL

1: Desabilitação dos componentes do MySQL para permitir uma melhor instalação:

```
[root@mysql-inovoice opc]# sudo yum module disable mysql
Last metadata expiration check: 0:00:57 ago on Sun 22 Sep 2024 05:38:58 PM GMT.
Dependencies resolved.

Transaction Summary

Is this ok [y/N]: y
Completed.
[root@mysql-inovoice opc]# sudo yum install mysql-community-server -y
Last metadata expiration check: 0:01:11 ago on Sun 22 Sep 2024 05:38:58 PM GMT.
Dependencies resolved.

Transaction Summary

Package           Architecture   Version      Repository    Size
=====
Installing:
  mysql-community-server          x86_64        8.0.39-1.el8   ole_MySQL80  65 M
  mysql-community-client          x86_64        8.0.39-1.el8   ole_MySQL80  16 M
  mysql-community-client-plugins x86_64        8.0.39-1.el8   ole_MySQL80  3.6 M
  mysql-community-common         x86_64        8.0.39-1.el8   ole_MySQL80  669 k
  mysql-community-icu-data-files x86_64        8.0.39-1.el8   ole_MySQL80  2.2 M
  mysql-community-lz4             x86_64        8.0.39-1.el8   ole_MySQL80  1.5 M

Transaction Summary

Install 6 Package(s)

Total download size: 89 M
Installed size: 16 M
Downloading Packages:
(1/6): mysql-community-common-8.0.39-1.el8.x86_64.rpm           5.8 MB/s | 669 kB  00:00
(2/6): mysql-community-client-plugins-8.0.39-1.el8.x86_64.rpm     19 MB/s | 3.6 MB  00:00
(3/6): mysql-community-client-8.0.39-1.el8.x86_64.rpm            14 MB/s | 669 kB  00:00
(4/6): mysql-community-common-libs-8.0.39-1.el8.x86_64.rpm        7.4 MB/s | 1.5 MB  00:00
(5/6): mysql-community-client-8.0.39-1.el8.x86_64.rpm            32 MB/s | 16 MB  00:00
(6/6): mysql-community-server-8.0.39-1.el8.x86_64.rpm           69 MB/s | 65 MB  00:00
Total                                         73 MB/s | 89 MB  00:01

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing :
    Installing : mysql-community-common-8.0.39-1.el8.x86_64
    Installing : mysql-community-client-plugins-8.0.39-1.el8.x86_64
  Installing  : mysql-community-common-libs-8.0.39-1.el8.x86_64
  Running scriptlet: mysql-community-libs-8.0.39-1.el8.x86_64
  Installing  : mysql-community-client-8.0.39-1.el8.x86_64
  Installing  : mysql-community-server-8.0.39-1.el8.x86_64

Total                                         1/1
 1/1
 1/1
 3/6
 3/6
 4/6
```

2: Instalação via wget:

```
complete!
[root@mysql-inovoice opc]# wget https://dev.mysql.com/get/mysql80-community-release-el7-3.noarch.rpm
--2024-09-22 17:40:37.149999  https://dev.mysql.com/get/mysql80-community-release-el7-3.noarch.rpm
Resolving dev.mysql.com (dev.mysql.com)... 123.108.20.40, 1419:4e00:291::2e31, 2606:1419:4e00:29c::2e31
Connecting to dev.mysql.com (dev.mysql.com)|123.108.20.40|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://repo.mysql.com//mysql80-community-release-el7-3.noarch.rpm [following]
--2024-09-22 17:40:38.149999  https://repo.mysql.com//mysql80-community-release-el7-3.noarch.rpm
Reusing existing connection 1...
HTTP request sent, awaiting response... 200 OK
Length: 26024 (25K) [application/x-redhat-package-manager]
Saving to: 'mysql80-community-release-el7-3.noarch.rpm.1'

mysql80-community-release-el7-3.noarch.rpm.1  100%[=====] 25.41K --.-KB/s  in 0s

2024-09-22 17:40:38 (105 MB/s) - 'mysql80-community-release-el7-3.noarch.rpm.1' saved [26024/26024]
```

```
[root@mysql-inovoice opc]# sudo rpm -Uvh mysql80-community-release-el7-3.noarch.rpm
warning: mysql80-community-release-el7-3.noarch.rpm: Header V3 DSA/SHA1 Signature, key ID 5072e1f5: NOKEY
Verifying...
#####
[100%]
Preparing...
#####
[100%]
  package mysql80-community-release-el7-3.noarch is already installed
[root@mysql-inovoice opc]# sudo yum install mysql-community-server -y
Last metadata expiration check: 0:01:51 ago on Sun 22 Sep 2024 05:38:58 PM GMT.
Package mysql-community-server-8.0.39-1.el8.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

3: Atualização do repo:

```
complete!
[root@mysql-inovoice opc]# yum repolist all
repo id                                repo name                               status
mysql-cluster-7.5-community              MySQL Cluster 7.5 Community - Source  disabled
mysql-cluster-7.5-community-source       MySQL Cluster 7.5 Community - Source  disabled
mysql-cluster-7.6-community              MySQL Cluster 7.6 Community - Source  disabled
mysql-cluster-7.6-community-source       MySQL Cluster 7.6 Community - Source  disabled
mysql-connectors-community               MySQL Connectors Community - Source  disabled
mysql-connectors-community-source        MySQL Connectors Community - Source  enabled
mysql-tools-community                   MySQL Tools Community - Source        enabled
mysql-tools-community-source            MySQL Tools Community - Source        disabled
mysql-tools-preview                     MySQL Tools Preview - Source        disabled
mysql-tools-preview-source              MySQL Tools Preview - Source        disabled
mysql55-community                       MySQL 5.5 Community Server - Source  disabled
mysql55-community-source                 MySQL 5.5 Community Server - Source  disabled
mysql56-community                       MySQL 5.6 Community Server - Source  disabled
mysql56-community-source                 MySQL 5.6 Community Server - Source  disabled
mysql57-community                       MySQL 5.7 Community Server - Source  disabled
mysql57-community-source                 MySQL 5.7 Community Server - Source  enabled
mysql80-community                       MySQL 8.0 Community Server - Source  enabled
mysql80-community-source                 MySQL 8.0 Community Server - Source  disabled
old_MORHIC                           Latest RHC with fixes from Oracle for Oracle Linux 8 (x86_64)  enabled
old_MORHIC_64                         Latest RHC with fixes from Oracle for Oracle Linux 8 (x86_64)  disabled
old_MYSQL80_connectors_community        MySQL 8.0 Connectors Community for Oracle Linux 8 (x86_64)  enabled
old_MYSQL80_tools_community            MySQL 8.0 Tools Community for Oracle Linux 8 (x86_64)  enabled
old_UERK6                               Latest Unbreakable Enterprise Kernel Release 6 for Oracle Linux 8 (x86_64)  disabled
old_UERK6_64                           Oracle Linux 8 UERK6 (x86_64)                enabled
old_UERK7                               Latest Unbreakable Enterprise Kernel Release 7 for Oracle Linux 8 (x86_64)  enabled
old_UERK7_RDMA                         Oracle Linux 8 UERK7 RDMA (x86_64)            disabled
old_addons                            Oracle Linux 8 Addons (x86_64)              enabled
old_baseos                            Oracle Linux 8 BaseOS Stream (x86_64)          enabled
old_baseos_latest                      Oracle Linux 8 BaseOS Latest (x86_64)          enabled
old_codeready_builder                 Oracle Linux 8 CodeReady Builder (x86_64)        unsupported
old_developer                          Oracle Linux 8 Development Packages (x86_64)  disabled
old_developer_EPEL                     Oracle Linux 8 Development Packages (x86_64)  disabled
old_developer_EPEL_modular             Oracle Linux 8 EPEL Modular Packages for Development (x86_64)  disabled
old_developer_UERK6                    Oracle Linux 8 EPEL Modular Packages for Development (x86_64)  disabled
old_developer_UERK7                    Oracle Linux 8 EPEL Modular Packages for Development (x86_64)  disabled
old_ksplice                           Ksplice for Oracle Linux 8 (x86_64)            enabled
old_kvm_apstream                      Oracle Linux 8 KVM Application Stream (x86_64)  disabled
old_ocl_included                      Oracle Software for OCL users on Oracle Linux 8 (x86_64)  enabled
```

#### 4: Limpeza do cache:

```
[root@mysql-inovoice opc]# sudo yum-config-manager --enable mysql80-community
[root@mysql-inovoice opc]# sudo yum clean all
81 files removed
[root@mysql-inovoice opc]# sudo yum makecache
MySQL 8.0 Community Server
MySQL Connector Community
MySQL Tools Community
MySQL Utilities Oracle Linux 8 (x86_64)
MySQL 8.0 Tools Community for Oracle Linux 8 (x86_64)
MySQL 8.0 Connectors Community for Oracle Linux 8 (x86_64)
Oracle Software for OCI users on Oracle Linux 8 (x86_64)
Oracle Linux 8 Base Latest (x86_64)
Oracle Linux Application System (x86_64)
Oracle Linux 8 Addons (x86_64)
Latest Unbreakable Enterprise Kernel Release 7 for Oracle Linux 8 (x86_64)
Metadata cache created.
```

#### 5: Inicialização do MySQL:

```
[root@mysql-inovoice opc]# sudo systemctl start mysqld
[root@mysql-inovoice opc]# sudo systemctl status mysqld
● mysqld.service - MySQL Server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2024-09-22 17:44:00 GMT; 8s ago
     Docs: man:mysqld(8)
           http://dev.mysql.com/doc/refman/en/using-systemd.html
  Process: 95303 ExecStartPre=/usr/bin/mysqld_pre_systemd (code=exited, status=0/SUCCESS)
 Main PID: 95378 (mysqld)
   Status: "Server is operational"
   Tasks: 38 (limit: 22589)
  Memory: 565.6M
    CGroup: /system.slice/mysqld.service
            └─95378 /usr/sbin/mysqld

Sep 22 17:43:51 mysql-inovoice systemd[1]: Starting MySQL Server...
Sep 22 17:44:00 mysql-inovoice systemd[1]: Started MySQL Server.
```

6: Nessa fase foram feitas as configurações dos usuários root e admin e abaixo logamos no MySQL com o admin inovoice:

```
[root@mysql-inovoice opc]# mysql -u inovoice -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 8.0.39 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

#### 7: Criação do DW:

```
mysql> CREATE DATABASE dw_inovoice;
Query OK, 1 row affected (0.00 sec)
```

#### 8: Entrar no DW:

```
mysql> use dw_inovoice;
Database changed
```

## 10.2 SCRIPT SQL PARA CRIAÇÃO DAS TABELAS

Segue abaixo o script de criação das tabelas do DW considerando o MySQL.

-- Script convertido do Oracle para MySQL

-- Criação da tabela d\_cliente

```
CREATE TABLE d_cliente (
    sk_cliente INT NOT NULL AUTO_INCREMENT,
    id_cliente INT,
    nm_cliente VARCHAR(100),
    nr_telefone BIGINT,
    nm_email VARCHAR(50),
    nr_cnpj VARCHAR(14),
    nm_empresa VARCHAR(100),
    PRIMARY KEY (sk_cliente),
    UNIQUE KEY d_cliente_un (id_cliente, nr_cnpj)
);
```

-- Criação da tabela d\_data

```
CREATE TABLE d_data (
    sk_data INT NOT NULL AUTO_INCREMENT,
    nr_ano INT,
    nr_mes TINYINT,
    ds_mes VARCHAR(20),
    ds_mes_ano VARCHAR(20),
    ds_mes_ano_abv VARCHAR(20),
    ds_mes_abv VARCHAR(20),
    nr_semana TINYINT,
    ds_nr_semana VARCHAR(20),
    dat_ref DATE,
```

```
dat_extenso VARCHAR(40),
ds_dia_semana VARCHAR(20),
nr_dia_mes TINYINT,
nr_dia_semana TINYINT,
nr_hora VARCHAR(5),
ds_hora VARCHAR(50),
PRIMARY KEY (sk_data),
UNIQUE KEY d_data_un (nr_ano, nr_mes, nr_dia_mes)
);
```

-- Criação da tabela d\_nps\_nota

```
CREATE TABLE d_nps_nota (
sk_nps_nota INT NOT NULL AUTO_INCREMENT,
id_nt_nps INT,
nt_nps FLOAT,
cls_nps VARCHAR(10),
nt_sistema DECIMAL(3,2),
nt_exp_atualizacao DECIMAL(3,2),
nt_sat_suporte DECIMAL(3,2),
nt_ate_suporte DECIMAL(3,2),
nt_agi_suporte DECIMAL(3,2),
nt_exe_vendas DECIMAL(3,2),
nt_custo DECIMAL(3,2),
nt_financeiro DECIMAL(3,2),
nt_unidade DECIMAL(3,2),
PRIMARY KEY (sk_nps_nota),
UNIQUE KEY d_nps_nota_un (id_nt_nps)
);
```

-- Criação da tabela d\_sistema

```
CREATE TABLE d_sistema (
    sk_sistema INT NOT NULL AUTO_INCREMENT,
    id_sistema INT,
    nm_sistema VARCHAR(100),
    loc_unidade VARCHAR(100),
    nm_atendente VARCHAR(100),
    dur_chamada_min VARCHAR(5),
    PRIMARY KEY (sk_sistema),
    UNIQUE KEY d_sistema_un (id_sistema)
);
```

-- Criação da tabela fato\_nps

```
CREATE TABLE fato_nps (
    id_indicador INT NOT NULL AUTO_INCREMENT,
    d_nps_nota_sk_nps_nota INT NOT NULL,
    d_data_sk_data INT NOT NULL,
    d_cliente_sk_cliente INT NOT NULL,
    d_sistema_sk_sistema INT NOT NULL,
    prct_promotor DECIMAL(5,2),
    prct_detrator DECIMAL(5,2),
    prct_neutro DECIMAL(5,2),
    dt_hra_inicial DATETIME,
    dt_hra_final DATETIME,
    tx_abandono DECIMAL(5,2),
    tmp_atendimento VARCHAR(5),
    tmp_espera VARCHAR(5),
    rkn_sistema VARCHAR(100),
    qtd_lig_promotor INT,
    qtd_lig_neutro INT,
    qtd_lig_detrator INT,
```

```

PRIMARY KEY (id_indicador),
FOREIGN KEY (d_nps_nota_sk_nps_nota) REFERENCES d_nps_nota
(sk_nps_nota),
FOREIGN KEY (d_data_sk_data) REFERENCES d_data (sk_data),
FOREIGN KEY (d_cliente_sk_cliente) REFERENCES d_cliente (sk_cliente),
FOREIGN KEY (d_sistema_sk_sistema) REFERENCES d_sistema (sk_sistema)
);

```

## D\_DATA

```

mysql> CREATE TABLE d_data (
->     sk_data INT NOT NULL AUTO_INCREMENT,
->     nr_ano INT,
->     nr_mes TINYINT,
->     ds_mes VARCHAR(20),
->     ds_mes_ano VARCHAR(20),
->     ds_mes_ano_abv VARCHAR(20),
->     ds_mes_abv VARCHAR(20),
->     nr_semana TINYINT,
->     ds_nr_semana VARCHAR(20),
->     dat_ref DATE,
->     dat_extenso VARCHAR(40),
->     ds_dia_semana VARCHAR(20),
->     nr_dia_mes TINYINT,
->     nr_dia_semana TINYINT,
->     nr_hora VARCHAR(5),
->     ds_hora VARCHAR(50),
->     PRIMARY KEY (sk_data),
->     UNIQUE KEY d_data_un (nr_ano, nr_mes, nr_dia_mes)
-> );
Query OK, 0 rows affected (0.03 sec)

```

## D\_CLIENTE

```

mysql> CREATE TABLE d_cliente (
->     sk_cliente INT NOT NULL AUTO_INCREMENT,
->     id_cliente INT,
->     nm_cliente VARCHAR(100),
->     nr_telefone BIGINT,
->     nm_email VARCHAR(50),
->     nr_cnpj VARCHAR(14),
->     nm_empresa VARCHAR(100),
->     PRIMARY KEY (sk_cliente),
->     UNIQUE KEY d_cliente_un (id_cliente, nr_cnpj)
-> );
Query OK, 0 rows affected (0.03 sec)

```

## D\_NPS\_NOTA

```
mysql> CREATE TABLE d_nps_nota (
->     sk_nps_nota INT NOT NULL AUTO_INCREMENT,
->     id_nt_nps INT,
->     nt_nps FLOAT,
->     cls_nps VARCHAR(10),
->     nt_sistema DECIMAL(3,2),
->     nt_exp_atualizacao DECIMAL(3,2),
->     nt_sat_suporte DECIMAL(3,2),
->     nt_ate_suporte DECIMAL(3,2),
->     nt_agi_suporte DECIMAL(3,2),
->     nt_exe_vendas DECIMAL(3,2),
->     nt_custo DECIMAL(3,2),
->     nt_financeiro DECIMAL(3,2),
->     nt_unidade DECIMAL(3,2),
->     PRIMARY KEY (sk_nps_nota),
->     UNIQUE KEY d_nps_nota_un (id_nt_nps)
-> );
Query OK, 0 rows affected (0.03 sec)
```

## D\_SISTEMA

```
mysql> CREATE TABLE d_sistema (
->     sk_sistema INT NOT NULL AUTO_INCREMENT,
->     id_sistema INT,
->     nm_sistema VARCHAR(100),
->     loc_unidade VARCHAR(100),
->     nm_atendente VARCHAR(100),
->     dur_chamada_min VARCHAR(5),
->     PRIMARY KEY (sk_sistema),
->     UNIQUE KEY d_sistema_un (id_sistema)
-> );
Query OK, 0 rows affected (0.04 sec)
```

## FATO\_NPS

```
mysql> CREATE TABLE fato_nps (
->     id_indicador INT NOT NULL AUTO_INCREMENT,
->     d_nps_nota_sk_nps_nota INT NOT NULL,
->     d_data_sk_data INT NOT NULL,
->     d_cliente_sk_cliente INT NOT NULL,
->     d_sistema_sk_sistema INT NOT NULL,
->     prct_promotor DECIMAL(5,2),
->     prct_detratr DECIMAL(5,2),
->     prct_neutro DECIMAL(5,2),
->     dt_hra_inicial DATETIME,
->     dt_hra_final DATETIME,
->     tx_abandono DECIMAL(5,2),
->     tmp_atendimento VARCHAR(5),
->     tmp_espera VARCHAR(5),
->     rkn_sistema VARCHAR(100),
->     qtd_lig_promotor INT,
->     qtd_lig_neutro INT,
->     qtd_lig_detratr INT,
->     PRIMARY KEY (id_indicador),
->     FOREIGN KEY (d_nps_nota_sk_nps_nota) REFERENCES d_nps_nota (sk_nps_nota),
->     FOREIGN KEY (d_data_sk_data) REFERENCES d_data (sk_data),
->     FOREIGN KEY (d_cliente_sk_cliente) REFERENCES d_cliente (sk_cliente),
->     FOREIGN KEY (d_sistema_sk_sistema) REFERENCES d_sistema (sk_sistema)
-> );
Query OK, 0 rows affected (0.04 sec)
```

## 10.3 POPULANDO AS TABELAS DO DW

### Inserindo dados na tabela

#### D\_DATA:

```
mysql> INSERT INTO d_data (nr_ano, nr_mes, ds_mes, ds_mes_ano, ds_mes_ano_abv, ds_mes_abv, nr_semana, ds_nr_semana, dat_ref, dat_extenso, ds_dia_semana, nr_dia_mes, nr_dia_semana, nr_hora, ds_hora) VALUES
--> (2023, 12, 'Dezembro', 'Dezembro 2023', 'Dez', 'Dez', 1, 'Semana 1', '2023-12-03', 'Terceiro dia de Dezembro de 2023', 'Domingo', 3, 7, '05:52', 'Cinco horas e cinqüenta e dois minutos'),
--> (2023, 5, 'Maio', 'Maio 2023', 'Maí', 'Maí', 1, 'Semana 1', '2023-05-07', 'Sétimo dia de Maio de 2023', 'Domingo', 7, 1, '02:57', 'Duas horas e cinqüenta e sete minutos'),
--> (2023, 11, 'Novembro', 'Novembro 2023', 'Nov', 'Nov', 1, 'Semana 1', '2023-11-23', 'Vigésimo terceiro dia de Novembro de 2023', 'Quinta-feira', 23, 4, '04:44', 'Quatro horas e quarenta e quatro minutos'),
--> (2023, 2, 'Fevereiro', 'Fevereiro 2023', 'Fev', 'Fev', 1, 'Semana 1', '2023-02-14', 'Quatorze dias de Fevereiro de 2023', 'Terça-feira', 14, 2, '01:54', 'Uma hora e cinqüenta e quatro minutos'),
--> (2023, 9, 'Setembro', 'Setembro 2023', 'Set', 'Set', 1, 'Semana 1', '2023-09-09', 'Nono dia de Setembro de 2023', 'Sábado', 9, 6, '03:08', 'Três horas e oito minutos'),
--> (2023, 6, 'Junho', 'Junho 2023', 'Jun', 'Jun', 1, 'Semana 1', '2023-06-18', 'Dezesseis dias de Junho de 2023', 'Domingo', 18, 7, '05:42', 'Cinco horas e quarenta e dois minutos'),
--> (2023, 4, 'Abril', 'Abril 2023', 'Abr', 'Abr', 1, 'Semana 1', '2023-04-30', 'Trinta dias de Abril de 2023', 'Domingo', 30, 1, '03:10', 'Três horas e dez minutos'),
--> (2023, 1, 'Janeiro', 'Janeiro 2023', 'Jan', 'Jan', 1, 'Semana 1', '2023-01-02', 'Dois dias de Janeiro de 2023', 'Segunda-feira', 2, 1, '08:58', 'Oito horas e cinqüenta minutos');
Query OK, 8 rows affected (0.01 sec)
Records: 8 Duplicates: 0 Warnings: 0
```

#### D\_CLIENTE:

```
mysql> INSERT INTO d_cliente (id_cliente, nm_cliente, nr_telefone, nm_email, nr_cnpj, nm_empresa) VALUES
--> (1, 'Geraldo', NULL, 'geraldo.darwin@bosel.com', '4544', 'N/A'),
--> (2, 'Ana Carla', NULL, 'anacarla.pereira@amelia.com.br', '07354754000199', 'N/A'),
--> (3, 'Claudio', NULL, 'N/A', '000200', 'N/A'),
--> (4, 'Dantele Melo', NULL, 'compras@betbar.com.br', '02315905000186', 'N/A'),
--> (5, 'Gabriel', NULL, 'ti@resendesantos.com.br', '000127', 'N/A'),
--> (7, 'Vitória', NULL, 'vitoria@hotmail.com', '12345678000190', 'N/A'),
--> (8, 'Isabela', NULL, 'isabela@totvs.com', '98765432000181', 'N/A'),
--> (9, 'Bento', NULL, 'bento.monteiro123@gmail.com', '21234567000172', 'N/A'),
--> (10, 'Augusto', NULL, 'augusto.silva456@yahoo.co', '87654321000163', 'N/A'),
--> (11, 'Isabelle', NULL, 'isabelle.lima789@outlook.com', '34567890000154', 'N/A'),
--> (12, 'Caio', NULL, 'caio.santos@gmail.com', '76543210000145', 'N/A'),
--> (13, 'João Silva', NULL, 'joao.silva@confeitariajulta.com', '45678901000136', 'N/A'),
--> (14, 'Jonathan Costa', NULL, 'jonathan.costa@mksab.com', '65432109000127', 'N/A'),
--> (15, 'Débora Almeida', NULL, 'debora.almeida@sortevedjudia.com', '56789012000118', 'N/A'),
--> (16, 'Rebeca Lima', NULL, 'rebeca.lima@crefisa.com', '43210987000109', 'N/A'),
--> (17, 'Carlos Pereira', NULL, 'carlos.pereira@bancojulia.com', '12345678000190', 'N/A'),
--> (18, 'Claudia Ribeiro', NULL, 'claudia.ribeiro@granel.com', '23456789000191', 'N/A'),
--> (19, 'Felipe Fonseca', NULL, 'felipe.fonseca@supermercadoabc.com', '34567890000192', 'N/A'),
--> (20, 'Rafaela Costa', NULL, 'rafaela.costa@mercado123.com', '45678901000193', 'N/A'),
--> (21, 'Bruno Almeida', NULL, 'bruno.almeida@lojaabc.com', '01234567000189', 'N/A'),
--> (22, 'Patrícia Silva', NULL, 'patricia.silva@petshopxyz.com', '12345678000190', 'N/A'),
--> (23, 'Mário Santos', NULL, 'mario.santos@cafe123.com', '23456789000191', 'N/A'),
--> (24, 'Silvia Costa', NULL, 'silvia.costa@florariaabc.com', '34567890000192', 'N/A'),
--> (25, 'Joana Darc', NULL, 'joana.darc@lanchonetec.com', '45678901000193', 'N/A'),
--> (26, 'Tiago Lima', NULL, 'tiago.lima@refrigerantes.com', '56789012000194', 'N/A'),
--> (27, 'Ana Oliveira', NULL, 'ana.oliveira@oficinajulta.com', '67890123000195', 'N/A'),
--> (28, 'Roberto Souza', NULL, 'roberto.souza@petshopabc.com', '78901234000196', 'N/A'),
--> (29, 'Carla Pereira', NULL, 'carla.pereira@consultoria.com', '89012345000197', 'N/A'),
--> (30, 'Letícia Campos', NULL, 'leticia.campos@cursosabc.com', '90123456000198', 'N/A'),
--> (31, 'Lucas Martins', NULL, 'lucas.martins@hoteisjulia.com', '01234567000180', 'N/A'),
--> (32, 'Paula Souza', NULL, 'paula.souza@livrariaabc.com', '12345678000181', 'N/A'),
--> (33, 'Patrícia Ramos', NULL, 'patricia.ramos@consultoria.com', '23456789000182', 'N/A'),
--> (34, 'Lucas Souza', NULL, 'lucas.souza@supermercadoabc.com', '34567890000183', 'N/A'),
--> (35, 'Fernanda Silva', NULL, 'fernanda.silva@mercado123.com', '45678901000184', 'N/A'),
--> (36, 'Júlio Costa', NULL, 'julio.costa@cafe123.com', '56789012000185', 'N/A'),
--> (37, 'Carla Almeida', NULL, 'carla.almeida@florariaabc.com', '67890123000186', 'N/A'),
--> (38, 'Ricardo Mendes', NULL, 'ricardo.mendes@refrigerantes.com', '78901234000187', 'N/A'),
--> (39, 'Vanessa Oliveira', NULL, 'vanessa.oliveira@livrariaabc.com', '89012345000188', 'N/A'),
--> (40, 'Marcelo Costa', NULL, 'marcelo.costa@hoteisjulia.com', '90123456000189', 'N/A');
Query OK, 39 rows affected (0.00 sec)
Records: 39 Duplicates: 0 Warnings: 0
```

#### D\_NOTA\_NPS

```
mysql> INSERT INTO d_nps Nota (id_nt_nps, nt_nps, cls_nps, nt_sistema, nt_exp_atualizacao, nt_sat_suporte, nt_ate_suporte, nt_agi_suporte, nt_exe_vendas, nt_custos, nt_financiero, nt_unidade) VALUES
--> (1, 9, 'Total', 9, 99, 7, 9, 6, 9, 7, 9, 8, NULL),
--> (2, 10, 'Total', 9, 97, 6, NULL, NULL, 9, 9, NULL),
--> (3, 8, 'Total', 6, 7, 8, NULL, 8, 8, 8, NULL),
--> (4, 1, 'Parcial', 8, 6, 5, 5, 8, 7, 7, NULL),
--> (5, 10, 'Total', 9, 9, 9, 8, 7, 9, 9, NULL),
--> (6, 10, 'Total', 9, 9, 9, 9, 9, 9, 9, 9, NULL),
--> (7, 9, 'Total', 8, 8, 8, 8, 8, 9, 9, 9, NULL),
--> (8, 4, 'Parcial', 8, 6, 7, 6, 7, 8, 6, 6, NULL),
--> (9, 9, 'Total', 9, 99, 7, 9, 6, 9, 8, 6, 6, NULL),
--> (10, 6, 'Parcial', 7, 6, 7, 8, 8, 8, 7, 7, NULL),
--> (11, 8, 'Total', 9, 9, 8, 7, 8, 6, 6, 6, NULL),
--> (12, 7, 'Parcial', 8, 6, 7, 6, 7, 7, 6, 6, NULL),
--> (13, 9, 'Total', 9, 99, 8, 9, 9, 9, 8, 7, 7, NULL),
--> (14, 6, 'Parcial', 7, 6, 6, 7, 6, 6, 5, 5, NULL),
--> (15, 8, 'Total', 9, 7, 8, 8, 8, 8, 8, 8, NULL);
Query OK, 15 rows affected (0.01 sec)
Records: 15 Duplicates: 0 Warnings: 0
```

## D\_SISTEMA

```
mysql> INSERT INTO d_sistema (id_sistema, nm_sistema, loc_unidade, nm_atendente, dur_chamada_min) VALUES
-> (1, 'Protheus', 'Totvs Minas Gerais', 'Viviane', '5:52'),
-> (2, 'Protheus', 'Totvs Brasilia', 'Beatriz', '2:57'),
-> (3, 'Protheus', 'Totvs Bauru', 'Lilian', '4:44'),
-> (4, 'Protheus', 'Totvs Serra do Mar', 'Beatriz', '1:54'),
-> (5, 'Protheus', 'Totvs Minas Gerais', 'Lilian', '3:08'),
-> (7, 'TOTVS HCM', 'Totvs Datasul', 'Beatriz', '5:42'),
-> (8, 'TOTVS HCM', 'Totvs Datasul', 'Beatriz', '3:10'),
-> (9, 'Protheus', 'Totvs Brasilia', 'Beatriz', '3:42'),
-> (10, 'TOTVS HCM', 'Totvs Minas Gerais', 'Lilian', '2:50'),
-> (11, 'Protheus', 'Totvs Serra do Mar', 'Lilian', '3:28'),
-> (12, 'Protheus', 'Totvs Serra do Mar', 'Viviane', '3:12'),
-> (13, 'TOTVS RM', 'Totvs Minas Gerais', 'Viviane', '3:56'),
-> (14, 'Protheus', 'Totvs Bauru', 'Viviane', '2:52'),
-> (15, 'TOTVS RM', 'Totvs Datasul', 'Viviane', '3:31'),
-> (16, 'TOTVS RM', 'Totvs Bauru', 'viviane', '3:30'),
-> (17, 'Protheus', 'Totvs Rio de Janeiro', 'Ana', '5:45'),
-> (18, 'Protheus', 'Totvs Belo Horizonte', 'Roberto', '6:00'),
-> (19, 'Fluig', 'Totvs Porto Alegre', 'Julia', '5:55'),
-> (20, 'Protheus', 'Totvs Curitiba', 'Paulo', '6:05'),
-> (21, 'Protheus', 'Totvs Recife', 'Marcela', '5:50'),
-> (22, 'TOTVS RM', 'Totvs Salvador', 'João', '6:10'),
-> (23, 'Fluig', 'Totvs Fortaleza', 'Clara', '6:00'),
-> (24, 'TOTVS RM', 'Totvs Brasilia', 'Sérgio', '5:50'),
-> (25, 'Protheus', 'Totvs Goiânia', 'Mariana', '6:20'),
-> (26, 'Fluig', 'Totvs Ibirapuera', 'Helena', '5:55'),
-> (27, 'Protheus', 'Totvs Recife', 'Marcos', '6:25'),
-> (28, 'Protheus', 'Totvs Ibirapuera', 'Julia', '6:10'),
-> (29, 'Fluig', 'Totvs Goiânia', 'Rafael', '6:30'),
-> (30, 'Protheus', 'Totvs Datasul', 'Mariana', '5:55'),
-> (31, 'TOTVS RM', 'Totvs Serra do Mar', 'Sérgio', '6:15'),
-> (32, 'Protheus', 'Totvs Datasul', 'Laura', '6:05'),
-> (33, 'TOTVS HCM', 'Totvs Serra do Mar', 'Bruno', '6:20'),
-> (34, 'Protheus', 'Totvs Datasul', 'Pedro', '5:55'),
-> (35, 'TOTVS RM', 'Totvs Datasul', 'Thiago', '6:30'),
-> (36, 'Fluig', 'Totvs Ibirapuera', 'Verusa', '6:00'),
-> (37, 'Protheus', 'Totvs Goiânia', 'Daniel', '6:25'),
-> (38, 'TOTVS HCM', 'Totvs Salvador', 'Bruno', '5:50'),
-> (39, 'Protheus', 'Totvs Brasilia', 'Fernanda', '6:05'),
-> (40, 'Fluig', 'Totvs Ibirapuera', 'Fernanda', '6:05');
Query OK, 39 rows affected (0.00 sec)
Records: 39  Duplicates: 0  Warnings: 0
```

## FATO\_NPS

```
mysql> INSERT INTO fato_nps (
->     d_nps_nota_sk_nps_nota,
->     d_data_sk_data,
->     d_cliente_sk_cliente,
->     d_sistema_sk_sistema,
->     prct_promotor,
->     prct_detratr,
->     prct_neutro,
->     dt_hra_inicial,
->     dt_hra_final,
->     tx_abandono,
->     tmp_atendimento,
->     tmp_espera,
->     rkn_rkn_sistema,
->     qtd_lig_promotor,
->     qtd_lig_neutro,
->     qtd_lig_detratr
-> )
-> SELECT
->     n.sk_nps_nota,
->     d.id_dia,
->     c.sk_cliente,
->     s.sk_sistema,
->     (SELECT AVG(nt.nps) FROM d_nps_nota WHERE cls_nps = 'Promotor') AS prct_promotor,
->     (SELECT AVG(nt.nps) FROM d_nps_nota WHERE cls_nps = 'Detratr') AS prct_detratr,
->     (SELECT AVG(nt.nps) FROM d_nps_nota WHERE cls_nps = 'Neutro') AS prct_neutro,
->     NOW() AS dt_hra_inicial,
->     NOW() AS dt_hra_final,
->     (SELECT COUNT(*) FROM d_nps_nota WHERE cls_nps = 'Detratr') /
->     (SELECT COUNT(*) FROM d_nps_nota WHERE cls_nps = 'Promotor') AS tx_abandono,
->     (SELECT COUNT(*) FROM d_nps_nota) AS tmp_atendimento,
->     s.dur_chamada_min AS tmp_espera,
->     rnk.rkn AS rkn_sistema,
->     (SELECT COUNT(*) FROM d_nps_nota WHERE cls_nps = 'Promotor') AS qtd_lig_promotor,
->     (SELECT COUNT(*) FROM d_nps_nota WHERE cls_nps = 'Neutro') AS qtd_lig_neutro,
->     (SELECT COUNT(*) FROM d_nps_nota WHERE cls_nps = 'Detratr') AS qtd_lig_detratr
->     FROM
->     d_nps_nota n
->     JOIN
->     d_data d ON n.id_nt_nps = d.nr_ano
->     JOIN
->     d_cliente c ON i=1 -- Isso conta todos os clientes
->     JOIN
->     d_sistema s ON i=1 -- Isso conta todos os sistemas
->     JOIN (
->         SELECT
->             sk_nps_nota,
->             ROW_NUMBER() OVER (ORDER BY nt.nps DESC) AS rnk
->             FROM
->             d_nps_nota
->             ) rnk ON rnk.sk_nps_nota = n.sk_nps_nota
->     WHERE
->     n.id_nt_nps IS NOT NULL;
Query OK, 0 rows affected (0.00 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

## Consultas em todas as tabelas do Data Warehouse

### D\_DATA

```
mysql> select * from d_data;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| sk_data | nr_ano | nr_mes | ds_mes | ds_mes_ano | ds_mes_ano_abv | ds_mes_abv | nr_semana | ds_nr_semana | dat_ref | dat_extenso | ds_dia_semana | nr_dia_mes |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 9 | 2023 | 12 | Dezembro | Dezembro 2023 | Dez | | 1 | Semana 1 | 2023-12-03 | Terceiro dia de Dezembro de 2023 | Domingo | 3
| 10 | 2023 | 7 | Junho | Junho 2023 | Jun | | 1 | Semana 1 | 2023-06-07 | Sétimo dia de Junho de 2023 | Domingo | 7
| 11 | 2023 | 11 | Novembro | Novembro 2023 | Nov | | 1 | Semana 1 | 2023-11-23 | Vigésimo terceiro dia de Novembro de 2023 | Quinta-feira | 23
| 12 | 2023 | 2 | Fevereiro | Fevereiro 2023 | Fev | | 1 | Semana 1 | 2023-02-14 | Quatorze dias de Fevereiro de 2023 | Terça-feira | 14
| 13 | 2023 | 9 | Setembro | Setembro 2023 | Set | | 1 | Semana 1 | 2023-09-09 | Nono dia de Setembro de 2023 | Sábado | 9
| 14 | 2023 | 6 | Março | Março 2023 | Mar | | 1 | Semana 1 | 2023-03-18 | Dezoito dias de Março de 2023 | Domingo | 18
| 15 | 2023 | 4 | Abril | Abril 2023 | Abr | | 1 | Semana 1 | 2023-04-30 | Trinta dias de Abril de 2023 | Domingo | 30
| 16 | 2023 | 1 | Janeiro | Janeiro 2023 | Jan | | 1 | Semana 1 | 2023-01-02 | Dois dias de Janeiro de 2023 | Segunda-feira | 2
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

### D\_CLIENTE

```
mysql> select * from d_cliente;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| sk_cliente | id_cliente | nm_cliente | nr_telefone | nm_email | nr_cnpj | nm_empresa |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 40 | 1 | Geraldo | NULL | geraldo.darwin@bosel.com | 4544 | N/A
| 41 | 2 | Ana Carla | NULL | anacarla.pereira@amelia.com.br | 07354754000199 | N/A
| 42 | 3 | Claudio | NULL | N/A | 000200 | N/A
| 43 | 4 | Daniele Melo | NULL | compras@betar.com.br | 02315905000186 | N/A
| 44 | 5 | Gabriel | NULL | ti@resendesantos.com.br | 000127 | N/A
| 45 | 7 | Vitória | NULL | vitoria@hotmail.com | 12345678000190 | N/A
| 46 | 8 | Isabela | NULL | isabela@totvs.com | 98765432000181 | N/A
| 47 | 9 | Bento | NULL | bento.monteiro123@gmail.com | 21234567000172 | N/A
| 48 | 10 | Augusto | NULL | augusto.silva456@yahoo.co | 87654321000163 | N/A
| 49 | 11 | Isabelle | NULL | isabelle.lima7890@outlook.com | 34567890000154 | N/A
| 50 | 12 | Caió | NULL | caio.santos@gmail.com | 76543210000145 | N/A
| 51 | 13 | João Silva | NULL | joao.silva@confeitariajulia.com | 45678901000136 | N/A
| 52 | 14 | Jonathan Costa | NULL | jonathan.costa@mkssab.com | 65432109000127 | N/A
| 53 | 15 | Débora Almeida | NULL | debora.almeida@sortevesjudia.com | 56789012000118 | N/A
| 54 | 16 | Rebeca Lima | NULL | rebeca.lima@refreisaca.com | 43210987000109 | N/A
| 55 | 17 | Carlos Pereira | NULL | carlos.pereira@bancojulia.com | 123456780001-90 | N/A
| 56 | 18 | Claudia Ribeiro | NULL | claudia.ribeiro@granel.com | 23456789000191 | N/A
| 57 | 19 | Felipe Fonseca | NULL | felipe.fonseca@supermercadoabc.com | 34567890000192 | N/A
| 58 | 20 | Rafaela Costa | NULL | rafaela.costa@mercadol23.com | 45678901000193 | N/A
| 59 | 21 | Bruno Almeida | NULL | bruno.almeida@lojaabc.com | 01234567000189 | N/A
| 60 | 22 | Patrícia Silva | NULL | patricia.silva@petshopxyz.com | 12345678000190 | N/A
| 61 | 23 | Mário Santos | NULL | mario.santos@cafe123.com | 23456789000191 | N/A
| 62 | 24 | Silvia Costa | NULL | silvia.costa@florariaabc.com | 34567890000192 | N/A
| 63 | 25 | Joana Darc | NULL | joana.darc@lanchonetec.com | 45678901000193 | N/A
| 64 | 26 | Tiago Lima | NULL | tiago.lima@refrigerantes.com | 56789012000194 | N/A
| 65 | 27 | Ana Oliveira | NULL | ana.oliveira@oficinajulia.com | 67890123000195 | N/A
| 66 | 28 | Roberto Souza | NULL | roberto.souza@petshopabc.com | 78901234000196 | N/A
| 67 | 29 | Carla Pereira | NULL | carla.pereira@consultoria.com | 89012345000197 | N/A
| 68 | 30 | Letícia Campos | NULL | leticia.campos@cursoساب.com | 90123456000198 | N/A
| 69 | 31 | Lucas Martins | NULL | lucas.martins@hoteisjulia.com | 01234567000180 | N/A
| 70 | 32 | Paula Souza | NULL | paula.souza@livrariaabc.com | 12345678000181 | N/A
| 71 | 33 | Patrícia Ramos | NULL | patricia.ramos@consultoria.com | 23456789000182 | N/A
| 72 | 34 | Lucas Souza | NULL | lucas.souza@supermercadoabc.com | 34567890000183 | N/A
| 73 | 35 | Fernanda Silva | NULL | fernanda.silva@mercadol23.com | 45678901000184 | N/A
| 74 | 36 | Júlio Costa | NULL | julio.costa@cafe123.com | 56789012000185 | N/A
| 75 | 37 | Carla Almeida | NULL | carla.almeida@florariaabc.com | 67890123000186 | N/A
| 76 | 38 | Ricardo Mendes | NULL | ricardo.mendes@refrigerantes.com | 78901234000187 | N/A
| 77 | 39 | Vanessa Oliveira | NULL | vanessa.oliveira@livrariaabc.com | 89012345000188 | N/A
| 78 | 40 | Marcelo Costa | NULL | marcelo.costa@hoteisjulia.com | 90123456000189 | N/A
+-----+-----+-----+-----+-----+-----+-----+-----+
39 rows in set (0.00 sec)
```

### D\_NPS\_NOTA

```
mysql> select * from d_npsNota;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| sk_nps nota | id_nt_nps | nt_nps | cls_nps | nt_sistema | nt_exp_atualizacao | nt_sat_suporte | nt_ate_suporte | nt_agt_suporte | nt_exe_vendas | nt_custo | nt_financiamento | nt_unidade |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
16	1	9	Total	0.99	7.00	9.00	8.00	0.00	7.00	9.00	8.00	7.60	
17	2	10	Total	0.99	7.00	9.00	8.00	0.00	7.00	9.00	8.00	7.60	
18	3	8	Total	0.00	7.00	8.00	8.00	0.00	7.00	9.00	8.00	7.60	
19	4	1	Parcial	0.00	6.00	5.00	5.00	5.00	6.00	7.00	7.00	7.60	
20	5	10	Total	0.00	9.00	9.00	8.00	7.00	9.00	9.00	9.00	7.60	
21	6	10	Total	0.00	9.00	9.00	9.00	9.00	9.00	9.00	9.00	7.60	
22	7	10	Total	0.00	8.00	8.00	8.00	8.00	8.00	9.00	9.00	7.60	
23	8	4	1	Parcial	0.00	6.00	7.00	6.00	7.00	8.00	9.00	9.00	7.60
24	9	9	Total	0.99	7.00	9.00	9.00	9.00	8.00	6.00	6.00	6.00	7.60
25	10	6	Parcial	7.00	6.00	7.00	8.00	7.00	8.00	7.00	7.00	7.60	
26	11	8	Total	0.00	7.00	8.00	7.00	8.00	6.00	6.00	6.00	7.60	
27	12	7	Parcial	8.00	6.00	7.00	6.00	7.00	7.00	6.00	6.00	7.60	
28	13	9	Total	0.99	6.00	9.00	9.00	9.00	8.00	7.00	7.00	7.60	
29	14	6	Parcial	7.00	6.00	7.00	6.00	6.00	5.00	5.00	5.00	7.60	
30	15	8	Total	0.00	7.00	8.00	8.00	8.00	8.00	8.00	8.00	7.60	
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
15 rows in set (0.00 sec)
```

## D\_SISTEMA

| sk_sistema | id_sistema | nm_sistema | loc_unidade          | nm_atendente | dur_chamada_min |
|------------|------------|------------|----------------------|--------------|-----------------|
| 1          | 1          | Protheus   | Totvs Minas Gerais   | Viviane      | 5:52            |
| 2          | 2          | Protheus   | Totvs Brasília       | Beatriz      | 2:57            |
| 3          | 3          | Protheus   | Totvs Bauru          | Lilian       | 4:44            |
| 4          | 4          | Protheus   | Totvs Serra do Mar   | Beatriz      | 1:54            |
| 5          | 5          | Protheus   | Totvs Minas Gerais   | Lilian       | 3:08            |
| 6          | 7          | TOTVS HCM  | Totvs Datasul        | Beatriz      | 5:42            |
| 7          | 8          | TOTVS HCM  | Totvs Datasul        | Beatriz      | 3:10            |
| 8          | 9          | Protheus   | Totvs Brasília       | Beatriz      | 3:42            |
| 9          | 10         | TOTVS HCM  | Totvs Minas Gerais   | Lilian       | 2:50            |
| 10         | 11         | Protheus   | Totvs Serra do Mar   | Lilian       | 3:28            |
| 11         | 12         | Protheus   | Totvs Serra do Mar   | Viviane      | 3:12            |
| 12         | 13         | TOTVS RM   | Totvs Minas Gerais   | Viviane      | 3:56            |
| 13         | 14         | Protheus   | Totvs Bauru          | Viviane      | 2:52            |
| 14         | 15         | TOTVS RM   | Totvs Datasul        | Viviane      | 3:31            |
| 15         | 16         | TOTVS RM   | Totvs Bauru          | Viviane      | 3:30            |
| 16         | 17         | Protheus   | Totvs Rio de Janeiro | Ana          | 5:45            |
| 17         | 18         | Protheus   | Totvs Belo Horizonte | Roberto      | 6:00            |
| 18         | 19         | Fluig      | Totvs Porto Alegre   | Julia        | 5:55            |
| 19         | 20         | Protheus   | Totvs Curitiba       | Paulo        | 6:05            |
| 20         | 21         | Protheus   | Totvs Recife         | Marcela      | 5:50            |
| 21         | 22         | TOTVS RM   | Totvs Salvador       | João         | 6:10            |
| 22         | 23         | Fluig      | Totvs Fortaleza      | Clara        | 6:00            |
| 23         | 24         | TOTVS RM   | Totvs Brasília       | Sérgio       | 5:50            |
| 24         | 25         | Protheus   | Totvs Goiânia        | Mariana      | 6:20            |
| 25         | 26         | Fluig      | Totvs Ibirapuera     | Helena       | 5:55            |
| 26         | 27         | Protheus   | Totvs Recife         | Marcos       | 6:25            |
| 27         | 28         | Protheus   | Totvs Ibirapuera     | Julia        | 6:10            |
| 28         | 29         | Fluig      | Totvs Goiânia        | Rafael       | 6:30            |
| 29         | 30         | Protheus   | Totvs Datasul        | Mariana      | 5:55            |
| 30         | 31         | TOTVS RM   | Totvs Serra do Mar   | Sérgio       | 6:15            |
| 31         | 32         | Protheus   | Totvs Datasul        | Laura        | 6:05            |
| 32         | 33         | TOTVS HCM  | Totvs Serra do Mar   | Bruno        | 6:20            |
| 33         | 34         | Protheus   | Totvs Datasul        | Pedro        | 5:55            |
| 34         | 35         | TOTVS RM   | Totvs Datasul        | Thiago       | 6:30            |
| 35         | 36         | Fluig      | Totvs Ibirapuera     | Verusa       | 6:00            |
| 36         | 37         | Protheus   | Totvs Goiânia        | Daniel       | 6:25            |
| 37         | 38         | TOTVS HCM  | Totvs Salvador       | Bruno        | 5:50            |
| 38         | 39         | Protheus   | Totvs Brasília       | Fernanda     | 6:05            |
| 39         | 40         | Fluig      | Totvs Ibirapuera     | Fernanda     | 6:05            |

FATO\_NPS

## 10.4 BACKUP NO DATA WAREHOUSE MYSQL

No Data Warehouse do InoVoice, é crucial seguir boas práticas de backup e recovery, uma vez que o MySQL oferece diversas ferramentas para garantir a proteção dos dados. Neste caso, será utilizado o mysqldump, que é uma das práticas mais conhecidas no mercado. Essa ferramenta permite exportar bancos de dados inteiros ou partes deles para arquivos SQL, sendo uma ótima escolha para a otimização de custos.

Escolhemos essa estratégia de backup por ela ser flexível, podendo ser completa, incremental ou diferencial, dependendo das características dos dados e dos requisitos de recuperação. Dessa forma, podemos adaptá-la tanto para este DW quanto para futuros DWs de NPS. Acreditamos que a combinação de diferentes estratégias pode ser necessária para atender a diferentes situações.

Portanto, definimos uma política de retenção para os backups, garantindo a segurança desses arquivos, protegendo-os contra perdas, danos e acessos não autorizados. Além disso, utilizaremos a replicação como uma aliada estratégica importante para garantir alta disponibilidade, assegurando que os dados sejam replicados em tempo real para outros servidores quando necessário.

## 11. AMBIENTE DE BUSINESS INTELLIGENCE

Abaixo segue o link com o Power BI publicado:

<https://app.powerbi.com/view?r=eyJrIjoiMjQxMWU2MmQtYTMwYi00N2U2LWE5ZTgtY2YwYTY1OWJhMjVhliwidCI6IjExZGJiZmUyLTg5YjgtNDU0OS1iZTEwLWNIYzM2NGU1OTU1MSIsImMiOjR9>

### 1. Público Alvo

O público alvo do InoVoice é composto por coordenadores de áreas técnicas e de negócio, gerentes e diretores da TOTVS, como por exemplo a persona da Cassia Dutra, pois são profissionais que lideram equipes das áreas que consomem dados de NPS para melhorar os negócios da TOTVS, alguns exemplos das áreas são CX (Customer Experience), que trata de sucesso do cliente, Marketing, que cuida de toda parte estratégica online e offline, Vendas e Atendimento e estão diretamente envolvidos na melhoria da satisfação dos clientes. Eles buscam soluções estratégicas e tecnológicas para otimizar a performance de suas equipes, valorizando dashboards que ofereçam insights claros e objetivos, facilitando a visualização e a interpretação dos dados de NPS e outras métricas de desempenho. Esse público é caracterizado por ser analítico, engajado e orientado à melhoria contínua, enfrentando desafios com a obtenção de informações comprehensíveis e confiáveis para a tomada de decisões. Além disso, eles utilizam ferramentas como WhatsApp, Power BI e Office 365 no dia a dia, e buscam soluções inovadoras que contribuam para elevar o nível de atendimento ao cliente, maximizar a retenção e melhorar a eficiência operacional da equipe.



**Cassia Dutra**

**Idade:** 45 Anos  
**Endereço:** Av. Alegre  
**Ocupação:** Gerente de CX

**NECESSIDADES**

- Informações claras e objetivas;
- Valoriza Dashboards que tragam insights de forma clara e estratégica
- Precisa tomar decisões com base nos insights do NPS para melhorar a sua equipe de atendimento;

**INTERESSES**

- Comunicação empresarial
- Estratégia de negócios
- Gestão de pessoas

**PAIN POINTS**

- Dificuldade em acessar informações do NPS e comprehensíveis devido à falta de fontes confiáveis;
- Falta de compreensão em dashboards com dados que lhe auxiliam para tomar decisões;
- Frustração com processos de melhoria do atendimento da TOTVS devido a dificuldade em acompanhar a performance dos atendimentos..

**MOTIVAÇÕES**

- Analítica
- Enagajada
- Curiosa
- Crítica

*"Os dados revelam as necessidades dos nossos clientes e ajudam a melhorar a experiência. Dados é Solução!"  
Cassia Dutra*

## 2. Dados/Metadados

| COLUNAS                      | DESCRIÇÃO  |
|------------------------------|--|
| ID                           | Chave primária da tabela                           |
| Arquivo                      | Nome do arquivo transcrita .wav                    |
| Email                        | Email do representante                             |
| Possui CNPJ                  | Cliente possuir ou não CNPJ                        |
| Nome Atendente               | Nome do atendente realizando a ligação             |
| Nota Recomendação            | Nota de 0-10 para recomendação da TOTVS            |
| Nota Agilidade Suporte       | Nota de 0-10 da agilidade do suporte               |
| Atendimento Executivo Vendas | Nota de 0-10 do atendimento do executivo de vendas |
| Nota Custo Produto           | Nota de 0-10 do custo do produto                   |
| Nota Atendimento Financeiro  | Nota de 0-10 do atendimento financeiro             |
| Nota Sistema                 | Aprovação do Sistema                               |
| Nota Exp Atualizacao         | Nota de 0-10 da experiência de atualização         |
| Nota Unidade                 | Nota de 0-10 da Unidade TOTVS                      |
| Nota NPS                     | Nota de 0-10 para definir a classificação NPS      |
| Classificação                | Classificação NPS: Promotor, Neutro e Detrator     |
| Localizacao Unidade          | Nome da Unidade TOTVS e sua localização            |
| Produto                      | Nome do produto avaliado na ligação                |
| Qtde Palavras                | Total de palavras na transcrição                   |
| Palavras por minuto          | Número de Palavras por minuto                      |
| Duracao Chamada              | Duração da chamada em formato hh:mm:ss             |
| Duracao Chamada min          | Duração da chamada em minutos                      |
| Data Audio                   | Data em que o áudio foi registrado                 |

### 3. Métricas

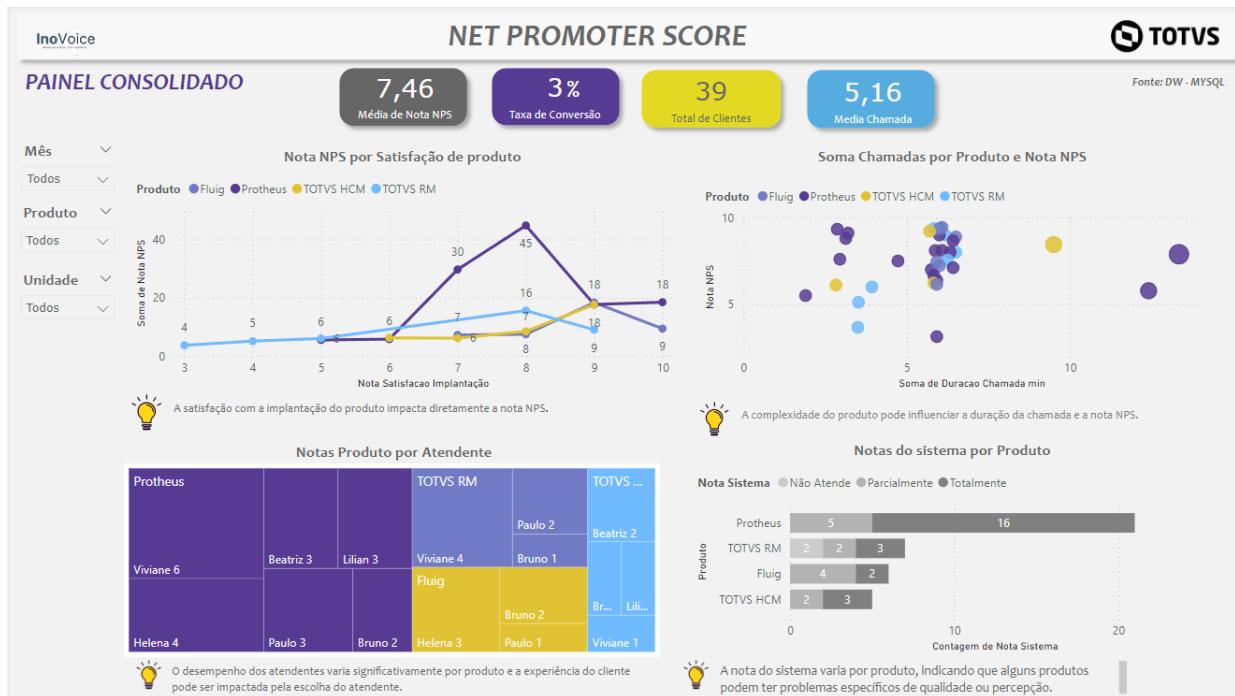
- **Número das ligações processadas por dia:** A quantidade de ligações transcritas e analisadas diariamente.
- **Quantidade de detratores, neutros e promotores:** Utilizado para verificação da distribuição das respostas dos clientes TOTVS.
- **Volume de Dados Gerados:** Quantidade de texto gerado e insights extraídos das transcrições.
- **Feedback de Uso Interno:** Avaliações qualitativas de como os insights extraídos estão ajudando a empresa a melhorar sua abordagem com os clientes detratores e identificando padrões e tendências nas respostas.

### 4. KPI's

- **Tempo de Processamento:** O tempo médio gasto para processar e transcrever uma ligação.
- **Taxa de Erro de Transcrição:** A proporção de erros nas transcrições (ex. palavras incorretas, falhas devido a sotaques ou ruídos).
- **Análise de Sentimento:** O percentual de ligações corretamente classificadas de acordo com o sentimento.
- **Classificação dos relatos:** A eficácia com que as ligações são categorizadas em seus motivos principais (ex. tipo de problema relatado).
- **Score NPS:** Faz a distribuição das respostas e como o NPS muda ao longo do tempo, oferecendo uma visão mais completa da satisfação do cliente

## 5. Prints Dashboards

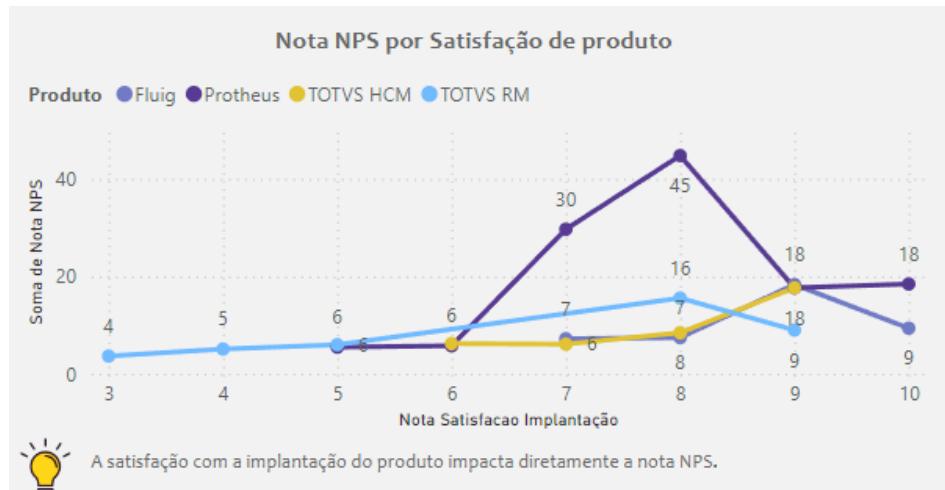
### Painel 1: Visão Consolidada



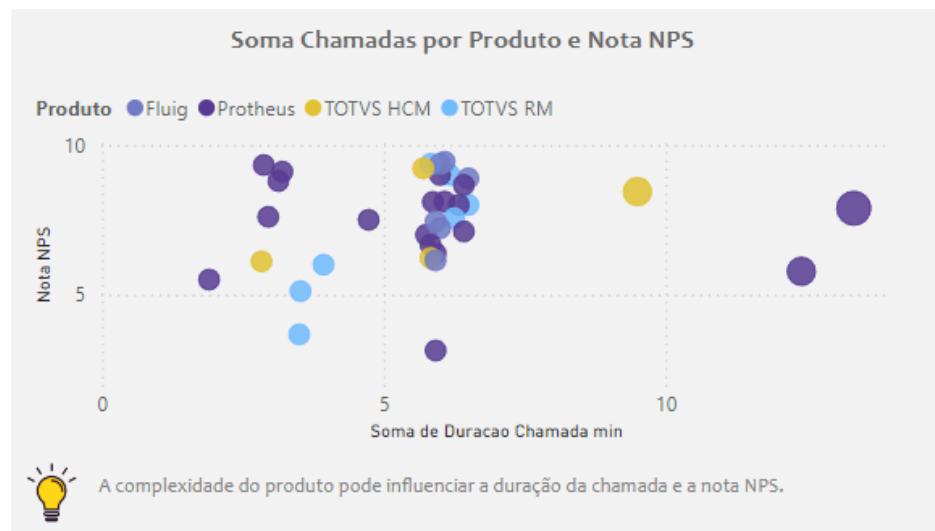
- Indicadores:** Usamos para destaque dos números e métricas, com o propósito de chamar a atenção do usuário. Esses indicadores mostram números rápidos e úteis para a análise.



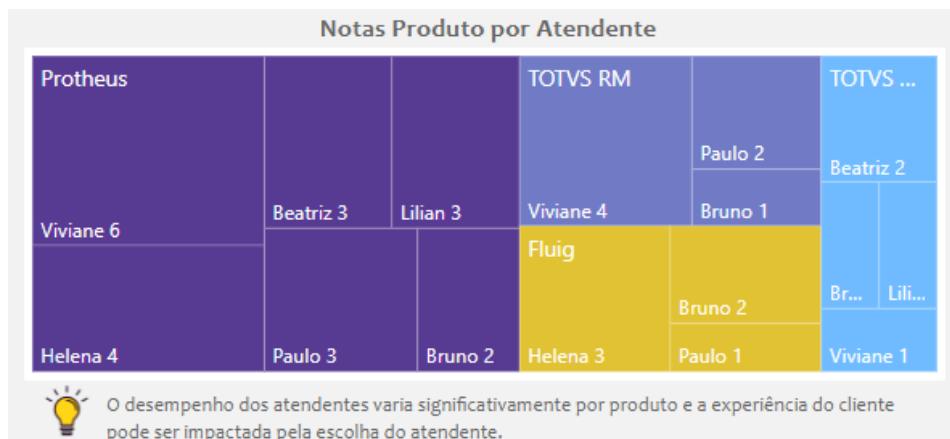
- Gráfico de linhas:** Nessa gráfico, queremos identificar quais produtos receberam melhor avaliação dentro do contexto de nota de Satisfação da Implantação do Software e da Nota NPS.



- **Gráfico de dispersão:** Gráfico de dispersão para verificar quais produtos estão mais relacionados com as ligações, ou seja, quais produtos foram mais avaliados pelos clientes nas ligações de NPS.



- **Tree map:** Tem o propósito de comparar as notas dos produtos por cada atendente da TOTVS. Isso só reforça o gráfico anterior, de que o Protheus foi o produto mais observado durante as entrevistas.



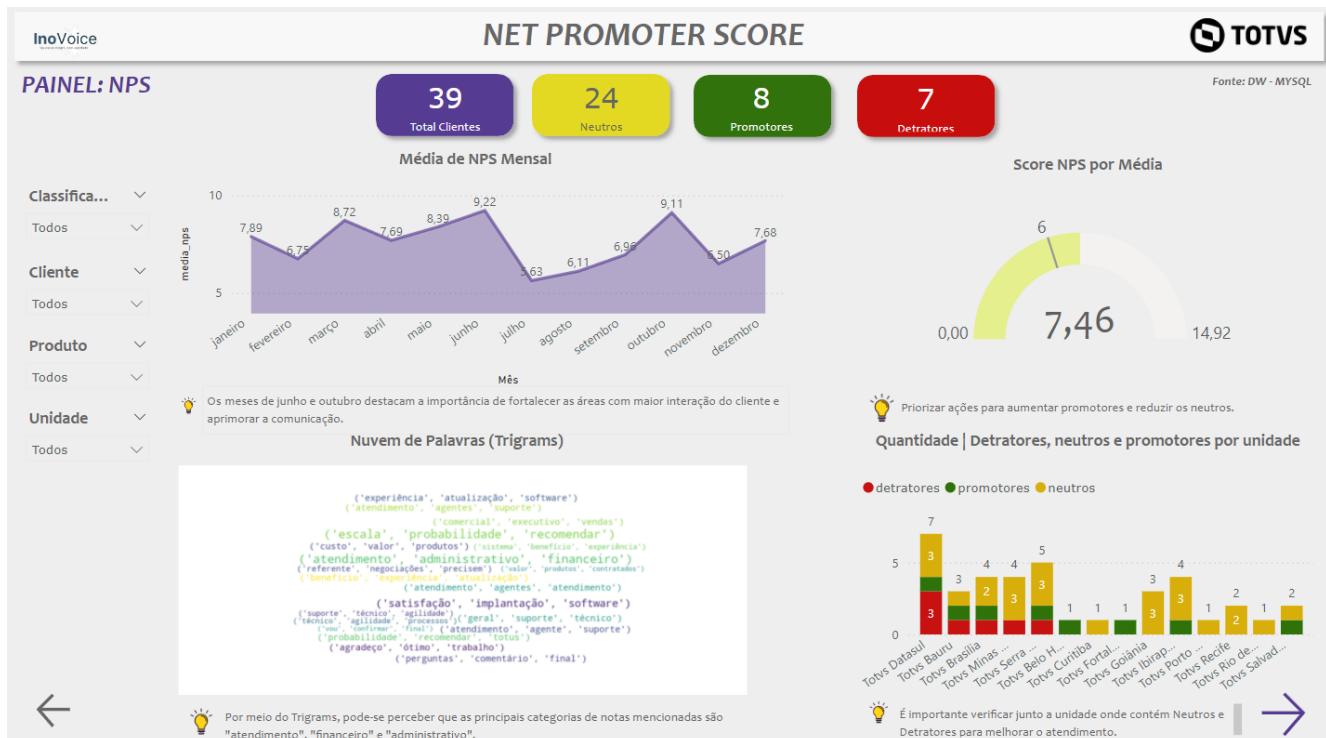
- **Gráfico de Barras Empilhadas:** Utilizamos para destacar se os produtos atendem ou não as necessidades dos clientes.



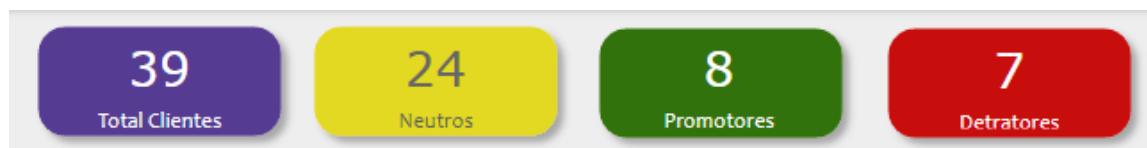
- **Segmentação:** Filtros/Segmentações por Mês, Produto e Unidade



**Painel 2 - Visão NPS:** Esse painel tem como objetivo gerar visão sobre como os clientes de fato foram classificados dentro da abordagem NPS, verificando como está a aprovação dos produtos TOTVS dentro da carteira de clientes.



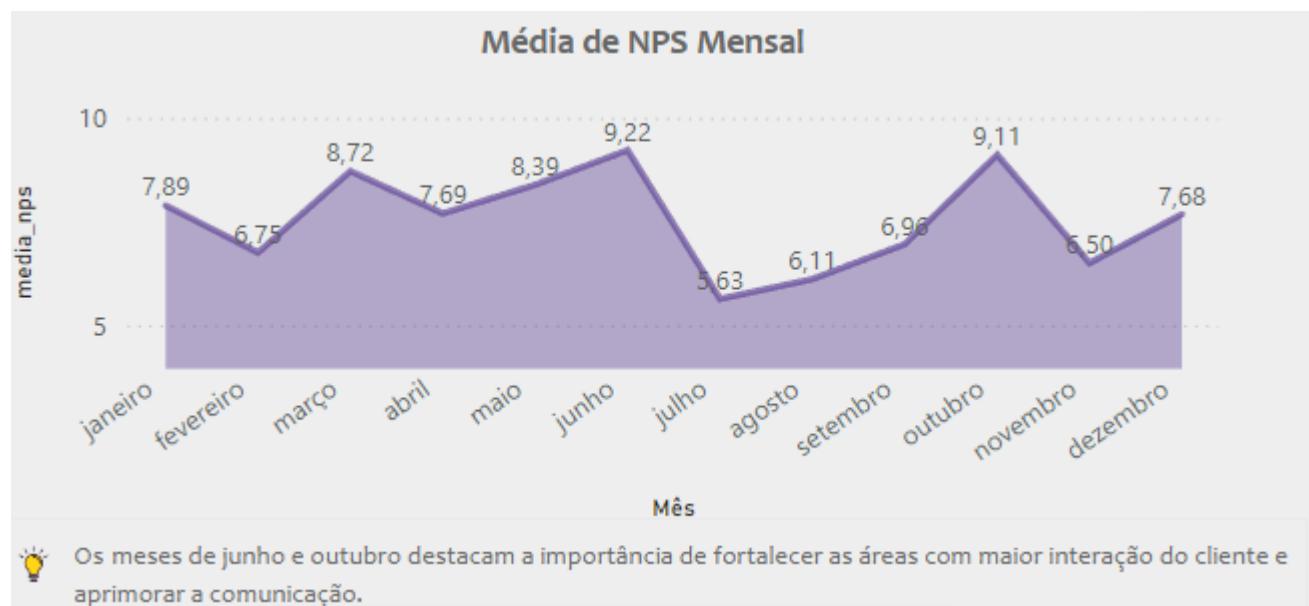
- Indicadores:** Usamos para destaque dos números e métricas, com o propósito de chamar a atenção do usuário. Esses indicadores mostram números rápidos e úteis para a análise.



- Gráfico de Indicador:** Utilizamos com o intuito de demonstrar onde se encontra a média de notas NPS, considerando o mínimo 6. Em outras palavras, a média de nota NPS indica que a maioria dos clientes são neutros.



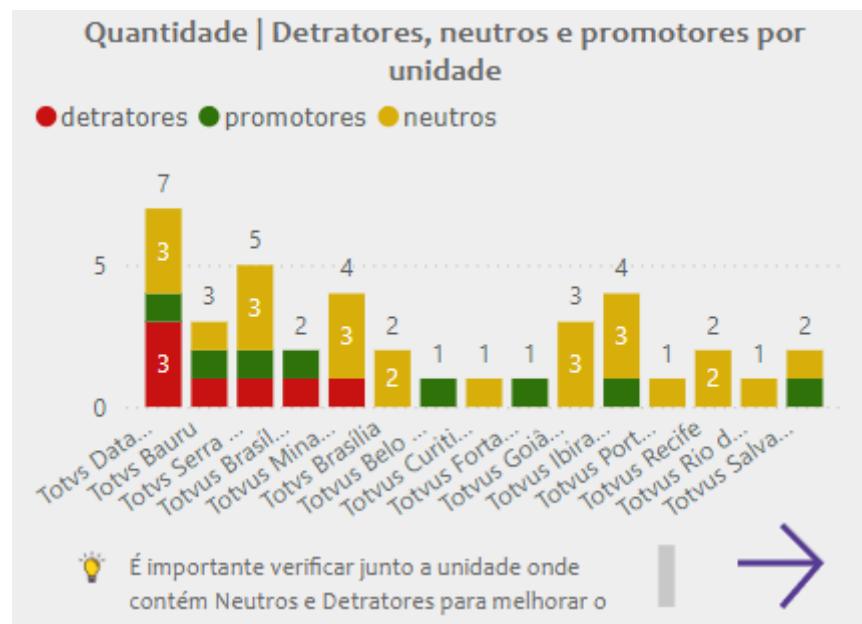
- **Gráfico de Área:** Gráfico para observar a média de Nota NPS ao longo dos meses.



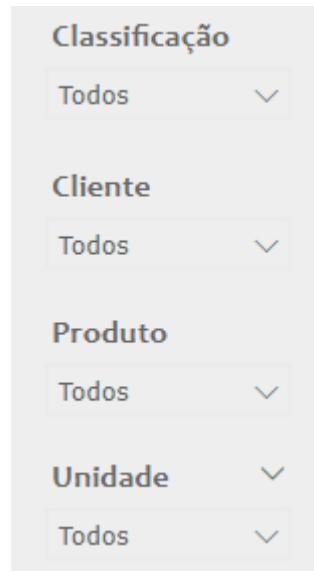
- **Nuvem de Palavras:** Nuvem de palavras contendo Ngrams para melhor performance e avaliação de sentenças, no caso, foi considerado Trigrams. (Código na próxima sessão dessa documentação e anexado na pasta .zip deste entregável)



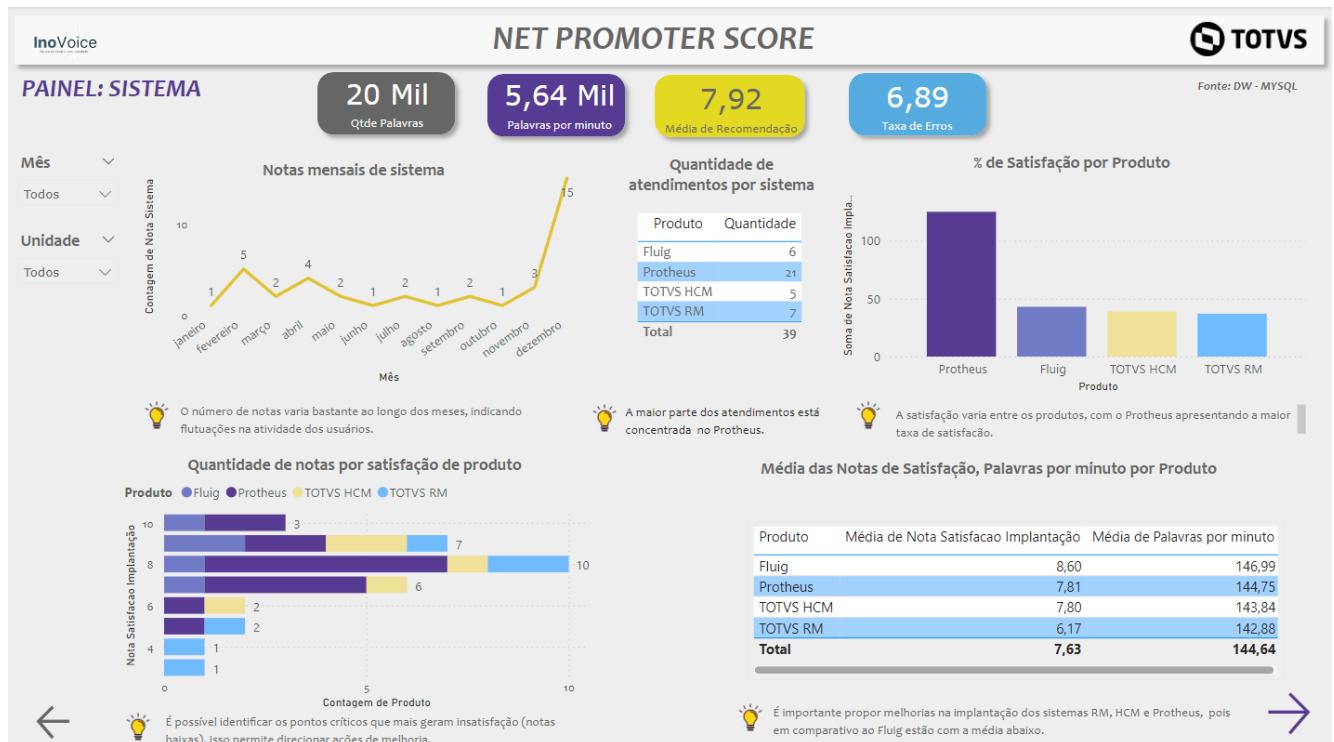
- **Gráfico de Colunas:** O objetivo deste gráfico é mostrar a quantidade de clientes detratores, neutros e promotores por unidade TOTVS.



- **Segmentação:** Filtros/Segmentações por Mês, Cliente, Produto e Unidade



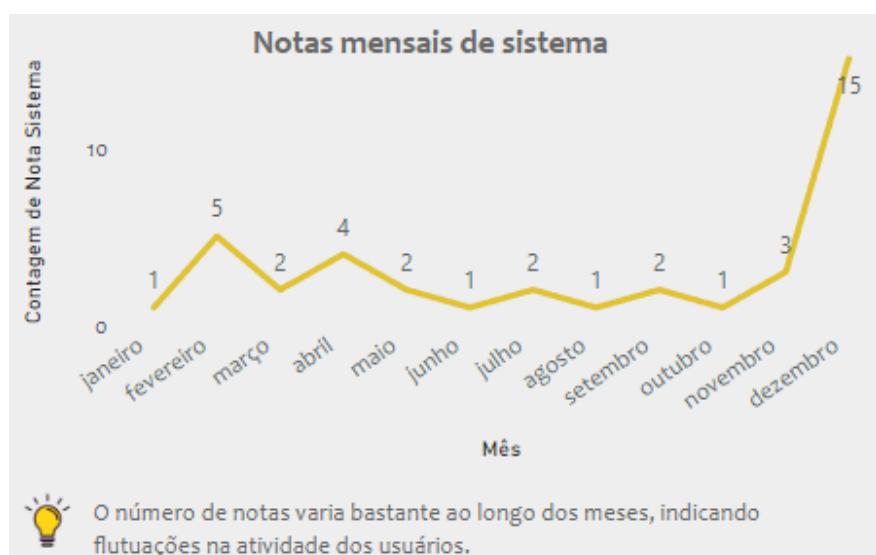
**Painel 3 - Visão Sistema:** Esse painel tem como objetivo auxiliar os times de desenvolvimento de softwares, tecnologia e até mesmo comercial



- Indicadores:** Usamos para destaque dos números e métricas, com o propósito de chamar a atenção do usuário. Esses indicadores mostram números rápidos e úteis para a análise.



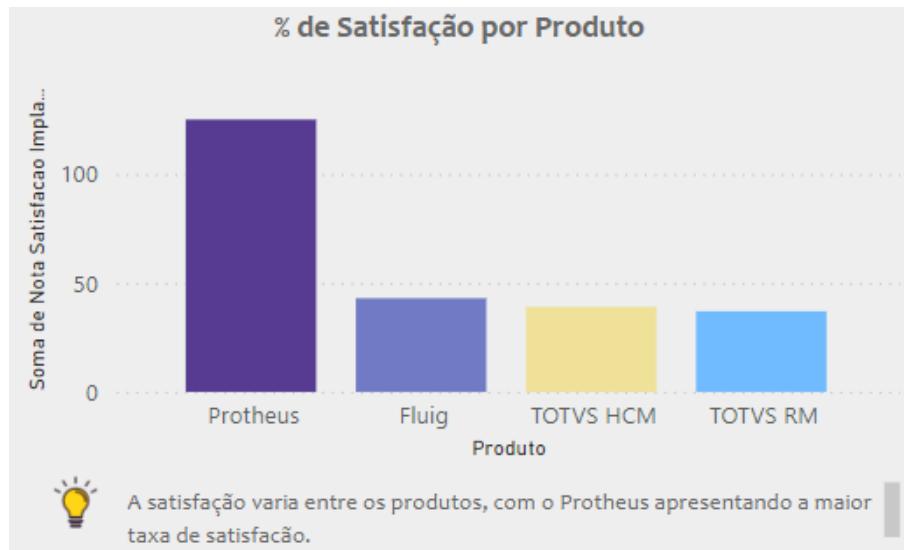
- Gráfico de linhas:** Esse gráfico demonstra a Nota de Sistema ao longo dos meses. Essa nota, é uma das que compõem a nota NPS ao final



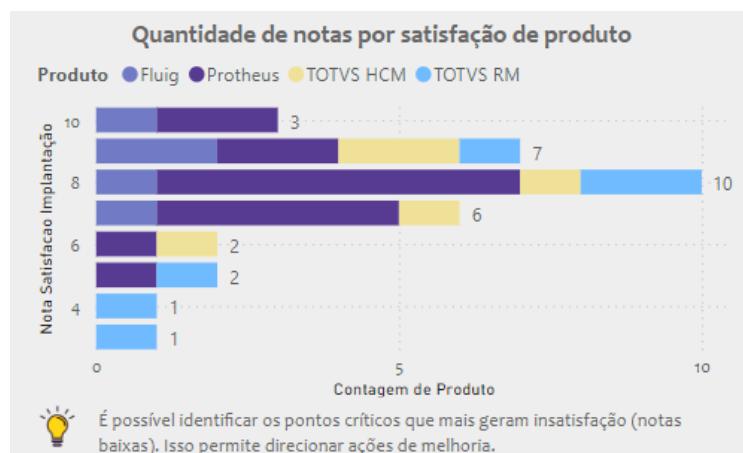
- **Tabela:** Tabela para demonstrar e ranquear os produtos TOTVS que mais foram mencionados e avaliados nas ligações.

| Quantidade de atendimentos por sistema |            |
|--|------------|
| Produto                                | Quantidade |
| Fluig                                  | 6          |
| Protheus                               | 21         |
| TOTVS HCM                              | 5          |
| TOTVS RM                               | 7          |
| Total                                  | 39         |

- **Gráfico de Barras:** Gráfico para demonstrar o percentual de satisfação com o produto, ou seja, percentual de clientes que avaliaram na entrevista que o produto atende totalmente às suas necessidades.



- **Gráfico de Barras Empilhadas:** Gráfico para demonstrar a quantidade de Notas de Satisfação de Implantação por produto TOTVS.

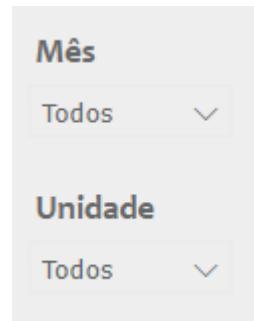


- **Gráfico de Tabela:** Gráfico para demonstrar a relação entre Média de Palavras por Minuto e Média de Nota de Satisfação.

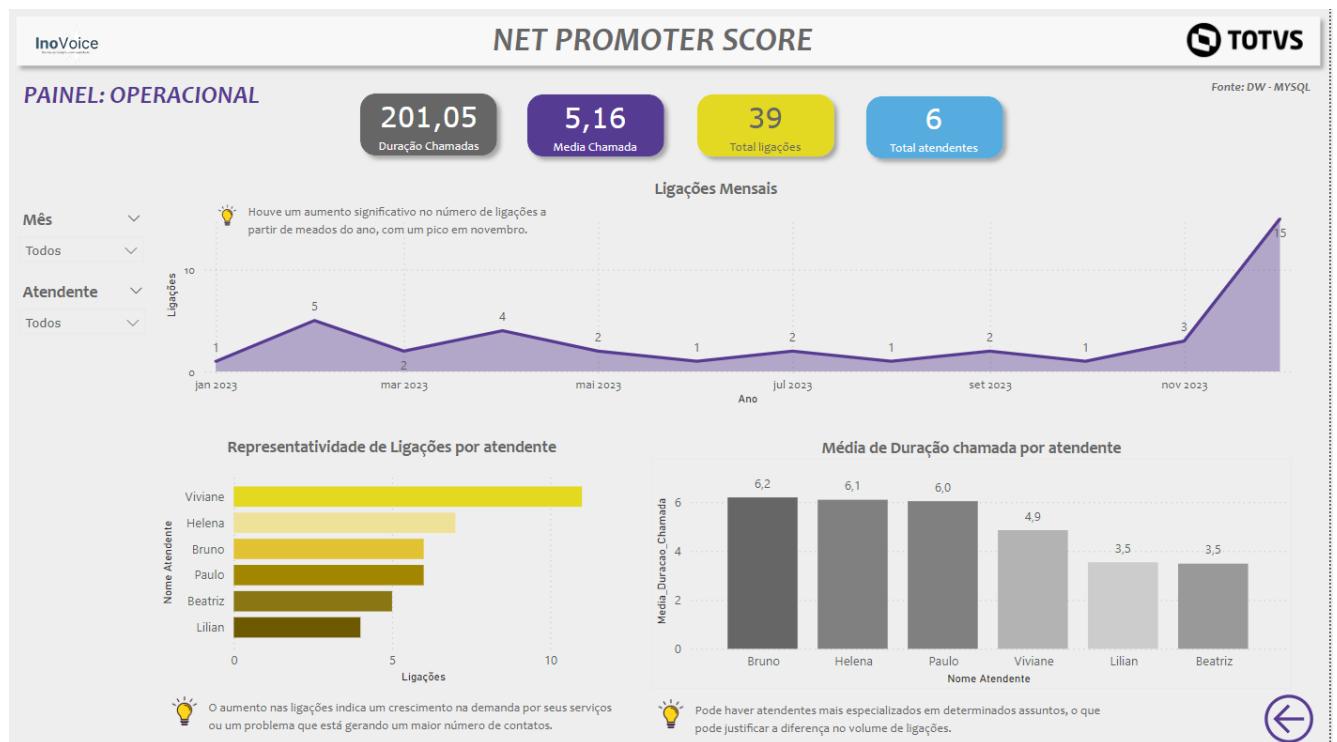
| Produto      | Média de Nota Satisfacao | Implantação | Média de Palavras por minuto |
|--------------|--------------------------|-------------|------------------------------|
| Fluig        | 8,60                     |             | 146,99                       |
| Protheus     | 7,81                     |             | 144,75                       |
| TOTVS HCM    | 7,80                     |             | 143,84                       |
| TOTVS RM     | 6,17                     |             | 142,88                       |
| <b>Total</b> | <b>7,63</b>              |             | <b>144,64</b>                |

 É importante propor melhorias na implantação dos sistemas RM, HCM e Protheus, pois em comparativo ao Fluig estão com a média abaixo. →

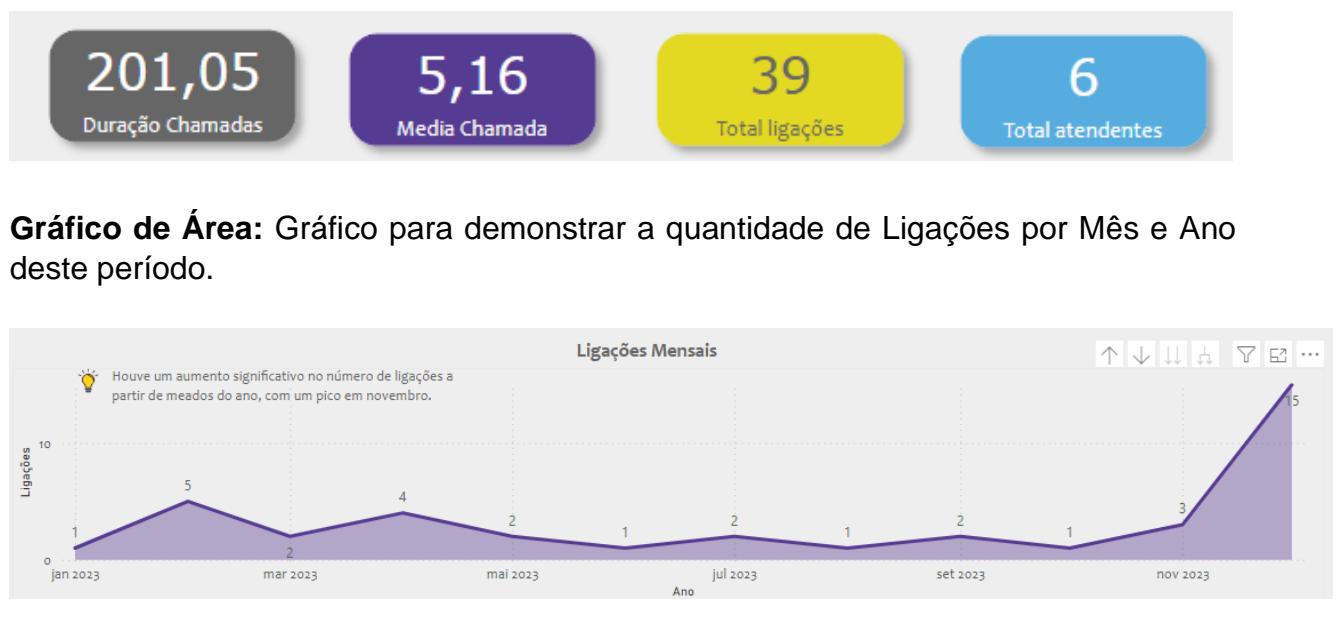
- **Segmentação:** Filtros/Segmentações por Mês e Unidade.



**Painel 4 - Visão Operacional:** Esse painel tem como intuito destacar as análises voltadas para os times de operação e atendimento ao cliente

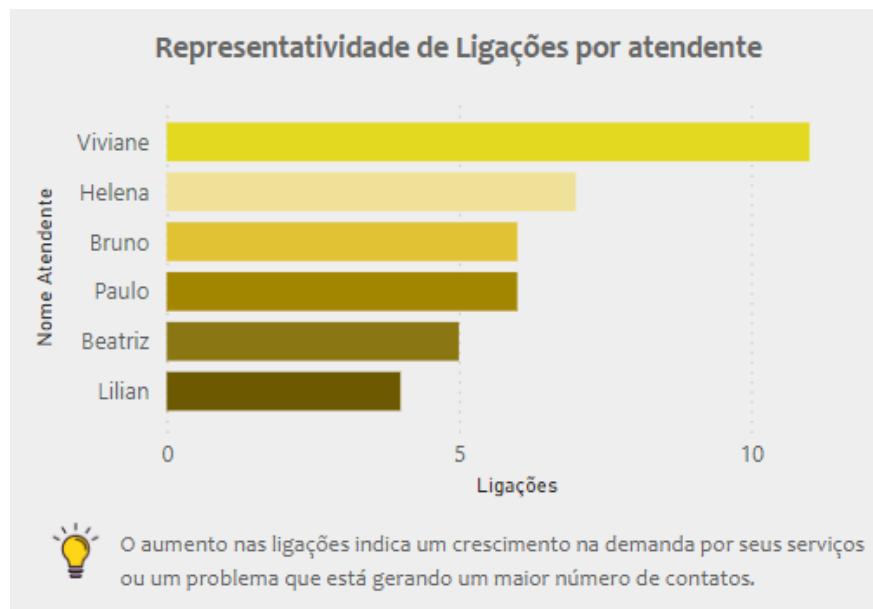


**Indicadores:** Usamos para destaque dos números e métricas, com o propósito de chamar a atenção do usuário. Esse indicadores mostram números rápidos e úteis para a análise

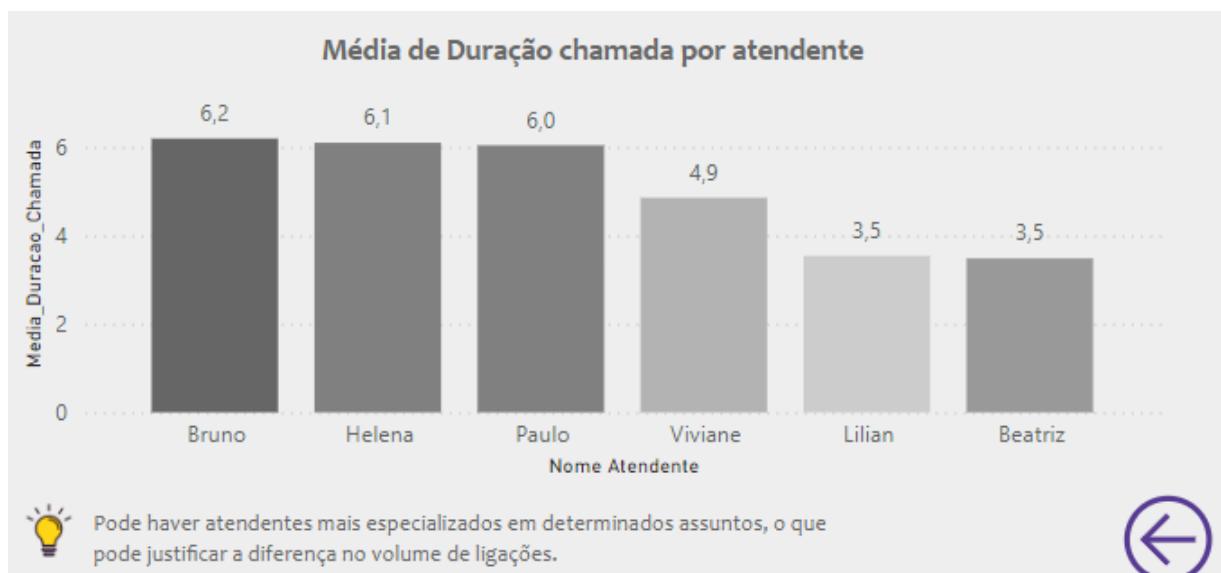


- **Gráfico de Área:** Gráfico para demonstrar a quantidade de Ligações por Mês e Ano deste período.

- **Gráfico de Barras:** Gráfico para demonstrar a quantidade de ligações por atendente.



- **Gráfico de Colunas:** Gráfico para demonstrar a quantidade média de duração de chamada por atendente.



- **Segmentação:** Filtros/Segmentações por Mês e Unidade.



## 11.1 CÓDIGO N-GRAMS

N-grams são sequências contínuas de "n" itens (ou "tokens") em um dado texto. Esses "itens" podem ser palavras, caracteres, ou outros componentes de uma sequência de texto. A técnica de N-grams é amplamente utilizada em processamento de linguagem natural (NLP), análise de texto e linguística computacional para entender o contexto e as relações entre palavras em um corpus de texto. N-grams são úteis para capturar o contexto local em textos, ou seja, as relações entre palavras adjacentes. São aplicados em várias tarefas de processamento de linguagem como Análise de Sentimento, Previsão de Texto, Classificação de Texto.

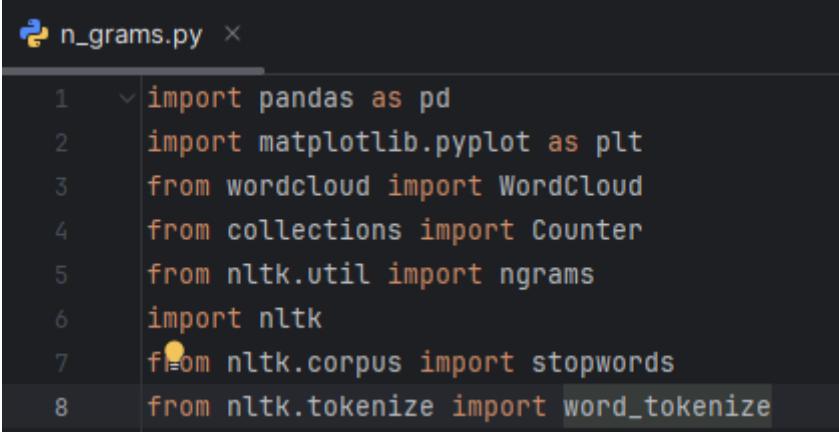
Abaixo segue o código para gerar os datasets de Bigrams e Trigrams. Esses dois datasets, foram extraídos a partir das transcrições realizadas, ou seja, compilamos as transcrições em um dataset e por meio dele, geramos dois datasets para serem importados no Power BI e assim plotar o Word Cloud via código python.

Configurações no PyCharm para executar o código:

Versão: Python 3.11

IDE: PyCharm Community Edition

Importação das bibliotecas, vale ressaltar, que no PyCharm deve-se instalá-las manualmente no Python Interpreter ou no Prompt Terminal



```
n_grams.py
1  import pandas as pd
2  import matplotlib.pyplot as plt
3  from wordcloud import WordCloud
4  from collections import Counter
5  from nltk.util import ngrams
6  import nltk
7  from nltk.corpus import stopwords
8  from nltk.tokenize import word_tokenize
```

Carregar o dataset “transcrição\_pbi.csv” que contém as transcrições armazenadas

```
# Carregar o dataset
df = pd.read_csv(filepath_or_buffer: 'transcricao_pbi.csv', encoding='latin-1')
```

## Baixar os datasets do modelo do NLTK

```
# Download NLTK datasets
nltk.download('punkt') # Faz o download do pacote de tokenização de palavras
nltk.download('stopwords') # Faz o download do pacote de stopwords (palavras irrelevantes)
```

## Dicionários adicionais de palavras para serem excluídas do modelo

```
# Lista adicional de preposições, conjunções, saudações e outras palavras desnecessárias para o modelo
additional_stopwords = {
    'a', 'ante', 'até', 'após', 'com', 'contra', 'de', 'desde', 'em', 'entre', 'para',
    'per', 'perante', 'por', 'sem', 'sob', 'sobre', 'trás', 'e', 'mas', 'ou', 'nem',
    'porque', 'pois', 'tchau', 'oi', 'olá', 'pra', 'tudo', 'bem', 'né', 'bom', 'tá', 'falar', 'minutinhos',
    'ah', 'ai', 'dia', 'caso', 'amigo', 'colega', 'hoje', 'pouquinho', 'ai', 'senhor', 'totos', 'nps', 'brasil',
    'bloco', 'então', 'gente', 'cnpj', 'sarah', 'bali', 'algum', 'nps', 'dp', 'peço', 'si', 'utiliza', 'nota', 'precisam',
    'agora', 'empresa', 'numero', 'número', 'gostaria', 'mais', 'meu', 'nossa',
    'nossa', 'você', 'vocês', 'eles', 'elas', 'ele', 'ela', 'dele', 'dela',
    'nos', 'nós', 'me', 'te', 'lhe', 'pro', 'sra', 'senhora', 'sra', 'seu',
    'dona', 'nome', 'gostaria', 'uns', 'umas', 'não', 'sim', 'está', 'estão',
    'foi', 'será', 'serão', 'ter', 'tive', 'tiver', 'tinha', 'tem', 'tenho', 'sendo',
    'estive', 'está', 'ficar', 'ficou', 'fiquei', 'podemos', 'poder', 'pode', 'fazer',
    'fazendo', 'feito', 'já', 'agora', 'tempo', 'tudo', 'todos', 'só', 'muito',
    'pouco', 'nada', 'mesmo', 'assim', 'também', 'ainda', 'quero', 'vamos', 'vai',
    'vem', 'veio', 'ir', 'indo', 'indo', 'ver', 'vendo', 'ouvir', 'ouvindo', 'preciso',
    'precisamos', 'precisa', 'faz', 'fazemos', 'quero', 'disse', 'dizer', 'falou',
    'teremos', 'terão', 'tiveram', 'poderemos', 'esperar', 'esperando', 'esperou',
    'ligar', 'ligando', 'ligou', 'mandar', 'mandou', 'enviar', 'enviou', 'estamos',
    'ajudar', 'ajudando', 'ajudou', 'resolver', 'resolvendo', 'resolvido', 'telefone', 'email', 'usuário', 'usuária',
    'usuário', 'alguma', 'algum', 'alguns', 'totalmente', 'parcialmente', 'total', 'parcial', 'usuária', 'chamo',
    'minutos', 'boleto', 'boleto', 'iniciando', 'contato', 'falando', 'dois', 'comigo', 'alguém', 'entrega'}
```

```
# Lista de nomes comuns em português (exemplo)
common_names = {
    'ana', 'maria', 'joão', 'jose', 'josé', 'antonio', 'francisco', 'carlos', 'paulo',
    'pedro', 'lucas', 'luiz', 'marcos', 'gabriel', 'rafael', 'daniel', 'vinicius',
    'eduardo', 'roberto', 'fernando', 'juliana', 'bruna', 'fernanda', 'camila', 'amanda',
    'jessica', 'leticia', 'beatriz', 'julio', 'rodrigo', 'renato', 'ricardo', 'tiago',
    'thiago', 'felipe', 'luis', 'sara', 'isabela', 'larissa', 'laura', 'giovana',
    'mariana', 'alice', 'hugo', 'luciana', 'marcio', 'sergio', 'andrê', 'carla', 'carol',
    'sandra', 'vanessa', 'gabriela', 'priscila', 'patricia', 'aline', 'daniela',
    'andreia', 'roberta', 'bianca', 'eduarda'
}
```

## Atualizar o modelo do NLTK

```
# Atualizar o conjunto de stopwords
stop_words.update(additional_stopwords) # Adiciona as stopwords personalizadas à lista principal
stop_words.update(common_names) # Adiciona os nomes próprios à lista de stopwords
```

## Funções para executar o modelo

```
# Função para tokenizar e remover stopwords e letras únicas
def preprocess_text(text): 1 usage
    tokens = word_tokenize(text.lower()) # Converte o texto em tokens e coloca tudo em letras minúsculas
    filtered_tokens = [word for word in tokens if word.isalpha() and word not in stop_words and len(word) > 1] # Remove stopwords, n
    return filtered_tokens

# Preprocessando o texto
df['Filtered_Tokens'] = df['texto'].apply(preprocess_text) # Aplica o pré-processamento de texto à coluna "texto"

# Função para gerar e contar n-grams
def generate_ngrams(tokens_list, n): 2 usages
    ngram_list = list(ngrams(tokens_list, n)) # Gera uma lista de n-grams (bigrams, trigrams, etc.)
    ngram_counts = Counter(ngram_list) # Conta a frequência de cada n-gram
    return ngram_counts

# Gerando bigrams e trigrams
df['Bigrams'] = df['Filtered_Tokens'].apply(lambda x: generate_ngrams(x, n=2)) # Gera bigrams a partir dos tokens filtrados
df['Trigrams'] = df['Filtered_Tokens'].apply(lambda x: generate_ngrams(x, n=3)) # Gera trigrams a partir dos tokens filtrados

# Inicializando contadores para bigrams e trigrams
bigram_counts = Counter() # Cria um contador vazio para bigrams
trigram_counts = Counter() # Cria um contador vazio para trigrams

# Somando os contadores
for bigram_counter in df['Bigrams']:
    bigram_counts.update(bigram_counter) # Atualiza o contador de bigrams com os valores da coluna
for trigram_counter in df['Trigrams']:
    trigram_counts.update(trigram_counter) # Atualiza o contador de trigrams com os valores da coluna

# Remover n-grams que contêm stopwords ou palavras de uma letra
def filter_ngrams(ngram_counts): 2 usages
    filtered_ngram_counts = Counter() # Cria um novo contador para os n-grams filtrados
    for ngram, freq in ngram_counts.items():
        if not any(word in stop_words or len(word) == 1 for word in ngram):
            filtered_ngram_counts[ngram] = freq # Mantém apenas os n-grams que não contêm stopwords ou palavras de uma letra
    return filtered_ngram_counts

# Aplicar filtro nos bigrams e trigrams
bigram_counts = filter_ngrams(bigram_counts) # Aplica o filtro nos bigrams
trigram_counts = filter_ngrams(trigram_counts) # Aplica o filtro nos trigrams
```

## Gerar o Word Cloud e salvar os datasets bigram.csv e trigram.csv para importar no Power BI.

```
# Função para gerar a word cloud de n-grams
def generate_wordcloud(ngram_counts, title): 2 usages
    # Criando a string combinada de n-grams para a word cloud
    word_freq = ' '.join(k: v for k, v in ngram_counts.items()) # Converte os n-grams em string e cria um dicionário de frequências
    wordcloud = WordCloud(width=800, height=400, background_color='white').generate_from_frequencies(word_freq) # Gera a word cloud

    # Plotando a word cloud
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.title(title, fontsize=18)
    plt.axis('off')
    plt.show()

# Gerando e exibindo as word clouds
generate_wordcloud(bigram_counts, title='Word Cloud de Bigramas')
generate_wordcloud(trigram_counts, title='Word Cloud de Trigramas')

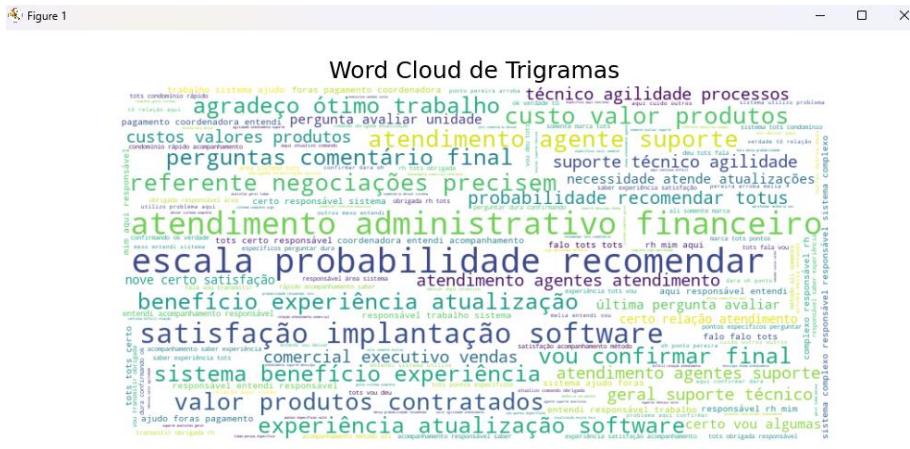
# Convertendo os bigrams e trigrams para DataFrame e salvando como CSV
bigrams_df = pd.DataFrame(bigram_counts.most_common(20), columns=['Bigram', 'Frequency'])
trigrams_df = pd.DataFrame(trigram_counts.most_common(20), columns=['Trigram', 'Frequency'])

# Salvando os DataFrames como CSV
bigrams_df.to_csv(path_or_buf='bigrams.csv', index=False)
trigrams_df.to_csv(path_or_buf='trigrams.csv', index=False)
```

### Exemplo output Bigrams



### Exemplo output Trigrams



Vale ressaltar que para gerar um gráfico utilizando Python dentro do Power BI, você deve ter instalado alguma versão Python em sua máquina e instalar as bibliotecas utilizadas, como pandas, matplotlib, wordcloud e nltk. Instale-as no prompt de comando do seu sistema operacional.

## Instalação Matplotlib

```
C:\Users\cesar>python -m matplotlib all
C:\Users\cesar>python -m matplotlib.downloader all
C:\Users\cesar>pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.9.2-cp310-cp310-win_amd64.whl (7.8 MB)
    7.8/7.8 MB 5.3 MB/s eta 0:00:00
Collecting packaging>=20.0
  Using cached packaging-24.1-py3-none-any.whl (53 kB)
Collecting fonttools>=4.22.0
  Downloading fonttools-4.53.1-cp310-cp310-win_amd64.whl (2.2 MB)
    2.2/2.2 MB 5.6 MB/s eta 0:00:00
Collecting pillow>=8
  Downloading pillow-10.4.0-cp310-cp310-win_amd64.whl (2.6 MB)
    2.6/2.6 MB 5.8 MB/s eta 0:00:00
Collecting cycler>=0.10
  Using cached cycler-0.12.1-py3-none-any.whl (8.3 kB)
Collecting kiwisolver>=1.3.1
  Downloading kiwisolver-1.4.7-cp310-cp310-win_amd64.whl (55 kB)
    55.9/55.9 kB ? eta 0:00:00
Collecting numpy>=1.23
  Downloading numpy-2.1.1-cp310-cp310-win_amd64.whl (12.9 MB)
    12.9/12.9 MB 5.7 MB/s eta 0:00:00
Collecting pyparsing>=2.3.1
  Using cached pyparsing-3.1.4-py3-none-any.whl (104 kB)
Collecting contourpy>=1.0.1
  Downloading contourpy-1.3.0-cp310-cp310-win_amd64.whl (216 kB)
    216.0/216.0 MB 13.7 MB/s eta 0:00:00
Collecting python-dateutil>=2.7
  Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
Collecting six>=1.5
  Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: six, pyparsing, pillow, packaging, numpy, kiwisolver, fonttools, cycler, python-dateutil, contourpy, matplotlib
Successfully installed contourpy-1.3.0 cycler-0.12.1 fonttools-4.53.1 kiwisolver-1.4.7 matplotlib-3.9.2 numpy-2.1.1 packaging-24.1 pillow-10.4.0 pyparsing-3.1.4 python-dateutil-2.9.0.post0 six-1.16.0
[notice] A new release of pip is available: 23.0.1 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

## Instalação Pandas

```
C:\Users\cesar>pip install pandas
Collecting pandas
  Downloading pandas-2.2.2-cp310-cp310-win_amd64.whl (11.6 MB)
    11.6/11.6 MB 5.3 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.22.4 in c:\users\cesar\appdata\local\programs\python\python310\lib\site-packages (from pandas) (2.1.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\cesar\appdata\local\programs\python\python310\lib\site-packages (from pandas) (2.9.0.post0)
Collecting pytz>=2020.1
  Using cached pytz-2024.2-py2.py3-none-any.whl (508 kB)
Collecting tzdata>=2022.7
  Using cached tzdata-2024.1-py2.py3-none-any.whl (345 kB)
Requirement already satisfied: six>=1.5 in c:\users\cesar\appdata\local\programs\python\python310\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Installing collected packages: pytz, tzdata, pandas
Successfully installed pandas-2.2.2 pytz-2024.2 tzdata-2024.1
[notice] A new release of pip is available: 23.0.1 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

## Instalação NLTK

```
C:\Users\cesar> python -m nltk.download all
C:\Users\cesar>AppData\Local\Programs\Python\Python310\lib\runpy.py:126: RuntimeWarning: 'nltk.download' found in sys.modules after import of package 'nltk', but prior to execution of 'nltk.download'; this may result in unpredictable behaviour
  warn(RuntimeWarning(msg))
[nltk_data] Downloading collection 'all'
```

## Instalação Wordcloud

```
C:\Users\cesar>pip install wordcloud
Collecting wordcloud
  Downloading wordcloud-1.9.3-cp310-cp310-win_amd64.whl (299 kB)
    300.0/300.0 kB 4.7 MB/s eta 0:00:00
Requirement already satisfied: pillow in c:\users\cesar\appdata\local\programs\python\python310\lib\site-packages (from wordcloud) (10.4.0)
Requirement already satisfied: numpy>=1.6.1 in c:\users\cesar\appdata\local\programs\python\python310\lib\site-packages (from wordcloud) (2.1.1)
Requirement already satisfied: matplotlib in c:\users\cesar\appdata\local\programs\python\python310\lib\site-packages (from wordcloud) (3.9.2)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\cesar\appdata\local\programs\python\python310\lib\site-packages (from matplotlib>wordcloud) (3.1.4)
Requirement already satisfied: packaging>=20.0 in c:\users\cesar\appdata\local\programs\python\python310\lib\site-packages (from matplotlib>wordcloud) (24.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\cesar\appdata\local\programs\python\python310\lib\site-packages (from matplotlib>wordcloud) (1.3.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\cesar\appdata\local\programs\python\python310\lib\site-packages (from matplotlib>wordcloud) (1.4.7)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\cesar\appdata\local\programs\python\python310\lib\site-packages (from matplotlib>wordcloud) (4.53.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\cesar\appdata\local\programs\python\python310\lib\site-packages (from matplotlib>wordcloud) (2.9.0.post0)
Requirement already satisfied: cycler>=0.10 in c:\users\cesar\appdata\local\programs\python\python310\lib\site-packages (from matplotlib>wordcloud) (0.12.1)
Requirement already satisfied: six>=1.5 in c:\users\cesar\appdata\local\programs\python\python310\lib\site-packages (from python-dateutil>=2.7->matplotlib>wordcloud) (1.16.0)
Installing collected packages: wordcloud
Successfully installed wordcloud-1.9.3
[notice] A new release of pip is available: 23.0.1 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Com o Python e bibliotecas instaladas e os datasets gerados e salvos, basta importá-los para o Power BI e executar o código abaixo clicando na visualização de Python e assim o wordcloud será gerado dentro do painel.

The screenshot shows a Power BI dashboard with the following components:

- PAINEL: NPS**: A section with four colored boxes displaying NPS metrics:
  - Total Clientes: 39
  - Neutros: 24
  - Promotores: 8
  - Detratores: 7
- Média de NPS Mensal**: A line chart showing monthly NPS scores from January to December. The chart highlights June and October as months of importance for communication improvement.
- Score NPS por Média**: A gauge chart showing a score of 7,46.
- Nuvem de Palavras (Trigrams)**: A word cloud visualization showing the most frequent trigrams. The text in the cloud includes: 'satisfação', 'implantação', 'software', 'atendimento', 'administrativo', 'financeiro', 'agradeço', 'ótimo', 'trabalho', 'atendendo', 'agente', 'suporte', 'atendimentos', 'agentes', 'atendimento', 'vender', 'comercial', 'executivo', 'vendas', 'benefício', 'experiência', 'atualização', 'escala', 'probabilidade', 'recomendar'.
- Editor de scripts Python**: An open Python script window with the following code:

```

5 As linhas duplicadas serão removidas dos dados.
6 # Cole ou digite aqui seu código de script:
7
8 import matplotlib.pyplot as plt
9 from wordcloud import WordCloud
10
11 # Converter os dados em um dicionário
12 word_freq = dict(zip(dataset['Trigram'], dataset['Frequency']))
13
14 # Gerar a nuvem de palavras com o tamanho máximo da fonte ajustado
15 wordcloud = WordCloud(width=800, height=400, background_color='white', max_words=10, max_font_size=500).generate_from_frequencies(word_freq)
16
17 # Plotar a nuvem de palavras
18 plt.imshow(wordcloud, interpolation='bilinear')
19 plt.axis('off')
20 plt.show()
21
22
23

```

## **12. GOVERNANÇA DE DADOS E LGPD**

Sob a Lei Geral de Proteção de Dados (LGPD) no Brasil, o tratamento de dados pessoais deve seguir princípios e bases legais específicas. Inserir conversas gravadas entre clientes e atendentes em plataformas como o ChatGPT envolve o processamento de dados pessoais, podendo incluir informações sensíveis.

Para estar em conformidade com a LGPD, a empresa precisa assegurar que possui uma base legal adequada para esse processamento, como o consentimento explícito dos titulares dos dados ou o legítimo interesse, entre outras. Além disso, é essencial garantir que qualquer serviço terceirizado utilizado para processar esses dados também esteja em conformidade com os requisitos de proteção e privacidade da LGPD.

Considerando que o ChatGPT é um serviço fornecido pela OpenAI, que pode processar dados fora do Brasil, é necessário prestar atenção às regras sobre transferência internacional de dados previstas na LGPD. A empresa deve implementar salvaguardas apropriadas para proteger esses dados durante a transferência e processamento.

Portanto, inserir conversas gravadas que contenham dados pessoais no ChatGPT sem as devidas precauções e bases legais pode resultar em violação da LGPD. Recomenda-se consultar um profissional jurídico ou um encarregado de proteção de dados para garantir que todas as práticas estejam em conformidade com a legislação vigente.

Dado o foco do nosso projeto em garantir a máxima proteção e privacidade dos dados, o uso de IA Generativa não se alinha com nossos objetivos. Essa tecnologia pode introduzir incertezas em relação ao controle sobre o processamento e armazenamento de informações, o que pode comprometer a segurança e a confiança que buscamos oferecer com nossa solução. Por isso, decidimos não adotar IA Generativa, priorizando alternativas que assegurem um maior controle sobre os dados e minimizem riscos.

## **LINKS PARA O PITCH COMERCIAL E VÍDEO TECNICO**

Link Pitch Comercial

<https://www.youtube.com/watch?v=ichbLMw37Fs>

Link Vídeo Técnico

[https://www.canva.com/design/DAGRyjiKync/r1qRn4gGgEU5XIO8TpXfHQ/watch?utm\\_content=DAGRyjiKync&utm\\_campaign=designshare&utm\\_medium=link&utm\\_source=editor](https://www.canva.com/design/DAGRyjiKync/r1qRn4gGgEU5XIO8TpXfHQ/watch?utm_content=DAGRyjiKync&utm_campaign=designshare&utm_medium=link&utm_source=editor)

## CONCLUSÃO

Nossa solução final proporciona à TOTVS uma infraestrutura robusta e escalável para otimizar o processo de avaliação de satisfação dos clientes via NPS. Utilizando ferramentas modernas de transcrição de áudio, análise de sentimentos e armazenamento em nuvem, conseguimos integrar diferentes tecnologias de maneira eficiente, garantindo maior precisão e velocidade nas análises. Isso permite que a TOTVS tome decisões mais embasadas e assertivas em relação à satisfação de seus clientes, identificando promotores, neutros e detratores com maior clareza.

Além disso, nossa solução aborda com sucesso os desafios de processamento de grandes volumes de dados e integrações entre plataformas, como o MongoDB e o Google Colab. A automação com Apache Airflow e a flexibilidade da infraestrutura em nuvem oferecem uma operação otimizada e de baixo custo, facilitando a escalabilidade e a manutenção do sistema. Dessa forma, a TOTVS tem à disposição uma solução capaz de acompanhar seu crescimento e suas demandas crescentes, mantendo a eficiência operacional.

Por fim, priorizamos a conformidade com as normas da LGPD, garantindo a segurança e o controle sobre os dados dos clientes. Ao focar em tecnologias que oferecem proteção e governança de dados, nossa solução assegura que o processamento de informações siga os padrões legais, minimizando riscos e aumentando a confiança no sistema. Com essa abordagem, a TOTVS estará bem equipada para melhorar a experiência de seus clientes e aprimorar suas estratégias de negócio com base em dados concretos e confiáveis.

## FONTES

[Compatibilidade e instalação do mongodump - MongoDB Database Tools](#) - Acessado em 14/09/2024

[Mongorestore - MongoDB Database Tools](#) - Acessado em 14/09/2024

[Mongodump - MongoDB Database Tools](#) - Acessado em 15/09/2024

[Preços da Oracle Cloud Infrastructure \(OCI\)](#) - Acessado em 18/09/2024

[Leveraging Reviews — N Grams and Word Clouds: An NLP Walk-through | by DatavisionDallas | Medium](#) - Acessado em 18/09/2024

[MySQL 8.4 Reference Manual : 6.5.4 mysqldump - A Database Backup Program](#) - Acessado em 20/09/2024

[Sched — Event scheduler — Python 3.12.6 documentation](#) - Acessado em 20/09/2024

<https://github.com/datavisionDallas/NLP-Walkthrough> - Acessado em 20/09/2024

[LGPD - Planalto.gov](#) - Acessado em 20/09/2024