

## Tech Challenge 6IADT - Fase 1

### GRUPO 17

<b>Ana Maria Silva</b>	<b>RM: 366289</b>
<b>Débora Dâmaso</b>	<b>RM: 366501</b>
<b>Juliana Conde</b>	<b>RM: 366402</b>
<b>Marcelo Fernandes</b>	<b>RM: 366035</b>
<b>Mariana Santana</b>	<b>RM: 366262</b>

Pós Tech IA para Devs  
São Paulo - SP  
2025

## SUMÁRIO

<b>1 - APRESENTAÇÃO DA EQUIPE.....</b>	<b>4</b>
<b>2 - INTRODUÇÃO DO PROJETO.....</b>	<b>5</b>
<b>2-1 Justificativa.....</b>	<b>5</b>
<b>3 – ESCOLHA DOS DATASETS.....</b>	<b>6</b>
<b>4 – DATASET DE REGISTRO DE PACIENTES.....</b>	<b>7</b>
<b>4-1 Descrição do Dataset de Registro de Pacientes .....</b>	<b>7</b>
<b>4-2 Análise Exploratória do Dataset de Registro de Pacientes.....</b>	<b>7</b>
<b>4-3 Pré-Processamento de Dados para o Dataset de Registro de Pacientes .....</b>	<b>19</b>
<b>4-4 Desenvolvimento do Modelo para o Dataset de Registro de Pacientes .....</b>	<b>22</b>
<b>4-4.1 Modelo de Regressão Logística e Resultados.....</b>	<b>26</b>
<b>4-4.2 Modelo de Random Forest e Resultados.....</b>	<b>29</b>
<b>4-4.3 Comparação entre os modelos de Regressão Logística e Random Forest .....</b>	<b>31</b>
<b>4-4.4 Modelo Árvore de Classificação.....</b>	<b>31</b>
<b>4-4.5 Comparação dos resultados dos 3 Modelos (Regressão Logística, Random Forest e Árvore de Classificação).....</b>	<b>32</b>
<b>4-5 Conclusão do estudo realizado na base tabular .....</b>	<b>33</b>
<b>4-6 Limitações encontradas no estudo e oportunidades de melhoria .....</b>	<b>34</b>
<b>5 – DATASET DE IMAGENS DE PULMÃO.....</b>	<b>34</b>
<b>5-1 Descrição do Dataset de Imagens .....</b>	<b>34</b>
<b>5-2 Pré-processamento de Imagens.....</b>	<b>37</b>
<b>5-3 Avaliação Geral dos 3 modelos de imagem.....</b>	<b>43</b>
<b>5-4 Conclusão dos Modelos de Imagens .....</b>	<b>44</b>
<b>6 – APLICABILIDADE PRÁTICA GERAL DOS ESTUDOS .....</b>	<b>45</b>
<b>7 – LIÇÕES APRENDIDAS E POSSÍVEIS MELHORIAS .....</b>	<b>46</b>
<b>8 – CONCLUSÃO GERAL DO TRABALHO .....</b>	<b>47</b>
<b>9 – ENTREGÁVEIS.....</b>	<b>47</b>
<b>9-1 Repositório no GitHub .....</b>	<b>47</b>
<b>9-2 Links dos Datasets .....</b>	<b>47</b>
<b>9-3 Notebooks ipynb.....</b>	<b>48</b>
<b>9-4 Vídeo de Demonstração.....</b>	<b>48</b>
<b>9-5 Pasta do Google Drive .....</b>	<b>49</b>
<b>10 – REFERÊNCIAS .....</b>	<b>50</b>



# 1 - APRESENTAÇÃO DA EQUIPE

A organização Artificial Intelligence & Business Insights - AI&BI composta pelos cinco integrantes do grupo 17, é uma organização acadêmica inovadora com foco em pesquisa e desenvolvimento de soluções baseadas em Inteligência Artificial para atuação em um variedade de campos de negócios: tecnologia, saúde, educação, entre outras.

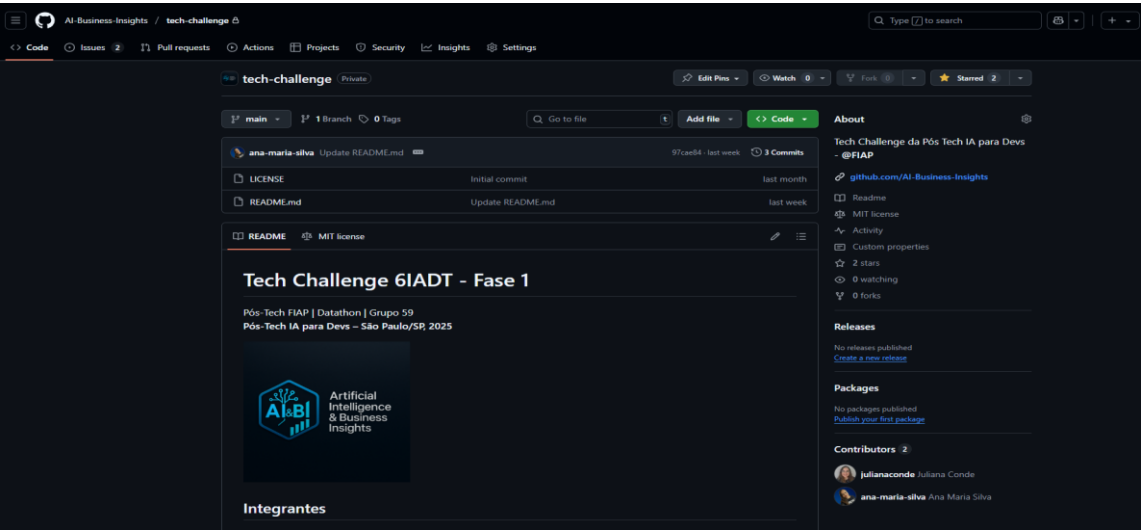
A Artificial Intelligence & Business Insights - AI&BI busca integrar conhecimento acadêmico com aplicações reais, explorando o potencial da IA para gerar insights estratégicos e transformar processos empresariais. O Estudo de caso descrito neste Relatório tem por objetivo construir soluções assistidas por Inteligência Artificial, para diagnósticos médicos, aplicando fundamentos essenciais de Machine Learning e Visão Computacional.

<b>Missão:</b>	Desenvolver soluções em IA que gerem impacto real em diversos campos de negócios.
<b>Visão:</b>	Ser referência em pesquisa aplicada de Inteligência Artificial no ambiente acadêmico.
<b>Valores:</b>	Inovação, Ética, Colaboração e Impacto Social.

Informações e projetos da AI&BI estão disponíveis no repositório de organização no GitHub em: [AI&Business Insights](#).



Figura 1 - Logomarca da Organização Artificial Intelligence & Business Insights



## 2 - INTRODUÇÃO DO PROJETO

A transformação digital no setor da saúde tem se intensificado nos últimos anos, impulsionada por uma crescente demanda por agilidade, precisão e eficiência nos processos clínicos. O desafio proposto na Fase 1 do Tech Challenge consiste em desenvolver um sistema inteligente de suporte ao diagnóstico para um hospital universitário. Com a proposta de aplicar técnicas de Inteligência Artificial (IA), com foco em Machine Learning e Visão Computacional, para auxiliar na análise de exames médicos e no processamento de dados clínicos, contribuindo para decisões mais rápidas e eficazes em contextos de alta demanda hospitalar.

Ao falar sobre demanda hospitalar, é notável que a pandemia da COVID-19 expôs gargalos críticos na saúde: escassez de leitos, deficiência de materiais e equipamentos, sobrecarga de profissionais e necessidade de decisões rápidas. O uso de inteligência artificial poderia auxiliar profissionais da saúde na triagem e diagnóstico de pacientes em casos como este, especialmente em cenários de sobrecarga hospitalar e escassez de recursos.

De acordo com fontes do Ministério da Saúde, do Portal Coronavírus Brasil e OpenDATASUS, que em 2020, no auge da pandemia, tinha como taxa de mortalidade por 100 mil habitantes igual a 92,77 e atualmente a taxa é de 0,82.

Com o aumento do volume de pacientes e exames, como radiografias, tomografias, ressonâncias magnéticas e prontuários digitalizados, a adoção de tecnologias de IA torna-se essencial. Modelos de aprendizado de máquina aplicados a dados estruturados podem identificar padrões sutis em registros clínicos, enquanto algoritmos de Deep Learning, como redes neurais convolucionais (CNNs), têm se mostrado altamente eficazes na análise de imagens médicas, como as radiografias de tórax — um exame-chave na detecção de complicações respiratórias causadas pela COVID-19.

É crucial buscar soluções que agilizem a triagem, reduzam erros e otimizem o tempo dos profissionais de saúde, como um sistema de IA capaz de analisar automaticamente resultados de exames e destacar informações relevantes para o diagnóstico, e principalmente identificar fatores de risco que possam influenciar no desenvolvimento do estado grave da doença.

### 2-1 Justificativa

O objetivo deste trabalho acadêmico é criar a base de um sistema de IA capaz de analisar automaticamente resultados de exames e destacar informações relevantes para o diagnóstico para agilizar a triagem, reduzir erros, otimizar o tempo dos profissionais de saúde e dar maior atenção à fatores que podem levar o paciente à desenvolver um estado grave da doença.

O Portal Coronavírus Brasil destaca limitações no registro dos dados: “Além de todos os desafios que a pandemia de COVID-19 impõe, é possível que haja mudanças no número de casos ou óbitos em decorrência de erros ou atrasos no repasse das informações.”

A proposta visa integrar duas frentes — dados estruturados e imagens — para construir uma solução de apoio à decisão clínica baseada em IA. Buscamos treinar modelos que não apenas classifiquem corretamente a presença da doença, mas também ofereçam interpretações visuais e estatísticas que possam apoiar o julgamento médico, sem jamais substituí-lo.

### 3 – ESCOLHA DOS DATASETS

O grupo decidiu trabalhar com estas fontes de dados distintas (dados tabulares e imagens) para ter a possibilidade de testar o uso de diferentes algoritmos de detecção de padrões, assim como aproximar-se da realidade de um hospital ou centro de diagnóstico, que certamente tem à disposição dados de diferentes naturezas e formatos, que podem ou não ser combinados para uma avaliação mais abrangente do estado de saúde de um paciente, permitindo ações que possam personalizar seu atendimento, de acordo com a tendência de evolução da doença, identificada a partir dos dados e modelos estatísticos e matemáticos empregados na análise dos dados.

Para a fase 1 do Tech Challenge, as bases de dados trabalhadas pelo grupo foram os seguintes:

- **Base de Dados I - Dataset de Registros de Pacientes** (dados tabulares, referentes a 1.048.576 pacientes únicos, provenientes do sistema de saúde do México): Contém dados estruturados, como idade, sexo e outras variáveis clínicas, para prever a probabilidade de evolução para quadros graves de COVID-19. Essa análise visa identificar fatores de risco que auxiliem na priorização de pacientes.
- **Base de dados II - Dataset de Imagens de Pulmão** (dados de 21.165 radiografias do pulmão, trabalhadas por pesquisadores das universidades de Doha, no Catar, e de Daca, em Bangladesh): Inclui imagens (ex.: radiografias) classificadas em estados como opacidade, pneumonia, COVID-19 ou saudável. Utilizando redes neurais convolucionais (CNNs), buscamos detectar anormalidades pulmonares associadas ao COVID-19, oferecendo suporte visual ao diagnóstico.

Vale a pena ressaltar que, ainda que sejam referentes ao mesmo tema, os dados dessas duas bases de dados são completamente independentes e trazem dados de pacientes totalmente distintos e de regiões completamente diferentes. Eles foram usados nesse trabalho para diversificar as análises de diagnósticos de doenças pulmonares, bem como, ao final do trabalho, traçar recomendações de protocolos e estratégias que possam ser utilizados a partir do resultado dos modelos, analisando o paciente de forma mais holística.

Dessa forma, o trabalho realizado consistiu em DUAS ANÁLISES INDEPENDENTES das bases de dados e, ao final, insights e recomendações convergiram para uma visão única. As etapas realizadas seguiram um fluxo sequencial de análise que variou de acordo com a base de dados, muito em função da natureza dos dados (dados tabulares, já minimamente estruturados, e dados de imagens, que requereram um pré-processamento prévio).

Para ambas as bases de dados, as análises resultaram em modelos matemáticos que prediziam dois eventos específicos:

- Probabilidade de o paciente evoluir para um estado grave ou morte, em decorrência da Covid19 (base de dados I)
- Probabilidade de a imagem representar um paciente com um pulmão não saudável (base de dados II)

A escolha do tema COVID-19 reflete sua importância global e a pressão sobre sistemas de saúde para diagnósticos rápidos e eficazes. O dataset de registros de pacientes permite explorar fatores clínicos e demográficos que influenciam a gravidade da doença, essencial para triagem e

alocação de recursos, enquanto o dataset de imagens complementa a abordagem ao possibilitar a análise automatizada de condições pulmonares, um componente crítico no diagnóstico de COVID-19, especialmente para identificar pneumonia ou outras complicações. Essa abordagem dupla combina técnicas de machine learning tradicional e visão computacional, atendendo aos objetivos do hospital de reduzir erros e melhorar a eficiência da triagem.

A partir dos modelos preditivos, o grupo propôs recomendações de estratégias de ação, assim como sugestões de melhoria no processo de análise e limitações identificadas nos dados. Os modelos preditivos foram desenvolvidos na linguagem Python e todo o processo de análise foi documentado e comentado em notebooks Python, construídos na plataforma Google Colab. Todos os dados, de alguma forma, utilizados em alguma etapa do processo de desenvolvimento do trabalho do grupo (bases de dados originais, dicionários de dados, scripts Python, relatório de entrega do trabalho, vídeo de apresentação dos resultados), foram armazenados em um repositório GitHub e compartilhados com a FIAP, para efeito de avaliação do trabalho e consulta a seus conteúdos.

## 4 – DATASET DE REGISTRO DE PACIENTES

### 4-1 Descrição do Dataset de Registro de Pacientes

Nesta primeira análise, o **objetivo foi criar um modelo preditivo de propensão do paciente a evoluir a um estado grave ou falecer por causa da Covid-19**. A base de dados utilizada desta análise foi encontrada no Kaggle, clássico portal de competições de ciência de dados e que contém um vasto repositório de dados e scripts, altamente relevante para estudo, teste e aprendizado das mais distintas técnicas de análise existentes no mundo da inteligência artificial e da ciência de dados. Esse conjunto de dados foi fornecido pelo governo mexicano ([Datos Abiertos de México - Información referente a casos COVID-19 en México](#)). Ele apresenta uma quantidade interessante de informações anônimas relacionadas a pacientes, incluindo condições pré-existent.

### 4-2 Análise Exploratória do Dataset de Registro de Pacientes

**O conjunto de dados bruto é composto por 21 variáveis únicas e 1.048.576 pacientes únicos.** O layout desse conjunto de dados é apresentado a seguir:

Variável	Descrição
SEX	Sexo do paciente (1 para FEMININO e 2 para MASCULINO)
AGE	Idade do paciente
CLASSIFICATION	Valores entre 1 e 3 o paciente foi diagnosticado com COVID em diferentes graus. Valores >=4 o paciente não carregava o vírus ou o teste foi inconclusivo.
PATIENT TYPE	1 se o paciente retornou para casa e 2 se foi hospitalizado
**PNEUMONIA	1 se o paciente já apresenta inflamação nos alvéolos pulmonares e 2 se não apresenta.
**PREGNANCY	1 se a paciente é uma gestante e 2 se não é.
**DIABETES	1 se o paciente tem diabetes e 2 se não tem.
**COPD	1 indica se o paciente tem DPOC (doença pulmonar obstrutiva crônica) e 2 se não tem.
**ASTHMA	1 se o paciente tem asma e 2 se não tem.

**INMSUPR	1 se o paciente é imunodeprimido e 2 se não é.
**HYPERTENSION	1 se o paciente é hipertenso e 2 se não é.
**CARDIOVASCULAR	1 se o paciente possui alguma doença relacionada ao coração ou aos vasos sanguíneos e 2 se não tem.
**RENAL CHRONIC	1 se o paciente tem doença renal crônica e 2 se não tem.
**OTHER DISEASE	1 se o paciente tem outras doenças e 2 se não tem.
**OBESITY	1 se o paciente é obeso e 2 se não é.
**TOBACCO	1 se o paciente é tabagista e 2 se não é.
USMR	Indica se o paciente foi atendido em unidades médicas de primeiro, segundo ou terceiro nível.
*MEDICAL UNIT	Tipo de instituição do Sistema Nacional de Saúde que prestou o atendimento.
**INTUBED	1 indica se o paciente foi conectado a um ventilador mecânico (se foi intubado) e 2 se não foi.
**ICU	1 indica se o paciente foi internado em uma Unidade de Terapia Intensiva (UTI) e 2 se não foi.
DATE DIED	Se o paciente faleceu, indica a data do óbito; caso contrário, utiliza o valor 9999-99-99.

*Tabela – Descrição dos campos da base tabular de pacientes atendidos pelo governo mexicano*

\*Apesar de a documentação pública não apresentar diretamente o mapa de códigos, é possível inferir seu significado pelas informações do campo relacionado SECTOR, que indica o tipo de instituição. Veja abaixo uma tabela com os principais valores, baseada em catálogos oficiais que acompanham esse tipo de base (via pacote R covid19mx, dados SISVER) e também disponível na planilha “Diccionario de Datos” (entregue juntamente com o trabalho) pasta “Catálogo SECTOR”, disponibilizado pelo Governo Mexicano ( e anexado à documentação desse trabalho).

1 – Cruz Roja, 2 – DIF, 3 – Estatal, 4 -IMSS, 5 – IMSS – BIENESTAR, 6 – ISSSTE, 7 – MUNICIPAL, 8 – PEMEX, 9 – PRIVADA, 10 – SEDENA, 11 – SEMAR, 12 – SSA, 13 – UNIVERSITARIO, 14 – CIJ, 15 - IMSS BIENESTAR OPD, 16 – NÃO ESPECIFICADO.

\*\* 1 indica que o paciente tem a condição, 2 que não possui a condição, e os códigos 97 a 99 sinalizam que a informação não estava disponível, porém, não necessariamente é um dado faltante (*missing value*), devendo ser interpretado com cautela para cada variável.

### **Etapas do processo de desenvolvimento do modelo preditivo de propensão a complicações sérias**

O processo de análise seguiu um fluxo sequencial de etapas que passaram pelo entendimento dos dados, descrição das variáveis, tratamento dos dados, modelagem preditiva, análise e validação dos resultados, conforme o diagrama esquemático apresentado a seguir:



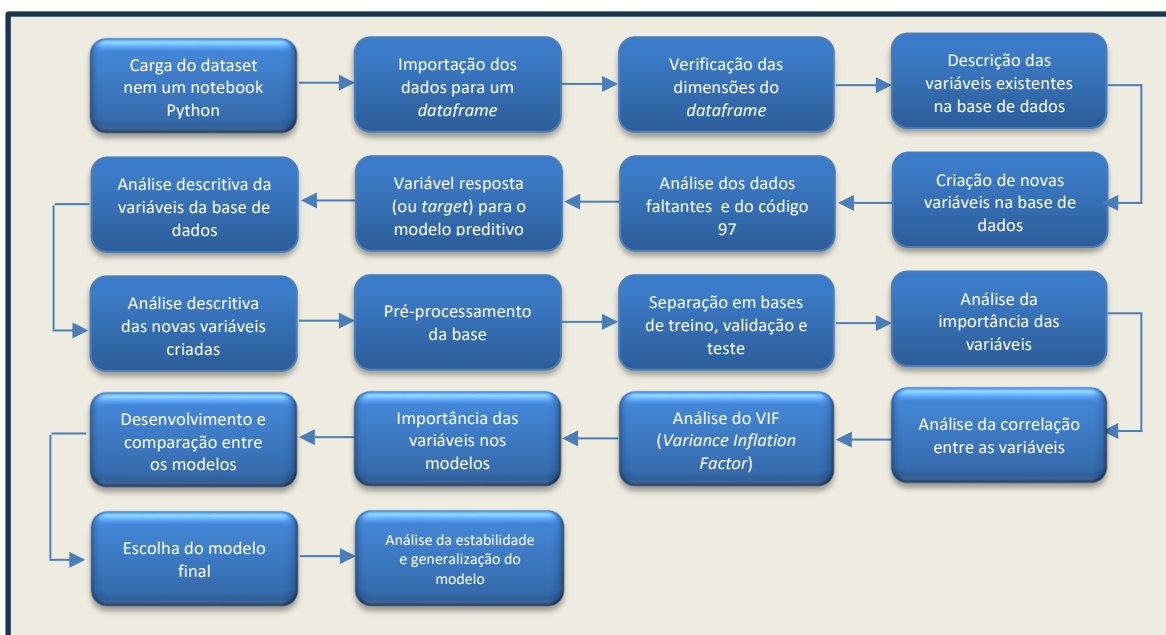


Diagrama – Fluxo sequencial de análise de dados da base de pacientes atendidos pelo governo mexicano

## Fase I – Carga de dados em um notebook Python

```

Etapa 1: Carga dos dados no notebook Python

[1] #Upload do dataset COVID19 para a nuvem Google, onde está o Google Colab
from google.colab import files
uploaded = files.upload()
  
```

A base de dados, extraída do Kaggle, como um arquivo no formato .txt, foi importada para um notebook Python, para que o processo de análise pudesse ser iniciado.

## Fase II – Importação dos dados para um dataframe

```

Etapa 2: Importação dos dados para um dataframe Pandas

[2] #Importação do dataset COVID19 para um formato dataframe
import pandas as pd

Covid = pd.read_csv('COVID19_Dados.csv', sep=',')
Covid.head()
  
```

	USHER	MEDICAL_UNIT	SEX	PATIENT_TYPE	DATE_DIED	INTUBED	PNEUMONIA	AGE	PREGNANT	DIABETES	...	ASTHMA	INSULIN	HIPERTENSION	OTHER_DISEASE	CARDIOVASCULAR	OBESITY	RENAL_CHRONIC	TOBACCO	CL
0	2	1	1	1	03/05/2020	97	1	65	2	2	...	2	2	1	2	2	2	2	2	2
1	2	1	2	1	03/06/2020	97	1	72	97	2	...	2	2	1	2	2	1	1	1	2
2	2	1	2	2	09/06/2020	1	2	55	97	1	...	2	2	2	2	2	2	2	2	2
3	2	1	1	1	12/06/2020	97	2	53	2	2	...	2	2	2	2	2	2	2	2	2
4	2	1	2	1	21/06/2020	97	2	68	97	1	...	2	2	1	2	2	2	2	2	2

5 rows x 21 columns

De acordo com a própria descrição da base de dados, no Kaggle, não há duplicações na base de dados e cada linha refere-se a dados de um paciente único e não havia nenhuma coluna de identificação do paciente, de forma que se considerou como verdade a informação fornecida na fonte de dados.

## Fase III – Verificação das dimensões do dataframe

**Etapa 3: Verificação das dimensões do dataframe**

```
# Contar número de linhas e colunas
linhas, colunas = Covid.shape

print(f"Número de linhas: {linhas}")
print(f"Número de colunas: {colunas}")
```

Número de linhas: 1048575  
Número de colunas: 21

A base de dados inicial tinha 1.048.575 linhas e 21 colunas

#### **Fase IV – Descrição das variáveis existentes na base de dados (como parte da documentação do notebook)**

```
"Variável": var,
"Tipo": descricao_completa[var][0],
"Descrição": descricao_completa[var][1]
}
for var in variaveis_dataset
})

# Exibir o DataFrame no Colab
Covid_descricao
```

	Variável	Tipo	Descrição
0	sex	Catégorica	Sexo do paciente (masculino ou feminino)
1	age	Numérica	Idade do paciente (em anos)
2	patient_type	Binária	Tipo de paciente: hospitalizado ou não
3	intubed	Binária	Se o paciente foi intubado
4	pneumonia	Binária	Presença de pneumonia
5	pregnant	Binária	Gestante (válido apenas para sexo feminino)
6	diabetes	Binária	Diagnóstico de diabetes
7	copd	Binária	Doença pulmonar obstrutiva crônica (DPOC)
8	asthma	Binária	Asma
9	inmsupr	Binária	Imunossupressão (condições ou tratamento)

Apenas a título de documentação e entendimento das variáveis durante o processo de análise, cada uma das variáveis iniciais presentes na base de dados foi descrita para que seu entendimento ficasse mais claro com a evolução das etapas.

#### **Fase V – Análise descritiva das variáveis da base de dados**

**Etapa 5: Análise descritiva das variáveis do dataset**

```
#Análise descritiva das variáveis da base (essencialmente variáveis categóricas, com exceção da variável idade)
import plotly.express as px
import pandas as pd

def plotly_distributions(Covid):
    for column in Covid.columns:
        if column in ['morreu', 'alto_risco']:
            continue

        Covid[column] = Covid[column].fillna('Desconhecido')

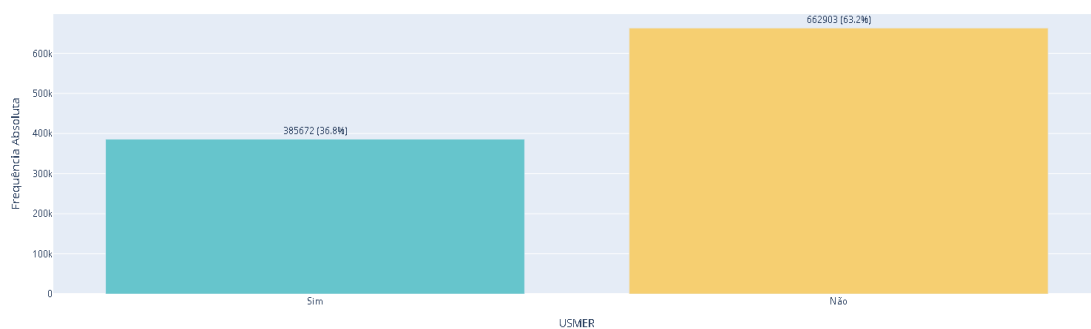
        label_map = None
        ordered_labels = None
        force_same_color = False
        column_label = column
        x_label = column

        # AGE
        if column == 'AGE':
            x_label = 'AGE (Em anos)'

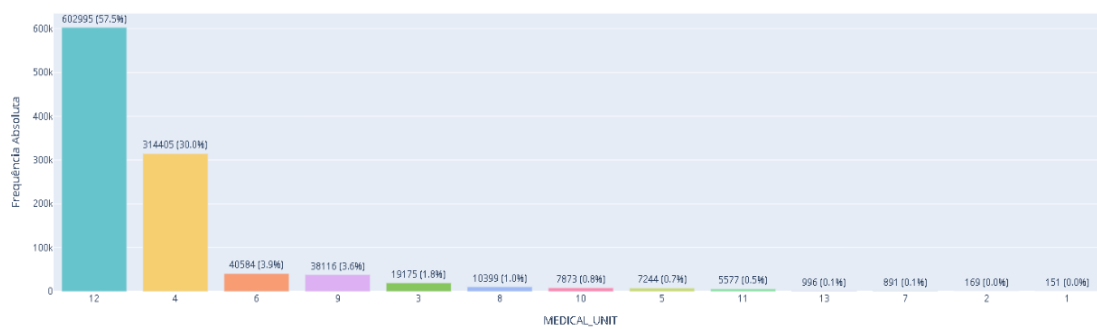
        # Substituições padronizadas para 97, 98, 99
        standard_map = {
            97: 'Não se aplica',
            98: 'Se ignora',
            99: 'Não especificado'
        }
```

Os gráficos a seguir refletem a distribuição de cada uma das variáveis presente na base de dados de pacientes atendidos pelo governo mexicano.

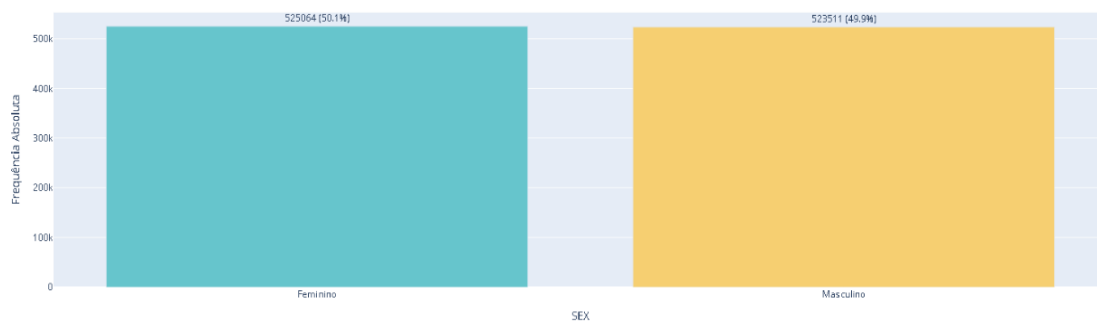
Distribuição da variável: **USMER**



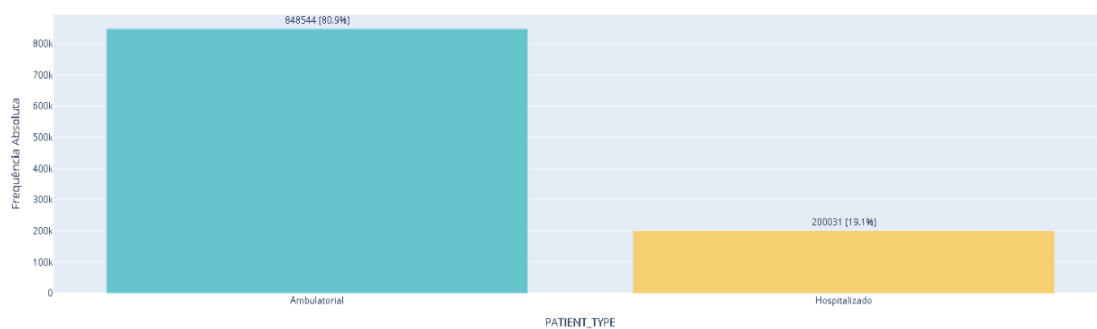
Distribuição da variável: **MEDICAL\_UNIT**



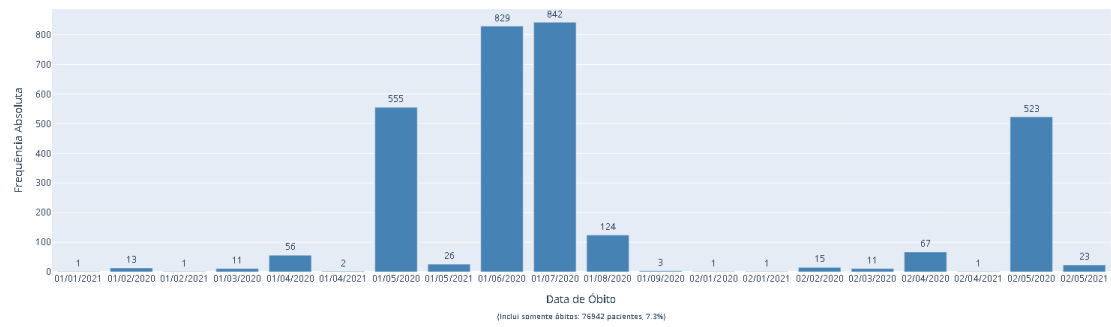
Distribuição da variável: **SEX**



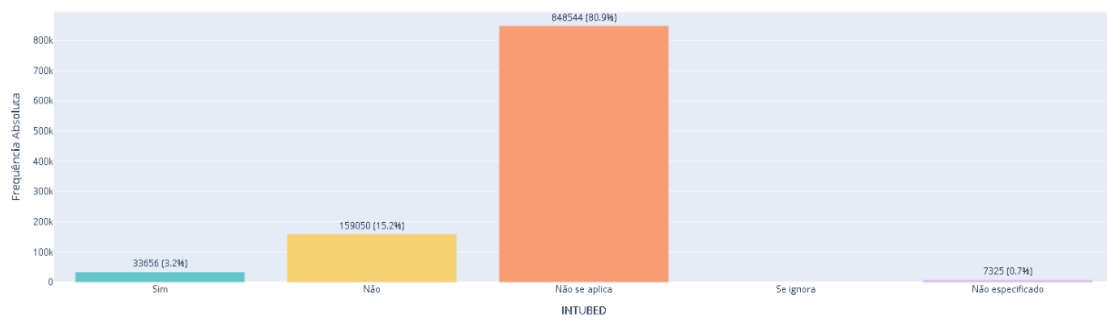
Distribuição da variável: **PATIENT\_TYPE**



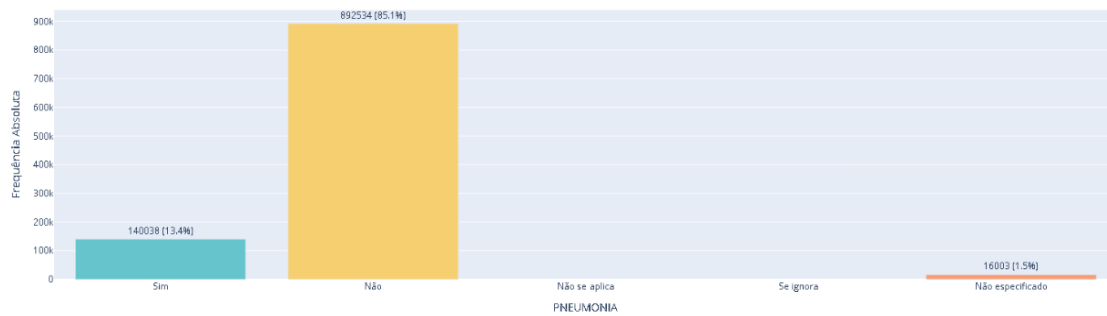
Distribuição da variável: **DATE\_DIED**



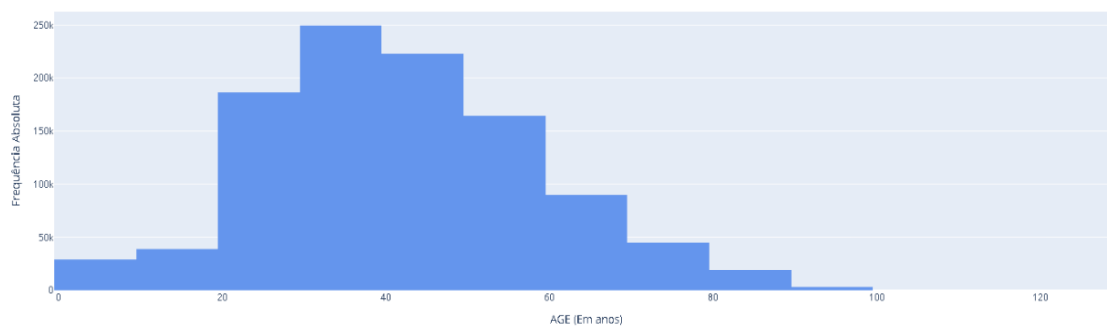
Distribuição da variável: **INTUBED**



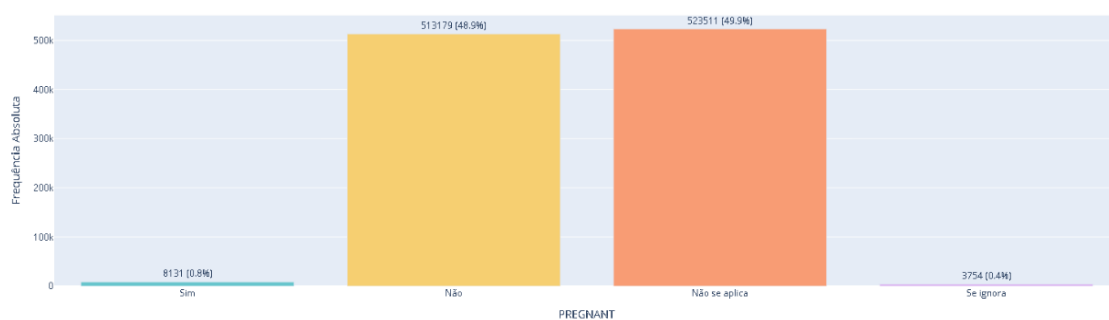
Distribuição da variável: **PNEUMONIA**



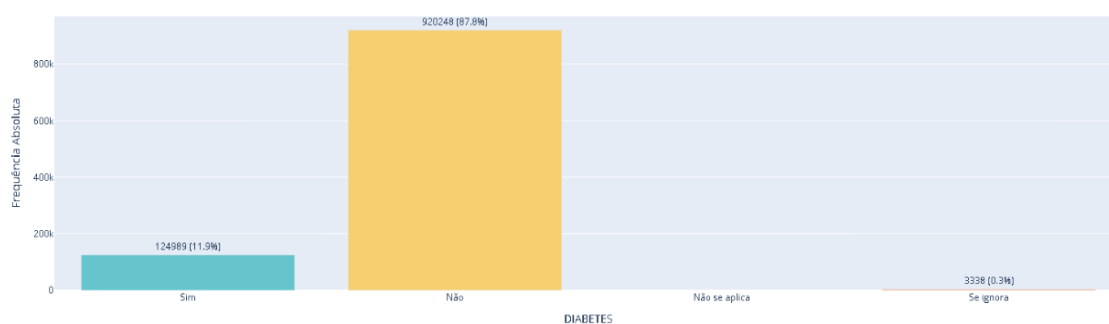
Distribuição da variável: **AGE**



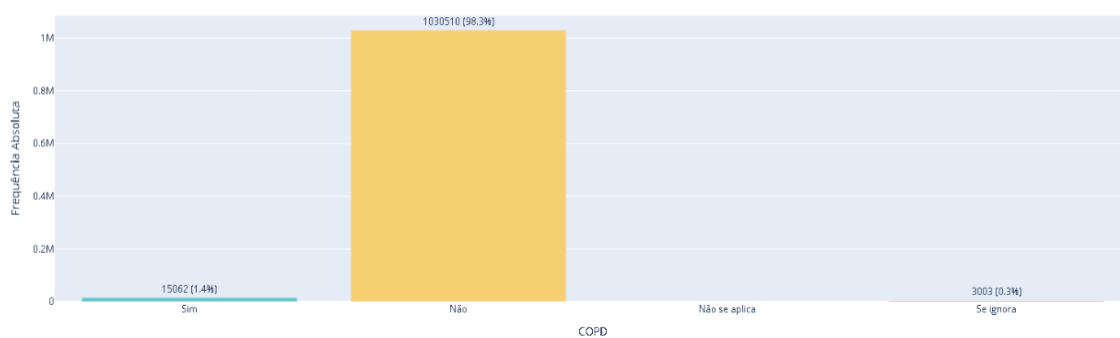
Distribuição da variável: **PREGNANT**



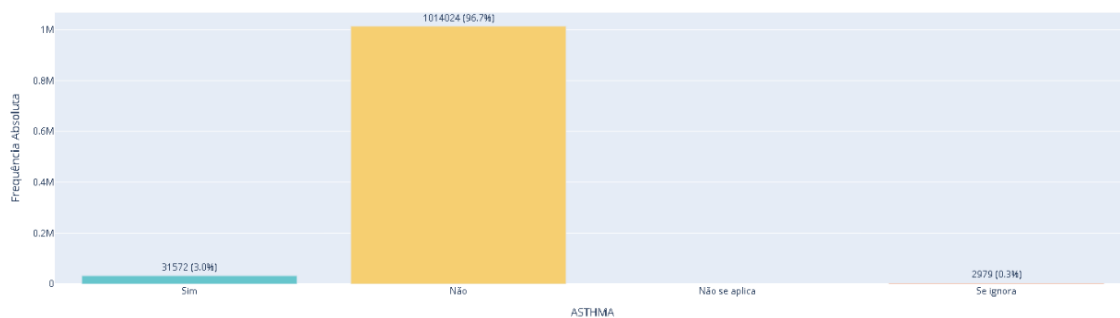
Distribuição da variável: **DIABETES**

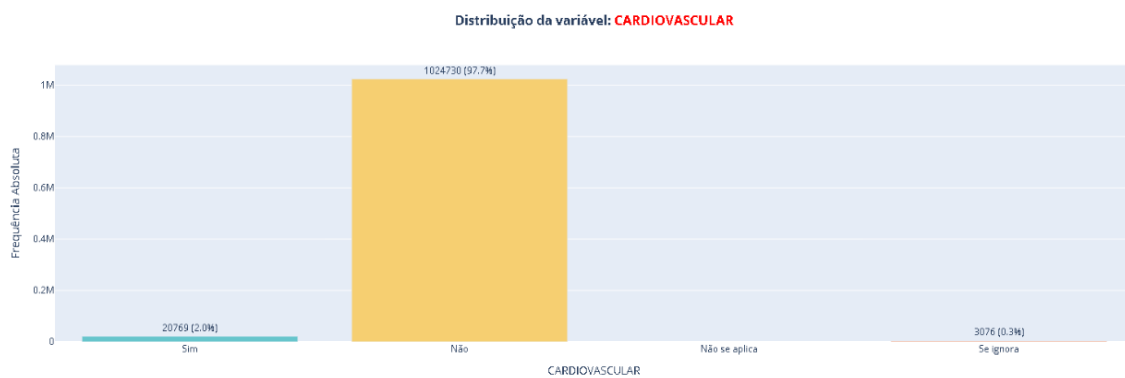
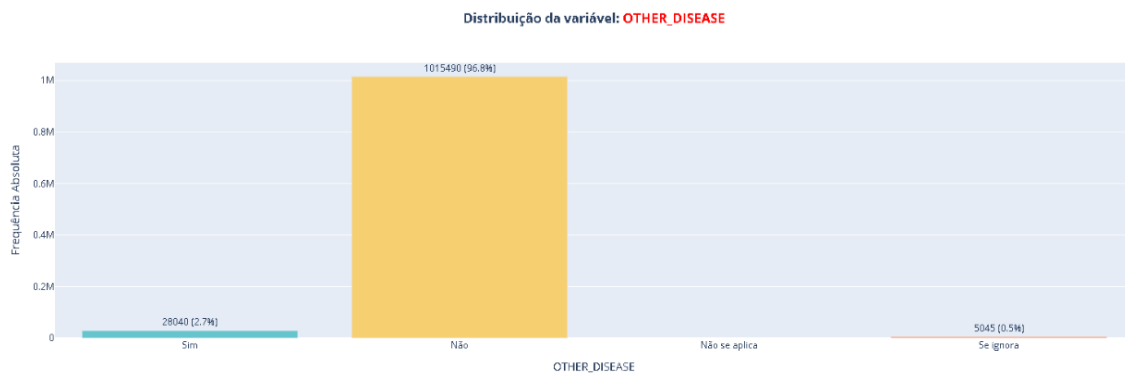
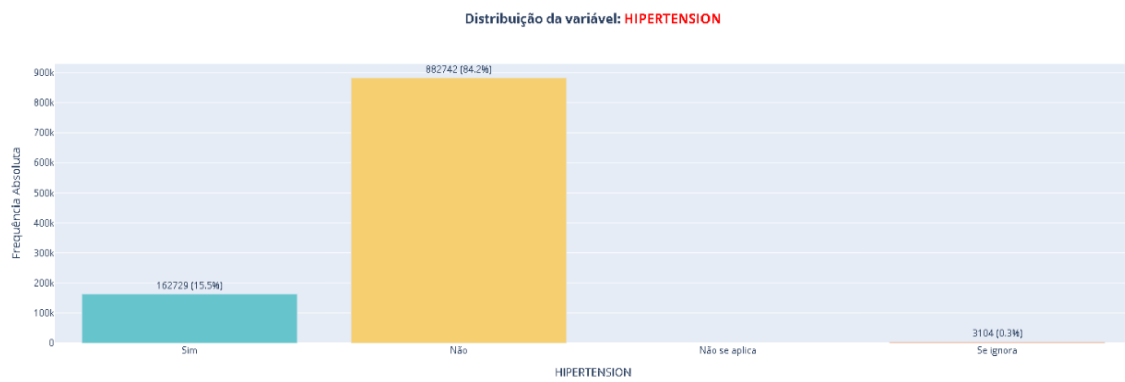
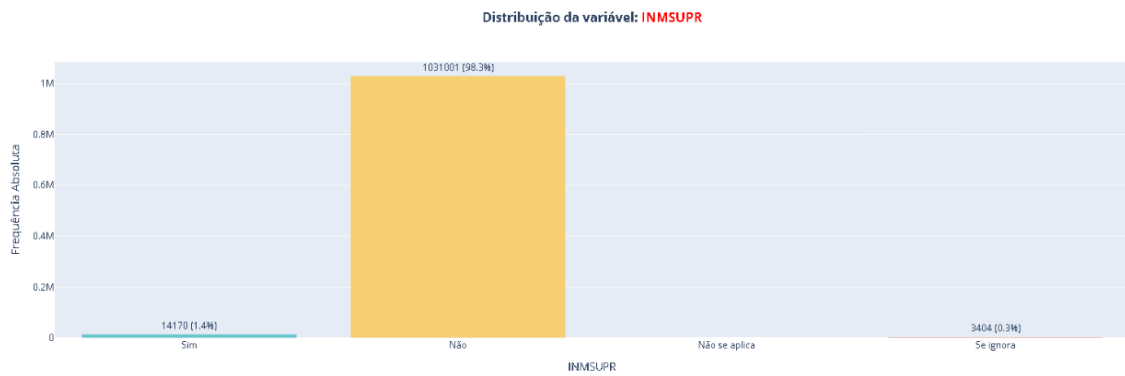


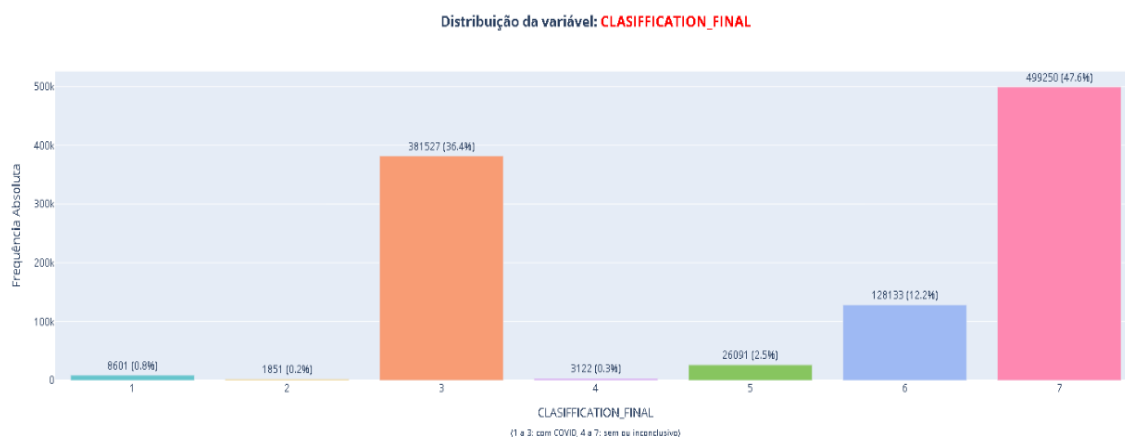
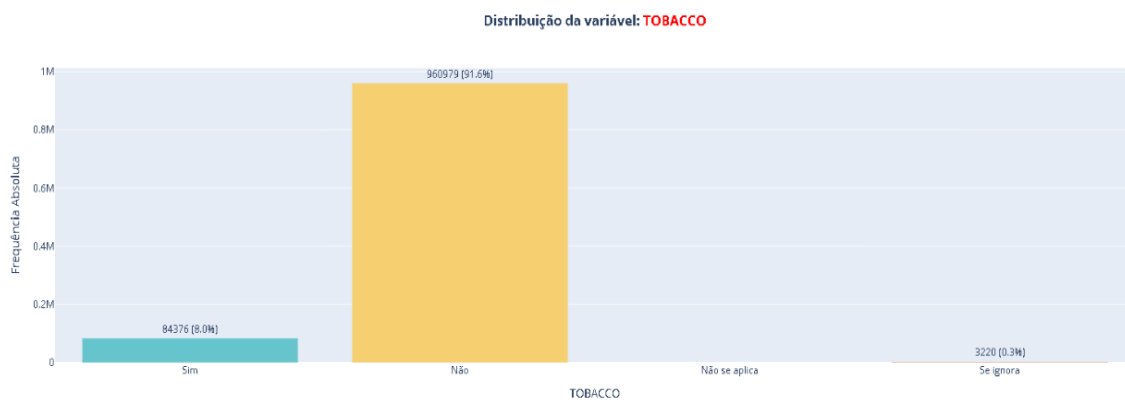
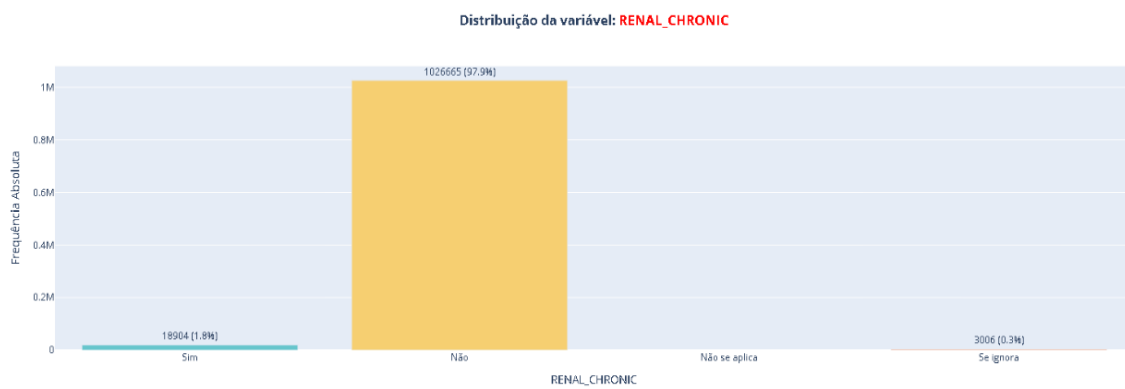
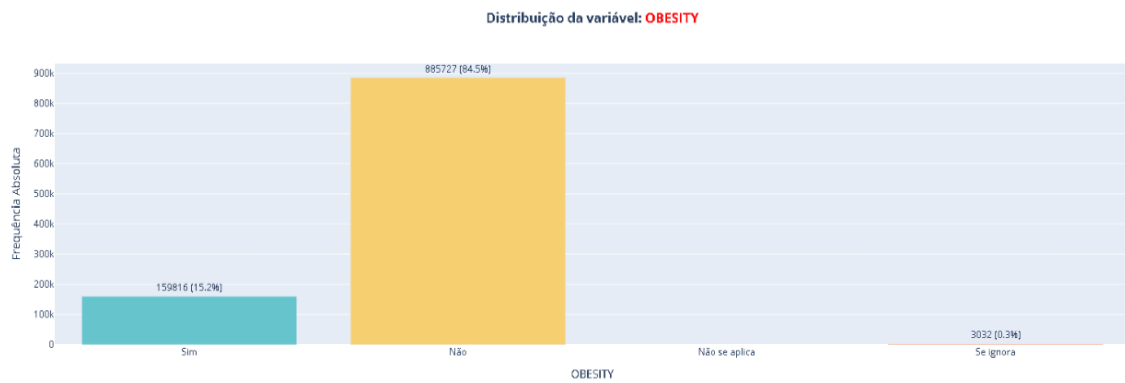
Distribuição da variável: **COPD**



Distribuição da variável: **ASTHMA**







Os principais **insights extraídos** dessa análise descritiva inicial das variáveis foram:

- ❖ Variável **USMER** - Cerca de **37%** dos pacientes foram tratados em uma unidade de vigilância epidemiológica do governo mexicano.
- ❖ Variável **MEDICAL\_UNIT** - Cerca de **87%** dos pacientes foram tratados em dois tipos distintos de unidades médicas, entre 14 existentes. No entanto, não foi possível ter clareza o que significam os domínios dessa variável.
- ❖ Variável **SEX** - A distribuição de homens e mulheres na base é extremamente equilibrada, cerca de **50%** para cada.
- ❖ Variável **PATIENT\_TYPE** - Cerca de **81%** dos pacientes não precisaram ser internados, receberam tratamento ambulatorial.
- ❖ Variável **DATE\_DIED** - Cerca de **7%** dos pacientes faleceram por conta da Covid-19 e a maioria de óbitos ocorreu entre mai/20 e jul/20.
- ❖ Variável **INTUBED** - Cerca de **3%** dos pacientes necessitaram de entubamento, como consequência de um agravamento da doença.
- ❖ Variável **PNEUMONIA** - Cerca de **13%** foram diagnosticados com pneumonia.
- ❖ Variável **AGE** - A distribuição da idade dos pacientes é simétrica, com uma maior concentração em cerca de **35-40 anos**.
- ❖ Variável **PREGNANT** - A incidência de gravidez na base foi de menos de **1%**.
- ❖ Variável **DIABETES** - Cerca de **12%** dos pacientes apresentavam quadro de diabetes.
- ❖ Variável **DPOC** - Pouco mais de **1%** dos pacientes apresentavam quadro de DPOC (distúrbio progressivo obstrutivo crônico).
- ❖ Variável **ASHTMA** - Cerca de **3%** dos pacientes apresentavam quadros de asma.
- ❖ Variável **INMSUPR** - Pouco mais de **1%** dos pacientes apresentavam quadro de imunossupressão.
- ❖ Variável **HIPERTENSION** - Ao redor de **16%** dos pacientes apresentavam quadro de hipertensão arterial.
- ❖ Variável **OTHER\_DISEASE** - Pouco menos de **3%** dos pacientes apresentavam registros de outras doenças.
- ❖ Variável **CARDIOVASCULAR** - Ao redor de **2%** dos pacientes apresentavam doenças cardiovasculares.
- ❖ Variável **OBESITY** - Cerca de **15%** dos pacientes apresentavam quadros de obesidade.
- ❖ Variável **RENAL\_CHRONIC** - Pouco menos de **2%** dos pacientes apresentavam contexto de doença renal crônica.
- ❖ Variável **TOBACCO** - Por volta de **8%** dos pacientes apresentavam quadro de tabagismo.
- ❖ Variável **CLASSIFICATION\_FINAL** - Cerca de **37%** dos pacientes foram diagnosticados com Covid-19.
- ❖ Variável **ICU** - Pouco menos de **2%** dos pacientes precisaram de UTI por precaução ou gravidade do quadro.

## **Fase VI – Definição da variável resposta do modelo preditivo**

Considerando que o objetivo central da Análise I é construir um modelo de aprendizado de máquina que, dado os sintomas atuais, o estado clínico e o histórico médico de um paciente com Covid-19, preveja um estado futuro do paciente, é possível pensar em diferentes configurações da variável resposta (que foi chamada de “TARGET”). Por exemplo, uma configuração bastante relevante do ponto de vista clínico é se o paciente possuía alto risco de apresentar complicações da doença ou não. Essa informação não aparecia de forma explícita na



base, mas uma forma de deduzi-la foi construir uma regra de negócio a partir de variáveis existentes na base. Esse exemplo recém citado poderia estar relacionado ao grau de severidade do paciente (alta, não alta), de forma que o modelo pudesse prever, de forma antecipada, a probabilidade de o paciente desenvolver a forma grave da doença e, a partir dessa predição, estabelecer um processo de triagem mais eficiente, priorizando pacientes com maior tendência a apresentarem complicações e, portanto, necessitassem de recursos especializados como respirador e UTI. Usando informações da base de dados, considerou-se seguinte regra de negócio para construção da variável resposta para o modelo de propensão: *O paciente é considerado grave se tiver sido intubado, ido para a UTI ou tenha falecido.* Assim, o modelo preditivo estimou a probabilidade de o paciente apresentar maior risco de complicações e, portanto, necessitar desses recursos.

#### Etapa 6: Teste de 2 configurações de variável resposta para o desenvolvimento de um modelo preditivo

```
#Teste de duas diferentes configurações da variável resposta
import pandas as pd
import plotly.express as px

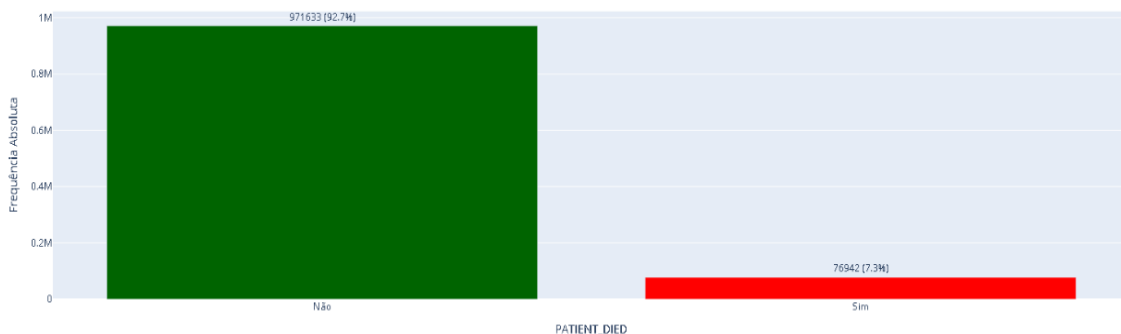
# Cenário I - Criar variável 'morreu'
Covid['PATIENT_DIED'] = Covid['DATE_DIED'].apply(lambda x: '0' if x == '9999-99-99' else '1')

# Cenário II - Criar variável 'alto_risco'
Covid['HIGH_RISK'] = (
    (Covid['INTUBED'] == 1) |
    (Covid['ICU'] == 1) |
    (Covid['PATIENT_DIED'] == '1')
).astype(int).astype(str)

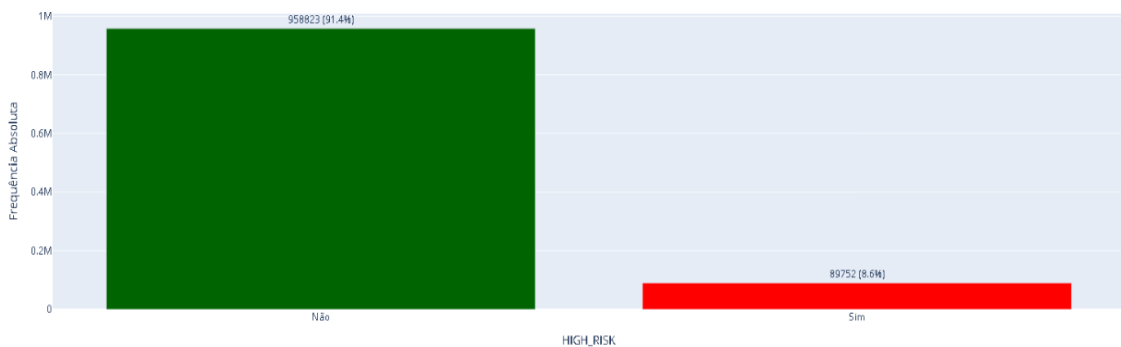
# Função com rótulos "Sim"/"Não" e títulos formatados
def plot_target_distribution(Covid, target_col):
    label_map = {'0': 'Não', '1': 'Sim'}
    series = Covid[target_col].map(label_map)

    freq_df = (
        series.value_counts(dropna=False)
        .rename_axis(target_col)
        .reset_index(name='Frequencia')
    )
```

Distribuição da variável: **PATIENT\_DIED**



Distribuição da variável: **HIGH\_RISK**



- ❖ Variável **PATIENT\_DIED** - Considera quem morreu ou não, o que ocorreu com cerca de **7%** dos pacientes.
- ❖ Variável **HIGH\_RISK** - Cerca de **9%** dos pacientes evoluíram a quadros graves ou vieram a falecer. Essa é a variável resposta a ser modelada. O grupo modelou a probabilidade de pacientes chegarem a estes estágios, de forma a priorizar atendimentos em um momento inicial, de uma maneira mais eficiente.

## Fase VII – Análise dos dados faltantes e interpretação do código 97

**Etapa 7: Análise do percentual de dados faltantes por variável**

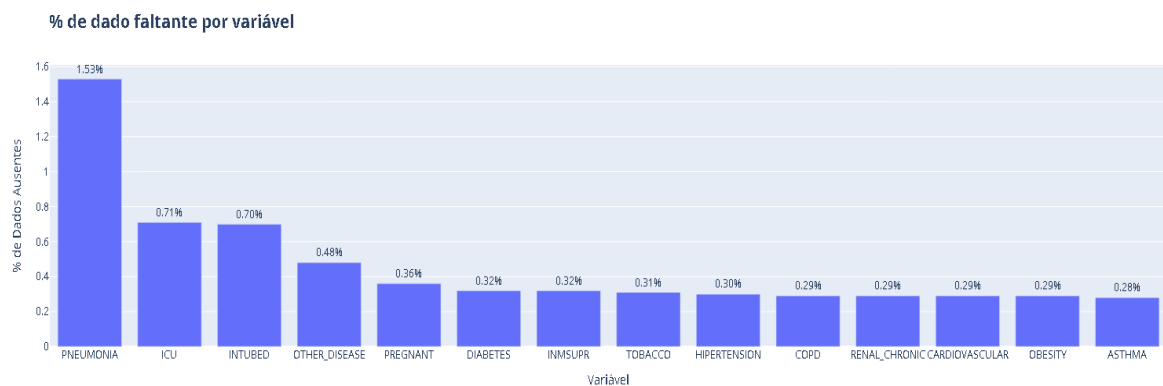
```
[ ] # Percentual de dado faltante por variável
import pandas as pd
import plotly.express as px

# Identifica o nome exato da coluna de data de óbito
col_died = next((col for col in Covid.columns if col.lower() == 'date_died'), None)

if not col_died:
    raise ValueError("A variável 'date_died' não foi encontrada no DataFrame.")

# Lista de colunas do DataFrame, exceto 'AGE'
variaveis = [col for col in Covid.columns if col != 'AGE']

# Função para contar dados ausentes
def contar_missing(col):
    if col == col_died:
        valores_str = Covid[col].astype(str).str.strip()
        missing = valores_str.eq('9999-99-99') | Covid[col].isna()
    else:
        missing = Covid[col].isin([98, 99])
```



Obs: As variáveis 'DATE\_DIED' e 'AGE' foram excluídas do gráfico.  
 'DATE\_DIED' possui 971,633 registros ausentes (92.66%),  
 que indicam pacientes que não morreram e, portanto, não têm data de óbito.  
 'AGE' foi removida pois os valores 98 e 99 representam idades válidas.

- ❖ As variáveis **DATE\_DIED** e **AGE** não foram consideradas nesse gráfico
- ❖ O % de dados ausentes foi relativamente pequeno no geral
- ❖ A variável **PNEUMONIA** foi a que teve maior % de dados ausentes
- ❖ O código **97** não é dado faltante e isso será mostrado a seguir.

#### Etapa 8: Análise do código 97 para as variáveis do dataset

```
[ ] # Cópia da base original
df = Covid.copy()

# Dicionários corrigidos de rótulos
sexo_labels = {
    1: 'Feminino',
    2: 'Masculino',
    99: 'Ignorado',
    9: 'Ignorado'
}

gravidez_labels = {
    1: 'Sim',
    2: 'Não',
    97: 'Não Aplicável',
    98: 'Ignorado',
    99: 'Desconhecido'
}

intubed_labels = {
    1: 'Sim',
```

Cruzamento entre PREGNANT e SEX (quantidades):

SEX_LABEL	Feminino	Masculino
PREGNANT_LABEL		
Ignorado	3754	0
Não	513179	0
Não Aplicável	0	523511
Sim	8131	0

Cruzamento entre INTUBED e PATIENT\_TYPE (quantidades):

PATIENT_TYPE_LABEL	Ambulatório	Internado
INTUBED_LABEL		
Desconhecido	0	7325
Não	0	159050
Não Aplicável	848544	0
Sim	0	33656

Cruzamento entre ICU e PATIENT\_TYPE (quantidades):

PATIENT_TYPE_LABEL	Ambulatório	Internado
ICU_LABEL		
Desconhecido	0	7488
Não	0	175685
Não Aplicável	848544	0
Sim	0	16858

- ❖ Para a variável **PREGNANT**, 100% dos pacientes classificados como **97** são do sexo **MASCULINO**.
- ❖ Para a variável **INTUBED**, 100% dos pacientes classificados como **97** não foram internados (somente tratamento ambulatorial).
- ❖ Para a variável **ICU**, 100% dos pacientes classificados como **97** não foram internados (somente tratamento ambulatorial).
- ❖ A conclusão é que, para essas 3 variáveis, o código **97** pode ser interpretado como **NÃO**.

## 4-3 Pré-Processamento de Dados para o Dataset de Registro de Pacientes

### Fase VIII – Exclusão dos dados inválidos e recodificação da base

#### Etapa 9: Exclusão dos dados inválidos e recodificação do dataset

```
# Passo 1: Criar variável 'MORREU'
Covid['PATIENT_DIED'] = Covid['DATE_DIED'].apply(lambda x: '0' if x == '9999-99-99' else '1')

# Passo 2: Criar variável 'ALTO_RISCO'
Covid['HIGH_RISK'] = (
    (Covid['INTUBED'] == 1) |
    (Covid['ICU'] == 1) |
    (Covid['PATIENT_DIED'] == '1')
).astype(int).astype(str)

# Passo 3: Preparar lista de colunas para limpeza (exceto AGE)
cols_to_clean = [col for col in Covid.columns if col != 'AGE']

# Cópia da base original
Covid_rec = Covid.copy()

# Contagem inicial
total_inicial = len(Covid_rec)

# Substituir 97 por 2 em todas as colunas exceto AGE
Covid_rec[cols_to_clean] = Covid_rec[cols_to_clean].replace(97, 2)
```

Total de registros antes da limpeza: 1048575  
Total de registros após a limpeza: 1019666  
Total de registros excluídos: 28909

- ❖ A base teve **28.909** linhas excluídas por conta dos códigos 98 e 99, que não tem informação.
- ❖ O código **97** foi recodificado para **2**, que significa **NÃO**

- ❖ As variáveis **PATIENT\_DIED** e **HIGH\_RISK** foram incorporadas à base recodificada.
- ❖ A base recodificada tem **1.019.666** linhas

### Fase IX – Criação de novas variáveis na base de dados

**Etapa 10: Criação de novas variáveis na base**

```

import pandas as pd

# Lista de comorbidades
COMORBIDITIES = [
    'DIABETES', 'COPD', 'ASTHMA', 'INMSUPR', 'HIPERTENSION',
    'CARDIOVASCULAR', 'OBESITY', 'RENAL_CHRONIC', 'TOBACCO'
]

# COMORBIDITIES_TOTAL: 1 se tiver alguma comorbidade, 2 se não tiver nenhuma
Covid_rec['COMORBIDITIES_TOTAL'] = Covid_rec[COMORBIDITIES].apply(
    lambda x: 1 if (x == 1).any() else 2, axis=1
)

# RESPIRATORY_DISEASE: 1 se tiver ASTHMA ou COPD, 2 se não tiver nenhuma
Covid_rec['RESPIRATORY_DISEASE'] = (
    ((Covid_rec['ASTHMA'] == 1) | (Covid_rec['COPD'] == 1))
    .apply(lambda x: 1 if x else 2)
)

# RISK_PREGNANT: 1 se SEX == 1 (feminino) e AGE > 35, caso contrário 0

```

TENSION	OTHER_DISEASE	CARDIOVASCULAR	OBESITY	RENAL_CHRONIC	TOBACCO	CLASSIFICATION_FINAL	ICU	PATIENT_DIED	HIGH_RISK	COMORBIDITIES_TOTAL	RESPIRATORY_DISEASE	RISK_PREGNANT	AGE_GROUP	AGE_RISK
1	2	2	2	2	2	3	2	1	1	1	2	1	acima de 60	1
1	2	2	1	1	2	5	2	1	1	1	2	0	acima de 60	1
2	2	2	2	2	2	3	2	1	1	1	2	0	46-60	0
2	2	2	2	2	2	7	2	1	1	2	2	1	46-60	0
1	2	2	2	2	2	3	2	1	1	1	2	0	acima de 60	1

Foram criadas variáveis para descrever grupos de risco, bem como resumir comorbidades e doenças respiratórias.

### Fase X – Análise descritiva das novas variáveis da base

**Etapa 11: Análise descritiva das novas variáveis**

```

import pandas as pd
import plotly.express as px

# Garantir que todas as colunas sejam exibidas
pd.set_option('display.max_columns', None)

# Converter AGE_GROUP para string, se estiver como categoria
Covid_rec['AGE_GROUP'] = Covid_rec['AGE_GROUP'].astype(str)

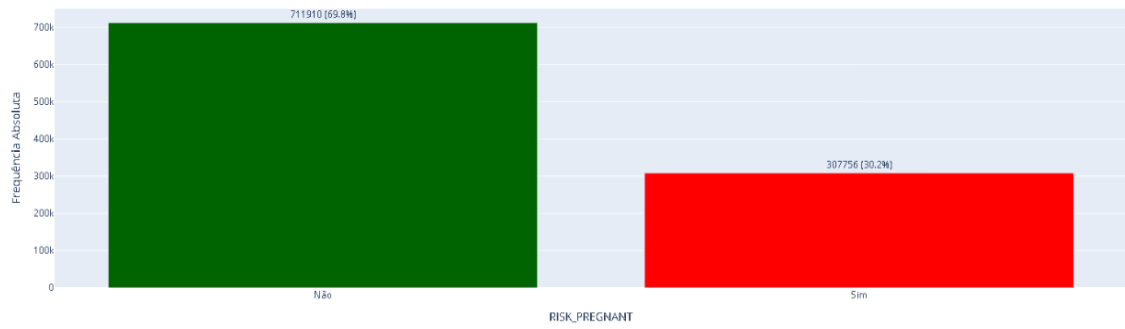
# Função genérica para variáveis binárias ou 1/2 (Sim/Não)
def plot_target_distribution(Covid, target_col, labels_map=None):
    if labels_map is None:
        labels_map = {0: 'Não', 1: 'Sim'}
    series = Covid[target_col].map(labels_map)

    freq_df = (
        series.value_counts(dropna=False)
        .rename_axis(target_col)
        .reset_index(name='frequencia')
    )
    freq_df['percentual'] = (freq_df['frequencia'] / freq_df['frequencia'].sum()) * 100

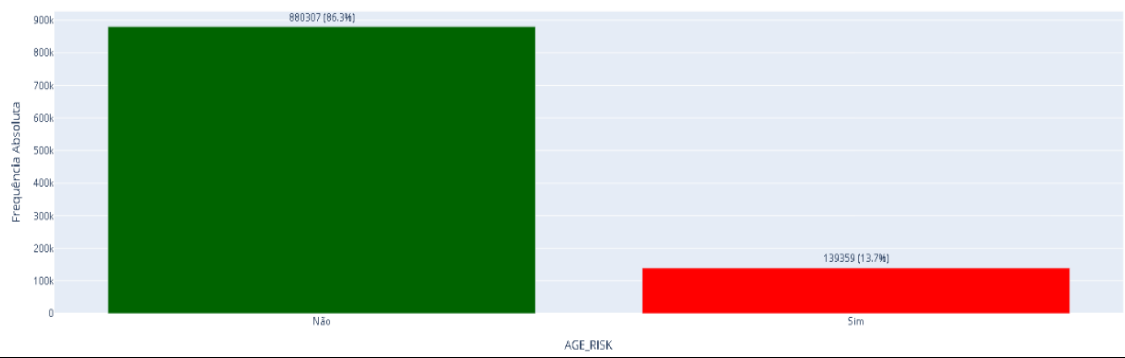
```

Nos gráficos a seguir refletem a distribuição de cada uma das variáveis adicionais criadas na base de dados de pacientes atendidos pelo governo mexicano.

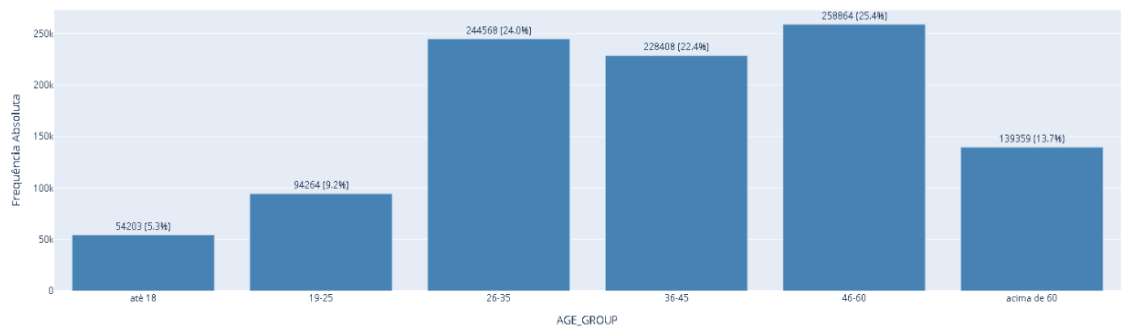
Distribuição da variável: **RISK\_PREGNANT**



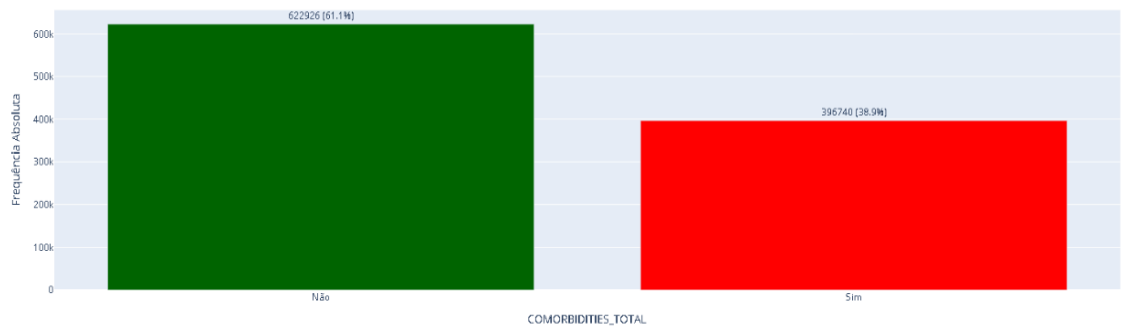
Distribuição da variável: **AGE\_RISK**

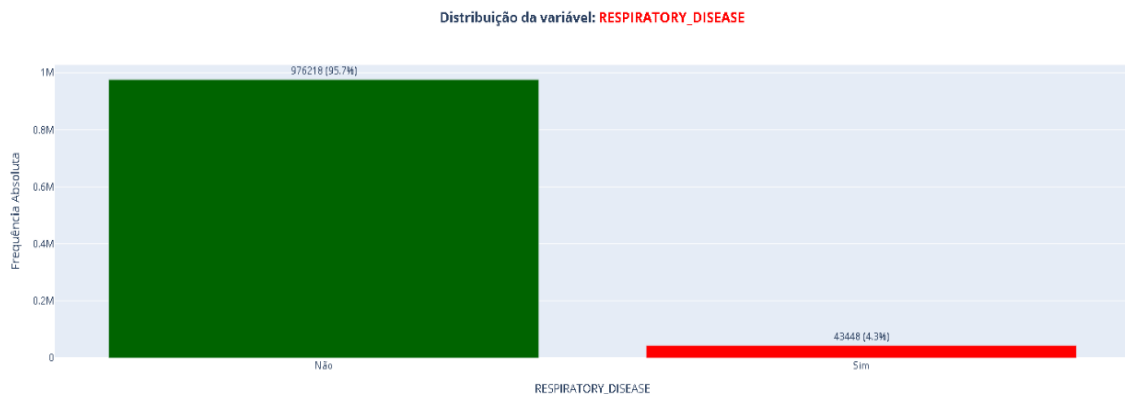


Distribuição da variável: **AGE\_GROUP**



Distribuição da variável: **COMORBIDITIES\_TOTAL**





## Fase XI – Pré-processamento da base

### Etapa 12: Pré-processamento adicional da base (recodificação das variáveis)

```
import pandas as pd

# 1. Remover colunas indesejadas
colunas_excluir = ['AGE', 'DATE_DIED', 'INTUBED', 'ICU', 'CLASSIFICATION_FINAL', 'PATIENT_DIED']
Covid_rec_II = Covid_rec.drop(columns=colunas_excluir)

# 2. Aplicar One-Hot Encoding apenas em IDADE_GRUPO
Covid_rec_II = pd.get_dummies(Covid_rec_II, columns=['AGE_GROUP'], drop_first=True)

# 3. Reorganizar colunas para colocar ALTO_RISCO no início
colunas_ordenadas = ['HIGH_RISK'] + [col for col in Covid_rec_II.columns if col != 'HIGH_RISK']
Covid_rec_II = Covid_rec_II[colunas_ordenadas]

# 4. Convertendo False/True para binários numéricos
Covid_rec_II = Covid_rec_II.astype({col: int for col in Covid_rec_II.select_dtypes(include='bool').columns})
```

	HIGH_RISK	USP/R	MEDICAL_UNIT	SEX	PATIENT_TYPE	PNEUMONIA	PREGNANT	DIABETES	COPD	ASTHMA	IN/SUPR	HIPERTENSION	OTHER_DISEASE	CARDIOVASCULAR	OBESITY	RENAL_CHRONIC	TOBACCO	COMORBIDITIES_TC
0	1	0	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0
1	1	0	1	0	1	1	0	0	0	0	0	1	0	0	1	1	1	0
2	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
3	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
4	1	0	1	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0

Nessa fase, as variáveis foram todas recodificadas para 0 e 1, bem como foi aplicado o processo de *one hot encoding* para variáveis categóricas com mais níveis, para preparar o processo para a etapa de desenvolvimento de modelos preditivos.

## 4-4 Desenvolvimento do Modelo para o Dataset de Registro de Pacientes

### Fase XII - Separação em bases de treino, validação e teste

#### Etapa 13: Separação da base em treino, validação e teste

```
from sklearn.model_selection import train_test_split
import pandas as pd

# 1. Separar X e y
X = Covid_rec_II.drop(columns=['HIGH_RISK'])
y = Covid_rec_II['HIGH_RISK']

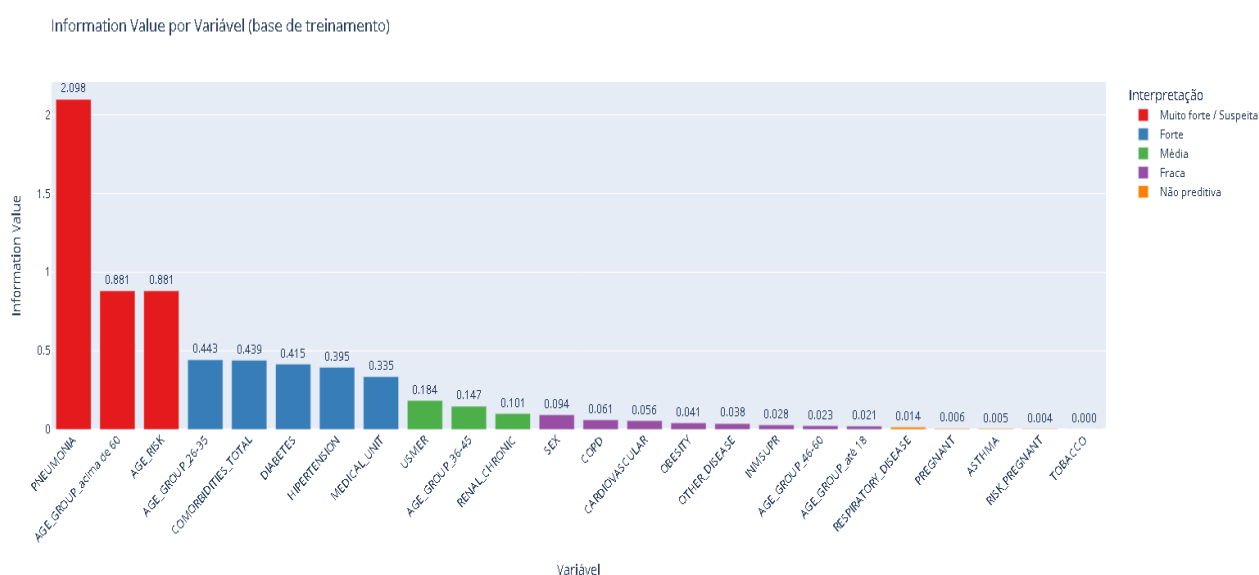
# 2. Separar treino (60%) e teste (40%)
X_train, X_temp, y_train, y_temp = train_test_split(
    X, y, test_size=0.4, random_state=42, stratify=y
)

# 3. Separar temp em validação (20%) e teste (20%)
X_valid, X_test, y_valid, y_test = train_test_split(
    X_temp, y_temp, test_size=0.5, random_state=42, stratify=y_temp
)
```

ILAR	OBSIDITY	RENAL_CHRONIC	TOBACCO	COMORBIDITIES_TOTAL	RESPIRATORY_DISEASE	RISK_PREGNANT	AGE_RISK	AGE_GROUP_26-35	AGE_GROUP_36-45	AGE_GROUP_46-60	AGE_GROUP_acima de 60	AGE_GROUP_até 18	HIGH_RISK	SAMPLE
0	0	0	0	0	0	0	0	0	0	0	0	1	0	TRAINING
0	0	0	0	0	0	1	0	0	0	1	0	0	0	TRAINING
0	0	0	0	0	0	1	0	0	1	0	0	0	0	TRAINING
0	0	0	0	0	0	0	1	0	0	0	1	0	0	TRAINING
0	0	0	0	0	0	0	0	0	0	1	0	0	0	TRAINING

A base de dados foi particionada em 3 grupos: 60% foram destinados a treinamento do modelo, 20% foram destinados a teste para avaliação dos resultados do modelo preditivo e outros 20% foram dedicados à validação do modelo, para efeito de estabilidade e capacidade de generalização.

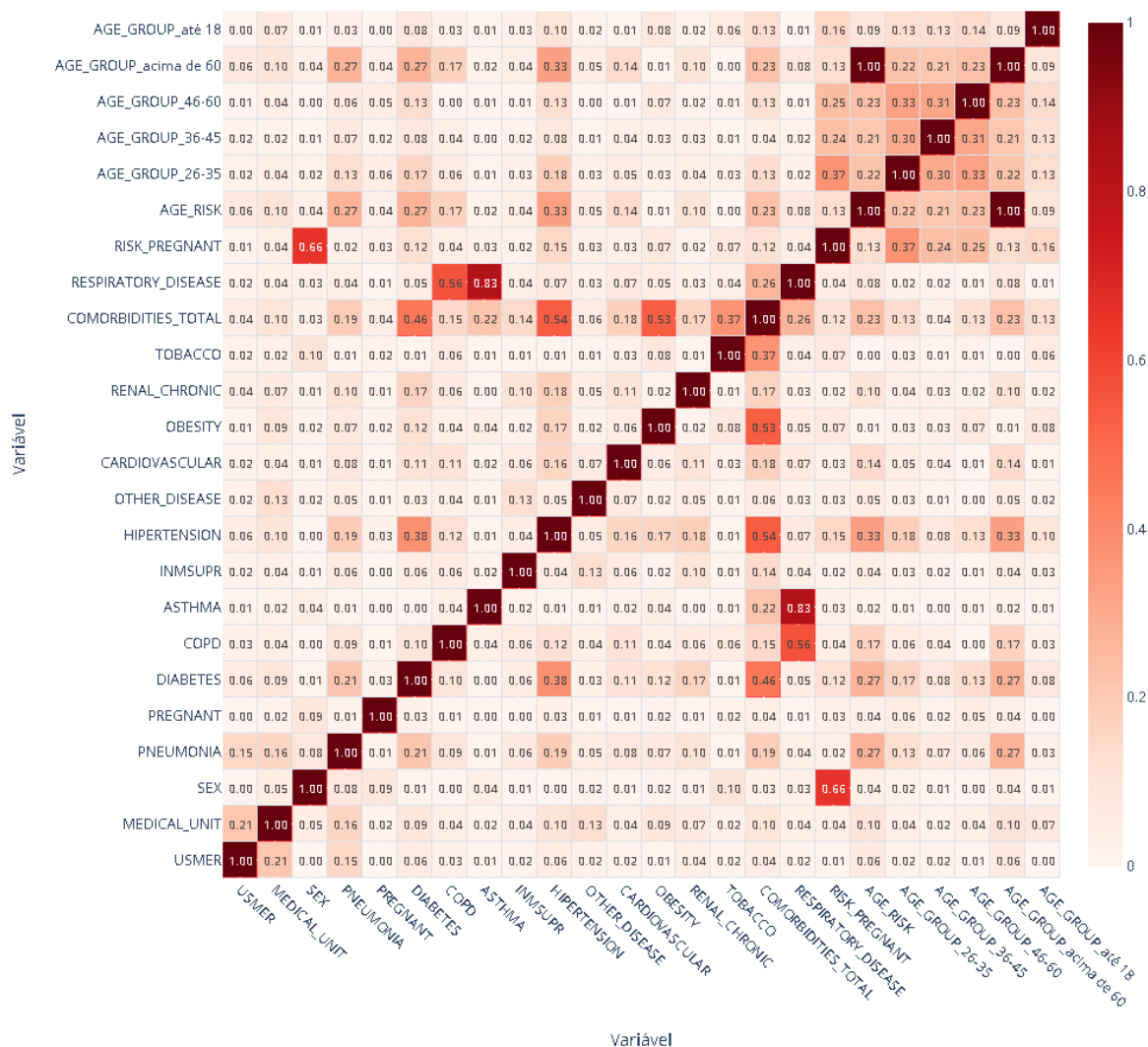
### Fase XIII – Análise da importância individual das variáveis na base de treinamento



- ❖ A métrica **INFORMATION VALUE (IV)** é fantástica para a avaliação da força preditiva individual de variáveis categóricas, sobretudo quando a variável resposta do modelo também é categórica, que é o caso em questão.
- ❖ As variáveis foram ordenadas de acordo com sua força preditiva (FORTE, MÉDIA E FRACA), conforme sugestão em Naeem Siddiqi no artigo publicado por LIN (2013), no SAS Global Forum, em que valores de IV acima de 0,5 retratam variáveis com altíssimo poder preditivo, valores de IV entre 0,3 e 0,5 referem-se a variáveis com forte poder preditivo, valores de IV entre 0,1 e 0,3 destacam variáveis com médio poder preditivo, enquanto variáveis com IV inferior a 0,1 refletem baixo poder preditivo. Há variáveis como PNEUMONIA, com altíssimo poder preditivo e variáveis como AGE\_GROUP\_acima\_60anos e AGE\_RISK que possuem o mesmo valor de IV, sugerindo que possam estar trazendo exatamente a mesma informação e, portanto, podendo incorporar informações redundantes a um futuro modelo preditivo. Isso foi analisado mais adiante.

## Fase XIV – Análise da correlação entre as variáveis

Matriz de Correlação (Cramer's V) — Base de Treinamento



- ❖ A métrica **V DE CRAMER** é a mais indicada quando se deseja avaliar correlação entre variáveis categóricas nominais, que é o caso das variáveis da base de estudo.
- ❖ Há alguns pares de variáveis com correlação acima de **0.4** (convenção) que merecem mais atenção.
- ❖ Essas variáveis podem estar trazendo a mesma informação, o que poderia sugerir que uma delas pode ser desconsiderada da modelagem.
- ❖ Uma análise de VIF (fatores de inflação da variância) pode ser útil para detectar potencial presença de multicolinearidade.



## Fase XV – Análise do VIF (Fator de Inflação da Variância) para detecção de multicolinearidade

Etapa 16: Análise do VIF (Variance Inflation Factor)		
<pre>import pandas as pd from statsmodels.stats.outliers_influence import variance_inflation_factor import statsmodels.api as sm from IPython.display import display  # Filtrar a base de treino df_treino = Covid_com_amostra[Covid_com_amostra['SAMPLE'] == 'TRAINING'].copy()  # Lista das variáveis preditoras (todas numéricas conforme informado) variaveis_preditoras = [     'USMER', 'MEDICAL_UNIT', 'SEX', 'PATIENT_TYPE', 'PNEUMONIA', 'PREGNANT',     'DIABETES', 'COPD', 'ASTHMA', 'INMSUPR', 'HIPERTENSION', 'OTHER_DISEASE',     'CARDIOVASCULAR', 'OBESITY', 'RENAL_CHRONIC', 'TOBACCO', 'COMORBIDITIES_TOTAL',     'RESPIRATORY_DISEASE', 'RISK_PREGNANT', 'AGE_RISK',     'AGE_GROUP_26-35', 'AGE_GROUP_36-45', 'AGE_GROUP_46-60',     'AGE_GROUP_acima de 60', 'AGE_GROUP_até 18' ]  # Subconjunto com as variáveis preditoras X = df_treino[variaveis_preditoras].copy()  # Adiciona constante (intercepto) X = sm.add_constant(X)</pre>		
Índice	Variável	VIF
0	AGE_GROUP_acima de 60	inf
1	AGE_RISK	inf
2	RESPIRATORY_DISEASE	41.317238
3	ASTHMA	27.948515
4	COPD	12.860825
5	AGE_GROUP_46-60	3.810816
6	COMORBIDITIES_TOTAL	3.603792
7	RISK_PREGNANT	3.599645
8	AGE_GROUP_36-45	3.469154
9	AGE_GROUP_26-35	2.746286
10	SEX	2.646693
11	PATIENT_TYPE	1.999885
12	PNEUMONIA	1.780403
13	HIPERTENSION	1.734353

A análise do VIF aponta que há variáveis com valores de VIF muito superiores a 5, como sugere a literatura internacional para detectar presença de um fenômeno chamado de multicolinearidade, que nada é mais do que a inflação de parâmetros de variabilidade das variáveis de modelos preditivos, muito fortemente causadas pela redundância de variáveis, que basicamente trazem conteúdos muito parecidos e, em grande parte dos casos, apresentam correlações elevadas entre si. Esse processo de inflação da variância deixa os modelos preditivos muito instáveis, pouco confiáveis e com baixa capacidade de generalização. Assim, evitar que variáveis com VIF maiores que 5 sejam usadas na modelagem é uma boa prática para obtenção de modelos mais robustos e confiáveis.

A métrica VIF (*Variance Inflation Factor*) aponta forte presença de multicolinearidade, principalmente em função de algumas variáveis que estão gerando redundância no processo, por trazerem a mesma informação (AGE\_GROUP\_acima de 60 vs AGE\_RISK, RESPIRATORY\_DISEASE vs ASHTMA). Vamos executar mais uma vez o VIF, porém sem as variáveis AGE\_GROUP e sem ASHTMA. Após a retirada de variáveis redundantes, a reanálise do VIF foi realizada e apresentou os seguintes resultados:

Etapa 17: Reanálise do VIF (Variance Inflation Factor) após retirada de algumas variáveis		
<pre>import pandas as pd from statsmodels.stats.outliers_influence import variance_inflation_factor import statsmodels.api as sm from IPython.display import display  # Filtrar a base de treino df_treino = Covid_com_amostra[Covid_com_amostra['SAMPLE'] == 'TRAINING'].copy()  # Lista das variáveis preditoras (sem ASTHMA e AGE_GROUP) variaveis_preditoras = [     'USMER', 'MEDICAL_UNIT', 'SEX', 'PATIENT_TYPE', 'PNEUMONIA', 'PREGNANT',     'DIABETES', 'COPD', 'INMSUPR', 'HIPERTENSION', 'OTHER_DISEASE',     'CARDIOVASCULAR', 'OBESITY', 'RENAL_CHRONIC', 'TOBACCO',     'COMORBIDITIES_TOTAL', 'RESPIRATORY_DISEASE', 'RISK_PREGNANT', 'AGE_RISK' ]  # Subconjunto com as variáveis preditoras X = df_treino[variaveis_preditoras].copy()  # Adiciona constante (intercepto) X = sm.add_constant(X)</pre>		
Índice	Variável	VIF
0	COMORBIDITIES_TOTAL	3.584321
1	PATIENT_TYPE	1.973203
2	RISK_PREGNANT	1.930761
3	SEX	1.898526
4	PNEUMONIA	1.776184
5	HIPERTENSION	1.719107
6	RESPIRATORY_DISEASE	1.679866
7	OBESITY	1.644655
8	COPD	1.561446
9	DIABETES	1.474774
10	TOBACCO	1.404022
11	AGE_RISK	1.295697
12	RENAL_CHRONIC	1.074276
13	MEDICAL_UNIT	1.073231
14	INMSUPR	1.065107
15	CARDIOVASCULAR	1.062033

Após a retirada das variáveis redundante AGE\_GROUP e ASHTMA, não foram mais detectados sinais de multicolinearidade (por conta de variáveis com VIF superiores a 5). Agora, o processo de modelagem pode ser conduzido de maneira mais segura.

#### 4-4.1 Modelo de Regressão Logística e Resultados

##### **Fase XVI(a) – Primeiro modelo desenvolvido – Modelo de regressão logística**

**Etapa 18:** Modelo de regressão logística na BASE DE TREINAMENTO para prever a probabilidade do paciente evoluir para um quadro grave (ou morte)

```
import pandas as pd
from sklearn.linear_model import LogisticRegression

# 1. Filtra apenas a base de treinamento
df_treino = Covid_com_amostra[Covid_com_amostra['SAMPLE'] == 'TRAINING'].copy()

# 2. Define variáveis
target = 'HIGH_RISK'

variaveis_explicativas = [
    'USMER', 'SEX', 'PNEUMONIA', 'PREGNANT',
    'DIABETES', 'COPD', 'INMSUPR', 'HIPERTENSION', 'OTHER_DISEASE',
    'CARDIOVASCULAR', 'OBESITY', 'RENAL_CHRONIC', 'TOBACCO',
    'COMORBIDITIES_TOTAL', 'RESPIRATORY_DISEASE', 'RISK_PREGNANT', 'AGE_RISK'
]

X_train = df_treino[variaveis_explicativas]
y_train = df_treino[target]

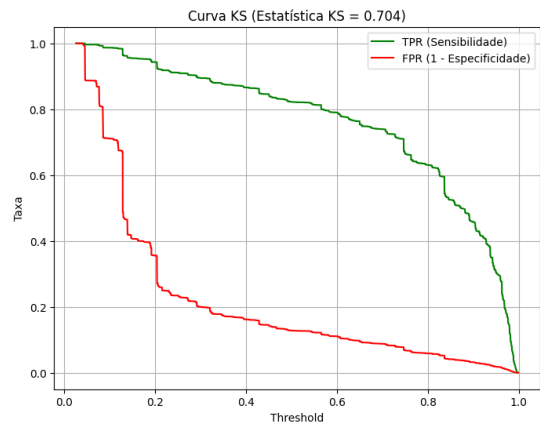
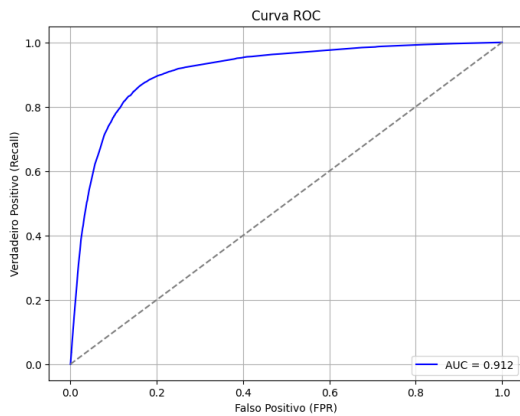
# 3. Modelo com penalização para lidar com classe desbalanceada
modelo = LogisticRegression(class_weight='balanced', max_iter=1000, solver='lbfgs')
modelo.fit(X_train, y_train)
```

Como o % de representatividade da categoria menos frequente da variável resposta (paciente com alto risco ou que morreu) era baixo, ao redor de 9%, isso caracterizava um evento raro o que tem como consequência um desequilíbrio nas taxas de acerto do modelo nas 2 categorias (alto risco e baixo risco), desfavorecendo a categoria com menos frequência, justamente a categoria de interesse para a modelagem do evento.

Dessa forma, na modelagem, é possível adotar estratégias de balanceamento da base de treinamento (seja via *oversampling* ou *undersampling*), ou trabalhar com estratégias de penalização dos erros para a categoria de alto risco, fazendo com que o modelo acerte de maneira mais equilibrada nas 2 categorias, consequentemente aumentando a taxa de acerto na categoria de interesse, que, nesse caso, é a categoria de pacientes que evoluíram para um estado grave ou vieram a falecer.

Essa mesma abordagem foi aplicada tanto nos modelos de regressão logística, como nos modelos testados na sequência, modelos de *random forest* e de árvore de classificação.

##### **Fase XVI(b) – Resultados do modelo de regressão logística**



✓ Taxa de acerto por classe (em %):

	% Predito 0	% Predito 1
HIGH_RISK		
0	87.19	12.81
1	17.75	82.25

Matriz de Confusão:

```
[[162844 23932]
 [ 3045 14113]]
```

Relatório de Classificação:


	precision	recall	f1-score	support
0	0.9816	0.8719	0.9235	186776
1	0.3710	0.8225	0.5113	17158
accuracy			0.8677	203934
macro avg	0.6763	0.8472	0.7174	203934
weighted avg	0.9303	0.8677	0.8888	203934

- ❖ O modelo de REGRESSÃO LOGÍSTICA é bastante recomendado nesse caso do estudo em questão, dado que tanto a variável resposta, quanto as variáveis preditoras são categóricas. O modelo estima a probabilidade de o paciente evoluir a um quadro grave da doença, ou mesmo ir a óbito.
- ❖ Tanto o K-S (Kolmogorov-Smirnov) do modelo, quanto o valor de AUC (Area under curve) são bastante significativos em termos de capacidade de diferenciação de pacientes com alto ou baixo potencial de evoluírem a um quadro grave de Covid-19 ou virem a falecer.
- ❖ O modelo está acertando mais de 82% dos casos de alto risco, o que é muito bom considerando que essa é a classe minoritária (evento raro) e essa taxa de acerto está bem equilibrada em ambas as classes (Pacientes com alto risco e com risco menor). Esse refere-se ao indicador chamado "Recall", a proporção de pacientes que realmente são graves/morrem e que foram corretamente identificados como tal pelo modelo.
- ❖ No entanto, o indicador "Precision" do modelo, em torno de 37% para a categoria HIGH\_RISK=1 é relativamente baixo, dado que, quando o modelo prevê "alto risco", só 37% são realmente alto risco. "Precision" é a proporção dos pacientes que o modelo classificou como graves e que realmente são graves.
- ❖ É possível tentar alterar o ponto de corte do modelo (cujo padrão para classificação é 0.5) e avaliar como isso afeta as taxas de acerto e, inclusive, o indicador de Precisão.
- ❖ Foi excluída da modelagem a variável MEDICAL\_UNIT por não ter sido possível identificar a coerência do poder preditivo dessa variável, por conta da ausência de descrição dos domínios da variável nas documentações existentes.

### Fase XVI(c) – Pesos das variáveis do modelo de regressão logística

	Variavel	Coefficiente	z	Odds_Ratio	p_valor	Significativo_5pct
0	const	-3.973883	-306.312225	0.018800	0.000000e+00	True
1	PNEUMONIA	2.771348	238.083665	15.980161	0.000000e+00	True
2	AGE_RISK	1.305563	101.474295	3.689766	0.000000e+00	True
3	SEX	-1.140290	-41.809343	0.319726	0.000000e+00	True
4	USMER	0.423521	36.784882	1.527330	3.221391e-296	True
5	RISK_PREGNANT	0.727825	25.537808	2.070573	7.500400e-144	True
6	DIABETES	0.378129	24.640880	1.459551	4.609305e-134	True
7	OTHER_DISEASE	0.613081	22.550628	1.846110	1.323946e-112	True
8	RENAL_CHRONIC	0.547859	19.671340	1.729547	3.796121e-86	True
9	COMORBIDITIES_TOTAL	0.364499	19.365695	1.439792	1.503136e-83	True
10	TOBACCO	-0.352082	-15.977964	0.703222	1.819950e-57	True
11	RESPIRATORY_DISEASE	-0.432879	-10.618418	0.648639	2.446723e-26	True
12	HIPERTENSION	0.157421	10.069973	1.170488	7.499837e-24	True
13	COPD	0.448007	8.852016	1.565189	8.595265e-19	True
14	INMSUPR	0.248433	6.956435	1.282015	3.489907e-12	True
15	OBESITY	0.107593	6.684029	1.113595	2.324601e-11	True
16	PREGNANT	0.477815	5.171695	1.612547	2.319806e-07	True
17	CARDIOVASCULAR	-0.006861	-0.237926	0.993162	8.119384e-01	False

### Fase XVI(d) – Comparação dos resultados do modelo de regressão logística, a partir do teste de distintos pontos de corte



Comparação das métricas por ponto de corte:

Threshold	Accuracy	Precision	Recall	F1-score	AUC
0	0.5	0.8677	0.3710	0.8225	0.5113
1	0.6	0.8808	0.3957	0.7898	0.5273
2	0.7	0.8977	0.4362	0.7394	0.5487

◆ Threshold = 0.5

Matriz de Confusão:

```
[[162844 23932]
 [ 3045 14113]]
```

Taxa de Acerto por Classe (%):

	% Predito 0	% Predito 1
HIGH_RISK		
0	87.19	12.81
1	17.75	82.25

◆ Threshold = 0.6

Matriz de Confusão:

```
[[166080 20696]
 [ 3606 13552]]
```

Taxa de Acerto por Classe (%):

	% Predito 0	% Predito 1
HIGH_RISK		
0	88.92	11.08
1	21.02	78.98

◆ Threshold = 0.7

Matriz de Confusão:

```
[[170376 16400]
 [ 4471 12687]]
```

Taxa de Acerto por Classe (%):

	% Predito 0	% Predito 1
HIGH_RISK		
0	91.22	8.78
1	26.06	73.94

O aumento do ponto de corte de 0.5 para 0.7 impacta em um aumento no valor da precisão (de 37% para 43%), mas consequentemente impacta na redução do recall (de 82% para 74%), ou seja, o aumento da precisão nos ajuda a reduzir os falsos positivos classificados como alto risco, mas impacta na redução de acerto do modelo em identificar pacientes de alto risco.

Entendemos que é melhor o modelo capturar um % mais alto dos casos reais de alto risco (recall), mesmo correndo o risco de que casos assinalados como alto risco sejam falsos positivos (precision). Assim, podemos manter o ponto de corte em 0.5.

#### 4-4.2 Modelo de Random Forest e Resultados

##### Fase XVI(e) – Segundo modelo desenvolvido – Modelo de Random Forest

**Etapla 22:** Modelo de random forest na BASE DE TREINAMENTO para prever a probabilidade do paciente evoluir para um quadro grave (ou morte)

```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier

# 1. Separar base de treino e base completa
df_train = Covid_com_amostra[Covid_com_amostra['SAMPLE'] == 'TRAINING'].copy()
df_full = Covid_com_amostra.copy()

# 2. Definir variáveis do modelo (sem MEDICAL_UNIT)
variaveis_modelo = [
    'USHER', 'SEX', 'PNEUMONIA', 'PREGNANT',
    'DIABETES', 'COPD', 'INMSUPR', 'HIPERTENSION', 'OTHER_DISEASE',
    'CARDIOVASCULAR', 'OBESITY', 'RENAL_CHRONIC', 'TOBACCO',
    'COMORBIDITIES_TOTAL', 'RESPIRATORY_DISEASE', 'RISK_PREGNANT', 'AGE_RISK'
]

X_train = df_train[variaveis_modelo]
y_train = df_train['HIGH_RISK'].astype(int)

# 3. Treinar modelo Random Forest
modelo_rf = RandomForestClassifier(
    n_estimators=100,
    random_state=42,
    class_weight='balanced'
```

  Avaliação do modelo Random Forest (ponto de corte = 0.5):

♦ Matriz de Confusão:

```
[[159053  27723]
 [   2723  14435]]
```

♦ Métricas de Performance:

```
Acurácia      : 0.8507
Precisão      : 0.3424
Recall        : 0.8413
F1-score      : 0.4867
AUC           : 0.9057
```

♦ Relatório de Classificação:

	precision	recall	f1-score	support
Classe 0	0.98	0.85	0.91	186776
Classe 1	0.34	0.84	0.49	17158
accuracy			0.85	203934
macro avg	0.66	0.85	0.70	203934
weighted avg	0.93	0.85	0.88	203934

♦ Taxa de Acerto por Classe (%):

	% Predito 0	% Predito 1
HIGH_RISK		
0	85.16	14.84
1	15.87	84.13

## Fase XVI(f) – Importância das variáveis de um modelo de Random Forest

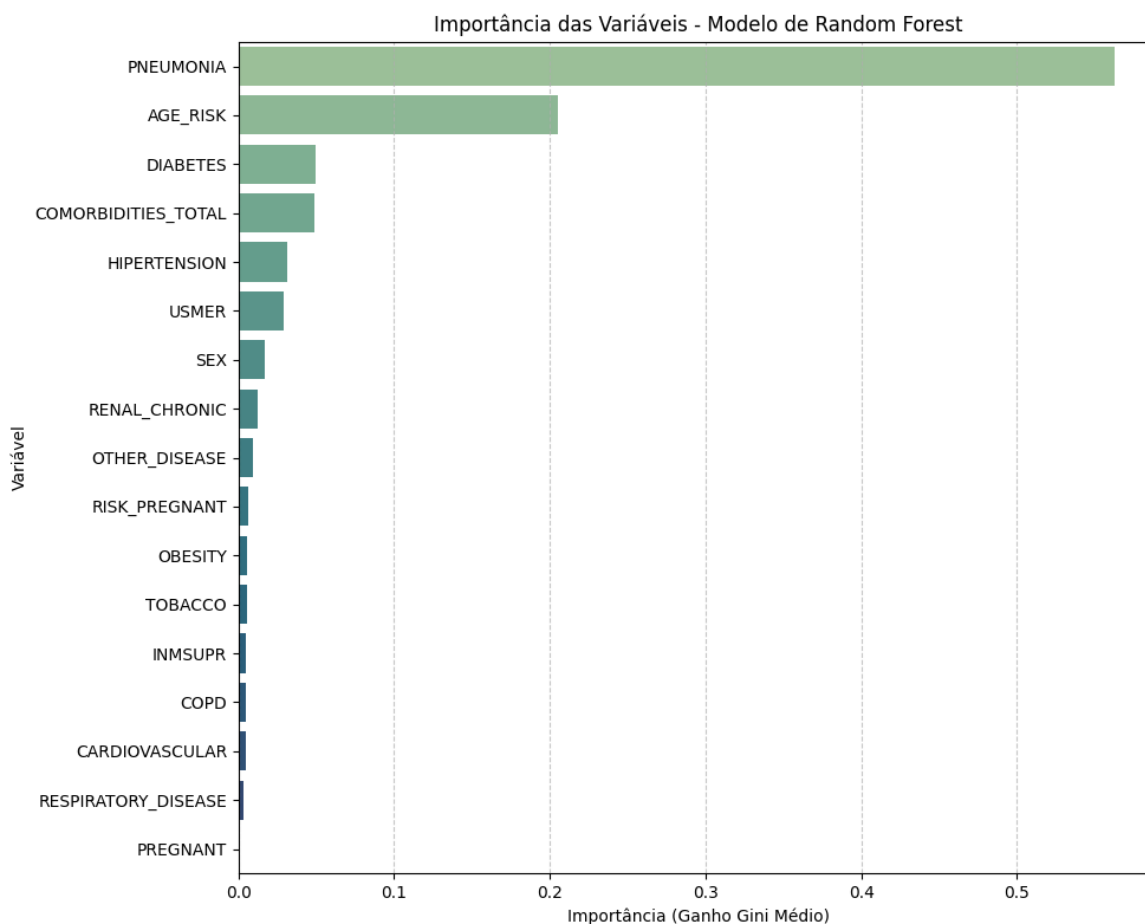
### **Etapa 23:** Importância das variáveis o modelo gerado a partir de Random Forest

```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# 1. Obter importâncias das variáveis
importancias = modelo_rf.feature_importances_
nomes_variaveis = np.array(variaveis_modelo)

# 2. Criar DataFrame e ordenar
df_importancia = pd.DataFrame({
    'Variável': nomes_variaveis,
    'Importância': importancias
}).sort_values(by='Importância', ascending=False)

# 3. Plotar gráfico
plt.figure(figsize=(10, 8))
sns.barplot(x='Importância', y='Variável', data=df_importancia, palette='crest')
plt.title('Importância das Variáveis - Modelo de Random Forest')
plt.xlabel('Importância (Ganho Gini Médio)')
plt.ylabel('Variável')
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



### 4-4.3 Comparação entre os modelos de Regressão Logística e Random Forest

#### Fase XVI(g) – Comparação entre os modelos de Regressão Logística e de Random Forest



Comparação das Métricas Gerais:

	Modelo	Accuracy	Precision	Recall	F1-score	AUC
0	Regressão Logística	0.8677	0.3710	0.8225	0.5113	0.9118
1	Random Forest	0.8507	0.3424	0.8413	0.4867	0.9057



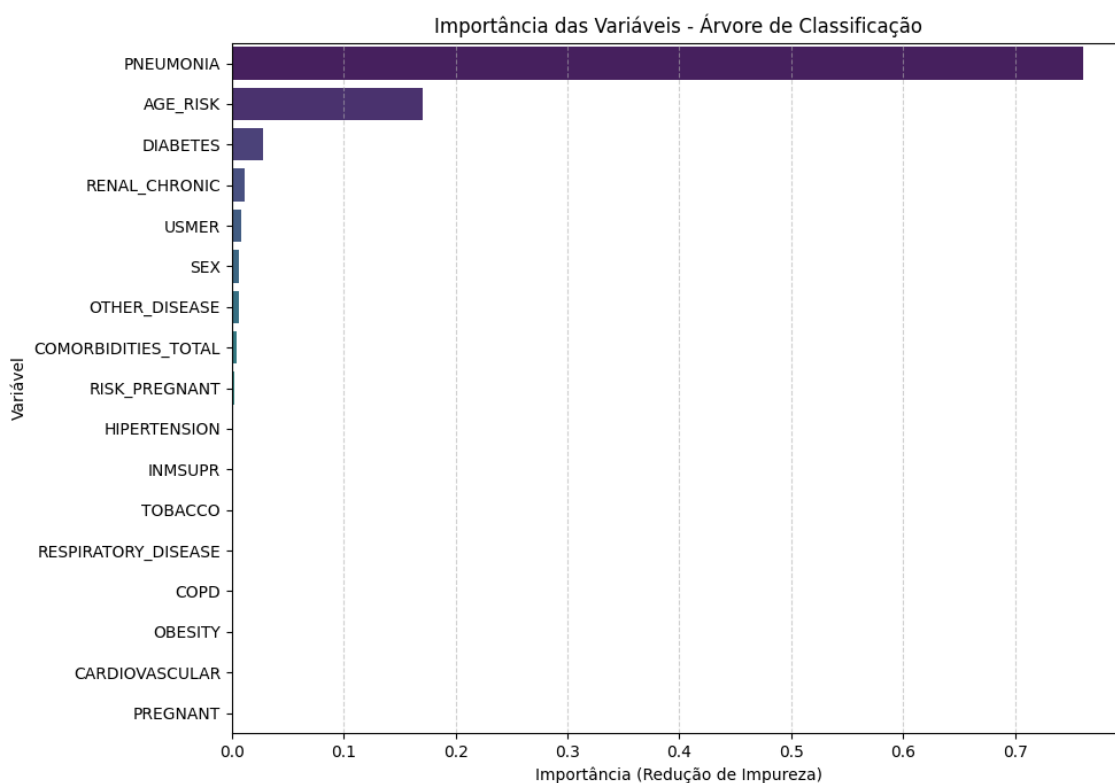
Comparação da Taxa de Acerto por Classe (%):

	% Predito 0	% Predito 1
Regressão Logística - Classe 0	87.19	12.81
Regressão Logística - Classe 1	17.75	82.25
Random Forest - Classe 0	85.16	14.84
Random Forest - Classe 1	15.87	84.13

De maneira geral, houve uma similaridade bastante importante em termos de performance dos 2 modelos. O modelo de regressão logística levou ligeira vantagem nos principais indicadores (precision, recall, AUC) e acaba sendo o modelo favorito, não apenas pelos resultados, como também pela simplicidade do modelo, em comparação ao modelo de random forest (que é uma mistura de árvores de decisão).

### 4-4.4 Modelo Árvore de Classificação

#### Fase XVI(h) – Terceiro modelo desenvolvido – Árvore de classificação



#### 4-4.5 Comparação dos resultados dos 3 Modelos (Regressão Logística, Random Forest e Árvore de Classificação)

##### Fase XVII – Comparação dos resultados dos 3 modelos (Regressão Logística, Random Forest e Árvore de Classificação)

Comparação das Métricas Gerais:

	Modelo	Accuracy	Precision	Recall	F1-score	AUC
0	Regressão Logística	0.8677	0.3710	0.8225	0.5113	0.9118
1	Random Forest	0.8507	0.3424	0.8413	0.4867	0.9057
2	Árvore de Decisão	0.8367	0.3243	0.8687	0.4723	0.9051

Comparação da Taxa de Acerto por Classe (%):

	% Predito 0	% Predito 1
Regressão Logística - Classe 0	87.19	12.81
Regressão Logística - Classe 1	17.75	82.25
Random Forest - Classe 0	85.16	14.84
Random Forest - Classe 1	15.87	84.13
Árvore de Decisão - Classe 0	83.37	16.63
Árvore de Decisão - Classe 1	13.13	86.87

De maneira geral, houve uma similaridade bastante importante em termos de performance dos 3 modelos. O modelo de árvore de regressão logística levou ligeira vantagem sobre o modelo de Random Forest e de regressão logística em quase todos os principais indicadores (accuracy, precision, F1-score, AUC). Por questões de robustez e simplicidade do modelo de regressão logística e consistência dos indicadores de performance, esse foi o modelo escolhido.

##### Fase XVIII – Avaliação da estabilidade e capacidade de generalização do modelo logístico

**Etapa 28:** Análise da estabilidade e capacidade de generalização do modelo logístico

```
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score,
    f1_score, roc_auc_score
)
import pandas as pd

# 1. Definir ponto de corte
threshold = 0.5

# 2. Inicializar lista para armazenar resultados
resultados_estabilidade = []

# 3. Avaliar por amostra
for sample in ['TRAINING', 'VALIDATION', 'TEST']:
    df_amostra = Covid_com_amostra[Covid_com_amostra['SAMPLE'] == sample].copy()
    y_true = df_amostra['HIGH_RISK'].astype(int)
    y_prob = df_amostra['HIGH_RISK_PROB'] # usa o modelo de Regressão Logística
    y_pred = (y_prob >= threshold).astype(int)

    # Calcular métricas
    auc = roc_auc_score(y_true, y_prob)
    acc = accuracy_score(y_true, y_pred)
    prec = precision_score(y_true, y_pred, zero_division=0)
    rec = recall_score(y_true, y_pred)
    f1 = f1_score(y_true, y_pred)
```

Avaliação da Estabilidade do Modelo DE Regressão Logística:

	Amostra	AUC	Accuracy	Precision	Recall	F1-score
0	TRAINING	0.9118	0.8683	0.3725	0.8254	0.5133
1	VALIDATION	0.9121	0.8682	0.3725	0.8272	0.5137
2	TEST	0.9118	0.8682	0.3710	0.8225	0.5113



**Etapa 30: Análise do % de pacientes de alto risco por faixa de score**

```

import pandas as pd
import plotly.express as px

# 1. Garantir que HIGH_RISK seja numérico
SCORE_DIST['HIGH_RISK'] = pd.to_numeric(SCORE_DIST['HIGH_RISK'], errors='coerce')

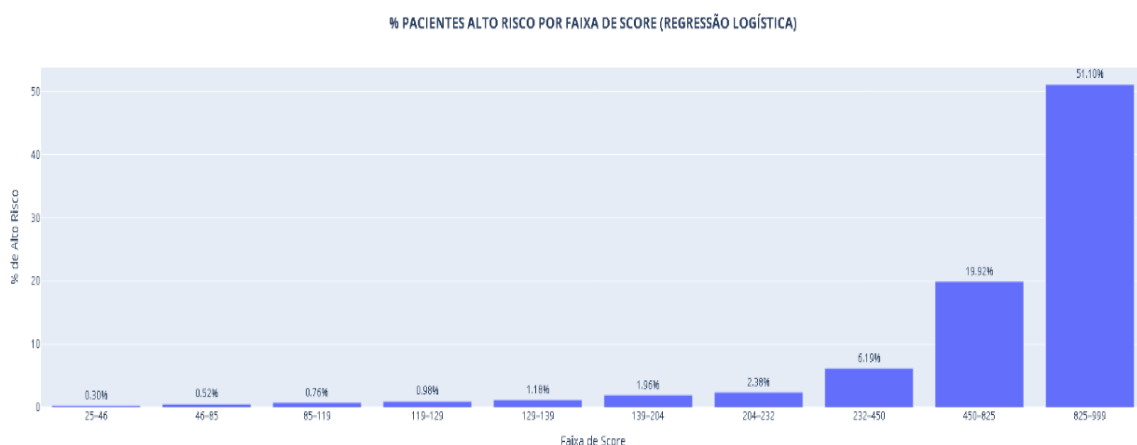
# 2. Criar as faixas reais com qcut (decis)
SCORE_DIST['FAIXA_SCORE'] = pd.qcut(SCORE_DIST['SCORE'], q=10, duplicates='drop')

# 3. Converter os intervalos para string (ex: '132-258')
SCORE_DIST['FAIXA_SCORE_LABEL'] = SCORE_DIST['FAIXA_SCORE'].apply(lambda x: f'{int(x.left)}-{int(x.right)}')

# 4. Calcular % de alto risco por faixa real
tabela_decis = (
    SCORE_DIST.groupby('FAIXA_SCORE_LABEL', observed=False)['HIGH_RISK']
    .mean()
    .reset_index()
    .rename(columns={'HIGH_RISK': 'PERCENTUAL_ALTO_RISCO'})
)
tabela_decis['PERCENTUAL_ALTO_RISCO'] *= 100

# 5. Gráfico com título centralizado na faixa branca
fig = px.bar(
    tabela_decis,
    x='FAIXA_SCORE_LABEL',
    y='PERCENTUAL_ALTO_RISCO',
    text='PERCENTUAL_ALTO_RISCO',

```



Não há qualquer sinal de *overfitting* do modelo. O mesmo apresentou performances bastante similares nas 3 amostras, confirmando sua boa capacidade de generalização.

Um dos sinais claros da robustez de um modelo de score é a coerência da tendência do resultado ao longo das faixas de score. Nesse caso, o % de pacientes com alto risco evolui de maneira progressiva ao longo das faixas, sem nenhuma interrupção da tendência. De maneira mais objetiva, o % de pacientes com alto risco (no geral) é de cerca de 9%, possuindo graus de incidência muito mais elevados (19,92% e 51,10%, respectivamente) nas 2 últimas faixas de score.

#### 4-5 Conclusão do estudo realizado na base tabular

A base de dados tabular analisada, de pacientes tratados no serviço de saúde do México, trouxe informações bastante relevantes sobre os pacientes, ainda que não houvesse um campo de identificação que permitisse cruzar os dados dessa tabela com outras fontes de dados.

Os resultados encontrados a partir do desenvolvimento dos modelos mostrou que existe um padrão de pacientes com maior tendência a evoluírem a um estado mais grave da Covid-19 ou a falecerem.

O modelo pode certamente ser usado como um *instrumento de priorização do atendimento de pacientes, ainda na fase de triagem*, de forma a identificar, de forma precoce, pacientes que possam necessitar de recursos mais especializados como processos de intubação ou unidades de tratamento intensivo (UTI), potencialmente reduzindo o volume de óbitos como consequência da Covid-19.

É fundamental ressaltar que a opinião e decisão final quanto ao tratamento a ser estabelecido, bem como o diagnóstico do paciente, é sempre do profissional de saúde. No entanto, contar com recursos que possam analisar padrões complexos a partir dos dados é fundamental para elevar as taxas de acerto do médico na condução dos procedimentos, assim como agilizar o diagnóstico que, em casos como o da Covid-19, podem significar a diferença entre a vida e a morte de um paciente.

#### **4-6 Limitações encontradas no estudo e oportunidades de melhoria**

A falta de uma variável de identificação do paciente limita a capacidade de considerar dados de distintas fontes adicionais, como dados de exames diagnósticos de imagem e exames de sangue. Todos os dados de cada paciente deveriam estar indexados a um código de identificação, facilitando e acelerando a integração de dados estruturados e não-estruturados.

Para a elevação dos índices de precisão e recall do modelo, é crucial obter, para cada paciente, dados de imagem que reflitam o nível de comprometimento do aparelho pulmonar, assim como indicadores que possam ser extraídos a partir de distintos exames de sangue que possam ser realizados no mesmo paciente, apontando possíveis estados inflamatórios ou possíveis preditores de uma evolução na gravidade da doença.

### **5 – DATASET DE IMAGENS DE PULMÃO**

#### **5-1 Descrição do Dataset de Imagens**

As imagens utilizadas nesta análise foram obtidas a partir do *COVID-19 Radiography Database*, disponível gratuitamente na plataforma Kaggle. Inclui imagens (ex.: radiografias) classificadas em estados como opacidade, pneumonia, COVID-19 ou saudável. Utilizando redes neurais convolucionais (CNNs), buscamos detectar anormalidades pulmonares associadas ao COVID-19, oferecendo suporte visual ao diagnóstico.

O link de acesso ao dataset é: <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database>

Este repositório foi desenvolvido com a colaboração de uma equipe de pesquisadores em colaboração com médicos, criaram um banco de dados de imagens de raios-X de tórax para casos positivos de COVID-19, juntamente com imagens de pneumonia normal e viral. O objetivo do dataset é fornecer uma base confiável para o desenvolvimento de modelos de aprendizado de máquina voltados ao diagnóstico por imagem da COVID-19 e outras condições pulmonares.

## 5-1 Análise Exploratória do Dataset de Imagens de Pulmão

O conjunto de dados é composto por imagens de radiografias torácicas (raio-X de tórax) categorizadas em quatro classes distintas:

- ❖ **COVID:** Imagens de raio-X de tórax de pacientes com diagnóstico confirmado de COVID-19.
- ❖ **Normal:** radiografias de indivíduos saudáveis, sem sinais de anormalidades pulmonares.
- ❖ **Viral Pneumonia:** imagens de pacientes com pneumonia viral não-COVID, utilizadas para diferenciar padrões entre tipos de infecção.
- ❖ **Lung Opacity:** categoria que inclui diversas condições com opacidades pulmonares, como derrames, infecções e outras anormalidades, mas sem confirmação de COVID-19 e nem de pneumonia viral.

Cada imagem é acompanhada de metadados limitados (como nome do arquivo, classe e caminho original da imagem), sendo a principal fonte de informação visual a própria radiografia. Essa estrutura favorece o uso em tarefas de classificação por visão computacional.

Para iniciar a análise, os arquivos de imagem foram carregados diretamente do dataset armazenado no drive, e organizados em subpastas conforme suas respectivas classes: **COVID**, **Normal**, **Viral Pneumonia** e **Lung Opacity**.

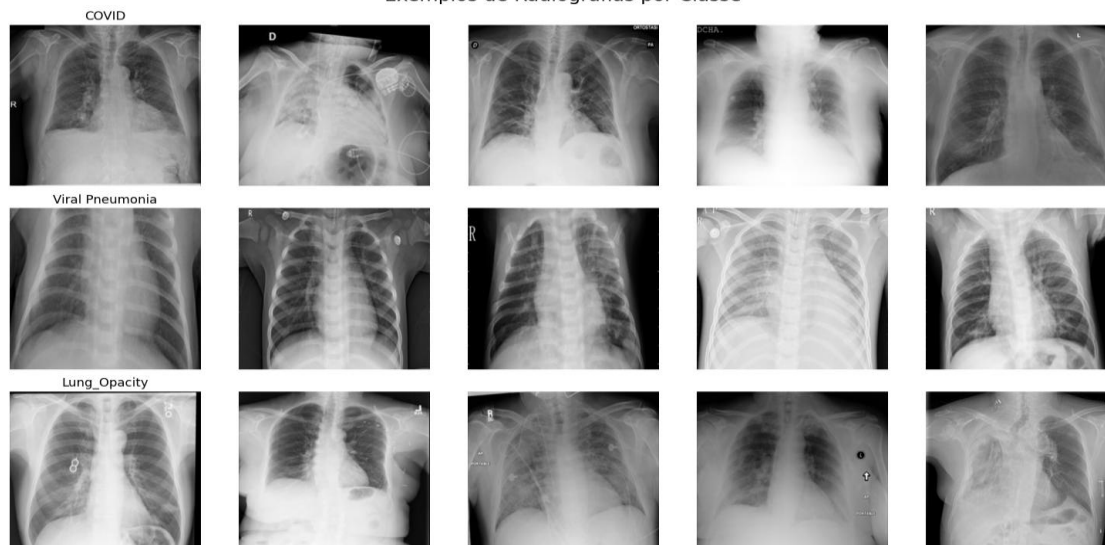
Durante o processo de carregamento, verificamos a integridade dos arquivos, removendo duplicatas e imagens corrompidas, garantindo que apenas dados válidos fossem utilizados na análise posterior.

Utilizamos bibliotecas como os, pandas, e matplotlib para leitura dos arquivos e exploração inicial dos dados. Cada imagem foi associada à sua respectiva classe e armazenada em uma estrutura tabular para facilitar a análise inicial e geração de alguns gráficos.

### Fase I – Visualização de Imagens

Para uma inspeção visual preliminar, selecionamos aleatoriamente algumas amostras de cada classe. A visualização dessas imagens auxilia na compreensão das variações visuais entre os diferentes diagnósticos.

Exemplos de Radiografias por Classe



Diante das imagens, é possível observar que:

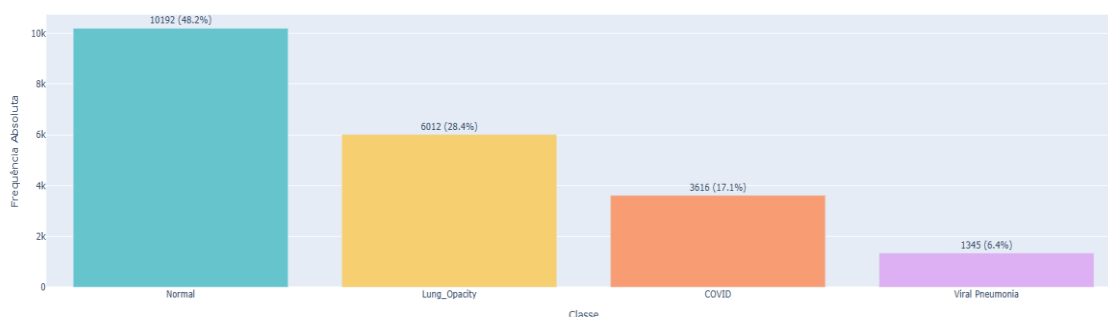
- ❖ As imagens da classe **COVID** frequentemente mostram opacidades bilaterais difusas.
- ❖ As imagens de **Viral Pneumonia** apresentam padrões de infiltração mais localizados.
- ❖ A classe **Lung\_Opacity** possui grande diversidade de padrões radiológicos, o que pode tornar a classificação mais desafiadora.

## Fase II – Visualização das Distribuições de Classes

### Visualização de Distribuições Relevantes

A distribuição das imagens por classe no dataset ocorre da seguinte forma:

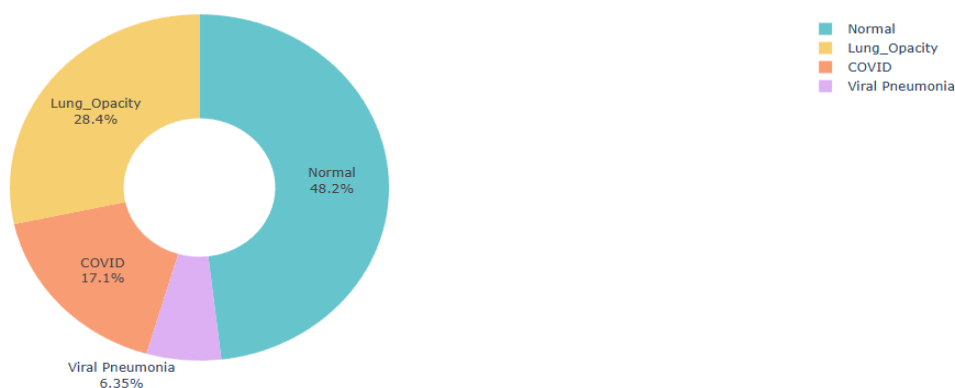
- ❖ **Normal:** X imagens
- ❖ **COVID:** 3616 imagens
- ❖ **Viral Pneumonia:** 1345 imagens
- ❖ **Lung\_Opacity:** 6012 imagens.



Distribuição Percentual por classe doente:

- ❖ **COVID:** 32.95%
- ❖ **Viral Pneumonia:** 12.26%
- ❖ **Lung\_Opacity:** 54.79%

Proporção de imagens por classe



A análise revela um desequilíbrio nas classes, especialmente com a classe **Viral Pneumonia** que é bem menor que as demais. Esse fator pode impactar o desempenho de modelos preditivos, o que nos fez pensar no uso de técnicas de balanceamento como *oversampling*, *undersampling* ou *data augmentation*.

No dataset de imagens não é possível identificar a quem pertence as radiografias. Esse banco de imagens foi criado com a finalidade de treinar modelos dentro dos padrões especificados: "Normal", "Pneumonia Viral" e "COVID". O autor da base no Kaggle, informa que houveram muitas validações e que essa combinação de diferentes idades e condições não seria um problema na identificação e não iria enviesar o modelo. Observando os metadados que contém no dataset, foi identificado que os únicos dados que são armazenados referente às imagens são: FILE NAME | FORMAT | SIZE | URL.

A etapa de exploração inicial nos revelou características importantes:

- ❖ **Desequilíbrio entre as classes:** requer atenção na modelagem para evitar viés de predição.
- ❖ **Qualidade visual das imagens:** a maioria possui boa resolução, mas há variações no contraste e ruído que podem influenciar a performance de algoritmos de visão computacional.
- ❖ **Diversidade dentro das classes:** principalmente na classe **Lung Opacity**, que agrupa diferentes condições, o que pode dificultar a generalização do modelo.

Essas observações guiam a definição das estratégias de pré-processamento, engenharia de atributos e avaliação de desempenho dos modelos que serão utilizados nas etapas seguintes.

## 5-2 Pré-processamento de Imagens

### Fase III – Pré-processamento geral

De forma geral, antes do treino de todos os modelos, foi realizado a remoção de imagens corrompidas, verificação de integridade com PIL (Image.verify), exclusão de duplicatas e as imagens foram utilizadas todas no mesmo formato (PNG), técnicas de aumento de dados de imagens (Augmentation).

Em seguida, para realizar nosso treinamento, optamos por três frentes diferentes, na qual realizamos diferentes tratamentos na base:

- ❖ A primeira é um modelo para triagem facilitada que classifica se o pulmão está saudável ou não (no qual a classe saudável é o pulmão normal, e a classe doente engloba covid, pneumonia e opacidade pulmonar). Nós iremos o denominar de "Modelo Binário". O objetivo deste modelo é agilizar a triagem em uma urgência, e respeitar a premissa de que o diagnóstico final cabe ao médico;
- ❖ A segunda abordagem é diferente da primeira, na qual vamos classificar em "normal, covid ou pneumonia". Considerando o contexto de que a opacidade pulmonar pode indicar uma característica inespecífica e não uma doença específica em si, neste modelo pensamos em desconsiderar esta classe para evitar confusão diagnóstica com outras doenças. O denominaremos de "Modelo Ternário". O seu objetivo é ser mais específico ao classificar as doenças em si, mas ainda desconsiderar esta condição que pode confundir o modelo.
- ❖ A terceira e última abordagem, o qual denominamos de "Modelo Quaternário" tem o objetivo de classificar todas as condições (classes: Normal, Covid, Pneumonia e Lung Opacity), visando englobar a maior especificidade de condições diagnóstica, ainda que opacidade pulmonar (Lung Opacity) seja inespecífico.

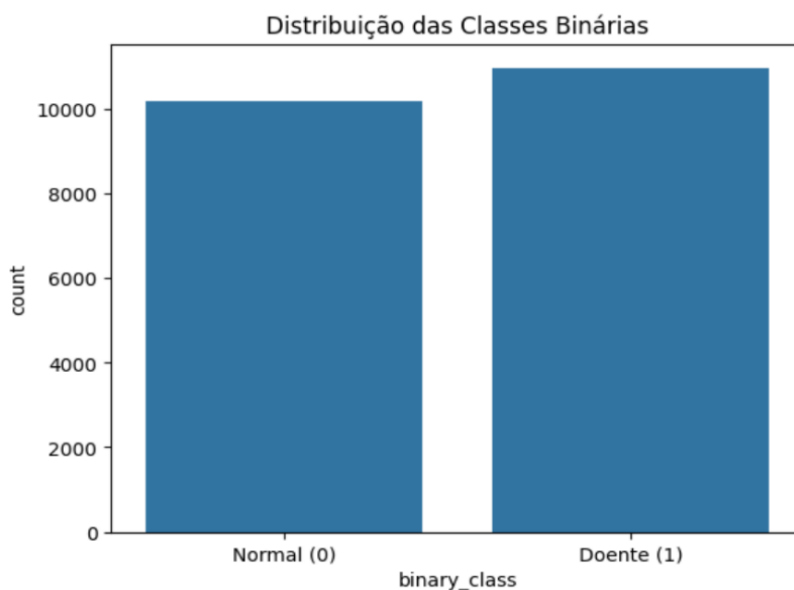
Cada modelo foi treinado para classificar as imagens de raio-x torácico com abordagens diferentes. Outras técnicas, como Undersampling das classes majoritárias para reduzir viés, Oversampling com aumento de dados nas minoritárias (COVID, Pneumonia viral), aumento de dados (data augmentation), class weight (peso das classes) foram utilizadas em alguns modelos, mas não em todos.

Abaixo estão as descrições de cada abordagem utilizada nos modelos e os principais tratamentos de dados aplicados em cada um, antes do treino do modelo:

#### **Fase IV – Pré-Processamento do Modelo Binário**

Este modelo foi pensado para ser usado em uma **triagem simplificada** de emergência, **maximizando o recall de pacientes doentes**. Foi treinado um **Modelo binário**, no qual as imagens foram separadas como 'normal' (saudável) ou 'doente' (a classe doente engloba Covid, Pneumonia e Opacidade).

A separação das classes originais para duas categorias (**normal = 0, doente = 1**), fez com que o “desbalanceamento” anterior (no qual a quantidade de “normais” era muito superior à das doenças específicas) fosse minimizado, visto que a soma das “doenças” equiparou a quantidade de imagens das classes. Na imagem abaixo, é possível verificar a nova distribuição mais equilibrada:



Diante disso, a distribuição dos conjuntos de treino, validação e teste englobou todas as imagens do dataset do seguinte modo:

Tamanho do conjunto de treino: 12699

- ❖ Indica que 12.699 imagens serão usadas para treinar o modelo.

Tamanho do conjunto de validação: 4233

- ❖ 4.233 imagens serão usadas durante o treinamento para validar o desempenho do modelo em dados que ele não viu durante a etapa de aprendizado. Isso ajuda a monitorar o overfitting.

Tamanho do conjunto de teste: 4233

- ❖ 4.233 imagens serão usadas após o treinamento para avaliar o desempenho final do modelo em dados completamente novos e não vistos durante o treino ou validação.

Distribuição nas classes binárias:

- ❖ Esta seção mostra a proporção (percentual) de imagens da classe '1' (Doente) e '0' (Normal) em cada um dos conjuntos.
- ❖ Treino: Aproximadamente 51.85% das imagens de treino são da classe 'Doente' e 48.15% são da classe 'Normal'.
- ❖ Validação: A distribuição é muito similar, com cerca de 51.83% 'Doente' e 48.17% 'Normal'.
- ❖ Teste: Novamente, a distribuição é próxima, com cerca de 51.85% 'Doente' e 48.15% 'Normal'.

### **Fase V – Pré-Processamento do Modelo Ternário**

Este modelo, que optamos por nomeá-lo como “Modelo Ternário”, teve a intenção de ser mais abrangente (em termos de classificação diagnóstica) do que o Binário, no qual foi treinado para classificar imagens em três categorias — 'normal', 'pneumonia viral' e 'covid'. É voltado para uma classificação mais específica das doenças pulmonares mais relevantes, descartando Lung Opacity (Opacidade Pulmonar).

Entre as técnica de tratamento, estão: o mesmo processo de validação das imagens, com exclusão de duplicatas, mas foi realizado mais etapas específicas para este treinamento:

- ❖ A classe 'Lung Opacity' foi propositalmente descartada, por ser uma categoria ampla e pouco específica, pois poderia confundir o modelo, visto que Lung Opacity é uma característica inespecífica que pode indicar desde o uso de cigarro eletrônico ao surgimento de um câncer pulmonar, ou seja, ela é uma condição e não uma doença em específico.
- ❖ Foi realizado um pequeno balanceamento do data frame, limitando o número máximo de imagens de cada classe até 3616, o que diminui a quantidade de imagens da classe “normal”, mas manteve a quantidade de imagens das outras classes, o que ainda fez necessário o cálculo de pesos de cada classe depois.
- ❖ Geradores como augmentation para aumentar os dados das imagens;
- ❖ Os pesos de cada classe foram calculados utilizando o código da próxima imagem anexada:

```

▶ from collections import Counter

# Contagem de categorias
counts = Counter(df['category'])
total = sum(counts.values())

# Mapeamento de classe para nome
label_map = {
    '0': 'Normal',
    '1': 'Viral Pneumonia',
    '2': 'COVID'
}

print("📊 Distribuição das classes:")
for label, count in counts.items():
    percent = (count / total) * 100
    print(f"- {label_map[label]} ({label}): {count} imagens ({percent:.2f}%)")

```

⇒ 📊 Distribuição das classes:

- COVID (2): 3616 imagens (42.16%)
- Viral Pneumonia (1): 1345 imagens (15.68%)
- Normal (0): 3616 imagens (42.16%)

*Figura: Pesos Calculados no modelo ternário*

## **Fase VI – Pré-Processamento do Modelo Quaternário**

Neste modelo, o qual denominamos de “Modelo Quaternário”, nossa intenção foi de abranger todas as 4 classes como uma forma de maior cobertura diagnóstica. O modelo classificou imagens em quatro categorias — 'normal', 'covid', 'pneumonia viral' e 'opacidade pulmonar'.

A inclusão da classe 'Lung Opacity' inclui um grau de complexidade ou ambiguidade no treinamento, pois essa condição pode representar diversas causas. O modelo pode apresentar maior risco de overfitting e desempenho inferior comparado aos demais.

Foram utilizadas técnicas de tratamento semelhante aos demais: limpeza de dados, verificação de arquivos e remoção de duplicatas. Dentre as técnicas adicionais utilizadas especificamente neste notebook estão:

- ❖ Undersampling das classes majoritárias para reduzir viés;
- ❖ Oversampling com aumento de dados nas minoritárias (COVID, Pneumonia viral)
- ❖ Aumento de dados das imagens (data augmentation)
- ❖ Mesmo com o dataset mais balanceado em relação ao modelo anterior, o Class Weight (peso das classes) foi aumentado nas classes de doenças, para diminuir o risco de classificar doentes como saudáveis. Desta vez, ao invés de ser calculado por código, estes valores foram definidos manualmente à dedo respeitando as regras expostas na figura à seguir (*Figura - Definindo pesos manualmente no modelo quaternário*):



```

# Obtém os rótulos do gerador de treino
labels = train_gen.classes

# Calcula os pesos das classes
class_weights = class_weight.compute_class_weight(
    class_weight='balanced',
    classes=np.unique(labels),
    y=labels
)

# Converte para dicionário
class_weights = {
    0: 3.0, # COVID - mais crítico
    1: 1.0, # Lung Opacity - inespecífico, não queremos que o modelo use muito
    2: 1.0, # Normal - comum, mas não precisa reforçar
    3: 2.5 # Viral Pneumonia - importante distinguir de COVID
}

```

Figura - Definindo pesos manualmente no modelo quaternário

No exemplo de imagem à seguir (Figura - Balanceamento das classes por igual no modelo quaternário), é demonstrado o trecho de código que foi utilizado para realizar undersampling de acordo com a quantidade das classes, no qual resultou que todas as classes ficassem com exatamente 1.345 imagens em cada, visando um treinamento mais justo e eficaz em relação ao modelo anterior:

```

# Coletar imagens e descobrir a menor quantidade
all_images = []
min_count = float('inf')

for cls in classes:
    folder = os.path.join(base, cls, 'images')
    images = [os.path.join(folder, f) for f in os.listdir(folder)
               if f.lower().endswith(('.png', '.jpg', '.jpeg'))]
    min_count = min(min_count, len(images))
    all_images.append((cls, images))

print(f"Quantidade mínima para balanceamento: {min_count} imagens por classe")

# Balancear e dividir (undersampling + divisão)
for cls, images in all_images:
    np.random.shuffle(images)
    subset = images[:min_count] # undersampling
    train_imgs, valtest_imgs = train_test_split(subset, test_size=0.3, random_state=42)
    val_imgs, test_imgs = train_test_split(valtest_imgs, test_size=0.5, random_state=42)

    for split, img_list in zip(['train', 'val', 'test'], [train_imgs, val_imgs, test_imgs]):
        for img_path in img_list:
            try:
                img = Image.open(img_path)
                img.verify() # valida imagem
                dest = os.path.join(output_dir, split, cls, os.path.basename(img_path))
                shutil.copy(img_path, dest)
            except:
                print(f"Imagem inválida ignorada: {img_path}")

```

Quantidade mínima para balanceamento: 1345 imagens por classe

Figura - Balanceamento das classes por igual no modelo quaternário

## 5-2 Desenvolvimento dos Modelos

### Fase VII – Escolha das Arquiteturas de cada modelo

#### Descrição resumida da arquitetura de Rede Neural utilizada no processo de treinamento:

- ❖ No **modelo binário**: A CNN foi desenvolvida do zero, priorizando desempenho e velocidade no treinamento do modelo.
- ❖ No **modelo ternário**: Inicialmente tentou-se usar VGG16 e ResNet50, mas foram substituídos por **MobileNetV2** por limitações de infraestrutura.
- ❖ No modelo Quaternário: foi utilizado o **EfficientNetB0** visando comparar os resultados com o modelo anterior.
- ❖ **Justificativa**: Modelos mais leves foram escolhidos em detrimento de pré-treinados mais robustos devido ao tempo de treinamento elevado e limitações computacionais. O modelo CNN simples foi ideal para situações onde a agilidade de resposta é mais importante do que a acurácia máxima

#### Justificativa da Escolha da Arquitetura CNN de cada modelo:

Inicialmente, ao treinar os modelos, houve a tentativa de utilizar os modelos **ResNet50** e **VGG16** (visão computacional avançada), que são recomendados para dataset médicos, porém, **estes modelos foram descartados** devido às condições de processamento (infraestrutura) atuais, o tempo de execução do treinamento do modelo estava extremamente longo por serem modelos mais pesados e robustos, o que influenciou na troca para um modelo mais rápido e leve. Dentre as substituições, no treinamento do “modelo ternário” foi utilizado o **MobileNetV2** e no treinamento do “modelo quaternário” foi utilizado o **EfficientNetB0** como forma de abranger dois modelos pré-treinados diferentes.

No “**Modelo Binário**” optamos por uma abordagem diferente, no qual construímos uma Convolutional Neural Network (CNN) “**do zero**” com o objetivo de priorizar a velocidade de execução do modelo, em contraste com a utilização de modelos pré-treinados como EfficientNetB0 ou MobileNetV2 explorados em outros experimentos anteriores.

A principal motivação para esta abordagem foi novamente a busca pela otimização do tempo de execução e treinamento do modelo, ressaltando o fato de que, ao contrário dos outros modelos, **no modelo binário, todas as imagens foram utilizadas** no treinamento (como foi visto anteriormente na etapa anterior que a divisão em apenas duas classes, doente ou normal, já equilibrava a quantidade de imagens por classe neste notebook em si), **sem diminuição na quantidade de imagens por classe**: o que tornaria o treinamento ainda mais longo ao usar uma das arquiteturas anteriores.

Arquiteturas pré-treinadas, embora poderosas e capazes de transferir conhecimento de grandes datasets (como ImageNet), frequentemente possuem um número muito maior de parâmetros, o que pode resultar em:

- ❖ **Tempo de Treinamento Mais Longo**: Modelos maiores exigem mais recursos computacionais e tempo para convergir durante o treinamento.
- ❖ **Tempo de Inferência Mais Lento**: A execução do modelo em novas imagens (inferência) pode ser mais demorada devido à complexidade e ao número de operações.

Ao construir uma CNN personalizada, podemos projetar uma arquitetura mais leve e específica para a tarefa de classificação binária de radiografias torácicas. Isso permitiu um ciclo de experimentação mais rápido e um modelo potencialmente mais eficiente em termos de recursos para uma aplicação de triagem em cenários de emergência, onde a velocidade de processamento é crucial.

Embora modelos pré-treinados possam atingir acurácias ligeiramente superiores (o que se faz necessário em casos de precisão e especificidade diagnóstica como nos outros modelos mais abrangentes), a escolha de uma CNN do zero neste contexto priorizou a eficiência computacional e o tempo de resposta, aspectos fundamentais para a viabilidade prática da solução proposta no modelo binário que tem a premissa de auxiliar em uma triagem rápida e eficiente.

### 5-3 Avaliação Geral dos 3 modelos de imagem

Esta análise tem como objetivo comparar três abordagens de modelos de classificação para detecção de condições médicas em radiografias torácicas: binária, ternária e quaternária. **O foco principal é a triagem de emergência, onde o erro mais grave seria classificar um paciente doente como saudável.**

**Métricas Gerais dos 3 modelos:**

Modelo	Acurácia	F1-Score Médio	Doença com menor Recall	Risco de Falso Negativo
Binário	91%	0.91	Doente (0.90)	Baixo
Ternário	87%	0.88	Covid (0.80)	Médio
Quaternário	85%	0.85	Normal (0.73)	Médio-Baixo

#### **Modelo Binário (Normal x Doente):**

- ❖ Alta acurácia geral (91%) e excelente recall da classe doente (0.90).
- ❖ Melhor escolha para triagem rápida, pois minimiza o risco de enviar um paciente doente para casa.
- ❖ Classificação simplificada ajuda na tomada de decisão imediata.
- ❖ Diagnóstico final da condição (Covid, Pneumonia etc.) pode ser delegado ao médico.

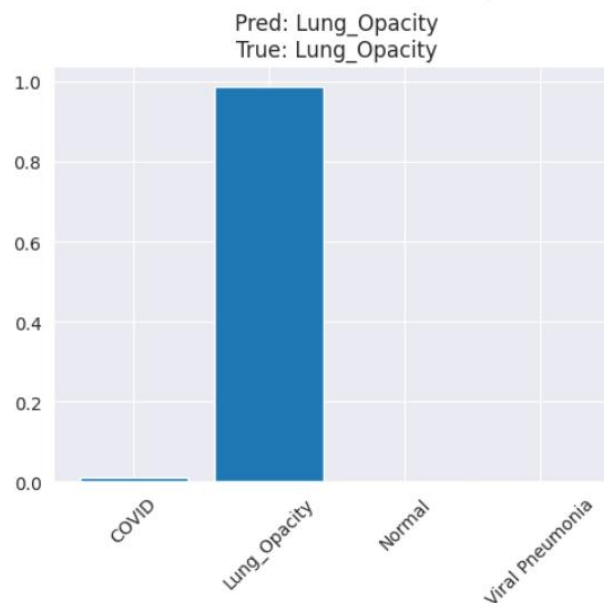
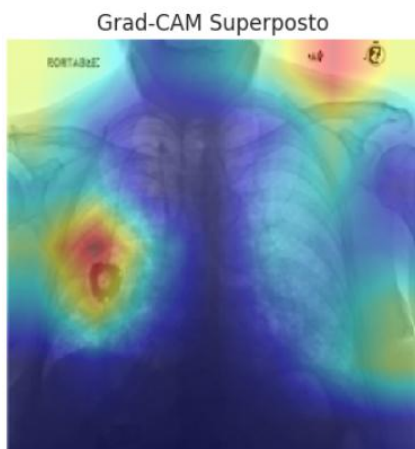
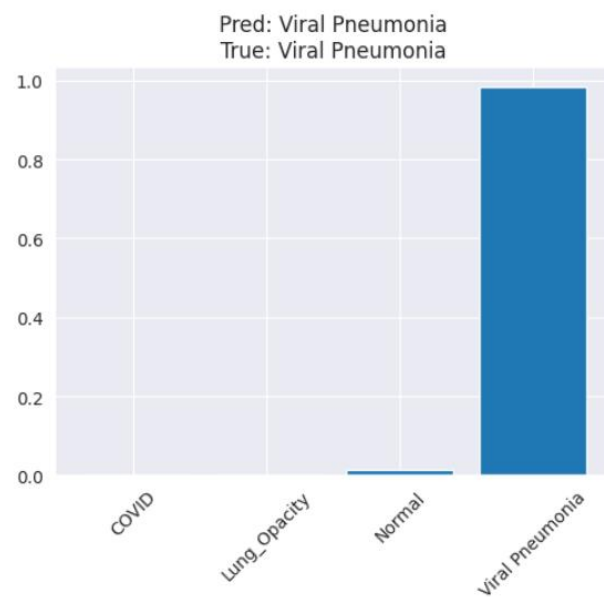
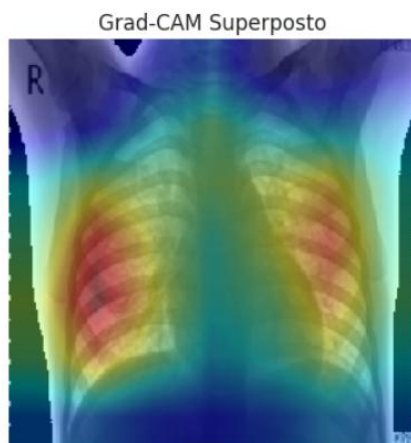
#### **Modelo Ternário (Normal x Pneumonia x Covid):**

- ❖ Boa performance geral (acurácia de 87%, F1-score de 0.88).
- ❖ Recall da classe Covid (0.80) é preocupante para triagem, pois pode deixar passar casos graves.
- ❖ Boa detecção de pneumonia, mas menos seguro em contextos onde o erro deve ser minimizado.

### Modelo Quaternário (Normal x Pneumonia x Covid x Opacidade Pulmonar):

- ❖ Maior complexidade de classificação e acurácia mais baixa (86%).
- ❖ Classe 'Normal' tem baixo recall (0.73), pode levar a sobrecarga no atendimento por falsos positivos.
- ❖ A classe 'Opacidade' pode introduzir ambiguidade clínica, pois representa múltiplas condições possíveis.

### Previsões Aleatórias do Modelo Quaternário:



## 5-4 Conclusão dos Modelos de Imagens

Em relação ao dataset de imagens, podemos fazer a seguinte comparação de resultados dos 3 modelos treinados:

Critério	Binário	Ternário	Quaternário
Acurácia geral	91%	88%	86%
Especificidade clínica	Baixa	Alta	Média
Complexidade / Overfitting	Menor risco	Balanceado	Maior risco
Interpretação dos resultados	Fácil	Média	Difícil
Utilidade prática (triagem)	Alta	Alta	Média

Conclui-se que, para fins de triagem médica em contexto de emergência, onde o foco é identificar rapidamente pacientes que precisam de atenção médica, **o modelo binário (normal ou doente) se mostra o mais adequado para uma triagem inicial**. Ele oferece alta sensibilidade, reduzindo o risco de falsos negativos, e apresenta uma estrutura de decisão simples, evita liberar doentes e facilita sua adoção em ambientes hospitalares. Já o modelo “ternário” ou “quaternário” seriam melhores em termos de especificidade clínica ou diagnóstica. No sentido diagnóstico, a opinião do médico prevalece sempre, como por exemplo: o modelo confirma a opinião do médico, mas caso contrário, o médico poderá pedir exames adicionais ou consultar outro médico para uma opinião adicional.

É importante ressaltar que o objetivo do nosso trabalho não é diagnosticar, mas sim criar ferramentas para facilitar e agilizar a triagem, e também respeitar o fato de que o diagnóstico final cabe somente ao médico. Por esta razão, o modelo binário também é mais adequado à este propósito.

## 6 – APLICABILIDADE PRÁTICA GERAL DOS ESTUDOS

Ambos os projetos desenvolvidos neste Tech Challenge possuem aplicação direta em ambientes hospitalares e sistemas de saúde.

- ❖ O **modelo baseado em dados tabulares** pode ser incorporado a sistemas de prontuário eletrônico, oferecendo alertas automáticos sobre o risco de agravamento ou óbito de pacientes internados com sintomas respiratórios. Isso permite aos gestores antecipar recursos e tomar decisões mais assertivas sobre internação, encaminhamento e suporte intensivo.
- ❖ Já o **modelo baseado em imagens** pode ser utilizado em conjunto com aparelhos de raio-x digital em unidades de pronto atendimento. Ao processar a imagem em tempo real, o modelo indicaria a presença de padrões compatíveis com Covid-19 ou outras condições pulmonares, auxiliando equipes médicas em contextos com escassez de especialistas.

Além disso, ambos os modelos podem ser integrados a **painéis de triagem baseados em IA**, priorizando pacientes com base em múltiplos fatores clínicos e visuais. Essa abordagem multissensorial pode ser decisiva em pandemias, campanhas de vacinação ou outras situações de crise sanitária.

**Recomenda-se**, portanto, o avanço em soluções interoperáveis e modelos auditáveis, garantindo que a inteligência artificial atue como aliada do profissional de saúde — apoiando decisões clínicas, salvando vidas e otimizando recursos.

### **Viabilidade de Implantação em Ambiente Clínico**

A adoção dos modelos desenvolvidos é tecnicamente viável, especialmente em instituições que já utilizam prontuários eletrônicos e possuem infraestrutura básica de TI.

- ❖ O modelo de dados tabulares pode ser integrado a sistemas hospitalares para emitir alertas de risco com base em registros clínicos.
- ❖ O modelo de imagens, por sua vez, pode ser incorporado a fluxos de radiologia digital, desde que exista conectividade mínima e equipamento compatível.
- ❖ Ambos podem ser executados em ambientes on-premise (hospitalar) ou em nuvem, com ajustes de segurança e anonimização dos dados, respeitando normas éticas e legais como a LGPD.

### **Papel dos Profissionais Médicos no Diagnóstico Final**

Em nenhuma hipótese os modelos propostos substituem o julgamento clínico. Eles são **ferramentas de apoio à decisão**, cujo objetivo é aumentar a agilidade na triagem, sugerir padrões de risco ou anormalidade e priorizar pacientes em contextos de alta demanda.

A responsabilidade final pelo diagnóstico e conduta permanece com os médicos, que podem usar os modelos como **segunda opinião automatizada** ou como filtro inicial em sistemas de triagem.

## **7 – LIÇÕES APRENDIDAS E POSSÍVEIS MELHORIAS**

### **Limitações Identificadas:**

- ❖ Os modelos foram treinados com dados de contextos específicos (México e bancos internacionais), o que pode reduzir a generalização para outras populações.
- ❖ O modelo de imagem depende da qualidade e padronização das radiografias.
- ❖ A base tabular apresenta viés de preenchimento, com campos faltantes ou inconsistentes.

### **Possíveis Melhorias:**

- ❖ Integração dos modelos com bases clínicas reais e contínuas (dados em tempo real)

- ❖ Aplicação de técnicas de explainability (ex: SHAP, Grad-CAM) para maior transparência nas decisões
- ❖ Treinamento de modelos com dados nacionais para aumentar precisão local
- ❖ Testes de usabilidade com profissionais da saúde para ajuste da interface e fluxos de uso

## 8 – CONCLUSÃO GERAL DO TRABALHO

Este trabalho integrou dois eixos analíticos distintos com foco no enfrentamento da Covid-19: um baseado em **dados tabulares** de registros clínicos de pacientes e outro com foco em **imagens radiográficas** de tórax. A combinação desses estudos evidencia o potencial da Inteligência Artificial para ampliar a capacidade de triagem médica, seja pela predição de agravamento com base em histórico clínico ou pela leitura automatizada de exames por imagem.

O estudo com dados tabulares permitiu construir modelos preditivos capazes de indicar pacientes com maior risco de óbito, contribuindo para a gestão de recursos e alocação de leitos em hospitais. Já o estudo com imagens trouxe contribuições significativas para triagem imediata, mostrando que modelos leves como CNNs específicas, MobileNetV2 ou EfficientNetB0 podem identificar rapidamente alterações pulmonares associadas à Covid-19.

Apesar de não serem integradas, essas fontes se complementam. Um futuro sistema que una dados clínicos estruturados e imagens pode representar um avanço expressivo na capacidade de resposta do sistema de saúde em emergências.

Desta forma, o estudo reforça que integrar tecnologias avançadas, como a Inteligência Artificial, à expertise e ao conhecimento clínico de um(a) profissional médico/a representa uma excelente ferramenta na área da saúde. Embora a tecnologia ofereça suporte valioso na análise de dados e na tomada de decisões, ela não substitui o olhar humano, ético de um médico. Quando utilizada como aliada, torna-se uma ferramenta excepcional para aprimorar diagnósticos, tratamentos e o cuidado com o paciente.

## 9 – ENTREGÁVEIS

### 9-1 Repositório no GitHub

Um repositório público foi criado no perfil da organização disponível em <https://github.com/AI-Business-Insights/tech-challenge>

### 9-2 Links dos Datasets

Os datasets estão disponíveis para download na plataforma Kaggle:

- [COVID-19 Radiography Database](#)
- [COVID-19 Dataset](#)

### 9-3 Notebooks ipynb

Os arquivos .ipynb do Notebook no Google Colab estão disponíveis no repositório do GitHub em: <https://github.com/AI-Business-Insights/tech-challenge> e na pasta do Google Drive.

- **Base de Dados I - Dataset de Registros de Pacientes (dados tabulares):**

Google Drive:

[https://drive.google.com/file/d/1Hyv1IJPU0ucWhv3VZcKa500RzfaoICj/view?usp=drive\\_link](https://drive.google.com/file/d/1Hyv1IJPU0ucWhv3VZcKa500RzfaoICj/view?usp=drive_link)

GitHub:

[https://github.com/AI-Business-Insights/tech-challenge/blob/5c3f6a3bed09d5c30e181cb124775c31389a7888/notebooks/Covid\\_19\\_Dataset\\_Analise\\_completa\\_dos\\_dados\\_tabulares.ipynb](https://github.com/AI-Business-Insights/tech-challenge/blob/5c3f6a3bed09d5c30e181cb124775c31389a7888/notebooks/Covid_19_Dataset_Analise_completa_dos_dados_tabulares.ipynb)

- **Base de dados II - Dataset de Radiografias de Pulmão (imagens):**

Google Drive:

[https://drive.google.com/file/d/1szy8fW1bGh8UMia0uddRBXVn4ALwH0aL/view?usp=drive\\_link](https://drive.google.com/file/d/1szy8fW1bGh8UMia0uddRBXVn4ALwH0aL/view?usp=drive_link)

GitHub:

[https://github.com/AI-Business-Insights/tech-challenge/blob/5c3f6a3bed09d5c30e181cb124775c31389a7888/notebooks/Covid\\_19\\_Imagens\\_dataset.ipynb](https://github.com/AI-Business-Insights/tech-challenge/blob/5c3f6a3bed09d5c30e181cb124775c31389a7888/notebooks/Covid_19_Imagens_dataset.ipynb)

### 9-4 Vídeo de Demonstração

Demonstração do sistema em execução com breve explicação do fluxo em um vídeo disponível no Youtube. Link:

<https://www.youtube.com/playlist?list=PL1zapSlcAQTfqfXpcY-upK14391b0Z5Nw>





### 9-5 Pasta do Google Drive

Pasta com arquivos como Relatório e notebook .ipynb

[https://drive.google.com/drive/folders/1pEZHwCrVuLVRahdMEr89pPuFEcwg8mQI?usp=drive\\_link](https://drive.google.com/drive/folders/1pEZHwCrVuLVRahdMEr89pPuFEcwg8mQI?usp=drive_link)

## 10 – REFERÊNCIAS

FIAP. **Material da Fase 1 - Welcome to IA para Devs.** Disponível em: <https://postech.fiap.com.br/plataforma>. Acesso em: 05 jun. 2025.

KAGGLE. **COVID-19 Dataset:** COVID-19 patient's symptoms, status, and medical history. Disponível em: <https://www.kaggle.com/datasets/meirnazri/covid19-dataset>. Acesso em: 06 jun. 2025.

KAGGLE. **COVID-19 Radiography Database:** covid-19 chest x-ray images and lung masks database. COVID-19 Chest X-ray images and Lung masks Database. Disponível em: <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database>. Acesso em: 06 jun. 2025.

KALINOWSKI, Marcos. Análise Exploratória e Visualização de Dados. In: KALINOWSKI, Marcos; ESCOVADO, Tatiana; VILLAMIZAR, Hugo; LOPES, Hélio. **Engenharia de Software para Ciência de Dados/: um guia de boas práticas com ênfase na construção de sistemas de Machine Learning em Python**. São Paulo: Casa do Código, 2023. p. 195-240.

LIN, Alec Zhixiao. **Variable Reduction in SAS by Using Weight of Evidence and Information Value**. SAS Global Forum 2013 - Paper 095-213. 2013.

PORTAL, Coronavírus Brasil. Ministério da Saúde. Disponível em: [Coronavírus Brasil](#) Acesso em: em 12 de jul. 2025.