



UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”

Instituto de Ciência e Tecnologia de Sorocaba

MATEUS DA COSTA SOUZA

Integração de *Robotic Process Automation (RPA)* e *Business Intelligence (BI)* para Análise Dinâmica de Exames de Bioressonância

Sorocaba

2025

Mateus da Costa Souza

Integração de *Robotic Process Automation* (RPA) e *Business Intelligence* (BI)
para Análise Dinâmica de Exames de Bioressonância

Projeto Final de Curso apresentado à
Universidade Estadual Paulista
(UNESP), Instituto de Ciência e
Tecnologia, Sorocaba, como parte
dos requisitos para obtenção do grau
de Bacharel em Engenharia de
Controle e Automação.

Orientador: Prof. Dr. Márcio
Alexandre Marques

Coorientadora: Prof^a. Ma. Patricia
Moriguchi

Sorocaba

2025

S729i

Souza, Mateus da Costa

Integração de Robotic Process Automation (RPA) e Business Intelligence (BI) para análise dinâmica de exames de bioressonância / Mateus da Costa Souza. -- Sorocaba, 2025

93 p. : il., tabs., fotos

Trabalho de conclusão de curso (Bacharelado - Engenharia de Controle e Automação) - Universidade Estadual Paulista (UNESP), Instituto de Ciência e Tecnologia, Sorocaba

Orientador: Marcio Alexandre Marques

Coorientadora: Patricia Moriguchi

1. Automação. 2. Business Intelligence. 3. Python (Linguagem de programação de computador). I. Título.

MATEUS DA COSTA SOUZA

**INTEGRAÇÃO DE ROBOTIC PROCESS AUTOMATION (RPA) E BUSINESS
INTELLIGENCE (BI) PARA ANÁLISE DINÂMICA DE EXAMES DE
BIORESSONÂNCIA**

Trabalho de Conclusão de Curso apresentado ao Instituto de Ciência e Tecnologia de Sorocaba, Universidade Estadual Paulista (UNESP), como parte dos requisitos para obtenção do grau de Bacharel(a) em Engenharia de Controle e Automação.

Data da defesa: 20/08/2025

BANCA EXAMINADORA:

Prof. Dr. MARCIO ALEXANDRE MARQUES
Orientador/UNESP – Instituto de Ciência e Tecnologia – Campus de Sorocaba

Prof. Dr. EVERSON MARTINS
UNESP – Instituto de Ciência e Tecnologia – Campus de Sorocaba

Prof. Dr. IVANDO SEVERINO DINIZ
UNESP – Instituto de Ciência e Tecnologia – Campus de Sorocaba

Agosto de 2025

*Este trabalho é dedicado à minha família, aos meus amigos e a todos os
membros e futuros membros da minha amada república.*

AGRADECIMENTOS

Inicialmente eu gostaria de agradecer aos meus pais, que me sempre foram um farol de bons exemplos para mim e sempre se esforçaram para proporcionar as melhores condições dentro do que era possível. Seu amor incondicional, seus sacrifícios e seus bons exemplos formam as bases de que eu sou hoje e sua presença e seu apoio constante me fez acreditar que consigo concluir todos meus objetivos.

Agradeço ao meu irmão, meu melhor amigo e mentor, que sempre me ajudou em todas as dificuldades da minha vida, que sempre me inspirou a ser minha melhor versão, que me conhece como nenhuma outra pessoa e que eu confio independente de qualquer situação.

Aos meus irmãos da república Hentrometeu e meus companheiros da Turma XVI, minha eterna gratidão por serem pessoas incríveis. Suas parcerias, experiências, conselhos e apoio me auxiliaram ao longo da minha jornada acadêmica, me ensinando coisas novas, comemorando conquistas e auxiliando nos desafios.

Aos meus orientadores, Prof. Dr. Márcio Alexandre Marques e Prof^a. Me. Patricia Moriguchi e meus professores da UNESP Sorocaba, meu eterno agradecimento por me orientarem, auxiliarem e provocarem durante a minha jornada acadêmica, sem suas presenças eu não conseguiria cumprir com meus objetivos e me tornar uma parte ativa da sociedade.

Este trabalho não teria sido realizado sem a presença, influência e apoio de todos citados anteriormente. Cada uma destas pessoas é responsável por eu ser quem eu sou hoje e por me ajudarem a chegar até aqui.

Muito obrigado!

*“A verdadeira generosidade para com o
futuro consiste em dar tudo ao
presente”*

(Albert Camus, “O Homem Revoltado”)

RESUMO

Este trabalho apresenta o desenvolvimento de uma plataforma automatizada para visualização e análise de exames de biorressonância magnética, integrando técnicas de *Robotic Process Automation* (RPA), linguagem de programação Python e software de *Business Intelligence* (Power BI). O objetivo principal foi facilitar a interpretação dos dados extraídos dos exames, originalmente gerados em arquivos PDF com estrutura pouco acessível, por meio da criação de um sistema automatizado de coleta, tratamento e visualização das informações. A solução proposta contempla a automatização da extração dos dados, seu armazenamento em nuvem (Google Drive) e posterior transformação em gráficos e *dashboards* interativos. O sistema foi testado em exames realizados em um conjunto de dados coletados de funcionários da UNIP/Sorocaba. A análise dos dados permitiu identificar padrões relevantes de saúde coletiva, com destaque para anomalias em categorias como obesidade, lipídios no sangue e função gastrointestinal. O uso combinado das tecnologias mencionadas demonstrou que a ferramenta implementada é eficaz na redução do tempo de análise dos exames gerados, no aumento da precisão e interpretação dos resultados coletivos, bem como no suporte à proposição de melhores condições de rotina de trabalho para o público analisado. A plataforma também apresenta potencial para expansões futuras, como a aplicação de inteligência artificial e *machine learning* em análises preditivas.

Palavras-chave: Biorressonância magnética, Power BI, Python, RPA, Business Intelligence (BI), Visualização de Dados.

ABSTRACT

This work presents the development of an automated platform for the visualization and analysis of magnetic bioresonance test results, integrating Robotic Process Automation (RPA) techniques, the Python programming language, and the Business Intelligence software, Power BI. The primary objective was to facilitate the interpretation of data extracted from exams originally generated in PDF files with limited accessibility, by creating an automated system for data collection, processing, and visualization. The proposed solution includes the automation of data extraction, its storage in the cloud (Google Drive), and its subsequent transformation into interactive charts and dashboards. The system was applied to tests conducted on various employees of UNIP/Sorocaba. Data analysis enabled the identification of relevant public health patterns, particularly highlighting anomalies in categories such as obesity, blood lipids, and gastrointestinal function. The combined use of the aforementioned technologies proved effective in reducing the time required for test analysis, increasing the accuracy of interpreting collective results, and supporting the justification for improved routine conditions for UNIP's staff. The project also opens possibilities for future expansion, including the application of artificial intelligence and machine learning for predictive analysis.

Keywords: Magnetic Bioresonance, Power BI, Python, RPA, Business Intelligence, Data Visualization.

LISTA DE ILUSTRAÇÕES

Figura 1 – Pirâmide DIKW e suas explicações.....	17
Figura 2 – Ferramentas que podem ser utilizadas em conjunto com RPA via Python.....	25
Figura 3 – Modelo de comunicação padrão de banco de dados com aplicações.....	27
Figura 4 – Analisador de bioressonância magnética realizando um exame.....	31
Figura 5 – Fluxograma das etapas de execução do projeto.....	34
Figura 6 – Fluxograma das etapas de desenvolvimento do RPA.....	35
Figura 7 – Página inicial da página do Google Cloud.....	36
Figura 8 – Primeiros passos para criação de projeto no Google Cloud.....	37
Figura 9 – Criação de uma API dentro de um projeto do Google Cloud.....	38
Figura 10 – Demonstração da seleção do campo de ativação de APIs.....	39
Figura 11 – Seleção da API relacionada ao funcionamento do Google Drive...	39
Figura 12 – Demonstração do campo de ativação da API do Google Drive.....	40
Figura 13 – Criação das credenciais para utilização da API.....	40
Figura 14 – Seleção dos tipos de credenciais para a API do Google Drive.....	41
Figura 15 – Cadastro dos dados para a verificação via OAuth.....	42
Figura 16 – Seleção de todos os serviços que serão utilizados na aplicação...	43
Figura 17 – Criação do ID para o cliente OAuth.....	43
Figura 18 – Tela de download das credenciais de conexão da API do Google Drive com o programa.....	44
Figura 19 – Terminal de seleção do comando “auto-py-to-exe”.....	46
Figura 20 – API de conexão do Power BI com Google Planilhas.....	48
Figura 21 – Transformação de dados no Power Query.....	49

Figura 22 – Janela de criação de rotina de atualização.....	50
Figura 23 – Continuação da criação da rotina de atualização.....	50
Figura 24 – Apresentação dos quadros de informações dos exames de bioressonância	52
Figura 25 – Painel com dados dos Exames da categoria da Obesidade.....	57
Figura 26 – Painel com dados do Exame “Coeficiente de conteúdo anormal de triglicerídeos”	58
Figura 27 – Painel com dados do Exame “Coeficiente de metabolismo de lipídeos”	59
Figura 28 – Painel com dados dos Exames da categoria de Ácidos graxos.....	60
Figura 29 – Painel com dados dos Exames da categoria de Função Gastrointestinal.....	61
Figura 30 – Painel com dados do Exame “Coeficiente das funções de absorção gástricas”	61
Figura 31 – Painel com dados dos Exames da categoria de Lipídios no sangue.....	62
Figura 32 – Painel com dados do Exame “Gordura neutra”	63
Figura 33 – Painel com dados do Exame “Triglicerídeos”	63
Figura 34 – Painel com dados do Exame “Viscosidade do Sangue”	64
Figura 35 – Painel com dados dos Exames da categoria da Grande função do intestino.....	65
Figura 36 – Painel com dados dos Exames da categoria da Função Renal.....	66
Figura 37 – Painel com dados dos Exames da categoria da metais pesados..	67
Figura 38 – Painel com dados dos Exame “Chumbo”	67
Figura 39 – Painel com dados dos Exame “Mercúrio”	68
Figura 40 – Painel com dados dos Exames da categoria da toxina humana....	69

LISTA DE QUADROS

Quadro 1 – Apresentação da categoria dos exames de bioressonância.....	55
--	----

LISTA DE TABELAS

Tabela 1 – Apresentação dos filtros e quadros de informações dos exames de bioressonância.....	53
--	----

LISTA DE ABREVIATURAS E SIGLAS

DIKW – *Data, Information, Knowledge and Wisdom*

BI – *Business Intelligence*

RPA – *Robotic Process Automation*

P&D – *Pesquisa e Desenvolvimento*

IA – *Inteligencia Artificial*

IDE – *Integrated Development Enviroment*

GUI – *Graphic user interface*

API – *Application Programming Interface*

HTTP – *Hypertext Transfer Protocol*

UNATI – *Universade Aberta à Terceira Idade*

SUMÁRIO

1. INTRODUÇÃO.....	16
1.1. Contexto e desafios.....	16
1.2. Objetivo.....	17
1.3. Estrutura do trabalho.....	18
2. REVISÃO BIBLIOGRÁFICA E ASPÉCTOS TEÓRICOS.....	19
2.1. Linguagem de programação Python.....	19
2.1.1. <i>PyCharm</i>	20
2.1.2. Bibliotecas utilizadas.....	20
2.2. <i>Power Query</i>	24
2.3. <i>Robotic Process Automation</i>	24
2.4. Google Drive.....	26
2.5. <i>Business Intelligence (BI)</i>	28
2.5.1. Ferramentas de business intelligence.....	29
2.5.1.1. <i>Power BI</i>	29
3. MATERIAIS E MÉTODOS.....	31
3.1. Materiais.....	31
3.1.1. Analisador de bioressonancia magnética.....	31
3.1.2. Computador.....	32
3.2. Métodos.....	32
3.2.1. Introdução.....	33
3.2.2. Desenvolvimento da solução para coleta de dados.....	34
3.2.2.1. <i>Elaboração da conexão entre o RPA e o Google Drive</i>	36
3.2.2.2. <i>Desenvolvimento do código para conexão entre RPA e Google Drive</i>	44
3.2.2.3. <i>Elaboração das funções responsáveis por coletar e concatenar os dados dos exames</i>	45
3.2.2.4. <i>Arquivo executável</i>	46
3.2.3. Relatório no Microsoft <i>Power BI</i>	46
4. RESULTADOS E DISCUSSÕES.....	51
4.1. Desenvolvimento do painel de exibição.....	51
4.1.1. Desenvolvimento dos filtros para o <i>Power BI</i>	53
4.1.2. Elaboração dos quadros de dados.....	55

4.2. Compartilhamento da solução RPA e publicação do relatório desenvolvido no <i>Power BI</i>	56
4.3. Análise dos dados coletados.....	57
5. CONCLUSÃO.....	70
5.1. Trabalhos futuros.....	71
REFERÊNCIAS	72
APÊNDICE A – DOWNLOAD e UPLOAD.....	78
APÊNDICE B – FUNCIONAMENTO – ESQUELETO DO PROGRAMA.....	81
APÊNDICE C – FUNÇÕES DO PROGRAMA.....	82
APÊNDICE D – EXAMES.....	85

1. INTRODUÇÃO

1.1. Contexto e Desafios

Profissões e proficiências relacionadas a saúde são tão antigas quanto a própria humanidade. Ao longo da história, o modo como lidamos com diagnósticos e exames médicos evoluiu significativamente, refletindo avanços não apenas científicos, mas também sociais e tecnológicos. Nesse contexto, a forma como interpretamos os dados oriundos da prática médica se torna crucial para que o tempo de enfermidade do paciente seja o mais curto possível (Martins, 2014).

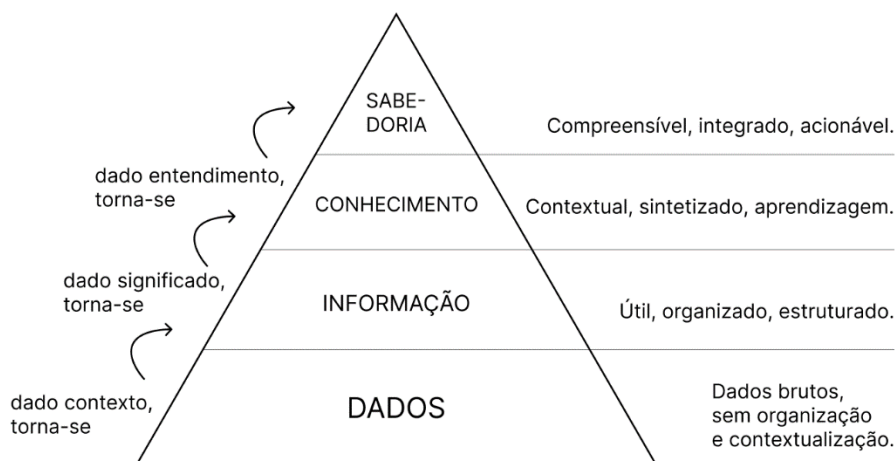
De acordo com Horita (2023), a evolução dos exames para diagnósticos realizados por esses profissionais demonstra, principalmente, o avanço das ciências médicas e as transformações sociais e tecnológicas que moldaram a compreensão do ser humano sobre saúde, longevidade e qualidade de vida.

Contudo, mais importante ainda que os resultados dos exames é a interpretação desta informação num âmbito mais amplo. Um exame analisado do ponto de vista de um indivíduo, terá como resultado apenas uma avaliação clínica e um plano de ação individual. Entretanto, quando se expande essa análise no âmbito de todo um grupo ou sociedade, é possível analisar comportamentos conjuntos e traçar possíveis hipóteses que, inicialmente, eram improváveis de serem analisadas. Philippa Fogarty (2019) cita que estamos vivendo um momento em que a análise de grupo está possuindo cada vez mais valor e, devido a isso, já existe a possibilidade de realizar um seguro saúde exponencialmente mais caro para doenças específicas, caso a seguradora tenha acesso aos dados de exames genéticos que demonstrem um aumento de pontuais enfermidades.

O exemplo de Philippa é justamente a aplicação do que é conhecido como construto de hierarquia do conhecimento (DIKW). De acordo com Silva (2017), este modelo descreve, como a informação é organizada e processada para criar conhecimento e, eventualmente, sabedoria. DIKW é um acrônimo para *Data*, *Information*, *Knowledge* e *Wisdom* que em português é traduzido para: dados, informação, conhecimento e sabedoria. Como mostrado na Figura 1, pode-se

organizar essa metodologia em forma de pirâmide, onde cada um dos seus níveis se constrói sobre o anterior, sendo este um processo contínuo.

Figura 1 - Pirâmide DIKW e suas explicações.



Fonte: Miranda (2021).

Nesse sentido, o uso deste modelo (DIKW) para gerar informações e conhecimentos através da análise dos resultados obtidos com os exames gerados por um aparelho de bioressonância magnética, torna-se uma alternativa viável, pois atualmente existe uma grande dificuldade de apresentar os dados dos exames obtidos pelo dispositivo para os pacientes. Isso se deve ao fato de que os documentos produzidos pelo equipamento possuem muitas páginas e muitas informações em formato de tabela, dificultando assim a análise individual e grupal destes exames.

1.2. Objetivo

Levando em consideração este contexto, o objetivo deste trabalho é desenvolver e aplicar uma solução plausível, envolvendo a Automação Robótica de processos, conhecido como RPA (*Robotic Process Automation*), e um software de inteligência de negócios, conhecido como BI (*Business Intelligence*), para que seja possível analisar com mais facilidade e dinamicidade os exames gerados por um aparelho de bioressonância e, com isso, conseguir traçar tendências sobre a saúde e o bem da população analisada.

Como objetivos específicos, tem-se:

- O desenvolvimento de um software de automação de processos, capaz de realizar a coleta dos dados dos exames (PDF) e salvá-los em uma plataforma em nuvem gratuita (tabelas).
- Desenvolver um painel de dados para receber os dados dessa plataforma em nuvem, reorganizá-los e distribuí-los em formatos gráficos.
- Realizar uma análise mais facilitada dos dados de um indivíduo ou de uma população estudada.

1.3. Estrutura do trabalho

Este trabalho está dividido em 5 capítulos:

No Capítulo 1, é feita a introdução ao tema, com a contextualização das dificuldades enfrentadas na análise de documentos provenientes de exames de bioressonância magnética. Também são apresentados o objetivo do trabalho, sua justificativa e os benefícios esperados com a aplicação de um painel de *Business Intelligence* (BI).

O Capítulo 2 aborda a revisão bibliográfica e a fundamentação teórica, reunindo os conceitos essenciais para o desenvolvimento do projeto. São discutidos temas como *Robotic Process Automation* (RPA) – incluindo sua definição, aplicação e vantagens – e *Business Intelligence*, com foco em sua importância, benefícios e principais ferramentas disponíveis no mercado.

O Capítulo 3 trata da metodologia utilizada na construção da solução proposta. São descritas as etapas de desenvolvimento da automação com RPA e a criação do painel de visualização de dados com o Microsoft Power BI, além dos procedimentos de implementação da solução.

No Capítulo 4, são apresentados os resultados obtidos a partir da análise dos dados coletados da população estudada, possibilitando a discussão e a avaliação da efetividade da solução desenvolvida.

Por fim, o Capítulo 5 traz as considerações finais do trabalho, destacando suas contribuições, limitações e sugestões para pesquisas futuras.

2. REVISÃO BIBLIOGRÁFICA E ASPÉCTOS TEÓRICOS

Este capítulo apresenta os fundamentos teóricos e as principais ferramentas tecnológicas que embasam o desenvolvimento do presente trabalho. São abordados os conceitos de automação de processos por meio da tecnologia RPA (*Robotic Process Automation*) e a utilização de soluções de *Business Intelligence* (BI), com foco no uso do software Microsoft Power BI. Além disso, são discutidas as características da linguagem de programação Python, empregada como base para a construção da solução automatizada.

2.1. Linguagem de programação Python

A linguagem Python destaca-se no cenário da Ciência da Computação e no mercado de trabalho por sua sintaxe simples, clara e de fácil compreensão, características que a tornam especialmente atrativa tanto para iniciantes quanto para profissionais experientes. Criada por Guido van Rossum e lançada oficialmente em 1991, Python é uma linguagem de alto nível, interpretada, o que significa que seu código é executado diretamente, sem a necessidade de uma etapa de compilação separada (Humphries, 2020).

Devido à sua clareza, concisão e versatilidade, Python tem sido amplamente utilizada na automação de processos, especialmente aqueles que envolvem tarefas repetitivas e baseadas em regras. Essas características a tornam uma das ferramentas preferidas para o desenvolvimento de soluções baseadas em RPA, sendo amplamente adotada por engenheiros de software e desenvolvedores que atuam com automação de processos empresariais. (Humphries, 2020).

A linguagem de programação Python foi escolhida como base para a construção do sistema devido à sua simplicidade sintática e versatilidade. Trata-se de uma linguagem de alto nível, com recursos que possibilitam desde o desenvolvimento de aplicações simples até sistemas complexos e robustos. Sua capacidade de integração com diferentes bibliotecas e plataformas a torna especialmente adequada para tarefas de automação de processos e manipulação de arquivos (Python Software Foundation, 2023).

2.1.1. PyCharm

O ambiente integrado de desenvolvimento (IDE) PyCharm foi desenvolvido pela empresa JetBrains. Este ambiente oferece uma série de funcionalidades que otimizam o processo de programação, como análise de código em tempo real, sugestões automáticas, gerenciamento de pacotes, integração com sistemas de controle de versão e a possibilidade de criação de ambientes virtuais personalizados. Além disso, o PyCharm é compatível com os principais sistemas operacionais, como Windows, Linux e macOS, facilitando sua adoção em diferentes contextos (Alura, 2023).

2.1.2. Bibliotecas Utilizadas

Diversas bibliotecas da linguagem Python foram empregadas na construção do sistema RPA, com o objetivo de automatizar tarefas, gerenciar janelas do sistema operacional, extrair dados de documentos PDF, entre outras funções. A seguir, estão descritas as principais bibliotecas utilizadas:

- **Auto-py-to-exe**

Responsável pela conversão do *script* Python em um executável (.exe), permitindo a distribuição e execução do sistema em máquinas que não possuem o interpretador Python instalado.

- **Threading**

Utilizada para executar múltiplas tarefas simultaneamente dentro do programa. Essa biblioteca gerencia múltiplas *threads* de execução, otimizando o desempenho e possibilitando que diferentes funções operem de forma paralela e independente (Digital Innovation one, 2023).

- **Pygetwindow (gw)**

Permite o controle de janelas e pastas abertas no sistema operacional Windows. Com esta biblioteca, é possível obter informações sobre janelas ativas, além de manipular seu posicionamento e dimensão, sendo frequentemente utilizada em conjunto com a biblioteca Pyautogui (Rosa, 2023).

- **Pyautogui**

Utilizada para simular ações humanas no computador, como movimentação e cliques do *mouse*, digitação no teclado e interação com a interface gráfica. Essa biblioteca é essencial para automações que demandam controle direto sobre a interface do sistema (Rosa, 2023).

- **Time**

Fornece funcionalidades para manipulação de informações temporais, como marcações de tempo e temporizadores, possibilitando o controle de intervalos de execução entre comandos, além da medição de tempo de processamento (Barbosa Filho, 2022).

- **Numpy (np)**

Biblioteca voltada ao tratamento de *arrays* e matrizes de forma eficiente. O uso da NumPy permite operações matemáticas e estatísticas de forma otimizada, ocupando menos memória e acelerando o processamento de grandes volumes de dados (Harve, 2023).

- **Tabula**

Ferramenta de código aberto utilizada para a extração de dados tabulares presentes em arquivos PDF. Além de realizar a conversão para formatos editáveis (como CSV e Excel), a biblioteca tabula também permite a manipulação desses dados, facilitando sua posterior análise e integração com outras ferramentas (Awari, 2021).

- **Fitz (PyMuPDF)**

A biblioteca Fitz, também conhecida como PyMuPDF, é utilizada para a leitura e extração de dados contidos em arquivos PDF. Ela permite localizar palavras-chave específicas dentro dos documentos e extrair conteúdos textuais de maneira estruturada, sendo uma ferramenta relevante para o tratamento automatizado de relatórios em formato digital (Villela, 2021).

- **OS**

O módulo OS, integrante da biblioteca padrão da linguagem Python, oferece recursos para a interação direta com o sistema operacional do usuário. Com ele, é possível realizar operações como criação e navegação por diretórios, execução de comandos do sistema e acesso a variáveis de ambiente, tornando-se essencial para aplicações que demandam gerenciamento de arquivos e pastas (Ramos, 2023).

- **IO**

Também pertencente à biblioteca padrão do Python, o módulo IO é voltado para o gerenciamento de operações de entrada e saída. Ele possibilita a manipulação de arquivos em diferentes formatos, como texto, binário ou dados brutos, sendo fundamental para a leitura e escrita eficiente de dados em arquivos (Mend, 2023).

- **Pandas (pd)**

A biblioteca Pandas é uma das mais utilizadas para análise e manipulação de dados estruturados. Com estruturas como *Series* e *DataFrames*, ela permite a realização de operações complexas de forma simples e rápida, como filtragens, agregações, junções e exportações de dados. Sua integração com outras bibliotecas potencializa ainda mais sua aplicabilidade em projetos de análise de dados (Harve, 2021).

- **Tkinter**

O Tkinter é uma biblioteca gráfica nativa do Python utilizada para o desenvolvimento de interfaces gráficas de usuário (GUI). Com seus componentes, como botões, caixas de entrada e menus, permite a criação de aplicações visuais de maneira prática e eficiente, facilitando a interação do usuário com o sistema desenvolvido (Hashtag treinamentos, 2024).

- **Google.oauth2.credentials**

Esta biblioteca faz parte do Google API Client Library e é responsável por gerenciar credenciais de autenticação e autorização de aplicações que interagem com os serviços da Google. Sua utilização é fundamental para garantir

a segurança e a conformidade do acesso às APIs oferecidas pela plataforma (GitHub, 2022).

- **Google.auth.transport.requests**

Integrando a Google Authentication Library, esta biblioteca fornece os meios necessários para a comunicação HTTP com os serviços de autenticação da Google. Ela é usada para realizar o transporte seguro das informações de autenticação entre a aplicação e os servidores da Google (GitHub, 2022).

- **Google_auth_oauthlib.flow**

Também pertencente à Google Authentication Library, essa biblioteca facilita a implementação do fluxo de autenticação OAuth 2.0. Sua principal função é permitir que aplicações Python acessem serviços da Google em nome de usuários específicos, automatizando o processo completo de autenticação (GitHub, 2022).

- **Googleapiclient.discovery**

Esta biblioteca é essencial para a construção de serviços de API da Google. Com ela, é possível gerar objetos de serviço que interagem com diversas aplicações da plataforma, como Google Drive, Google Sheets, Gmail e Google Calendar, permitindo o desenvolvimento de aplicações integradas e com alto grau de automação (GitHub, 2022).

- **Googleapiclient.http**

Utilizada em conjunto com as demais bibliotecas da Google API Client Library, essa ferramenta oferece suporte para requisições HTTP dentro das interações com APIs da Google. É especialmente útil para operações de envio e recebimento de dados, incluindo uploads e downloads de arquivos (GitHub, 2022).

- **Openpyxl**

A biblioteca Openpyxl permite a leitura e manipulação de arquivos do tipo Excel (.xlsx) diretamente por meio do Python, sem a necessidade de instalação de softwares da suíte Microsoft Office. Ela possibilita a leitura, modificação e

criação de planilhas, sendo particularmente útil para a manipulação de dados tabulares oriundos de relatórios e análises (Asimov Academy, 2023).

2.2. Power Query

A ferramenta Power Query foi desenvolvida pela Microsoft e é integrada ao Microsoft Excel e ao Power BI. Essa ferramenta oferece funcionalidades de extração, transformação e carregamento de dados (ETL), permitindo a integração com múltiplas fontes de dados, como arquivos no formato CSV, bancos de dados SQL e APIs (Retsum, 2023).

Ela possibilita a aplicação de filtros, a remoção de valores indesejados e a automação de etapas repetitivas, contribuindo significativamente para a organização e estruturação de conjuntos de dados originalmente desorganizados ou com inconsistências (Retsum, 2023).

2.3. Robotic Process Automation

O conceito de *Robotic Process Automation* (RPA) refere-se à criação e utilização de robôs de software, também conhecidos como *bots*, com o objetivo de automatizar tarefas rotineiras e repetitivas que, tradicionalmente, são executadas por seres humanos. Esses *bots* interagem com sistemas, aplicativos e interfaces de usuário de maneira semelhante à atuação humana, porém com maior velocidade, precisão e eficiência, além de não estarem sujeitos a erros por fadiga ou distração (Guerra, 2022).

A adoção do RPA tem se expandido significativamente no ambiente corporativo, consolidando-se como uma tecnologia estratégica para a transformação digital das organizações. Dentre os principais benefícios observados estão a redução de custos operacionais, o aumento da precisão em tarefas repetitivas, o ganho de agilidade nos processos e a liberação de recursos humanos para atividades de maior valor agregado e caráter estratégico (PRTI Digital, 2020).

A tecnologia se mostra particularmente eficaz em processos que exigem alto volume de operação, baixa tolerância a erros e necessidade de integração entre múltiplos sistemas. Exemplos comuns de aplicação do RPA incluem a manipulação de grandes volumes de dados, o processamento automatizado de

documentos e a comunicação entre plataformas corporativas diversas — um desafio comum em organizações que utilizam soluções de diferentes fornecedores (Bhardwaj, 2023).

A Figura 2 mostra que o ecossistema de desenvolvimento do RPA é amplamente compatível com a linguagem Python, devido à sua flexibilidade e capacidade de integração com uma grande variedade de aplicativos e serviços. Isso permite que aplicações automatizadas se comuniquem facilmente com ferramentas amplamente utilizadas tanto no meio empresarial quanto acadêmico, como WhatsApp, Slack, Microsoft Teams, além de serviços em nuvem como AWS e Google Drive (Botcity (2023)).

Figura 2 - Ferramentas que podem ser utilizadas em conjunto com RPA via Python.



Fonte: Botcity (2023)

O termo *Robotic Process Automation* começou a ganhar destaque no início dos anos 2000, impulsionado por empresas como Blue Prism, UiPath e Automation Anywhere, que foram pioneiras no desenvolvimento de plataformas voltadas à criação de *bots* com mínima dependência da infraestrutura de TI das companhias (PRTI Digital, 2020).

A popularização e a evolução do RPA foram aceleradas pela crescente demanda por eficiência operacional e pela necessidade de adaptação a um mercado cada vez mais dinâmico e competitivo. Além disso, a adoção de RPA

contribui significativamente para a transformação digital das organizações, ao permitir a orquestração de fluxos de trabalho automatizados em múltiplos sistemas corporativos (Bhardwaj, 2023).

Entretanto, a implementação do RPA também apresenta desafios. Um dos principais refere-se à identificação adequada dos processos passíveis de automação e à gestão eficiente da implementação da ferramenta em âmbito organizacional. De acordo com Katiyar (2024), é comum que a iniciativa parta de equipes de Pesquisa e Desenvolvimento (P&D) ou de parceiros externos, onde após implementado é exigido treinamentos específicos para a ferramenta e integração com sistemas legados já existentes na empresa.

Recentemente, o conceito de “Automação Inteligente” tem ganhado espaço no contexto de integração de novas tecnologias. Essa abordagem propõe a integração do RPA com tecnologias emergentes, como Inteligência Artificial (IA) e *Machine Learning*, permitindo que os *bots* passem a tomar decisões de forma autônoma e baseada em dados. Essa evolução representa um novo estágio na automação de processos, conhecido como “automação cognitiva”, o qual tem potencial para transformar profundamente a natureza do trabalho em diferentes setores da economia (Moraes, 2020).

2.4. Google Drive

O Google Drive é uma plataforma de armazenamento em nuvem desenvolvida pela empresa Google, lançada oficialmente em 2012. Desde então, consolidou-se como uma das soluções mais utilizadas globalmente para armazenamento e compartilhamento de arquivos *online*, competindo diretamente com outros serviços do mesmo segmento, como o OneDrive, da Microsoft (Santos, 2021).

Um dos fatores que contribuem para a popularidade do Google Drive é sua oferta gratuita de espaço de armazenamento, o que facilita sua integração com outras ferramentas e aplicações. Diversas soluções tecnológicas já incorporam o uso do Google Drive como meio de acesso e manipulação de dados, permitindo que programas externos utilizem seu potencial de armazenamento para leitura, escrita e atualização de arquivos de forma automatizada. Essa integração também beneficia ferramentas de análise e visualização de dados, como o Microsoft Power BI, que pode consumir dados

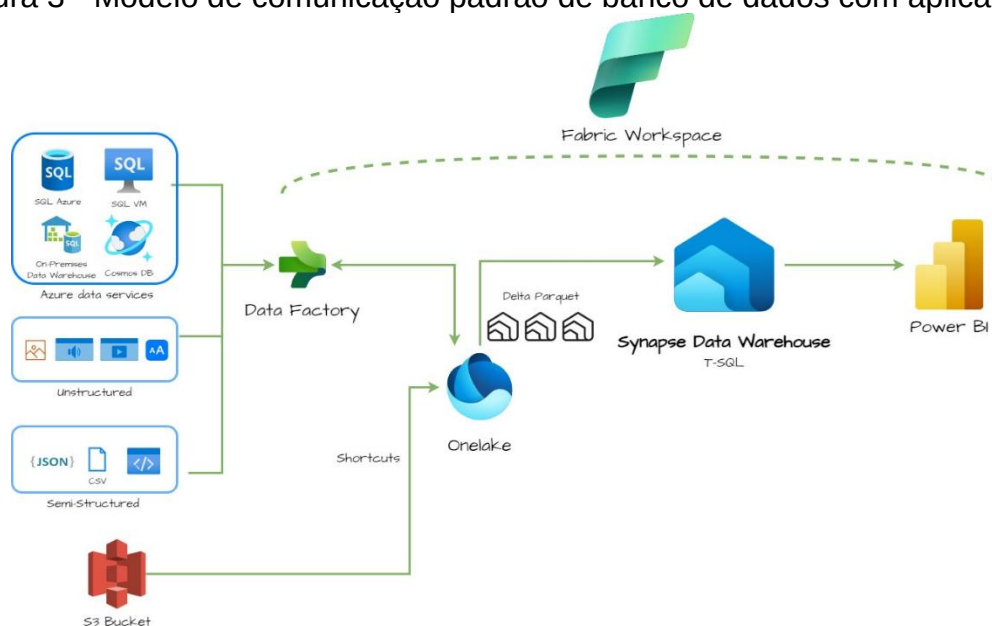
diretamente da nuvem para geração de painéis e relatórios dinâmicos (Santos 2021).

O Google Drive foi utilizado como plataforma de armazenamento em nuvem, possibilitando o armazenamento, compartilhamento e colaboração em arquivos de forma eficiente e segura. Por integrar-se ao pacote de serviços do Google, como Gmail e Google Calendar, a ferramenta proporcionou uma experiência integrada e facilitada na gestão dos documentos relacionados ao projeto (Prodanov, 2013).

Além disso, destaca-se que o Google Drive oferece suporte a APIs que permitem sua integração com diversas ferramentas e serviços, como plataformas de *Business Intelligence*, editores de mídia e sistemas de gerenciamento de tarefas, o que o torna uma solução versátil para o armazenamento e interoperabilidade dos dados utilizados no desenvolvimento da aplicação.

A Figura 3 ilustra o fluxo típico de transformação de dados em informação utilizando ferramentas do ecossistema Microsoft, o qual serviu como referência para o modelo adotado neste projeto, com a devida adaptação para o uso do Google Drive como repositório central de dados.

Figura 3 - Modelo de comunicação padrão de banco de dados com aplicações.



Fonte: Prefeitura da cidade do Rio de Janeiro (2021)

2.5. *Business Intelligence* (BI)

Conforme descrito por Davenport (2018), o conceito de *Business Intelligence* (BI), ou inteligência de negócios, tem ganhado destaque nas áreas de gestão e tecnologia da informação devido ao seu impacto direto na capacidade das organizações de tomarem decisões mais informadas, fundamentadas em dados concretos e análises aprofundadas (Davenport, 2018).

Mais do que um simples processo de coleta e processamento de dados, o BI envolve um conjunto de metodologias e ferramentas voltadas à análise estratégica da informação. Entre os recursos comumente utilizados, destacam-se os *dashboards* interativos, a mineração de dados e as análises preditivas, que têm se tornado indispensáveis para empresas inseridas em um cenário de transformação digital e competitivo (Davenport, 2018).

O avanço e a consolidação do *Business Intelligence* estão diretamente relacionados ao crescimento exponencial da geração de dados nas organizações e à necessidade de extrair valor prático dessas informações de forma rápida, precisa e estruturada (Chaudhuri, 2011). A habilidade de identificar padrões históricos e de antecipar tendências a partir da análise de dados contribui para uma atuação mais assertiva frente a contextos complexos, o que evidencia o papel estratégico do BI nas organizações contemporâneas.

Um dos aspectos mais relevantes abordados na literatura especializada é a capacidade do BI de integrar-se com tecnologias emergentes, como a Inteligência Artificial (IA) e a automação de processos. De acordo com Moraes (2020), essa integração amplia consideravelmente o potencial analítico das plataformas de BI, possibilitando, por exemplo, o desenvolvimento de *dashboards* que operam com dados em tempo real e são capazes de gerar previsões com base em algoritmos de *machine learning*. Um caso prático seria o monitoramento de equipamentos industriais, em que o sistema de BI pode prever falhas e recomendar manutenções preventivas com base no estágio da vida útil dos ativos (Moraes, 2020).

Entretanto, para que as soluções de BI sejam efetivas, é necessário atender a alguns pré-requisitos fundamentais. Em primeiro lugar, é imprescindível o investimento em ferramentas e práticas que assegurem a qualidade dos dados, uma vez que dados inconsistentes ou desatualizados

podem comprometer seriamente a confiabilidade das análises e, por consequência, as decisões tomadas (Negash, 2004). Além disso, a implementação de um modelo robusto de governança de dados é essencial, especialmente diante das exigências legais impostas pela Lei Geral de Proteção de Dados (LGPD), que estabelece diretrizes para a segurança, confidencialidade e uso responsável das informações (Ministério do esporte, 2018).

A aplicação do *Business Intelligence* não se limita ao contexto corporativo tradicional. Conforme cita Li (2024), atualmente, essa abordagem tem sido incorporada em diferentes setores, como o *e-commerce*, onde possibilita uma gestão mais eficiente de estoques e campanhas promocionais direcionadas, realizando análises e utilizando outras ferramentas como *big data* (Li, 2024). Já na área da saúde, Raghupathi (2014) descreve que o BI é utilizado para analisar prontuários eletrônicos, prever demandas por determinados atendimentos e otimizar a alocação de recursos humanos (Raghupathi, 2014).

2.5.1. Ferramentas de *Business Intelligence*

Devido ao momento atual, pode-se analisar no mercado diversos softwares e plataformas que oferecem o serviço de *Business Intelligence*. Essas ferramentas possuem a capacidade de interagir com diversos bancos de dados, onde ocorre uma coleta, tratamento e processamento destas informações. Algumas das principais soluções de *Business intelligence* são:

- Microsoft PowerBI
- Google Data Studio
- Tableau

2.5.1.1. Power BI

O Power BI é um software de *Business Intelligence* disponibilizado pela Microsoft. Seu modelo de negócios contempla uma versão gratuita — aplicada em parte deste trabalho — que apresenta funcionalidades mais enxutas, voltadas especialmente para estudantes e microempresas; e uma versão paga, destinada a companhias mais estruturadas (Microsoft, 2020).

A ferramenta se destaca como uma plataforma de análise de dados que permite aos usuários conectar-se a diversas fontes, como planilhas do Excel, bancos de dados, serviços *online* e plataformas de *Big Data*, transformando dados brutos em *insights* visuais e interativos (Microsoft, 2020).

Uma das principais vantagens do Power BI é sua facilidade de uso, especialmente quando comparada a outras ferramentas de *Business Intelligence*. Sua interface gráfica é amigável e intuitiva, remetendo a outros aplicativos da Microsoft, como o Word e o Excel. Isso possibilita que usuários sem experiência prévia em programação ou análise de dados possam criar relatórios e desenvolver *dashboards* com relativa facilidade (Mesquita, 2022).

O Power BI também se destaca pelo uso de modelos semânticos, que viabilizam a criação de uma camada lógica entre os dados brutos e os painéis informacionais. Essa estrutura favorece a compreensão dos dados, promovendo maior consistência e eficiência nas análises, além de permitir o uso de uma base comum, eliminando a necessidade de replicação de cálculos (Bansal, 2023).

Como mencionado anteriormente, é possível integrar soluções de *Business Intelligence* com diversos outros aplicativos. Nesse contexto, o Power BI pode ser considerado uma das ferramentas mais robustas do mercado. Ele se conecta não apenas a fontes de dados tradicionais, como SQL Server, Oracle e Excel, mas também a plataformas na nuvem, como Google Cloud — também utilizada neste trabalho —, Salesforce e Azure. Isso permite que diferentes departamentos de uma organização utilizem uma plataforma unificada (Sharda, 2017).

Adicionalmente, o software oferece funcionalidades como o recurso “Quick Insights”, baseado em algoritmos analíticos, que gera automaticamente interpretações relevantes a partir dos dados. Esse recurso reduz a necessidade de intervenção humana, contribuindo para a diminuição de erros em análises estatísticas (Krishnan, 2017).

3. MATERIAIS E MÉTODOS

Esta seção apresenta os recursos utilizados para o desenvolvimento do trabalho, bem como os procedimentos adotados para sua execução. Os materiais empregados estão diretamente relacionados às tecnologias utilizadas na automação e análise dos dados provenientes de exames de bioressonância magnética. A seguir, são descritos os principais materiais utilizados.

3.1. Materiais

3.1.1. Analisador de bioressonância magnética

Para a realização deste estudo, foi utilizado um analisador de bioressonância magnética T-Q01A, modelo 10th QRMA (Figura 4), disponibilizado pela Prof^a. Me. Patricia Moriguchi, da Universidade Paulista (UNIP), campus Sorocaba.

O equipamento necessita de um software específico para seu pleno funcionamento, responsável por gerar os resultados dos exames em arquivos no formato PDF, contendo os dados dos pacientes analisados.

Figura 4 - Analisador de bioressonância magnética realizando um exame.



Fonte: Foto de autoria própria.

Segundo Vaz (2020), o funcionamento do analisador baseia-se na identificação de frequências eletromagnéticas e possíveis desequilíbrios energéticos no corpo humano, possibilitando a quantificação de diversos índices

de maneira rápida e não invasiva. O exame é realizado por meio de uma haste metálica que o paciente deve segurar (ver Figura 4), enquanto sensores integrados captam sinais eletromagnéticos emitidos pelo corpo, permitindo a coleta de dados bioenergéticos para posterior interpretação e análise.

3.1.2. Computador

Para o desenvolvimento e execução das soluções propostas neste trabalho, foi utilizado um computador pessoal da marca Dell, modelo Latitude 3380. Este equipamento é composto pelas seguintes especificações técnicas: processador Intel Core i5-7200U de sétima geração, 8 GB de memória RAM DDR4, unidade de armazenamento em estado sólido (SSD) com capacidade de 256 GB e placa gráfica integrada Intel HD Graphics.

O referido equipamento foi utilizado como ambiente principal para o desenvolvimento da aplicação em linguagem Python, além de servir como estação de trabalho para a realização das conexões com os serviços de armazenamento em nuvem. Adicionalmente, foi por meio deste computador que se procedeu à criação e customização dos painéis de visualização de dados utilizando o software Microsoft Power BI, possibilitando a análise gráfica das informações extraídas dos exames de bioressonância magnética.

3.2. Métodos

O desenvolvimento da solução automatizada para extração e organização de dados dos exames de bioressonância magnética foi realizado por meio da criação de um sistema de *Robotic Process Automation* (RPA). Para isso, foi utilizado a linguagem de programação Python (seção 2.1), o ambiente de desenvolvimento PyCharm (seção 2.1.1) e bibliotecas responsáveis por estabelecer a conexão com a base de dados no Google Drive, coletar e tratar os dados dos exames (seção 2.1.2).

Com essas ferramentas foi possível desenvolver e implementar o programa, cujo código fonte se encontra nos apêndices A, B e C.

3.2.1. Introdução

A análise dos dados provenientes de exames de bioressonância magnética realizados com diferentes grupos populacionais é uma tarefa difícil devido a forma como eles são disponibilizados pelo equipamento. Assim, o desenvolvimento de uma ferramenta para tornar mais clara e acessível de visualização desses dados, de modo a facilitar a compreensão dos resultados e, consequentemente, auxiliar na promoção da saúde e do bem-estar físico dos indivíduos avaliados foi implementada.

Para a implementação e testes da ferramenta, exames de bioressonância foram realizados pela Patricia Moriguchi, com diversos grupos de pessoas pertencentes a contextos específicos, tais como: o corpo docente e técnico-administrativo da UNIP/Sorocaba, integrantes do grupo UNATI (Universidade Aberta à Terceira Idade), vinculado à UNESP/Sorocaba, e crianças atendidas pelo educandário Bezerra de Menezes, também localizado na cidade de Sorocaba. A população escolhida para realizar o estudo apresentado neste trabalho é o quadro de funcionários da UNIP/Sorocaba.

Inicialmente, a análise dos exames era conduzida de forma individualizada, com a avaliação sendo feita paciente por paciente. Tal abordagem não permitia uma visão consolidada do estado geral de um grupo, além de apresentar desafios operacionais significativos, especialmente nos exames de adultos, que tendem a ser mais extensos. Isso dificultava a priorização dos dados mais críticos e relevantes, uma vez que o processo era realizado manualmente, com base em arquivos PDF digitalizados ou impressos.

Devido a essas dificuldades, gerou-se um plano de ação para o trabalho que foram divididos em duas etapas principais, como mostra a Figura 5.

Figura 5 – Fluxograma das etapas de execução do projeto.



Fonte: Autoria própria.

Etapa 1 – Desenvolvimento de um sistema de coleta de dados: nesta fase, foi criada uma solução automatizada com base em RPA (*Robotic Process Automation*). Esse sistema realiza a leitura metódica dos exames gerados pelo aparelho de bioressonância, extrai as informações relevantes e as armazena em uma base de dados na nuvem. A base de dados é atualizada automaticamente sempre que a execução do programa é iniciado, garantindo que as informações estejam sempre sincronizadas com os exames mais recentes.

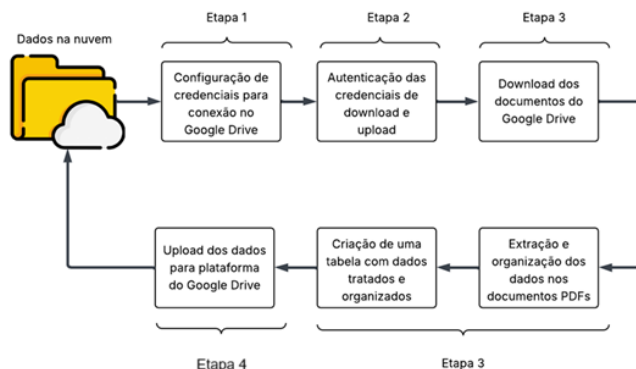
Etapa 2 – Criação de um painel de visualização gráfica: utilizando uma ferramenta de *Business Intelligence* (BI), foi desenvolvido um painel gráfico conectado à base de dados gerada na etapa anterior. Este painel possibilita a visualização dos dados de forma clara e interativa, com atualizações diárias, permitindo análises individuais e coletivas de maneira mais eficiente e acessível.

3.2.2. Desenvolvimento da solução para coleta de dados

A solução automatizada para coleta de dados foi desenvolvida utilizando a linguagem de programação Python, devido à sua versatilidade e ampla compatibilidade com bibliotecas voltadas para automação de processos. O objetivo foi criar um RPA (*Robotic Process Automation*) capaz de simular ações humanas, automatizando a extração e o tratamento dos dados contidos nos exames de bioressonância.

O desenvolvimento do algoritmo seguiu uma estrutura dividida em etapas, conforme apresentada na Figura 6.

Figura 6 – Fluxograma das etapas de desenvolvimento do RPA.



Fonte: Autoria própria.

Etapa 1 – Geração de credenciais de acesso ao Google Drive: inicialmente, foi estabelecida a conexão entre o RPA e o Google Drive, com a geração das credenciais de autenticação necessárias para permitir o acesso e manipulação de arquivos na nuvem.

Etapa 2 – Implementação da integração com o Google Drive: a segunda etapa consistiu no desenvolvimento do código responsável pela autenticação da aplicação junto ao Google Drive, utilizando as credenciais obtidas anteriormente. Esta conexão foi essencial para viabilizar o armazenamento e a atualização contínua dos dados extraídos dos exames, realizando downloads e uploads.

Etapa 3 – Download, extração e organização dos dados: durante a terceira etapa implementou-se o módulo responsável pela identificação da pasta que contém os documentos em formato PDF. O algoritmo realiza a leitura das tabelas presentes nos arquivos, extrai os dados relevantes, organiza as informações em estrutura tabular e as insere de forma automatizada no repositório definido na etapa anterior.

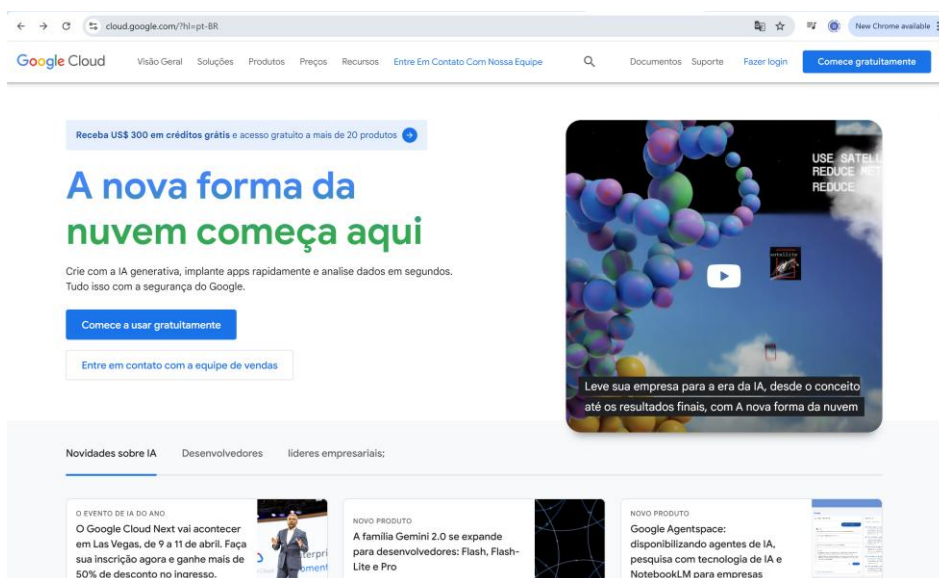
Etapa 4 – Upload dos dados para plataforma do Google Drive: Por fim, a quarta etapa consistiu em utilizar os dados de credenciais gerados anteriormente para que fosse possível subir o documento gerado pela Etapa 3 para o banco de dados do Google Drive, encerrando assim o ciclo e mantendo o arquivo atualizado na nuvem.

Essa estrutura permitiu a automação completa do processo de coleta, organização e armazenamento dos dados, reduzindo significativamente o tempo necessário para a manipulação manual dos exames e aumentando a confiabilidade e integridade dos dados gerados.

3.2.2.1. Elaboração da conexão entre o RPA e o Google Drive

Para realizar a conexão do programa com a plataforma Google Drive, primeiramente foi necessário acessar a conta Google e selecionar o ícone do Google Cloud, como mostrado pela Figura 7, onde foi realizado o cadastro e a configuração da API que conecta o RPA com a conta Google.

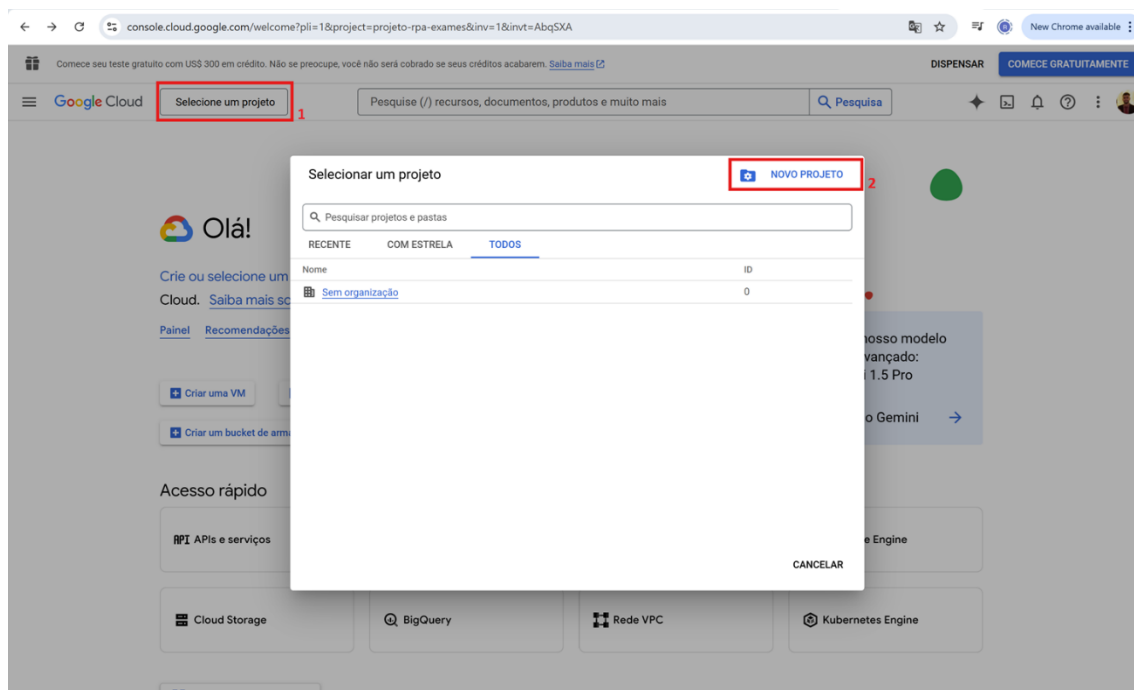
Figura 7 – Página inicial da página do Google Cloud



Fonte: Captura de tela de autoria própria.

Realizado o *login* da conta, foi necessário criar um projeto dentro da plataforma para que fosse possível coletar os dados necessários e realizar a conexão do RPA com a API. Para isso, inicialmente clicou-se no botão dentro da área denominada como 1, e logo após isso, na área denominada como 2, como mostra a Figura 8.

Figura 8 – Primeiros passos para criação de projeto no Google Cloud



Fonte: Captura de tela de autoria própria.

Com o projeto criado, é feita sua seleção no menu de projetos apresentado e, com isso, direcionado para a tela da Figura 9. Nesta tela, clicou-se no botão dentro da área 1 para abrir o menu de todos os produtos disponibilizados pela plataforma Google e, em seguida, no botão dentro da área 2 para ser direcionado ao menu de gerenciamento de APIs.

Figura 9 – Criação de uma API dentro de um projeto do Google Cloud

Google Cloud Projeto RPA - Exames Pesquise (/) recursos, documentos, produtos e muito mais Pesquisa

Soluções / Soluções

1 Todos os produtos

Encontre e implante mais de 4.500 produtos no Marketplace

ACESSAR O MARKETPLACE

2 APIs e serviços

Administração

Controle custos, defina o gerenciamento de identidade e acesso e use APIs

Nome	Descrição
APIs e serviços	Gerencie APIs para serviços em nuvem
Google Auth Platform	Configuração e credenciais do OAuth
Faturamento	Ferramentas para gerenciar faturas e custos
IAM e administrador	Controle de acesso a recursos
Configuração do Google...	Definir e implantar práticas recomendadas
Admin do Gemini	Compra e gestão de assinatura no Google Cloud

Computação

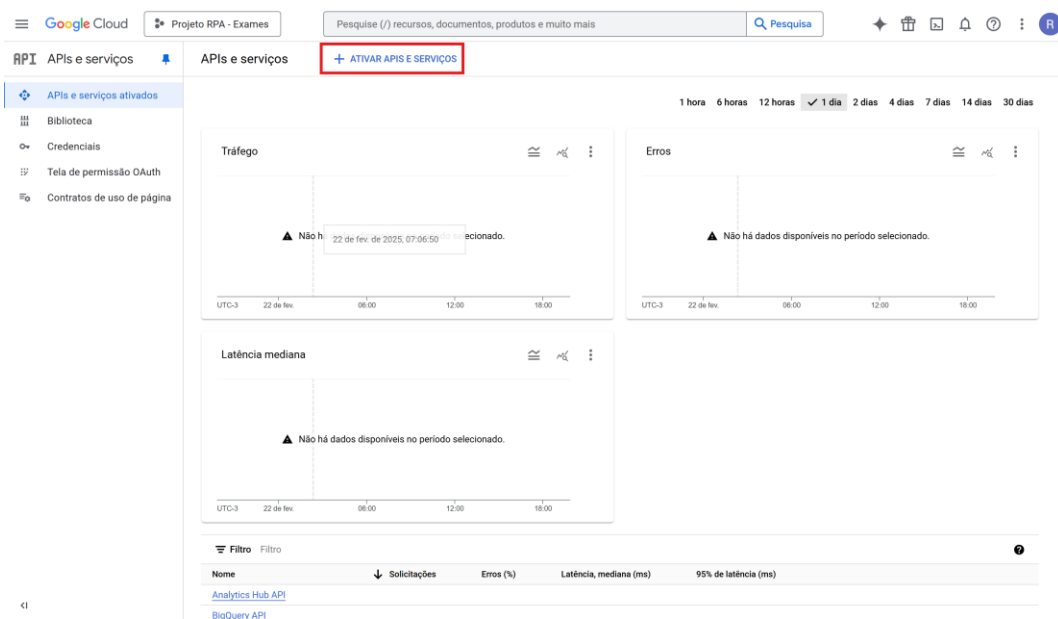
Execute máquinas virtuais e contêineres escalonáveis

Nome	Descrição
Compute Engine	VMs, GPUs, TPUs, discos
Kubernetes Engine	Contêineres/Kubernetes gerenciados
VMware Engine	VMware como serviço

Fonte: Captura de tela de autoria própria.

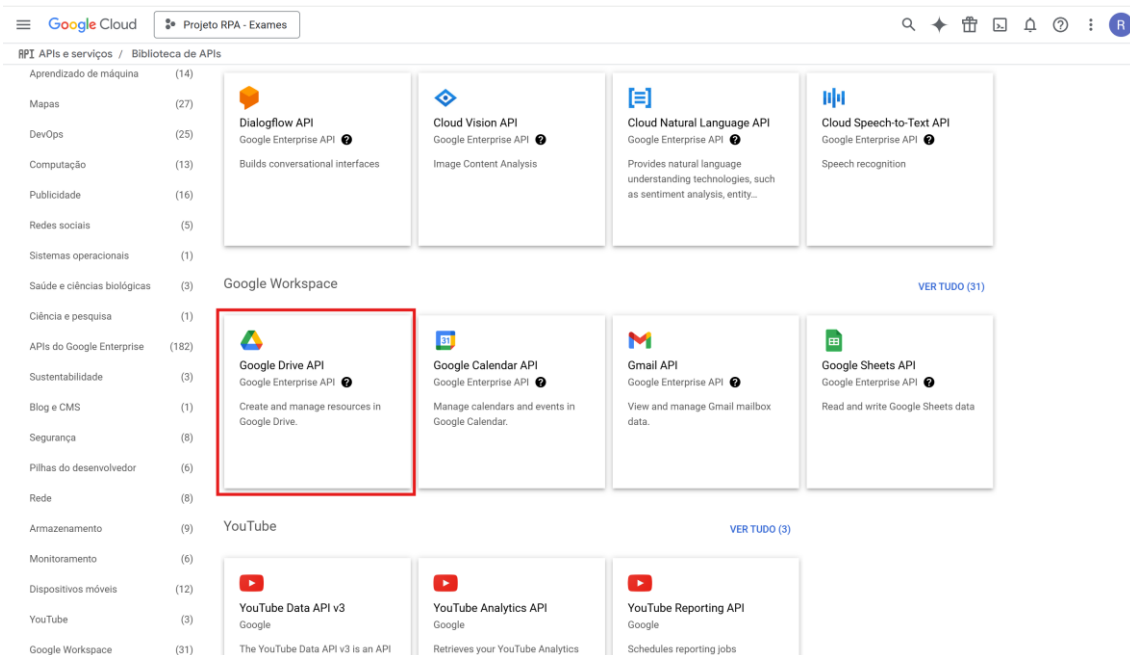
Após selecionado o campo de APIs e serviços, foi necessário ativá-los conforme mostra na Figura 10 e, em seguida, foi escolhida a opção Google Drive API como mostra a Figura 11.

Figura 10 – Demonstração da seleção do campo de ativação de APIs



Fonte: Captura de tela de autoria própria.

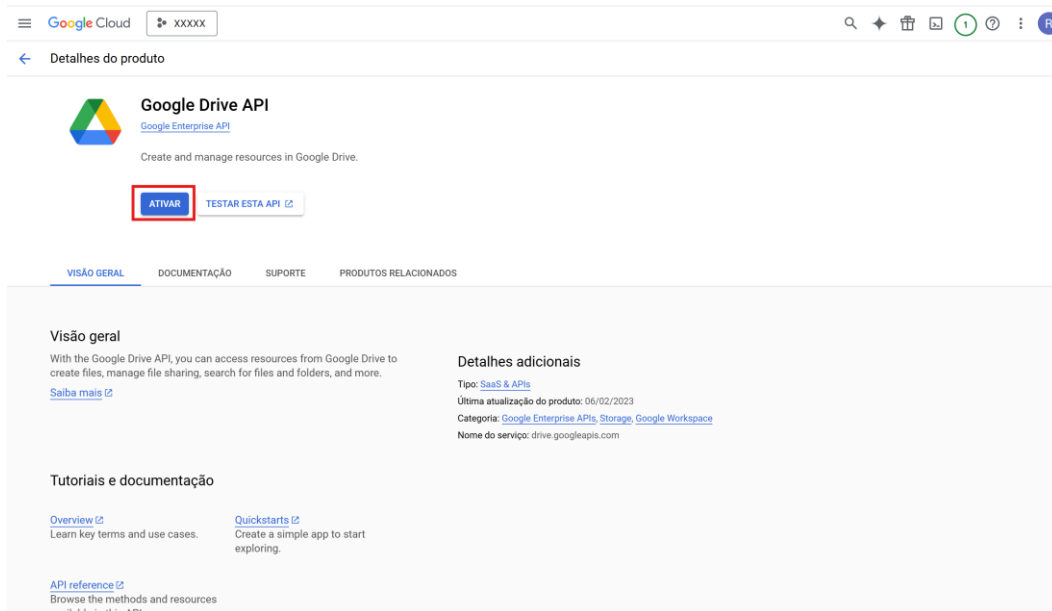
Figura 11 – Seleção da API relacionada ao funcionamento do Google Drive



Fonte: Captura de tela de autoria própria.

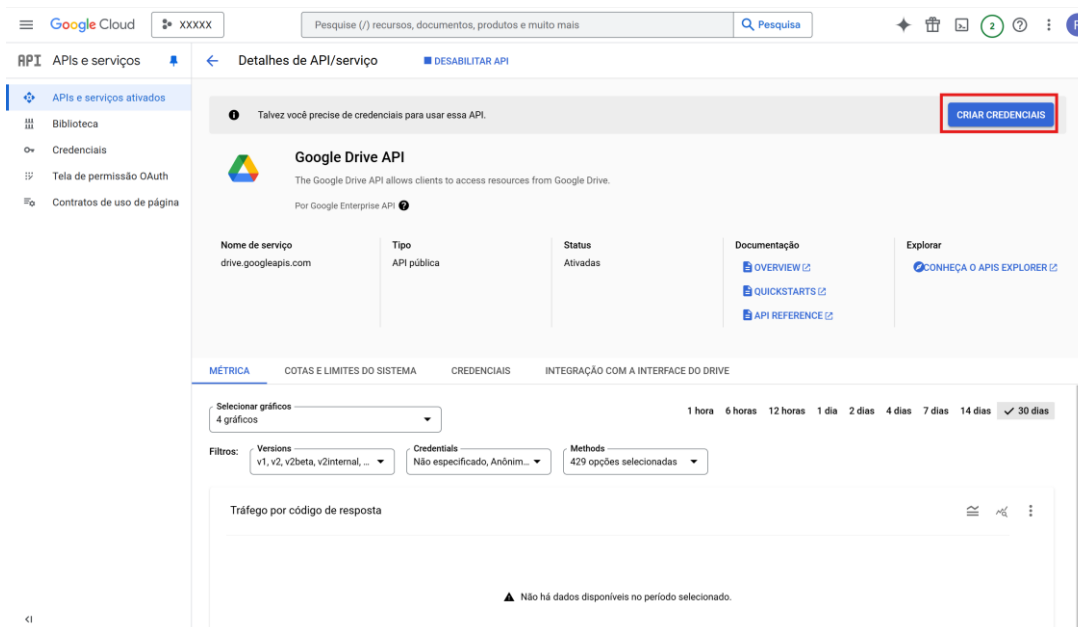
Com isso, uma nova janela é disponibilizada (Figura 12) onde foi apresentada a tela de ativação da API do Google drive, e quando ativada, o fluxo é direcionado para a janela de criação das credenciais (Figura 13).

Figura 12 – Demonstração do campo de ativação da API do Google Drive



Fonte: Captura de tela de autoria própria.

Figura 13 – Criação das credenciais para utilização da API



Fonte: Captura de tela de autoria própria.

Após selecionar a opção de criação das credenciais (Figura 13), ocorreu o direcionamento para a página de configuração da credencial, onde foi necessário selecionar a API que seria utilizada, neste caso, o Google Drive, bem como o tipo de dado, conforme mostra a Figura 14. O tipo de dado selecionado foi “dados do usuário”, devido a simplicidade e pelo trabalho não ser vinculado a uma companhia ou a um *e-mail* corporativo.

Figura 14 – Seleção dos tipos de credenciais para a API do Google Drive

A imagem mostra a interface de criação de credenciais no Google Cloud Console. No topo, há uma barra de busca e uma barra de navegação com o menu "APIs e serviços". O menu "Credenciais" está selecionado. A página principal é intitulada "Criar credenciais" e contém duas etapas:

- 1 Tipo de credencial**: Esta etapa pergunta "Qual API você usa?". Abaixo, há uma seção "Selecione uma API *" com uma lista suspensa que mostra "Google Drive API". Abaixo disso, há uma seção "Que dados você acessará? *". Nesta seção, há duas opções: "Dados do usuário" (selecionada com um botão de rádio) e "Dados do aplicativo". A opção "Dados do usuário" tem um ícone de informação e um texto explicativo: "Dados de um usuário do Google, como o endereço de e-mail ou a idade. O consentimento do usuário é obrigatório. Isso vai criar um cliente OAuth".
- 2 Suas credenciais**: Esta etapa não é visível na imagem, mas é a próxima etapa no processo.

Na parte inferior da primeira etapa, há um botão "PRÓXIMA". Na parte inferior da segunda etapa, há botões "CONCLUIR" e "CANCELAR".

Fonte: Captura de tela de autoria própria.

Na sequência, foi necessário preencher as informações requisitadas como o “endereço de *e-mail*” e “Nome do App”, conforme mostrados pelas caixas destacadas pelos números de 1 a 3 (Figura 15). Após os campos serem preenchidos, selecionou-se o botão mostrada pela caixa de número 4 (Figura 15).

Figura 15 – Cadastro dos dados para a verificação via OAuth

Google Cloud XXXXX Pesquisar (7) recursos, documentos, produtos e muito mais

APIs e serviços

- APIs e serviços ativados
- Biblioteca
- Credenciais**
- Tela de permissão OAuth
- Contratos de uso de página

2 Tela de permissão OAuth

Informações do app

Essas informações são exibidas na tela de consentimento. Elas informam aos usuários finais quem é você e como entrar em contato.

Nome do app *
Exemplo 123

O nome do aplicativo que aparece na permissão.

E-mail para suporte do usuário *
exemplo@gmail.com

Para que os usuários entrem em contato com você a respeito do consentimento. [Saiba mais](#)

Logotipo do app

Este é seu logotipo. Ele ajuda as pessoas a reconhecerem o app e aparece em destaque na tela de permissão OAuth. Depois de fazer upload de um logotipo, será necessário enviar o app para verificação, a menos que esteja configurado para uso interno ou tenha o status de publicação "Teste". [Saiba mais](#)

Arquivo de logotipo a fazer upload [PROCURAR](#)

Faça upload de uma imagem de até 1 MB na tela de permissão para ajudar os usuários a reconhecerem seu app. Os formatos de imagem permitidos são JPG, PNG e BMP. Para garantir o melhor resultado, os logotipos precisam ser quadrados e ter a resolução de 128 px por 128 px.

Dados de contato do desenvolvedor

Endereço de e-mail *
exemplo@gmail.com

O Google usa esses endereços de e-mail para notificar você sobre todas as alterações no projeto.

SALVAR E CONTINUAR

1 **2** **3** **4**

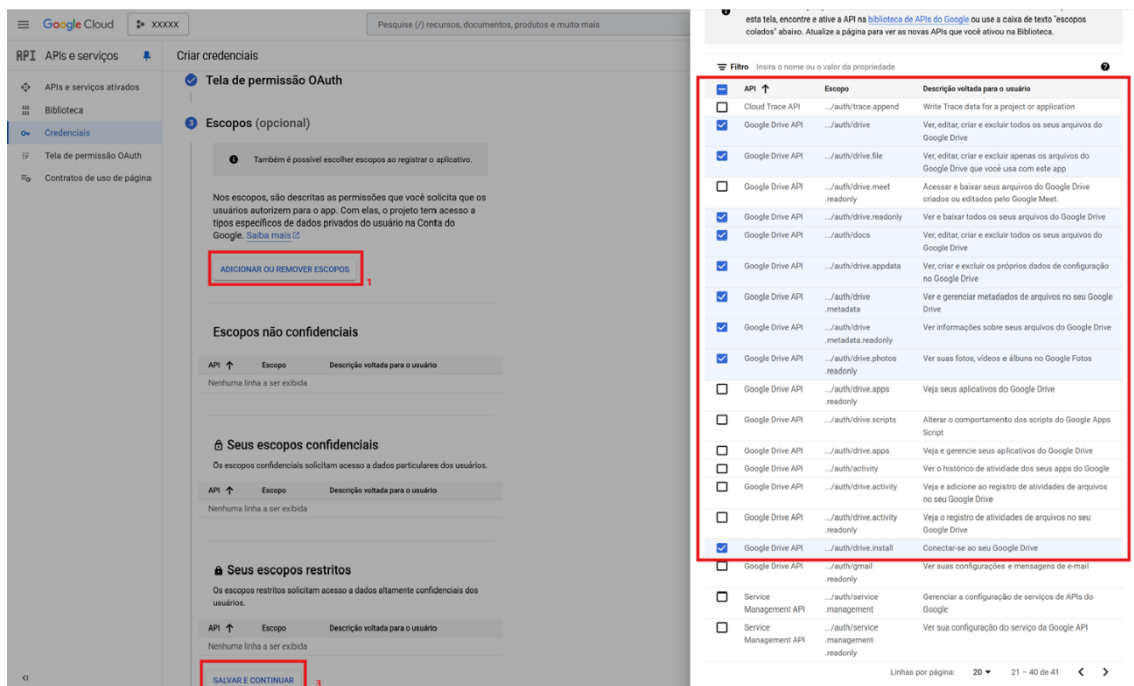
1 Escopos (opcional)

2 ID do cliente OAuth

Fonte: Captura de tela de autoria própria.

Na Figura 16, foi necessário selecionar os escopos de trabalho das credenciais, ou seja, o que será utilizado por aquela API, seja realizar *downloads* e *uploads* ou alterar dados de dentro do Google Drive. Primeiro, foi necessário acessar o botão demarcado como 1. A seguir, foram selecionadas as permissões do que será realizado pela API (destacada pelo número 2), e na sequência, clicou-se no botão mostrado pela caixa de número 3 para concluir.

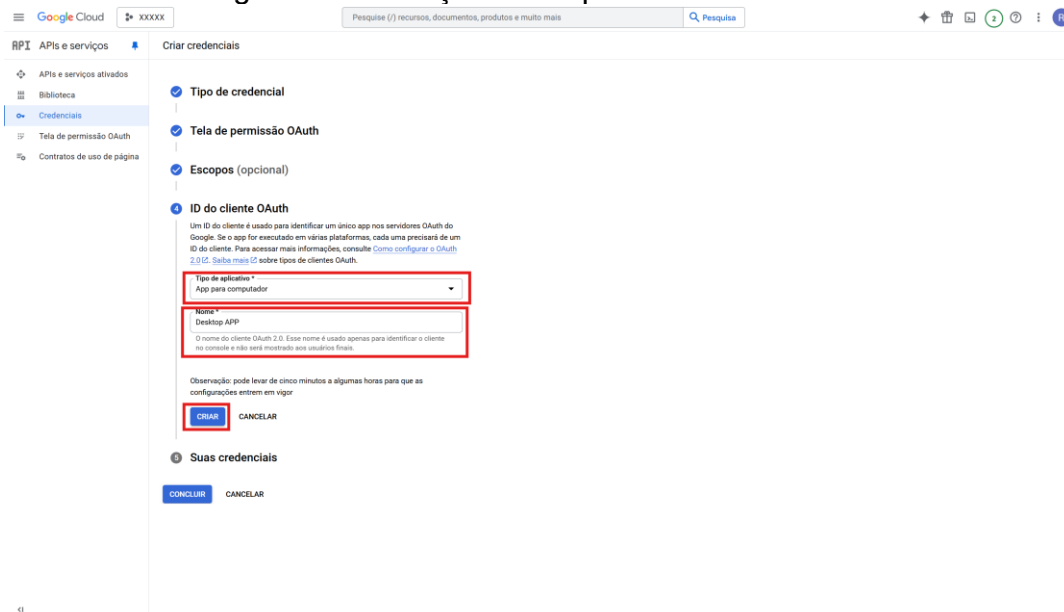
Figura 16 – Seleção de todos os serviços que serão utilizados na aplicação



Fonte: Captura de tela de autoria própria.

Por fim, na Figura 17, foi necessário selecionar a opção “App para computador” e nomear o cliente. Essa opção foi escolhida pois o RPA é um programa para computadores e, somente com essa opção selecionada, ele consegue acessar o Google Drive.

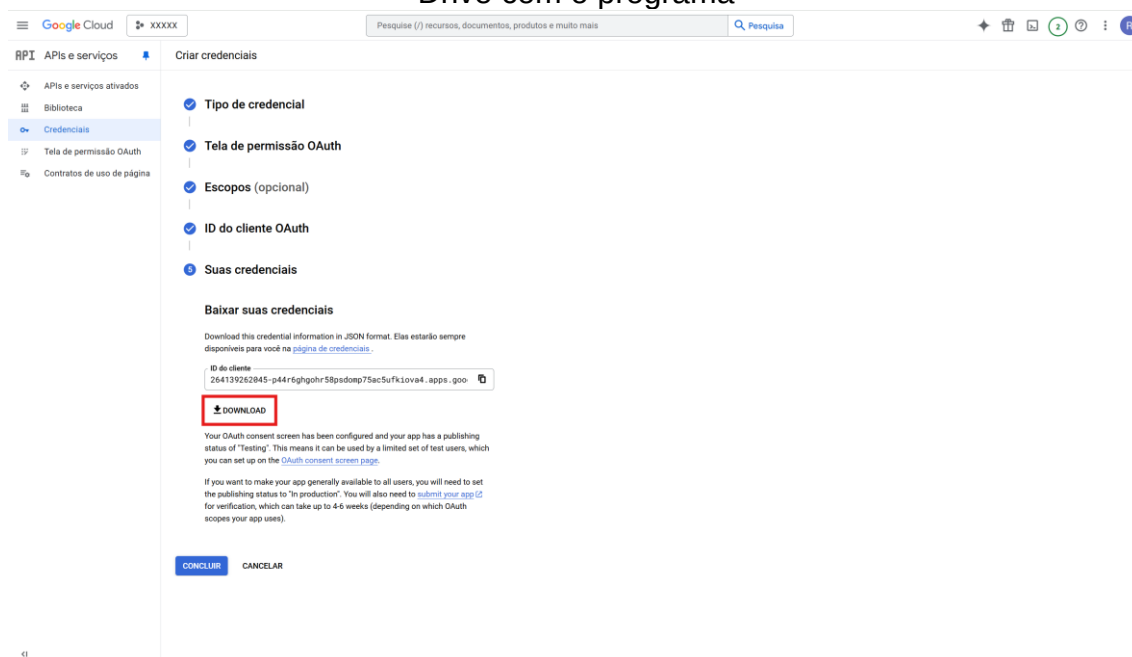
Figura 17 – Criação do ID para o cliente OAuth



Fonte: Captura de tela de autoria própria.

Após a execução dos passos anteriores, foi criada a credencial. Para obter os dados dessa credencial e adicioná-los no código do RPA foi necessário realizar o *download* do arquivo (Figura 18). Os dados da credencial que foram utilizados são as variáveis “*client_id*” e “*client_secret*”.

Figura 18 – Tela de *download* das credenciais de conexão da API do Google Drive com o programa



Fonte: Captura de tela de autoria própria.

3.2.2.2. Desenvolvimento do código para conexão entre RPA e Google Drive

Para a integração entre o *Robotic Process Automation* (RPA) e a plataforma Google Drive, foi desenvolvido um código específico (Apêndice A), cuja funcionalidade principal consiste em realizar o *download* e o *upload* de arquivos. Inicialmente, foram utilizadas variáveis provenientes da etapa anterior, como “*client_id*”, “*client_secret*”, “*FileId*”, “*TargetFolderId*” e “*TargetFieldId*”, além dos *links* correspondentes ao arquivo de origem e a pasta de destino no Google Drive.

A comunicação com a plataforma foi viabilizada por meio das funções “*generate_tokens*” e “*authenticate_with_token*”, responsáveis por gerar e autenticar as credenciais de acesso, utilizando as bibliotecas disponibilizadas pelo Google (conforme seção 2.1.2). Essas bibliotecas foram configuradas com

base na API definida na seção 3.2.2.1, que trata da elaboração da conexão com o Google Drive.

Adicionalmente, foram criadas as funções “*generate_tokens_upload*” e “*authenticate_with_token_upload*”, destinadas especificamente à autenticação de credenciais para o processo de *upload*. Para automatizar o processo de autenticação da conta Google, foi implementada a função “*open_browser*”, que utiliza a biblioteca “*Threading*” para executar, em paralelo, os cliques necessários para o *login* enquanto a autenticação é processada.

3.2.2.3. Elaboração das funções responsáveis por coletar e concatenar os dados dos exames

A coleta de dados oriundos dos exames de biorressonância foi realizada por meio de um programa principal (Apêndices B e C), que define inicialmente as variáveis “*paginas*”, “*paginas2*” e “*colunas_preenchimento*”, utilizadas nas etapas subsequentes.

Em seguida, é executada a função “*iniciar_ou_encerrar*”, que apresenta ao usuário uma interface de decisão, desenvolvida com a biblioteca Tkinter (ver seção 2.1.2), para iniciar a aplicação. Na sequência, a função “*select_save_location*” é acionada, permitindo ao usuário indicar o caminho para salvamento do arquivo Excel que irá armazenar os dados concatenados.

Após o *download* do arquivo Excel, conforme descrito anteriormente, o sistema solicita a seleção da pasta contendo os arquivos PDF dos exames, por meio da função “*selecionar_pasta*”. Com a pasta definida, inicia-se a execução da função principal “*processar_todos_pdfs*”, que cria um *DataFrame* vazio destinado ao armazenamento dos dados extraídos.

A função “*processar_pdf*”, em conjunto com a função “*extract_data*”, realiza a leitura das informações contidas na primeira página de cada arquivo PDF, extraindo o nome, o sexo e a idade do paciente. Posteriormente, é utilizado o comando “*tabula.read_pdf*”, da biblioteca *Tabula*, para a leitura das tabelas localizadas nas páginas especificadas pela variável “*paginas*”. Esta abordagem permite maior precisão na coleta de dados, considerando que os arquivos analisados possuem centenas de páginas.

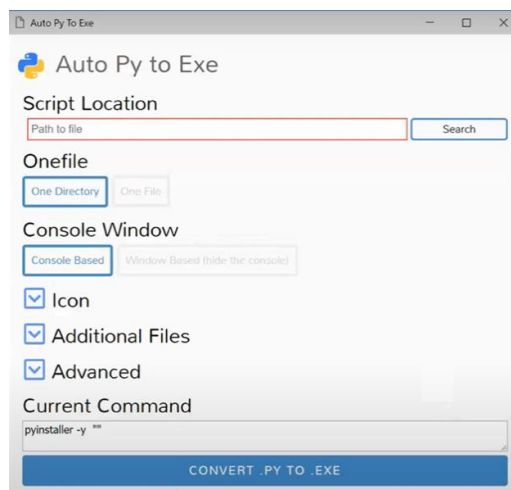
Finalizada a coleta, os dados são tratados, eliminando-se colunas desnecessárias e reorganizando as informações dos pacientes com base na estrutura definida por “*colunas_preenchimento*”. O resultado é um *DataFrame* estruturado, o qual é salvo em formato Excel. Por fim, a função de *upload* é acionada novamente, realizando a autenticação das credenciais e enviando o arquivo finalizado para o Google Drive.

3.2.2.4. Arquivo executável

Após finalizar o código para o RPA, para garantir a acessibilidade do programa por usuários não familiarizados com ambientes de desenvolvimento, foi realizada a conversão do código para um arquivo executável. Esse procedimento foi executado por meio do comando: `auto-py-to-exe`

Ao utilizar este comando no terminal do ambiente PyCharm, é aberta uma interface gráfica (Figura 19), na qual é possível selecionar o *script* Python desejado, configurar se o terminal deve ser exibido durante a execução, escolher entre a criação de um executável simples ou com diretório contendo os dados auxiliares, e definir uma imagem personalizada como ícone do executável.

Figura 19 – Terminal de seleção do comando “*auto-py-to-exe*”



Fonte: Captura de tela de autoria própria.

3.2.3. Relatório no Microsoft Power BI

Com a finalização do desenvolvimento do programa responsável pela coleta e organização dos dados dos exames, foi necessário construir um painel interativo para visualização e análise dessas informações. Para tal finalidade,

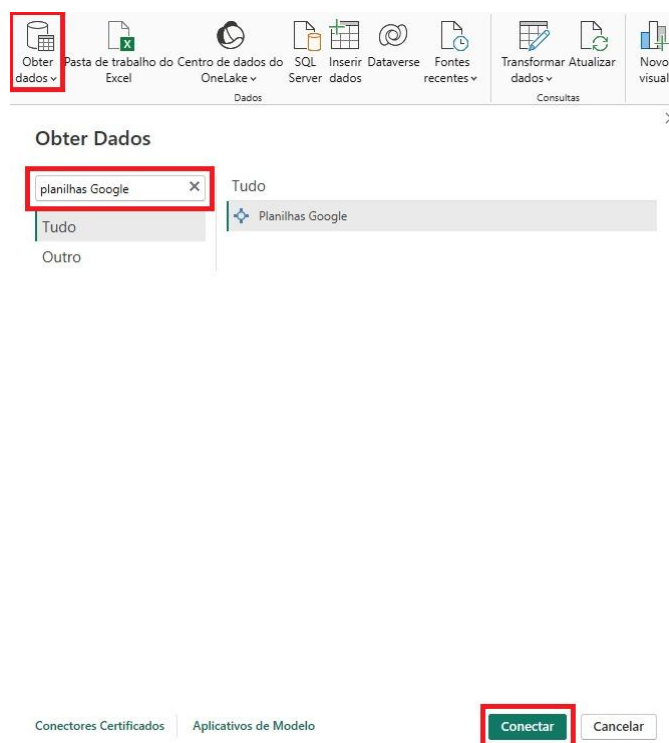
utilizou-se a ferramenta de *Business Intelligence* Microsoft Power BI, que permite a criação de relatórios dinâmicos com visualizações interativas.

O processo de construção do painel foi dividido em cinco etapas principais:

1. Conexão do Microsoft Power BI com as planilhas armazenadas no Google Drive.
2. Definição dos tipos de dados e aplicação de tratamentos às colunas provenientes dos exames.
3. Criação de colunas condicionais para categorização por faixa etária.
4. Desenvolvimento dos painéis de visualização.
5. Configuração de rotinas automatizadas para atualização da base de dados.

A primeira etapa consistiu na conexão da base de dados ao Power BI. Para isso, foi utilizada a API de integração do Power BI com o Google Sheets (Figura 20). Essa integração permitiu o carregamento dinâmico dos dados diretamente da nuvem, facilitando sua atualização contínua.

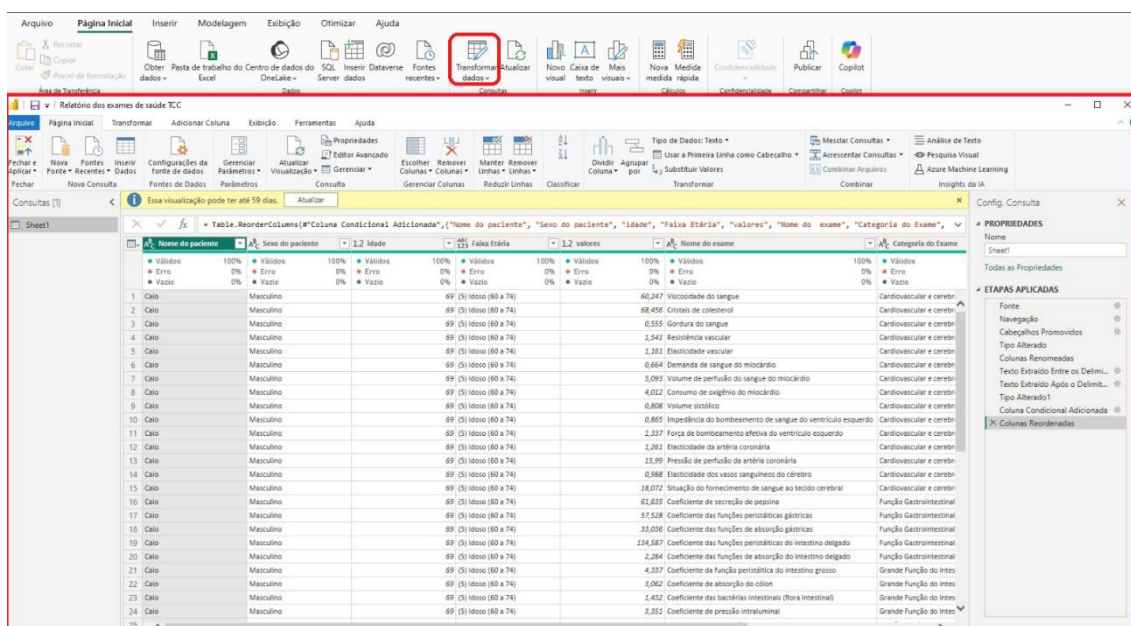
Figura 20 – API de conexão do Power BI com Google Planilhas



Fonte: Captura de tela de autoria própria.

Na etapa seguinte, procedeu-se à transformação e tratamento dos dados. Esse processo foi realizado por meio da funcionalidade "Transformar dados", presente na interface do Power BI (Figura 21). Nessa janela, utilizando-se o Power Query, foram definidos os tipos de dados de cada coluna, aplicadas correções de inconsistências, e criadas colunas com base em estruturas condicionais (etapa 3).

Figura 21 – Transformação de dados no Power Query

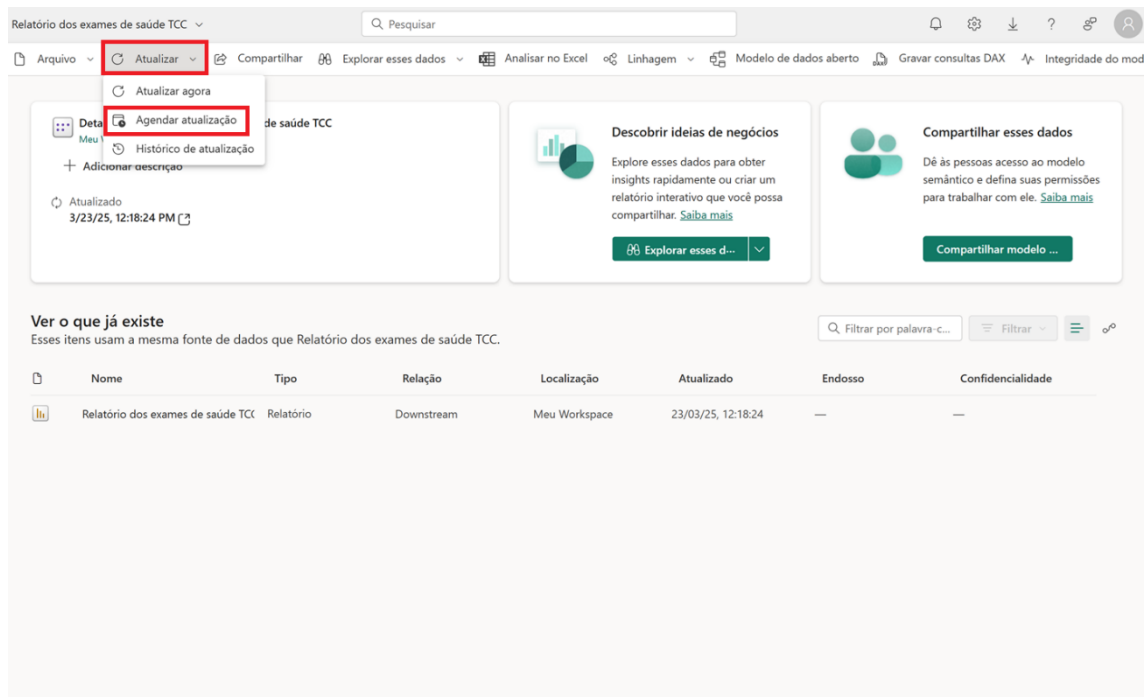


Fonte: Captura de tela de autoria própria.

Com os dados tratados e estruturados, foi possível construir os painéis interativos (etapa 4). O relatório desenvolvido inclui um gráfico geral com os limites de referência dos exames, permitindo a identificação visual de valores fora da normalidade, e uma tabela detalhada que apresenta a faixa etária, nome do exame, valores coletados e os respectivos limites. A proposta do painel é proporcionar uma visualização clara, rápida e objetiva, otimizando a análise dos resultados.

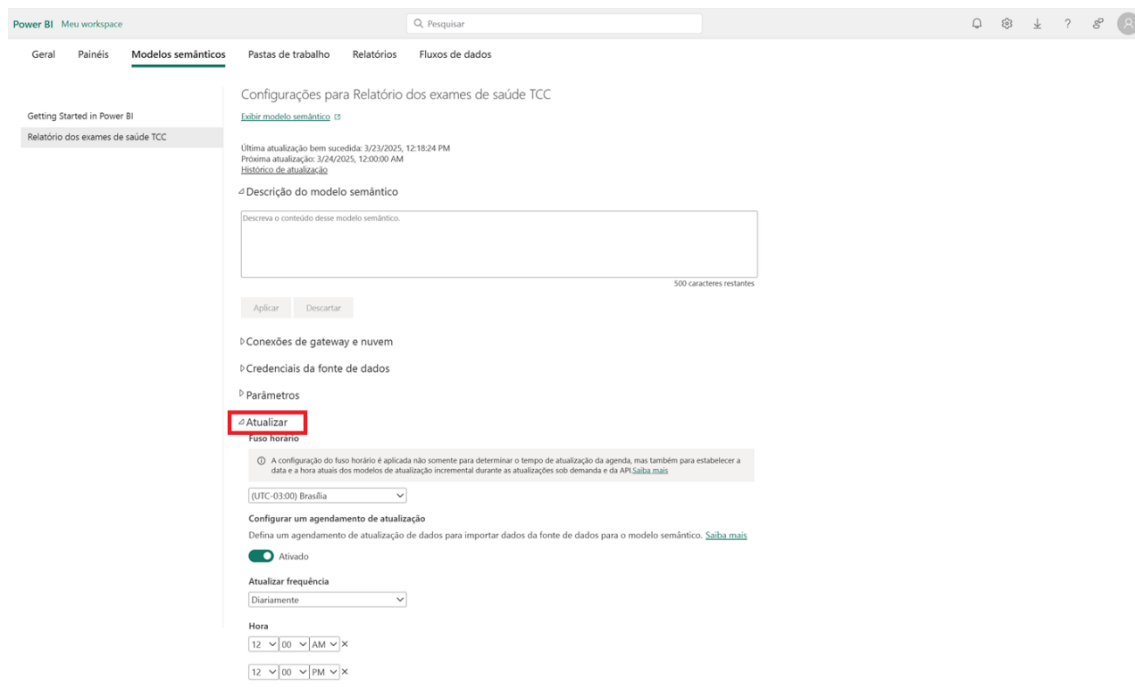
Por fim, para garantir que o relatório permaneça sempre atualizado com os dados mais recentes, foi configurada uma rotina automática de atualização (etapa 5). Esse processo envolve o acesso ao *workspace* do Power BI, a seleção do relatório desenvolvido e, posteriormente, a ativação das opções “Atualizar” e “Agendar Atualização”, conforme ilustrado nas Figuras 22 e 23. Dessa forma, o relatório se mantém sincronizado com a base de dados original, dispensando atualizações manuais frequentes.

Figura 22 – Janela de criação de rotina de atualização



Fonte: Captura de tela de autoria própria.

Figura 23 – Continuação da criação da rotina de atualização



Fonte: Captura de tela de autoria própria.

4. RESULTADOS E DISCUSSÕES

4.1. Desenvolvimento do painel de exibição

Após a coleta dos dados realizada por meio do RPA, bem como o devido tratamento e manipulação dessas informações, foi desenvolvido um painel dinâmico de visualização utilizando a ferramenta Microsoft Power BI.

O painel é estruturado em quatro quadros principais:

- Quadro 1: Exibe o número de pacientes conforme os filtros selecionados pelo usuário.
- Quadro 2: Apresenta uma tabela com os dados individuais de cada paciente.
- Quadro 3: Contém um gráfico customizado que compara os valores dos exames com os respectivos limites considerados saudáveis para adultos.
- Quadro 4: Demonstra a quantidade de exames realizados, segmentando-os entre aqueles que estão dentro ou fora dos limites adequados, de acordo com os filtros aplicados.

Para a customização dos quadros foram criados seis filtros, sendo eles:

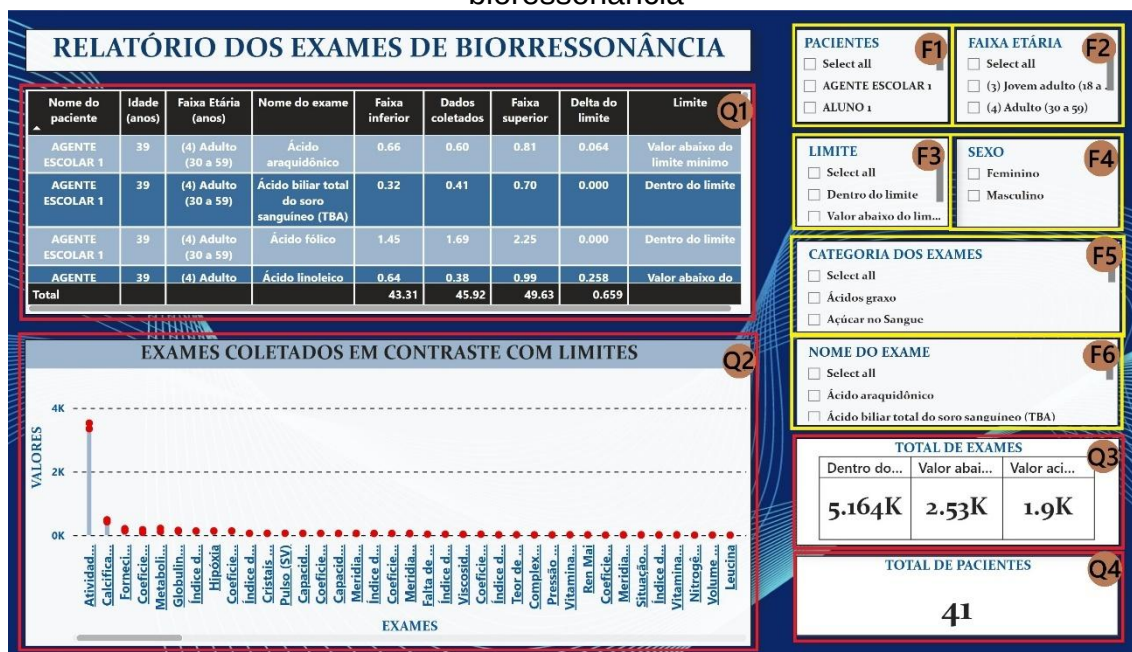
- Pacientes
- Faixa Etária
- Limite superior e inferior de valores considerados como “saudável” no exame
- Sexo
- Categoria dos exames
- Nome do exame

Quando os filtros selecionados resultam em múltiplos pacientes, os dados numéricos exibidos nos quadros representam a média dos valores coletados entre esses pacientes, permitindo, assim, uma análise representativa de um grupo populacional específico.

A Figura 24 ilustra a interface do painel, onde os filtros estão destacados com bordas amarelas e indicados como **F1**, **F2**, **F3**, **F4**, **F5** e **F6**, enquanto os

quadros com bordas vermelhas são identificados como **Q1**, **Q2**, **Q3** e **Q4**. Cada elemento do painel foi nomeado conforme assinalado na Tabela 1, facilitando a identificação e o entendimento da disposição visual dos dados.

Figura 24 – Apresentação dos quadros de informações dos exames de bioressonância



Fonte: Captura de tela de autoria própria.

Tabela 1 – Apresentação dos filtros e quadros de informações dos exames de bioressonância

Filtros e quadros	Número	Cor
Filtro para pacientes	F1	Amarelo
Filtro para faixa etária	F2	Amarelo
Filtro de limite	F3	Amarelo
Filtro para sexo	F4	Amarelo
Filtro para categoria dos exames	F5	Amarelo
Filtro para nome dos exames	F6	Amarelo
Quadro tabela de dados	Q1	Vermelho
Quadro gráfico customizado	Q2	Vermelho
Quadro total de exames	Q3	Vermelho
Quadro total de pacientes	Q4	Vermelho

Fonte: Autoria própria.

4.1.1. Desenvolvimento dos filtros para o Power BI

Com base nas necessidades de análise de dados extraídos via RPA, foram criados seis filtros interativos no Power BI, conforme detalhado na Tabela 1. Cada filtro foi desenvolvido com o objetivo de proporcionar uma análise personalizada, permitindo a segmentação e comparação dos dados de forma mais eficiente e intuitiva.

O primeiro filtro (**F1**), relacionado aos pacientes, permite a seleção de um indivíduo específico ou de múltiplos pacientes, possibilitando comparações diretas entre os dados coletados. Quando múltiplos pacientes são selecionados, os valores apresentados nos quadros correspondem à média dos resultados, o que favorece a análise estatística de um grupo populacional específico.

O segundo filtro (**F2**) refere-se à faixa etária e possibilita a seleção de pacientes com base em intervalos definidos de idade. Essa categorização facilita a análise por geração e é composta pelas seguintes faixas:

- Criança – 0 a 12 anos
- Adolescente – 13 a 17 anos
- Jovem adulto – 18 a 29 anos

- Adulto – 30 a 59 anos
- Idoso – 60 a 75 anos
- Longevo – Acima de 75 anos

O terceiro filtro (**F3**), denominado limite, foi projetado para permitir a análise de exames cujos resultados estejam fora dos parâmetros considerados saudáveis. Essa funcionalidade é importante para identificar rapidamente valores críticos e facilitar comparações entre pacientes com condições semelhantes.

O quarto filtro (**F4**) permite a seleção do sexo biológico dos pacientes, possibilitando análises comparativas entre homens e mulheres ou a identificação de tendências com base nessa variável.

O quinto filtro (**F5**) está relacionado à categoria dos exames. Os exames gerados pelo equipamento de biorressonância são organizados em categorias específicas, cada uma contendo um conjunto de exames correlacionados. A aplicação desse filtro permite a análise simultânea de exames relacionados dentro de uma mesma área de avaliação clínica. As categorias disponíveis estão listadas no Quadro 1.

Quadro 1 – Apresentação da categoria dos exames de bioressonância

Categoria dos exames		
Ácido Graxo	Colágeno	Lípideos no sangue
Função da vesícula biliar	Função pulmonar	Cardiovascular e cerebrovascular
Coenzima	Toxina humana	Sistema imunológico
Função Renal	Minerais	Pele
Olhos	Ossos e fator reumatóide	Sistema endócrino
Alergenos	Avaliação física básica	Função do fígado
Metais pesados	Função do intestino	Função pancreática
Obesidade	Função gastrointestinal	Tireoide
Aminoácidos	Açúcar no sangue	Vitaminas
Sistema nervoso	Densidade mineral óssea	

Fonte: Autoria própria.

Por fim, o sexto filtro (**F6**) permite a seleção de exames específicos. Com ele, o usuário pode visualizar com mais profundidade os resultados de um determinado exame, facilitando a análise detalhada de um parâmetro clínico particular. A relação completa entre exames e suas respectivas categorias encontra-se no Apêndice D.

4.1.2. Elaboração dos quadros de dados

O painel desenvolvido no Power BI apresenta quatro quadros principais, conforme indicado na Tabela 1 e ilustrado na Figura 24. A seguir, detalha-se a função e os elementos presentes em cada um deles.

O primeiro quadro denominado como **Q1**, corresponde as seguintes informações:

- Nome ou pseudônimo do paciente
- Idade em anos
- Faixa etária que o paciente se encontra
- Nome do exame observado
- Dados coletados do exame observado
- Limite superior e limite inferior do exame observado

- O delta do limite com os dados coletados
- Informação do limite

Especificamente, o dado definido como “Delta do limite” é calculado com base na distância do valor obtido em relação ao intervalo saudável. Esse recurso facilita a interpretação do quão distante o resultado está do parâmetro de normalidade estabelecido.

O segundo quadro (Q2) apresenta uma visualização gráfica comparativa entre os valores dos exames e seus respectivos limites saudáveis. Os limites são representados por pontos vermelhos, enquanto os valores dos pacientes aparecem em colunas azuis. Quando a coluna azul se encontra entre os pontos vermelhos, o exame é considerado dentro da normalidade; caso contrário, é classificado como fora do limite saudável. Esta representação facilita a análise visual da condição clínica do paciente ou da média do grupo.

O terceiro quadro (Q3) apresenta uma contagem agregada dos exames, categorizando-os conforme o *status* em relação aos limites: dentro, acima ou abaixo. Esse painel auxilia na identificação de padrões e prevalência de exames alterados dentro de um grupo filtrado.

O quarto e último quadro (Q4) exibe o número total de pacientes considerados na análise, de acordo com os filtros aplicados. Essa informação é fundamental para contextualizar a amostra e garantir a precisão da interpretação dos resultados.

4.2. Compartilhamento da solução RPA e publicação do relatório desenvolvido no Power BI

Após a conclusão do desenvolvimento tanto da automação com RPA quanto do painel analítico no Power BI, a solução completa foi compartilhada com a professora Patricia Moriguchi para análise da ferramenta. Também foram compartilhados a base de dados e as credenciais de acesso à conta responsável pela publicação do painel.

Além disso, o relatório foi disponibilizado na *web*, possibilitando o acesso público via *link*, tanto em computadores quanto em dispositivos móveis. Essa

funcionalidade amplia o alcance e facilita a consulta às informações por qualquer interessado que possua o *link* de visualização.

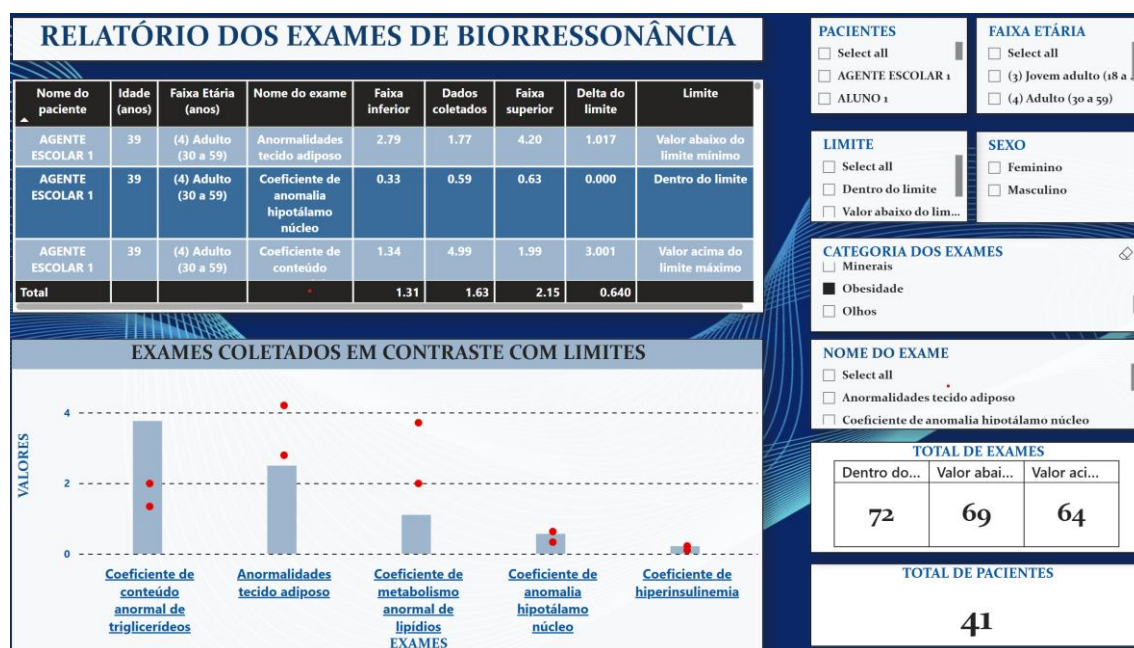
4.3. Análise dos dados coletados

Com a implementação e o estudo da plataforma de dados em conjunto com a Professora Patrícia, foi possível observar algumas tendências envolvendo certas categorias de exames.

- **Categoria obesidade**

Na análise dos exames voltados para essa categoria, como mostrado na Figura 25, observou-se que 65% dos exames (133 de 205 exames da categoria “Obesidade”) estão fora dos limites saudáveis. Dentre os parâmetros analisados, destacam-se dois exames com os piores desempenhos:

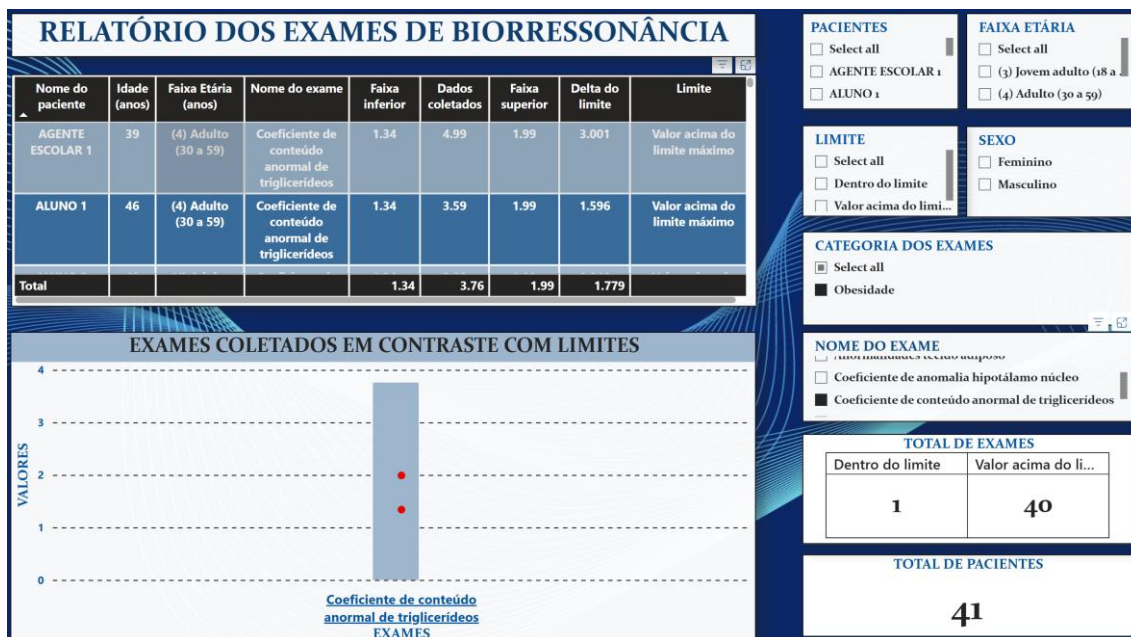
Figura 25 – Painel com dados dos Exames da categoria da Obesidade



Fonte: Captura de tela de autoria própria.

- Coeficiente de conteúdo anormal de triglicerídeos: Analisando a Figura 26, nota-se que apenas 1 paciente dentro do limite saudável.

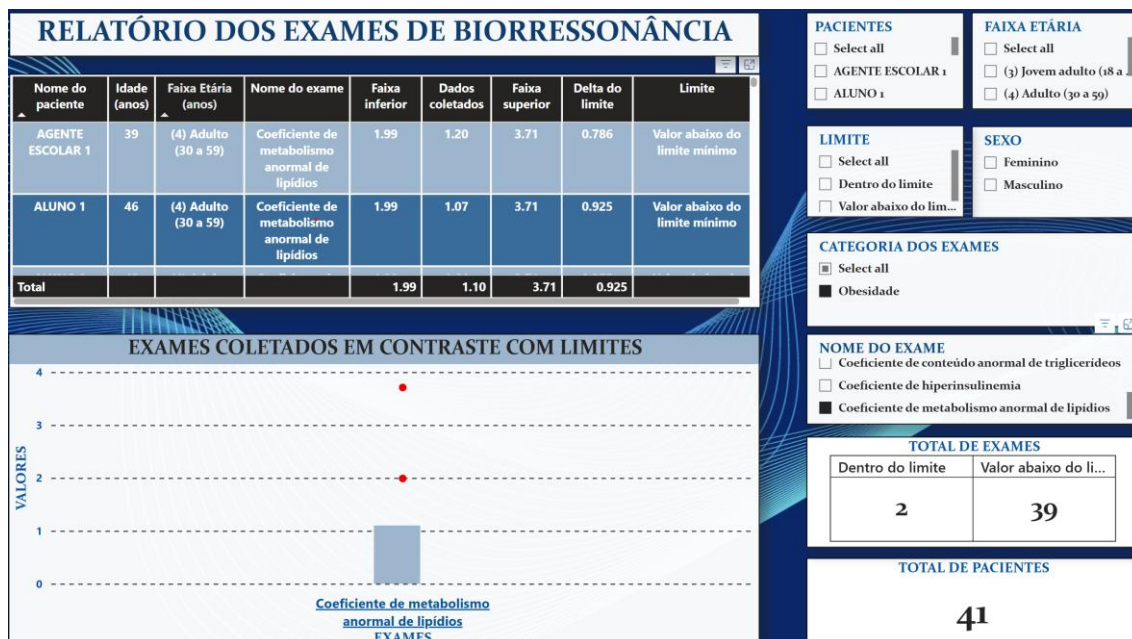
Figura 26 – Painel com dados do Exame “Coeficiente de conteúdo anormal de triglicerídeos”



Fonte: Captura de tela de autoria própria.

- Coeficiente de metabolismo anormal de lipídeos: Analisando a Figura 27, nota-se que apenas 2 pacientes com resultado dentro da normalidade.

Figura 27 – Painel com dados do Exame “Coeficiente de metabolismo de lipídeos”



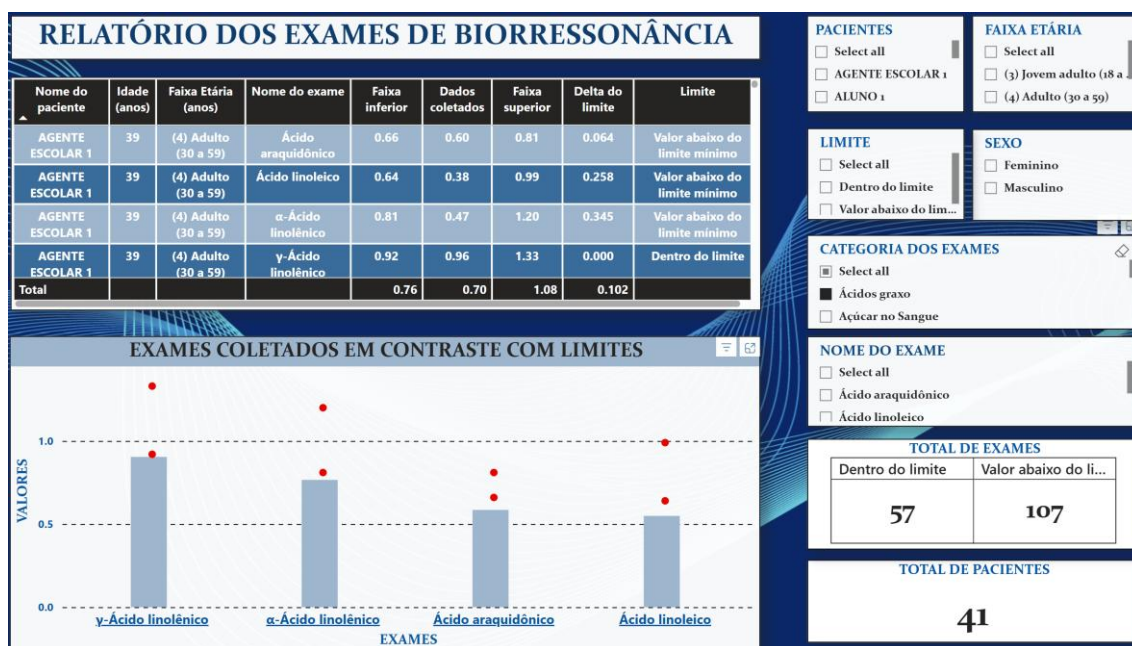
Fonte: Captura de tela de autoria própria.

- **Categoria ácidos graxos**

Nesta categoria, dos 164 exames analisados, 107 (65%) estavam com resultados alterados, indicando um padrão preocupante relacionado ao metabolismo de ácidos graxos.

De acordo com a Figura 28 é possível notar que a média de todos os exames da categoria dos ácidos graxos estão abaixo do limite considerável saudável em cada um dos exames.

Figura 28 - Painel com dados dos Exames da categoria de Ácidos graxos

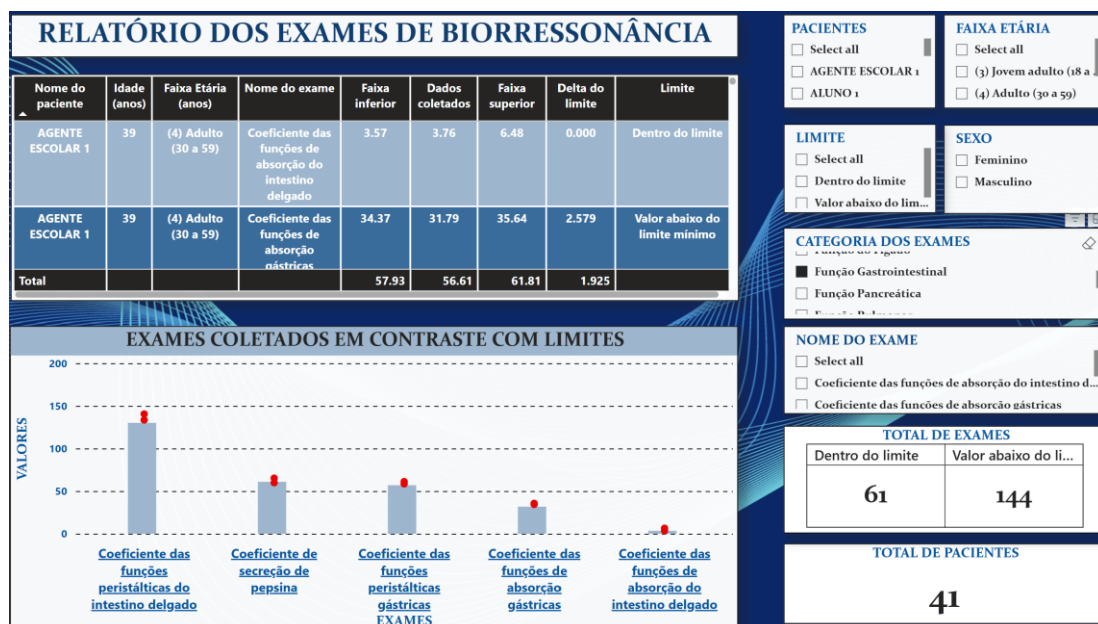


Fonte: Captura de tela de autoria própria.

- **Categoria função gastrointestinal**

A análise dos exames de função gastrointestinal, apresentados na Figura 29, revelou que 60% dos resultados obtidos estão fora do limite saudável. Apenas o coeficiente de secreção da pepsina apresentou desempenho dentro do limite, com 23 de 41 pacientes dentro da faixa esperada.

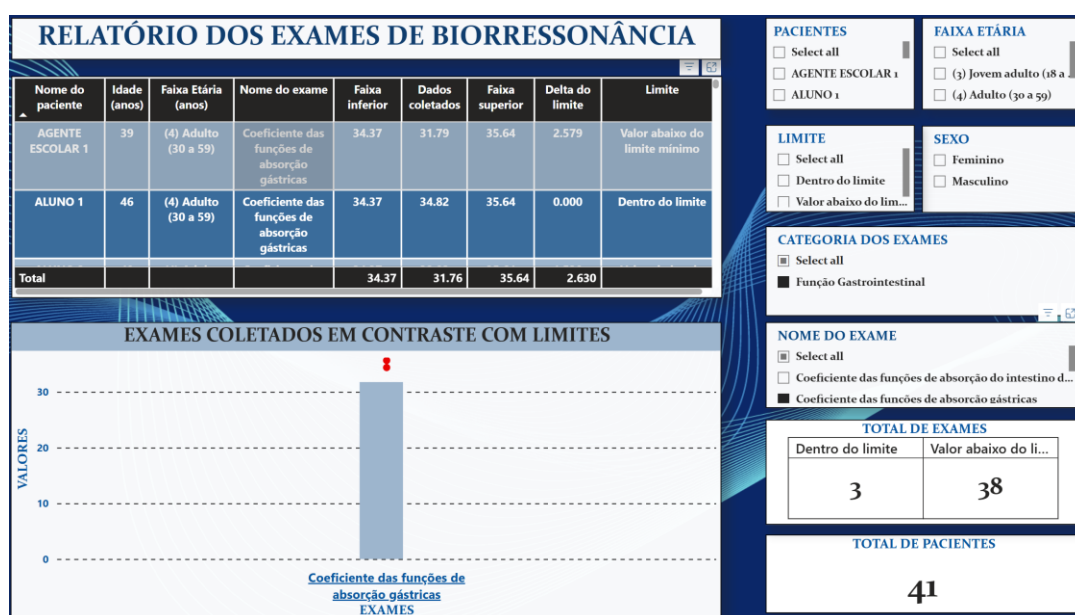
Figura 29 - Painel com dados dos Exames da categoria de Função Gastrointestinal



Fonte: Captura de tela de autoria própria.

Por outro lado, o coeficiente das funções de absorção gástricas, demonstrado na Figura 30, apresentou desempenho muito inferior, com apenas 3 pacientes com resultados normais.

Figura 30 - Painel com dados do Exame “Coeficiente das funções de absorção gástricas”

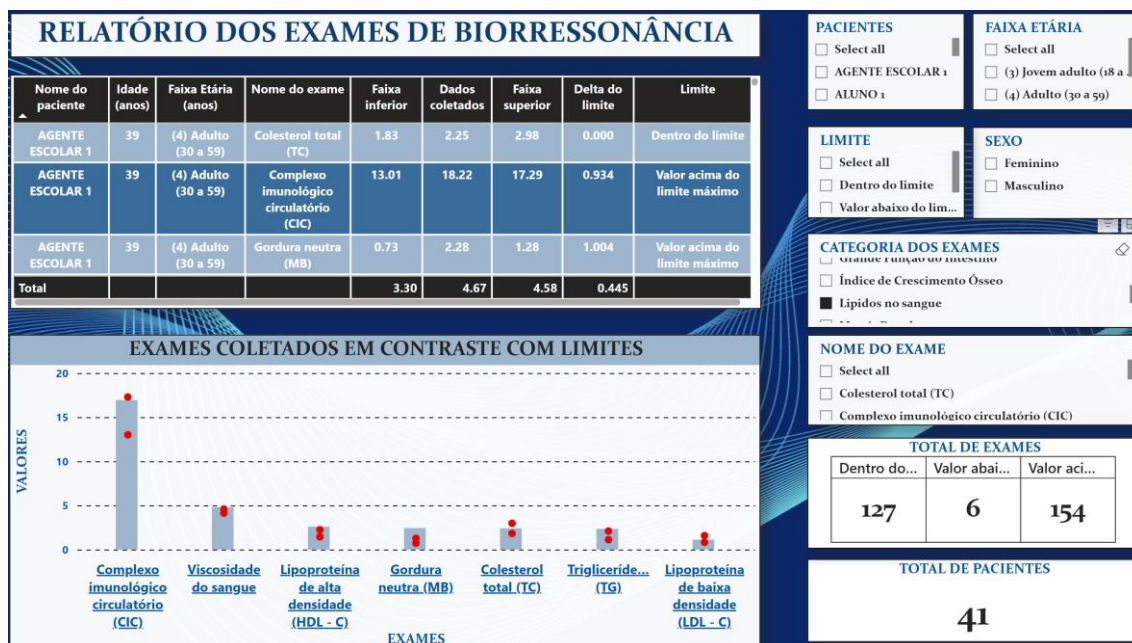


Fonte: Captura de tela de autoria própria.

- **Categoria lipídios no sangue**

Os exames referentes a esta categoria mostrados na Figura 31, possuem resultados surpreendentes e inquietantes, sendo eles:

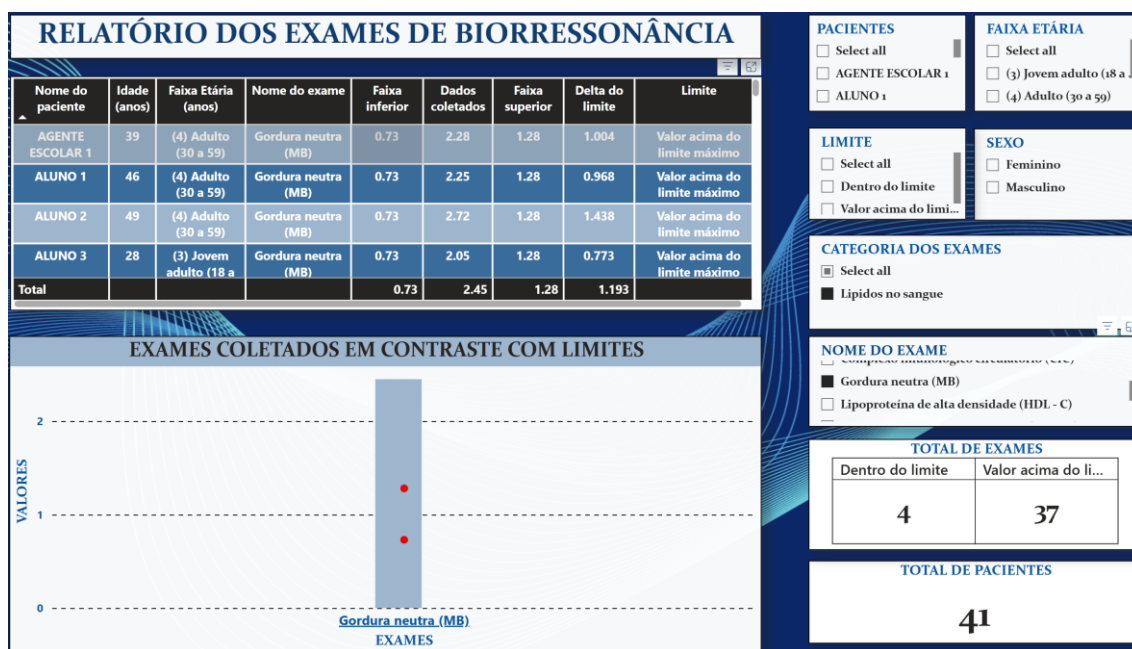
Figura 31 - Painel com dados dos Exames da categoria de Lipídios no sangue



Fonte: Captura de tela de autoria própria.

- Gordura neutra: apenas 4 pacientes estão com níveis dentro do padrão saudável, com a média geral próxima ao dobro do valor recomendado, conforme apresenta a Figura 32.

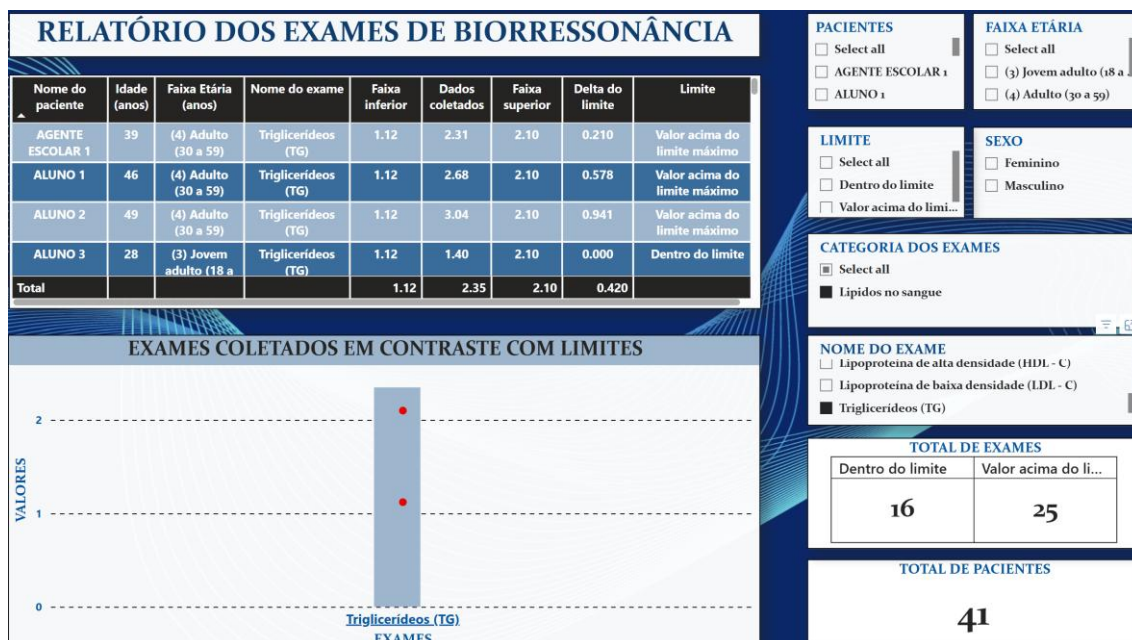
Figura 32 - Painel com dados do Exame “Gordura neutra”



Fonte: Captura de tela de autoria própria.

- Triglicerídeos: 60% (25 de 41 pacientes) apresentaram níveis acima do limite saudável conforme demonstra a Figura 33.

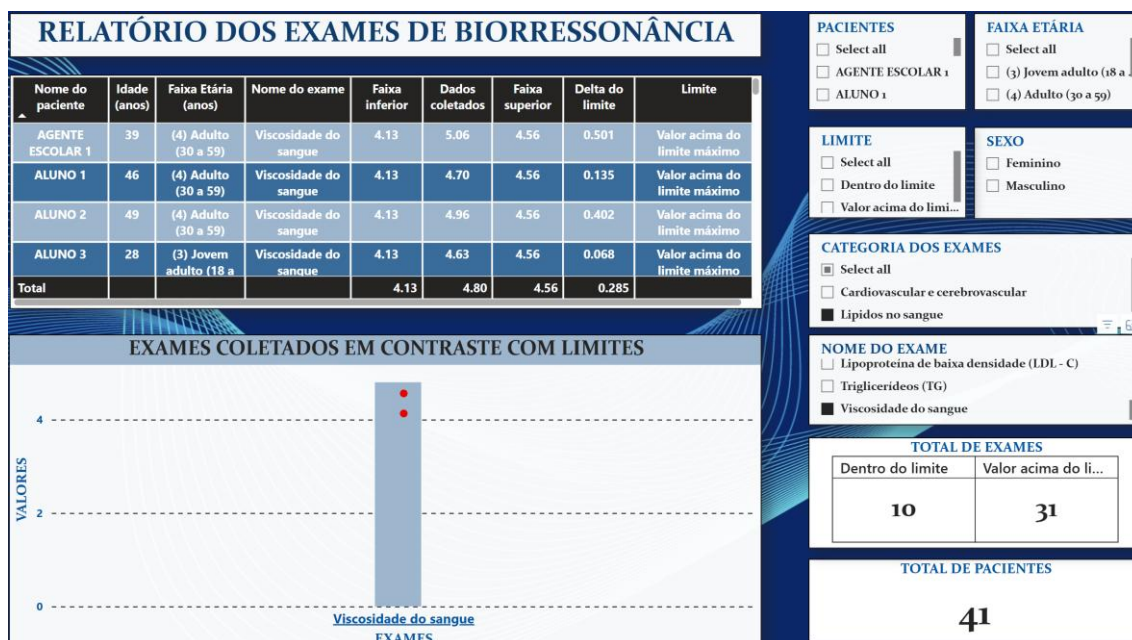
Figura 33 - Painel com dados do Exame “Triglicerídeos”



Fonte: Captura de tela de autoria própria.

- Viscosidade do sangue: somente 25% dos pacientes (10 de 41 pacientes), apresentados na Figura 34, estavam dentro do limite saudável.

Figura 34 - Painel com dados do Exame “Viscosidade do Sangue”



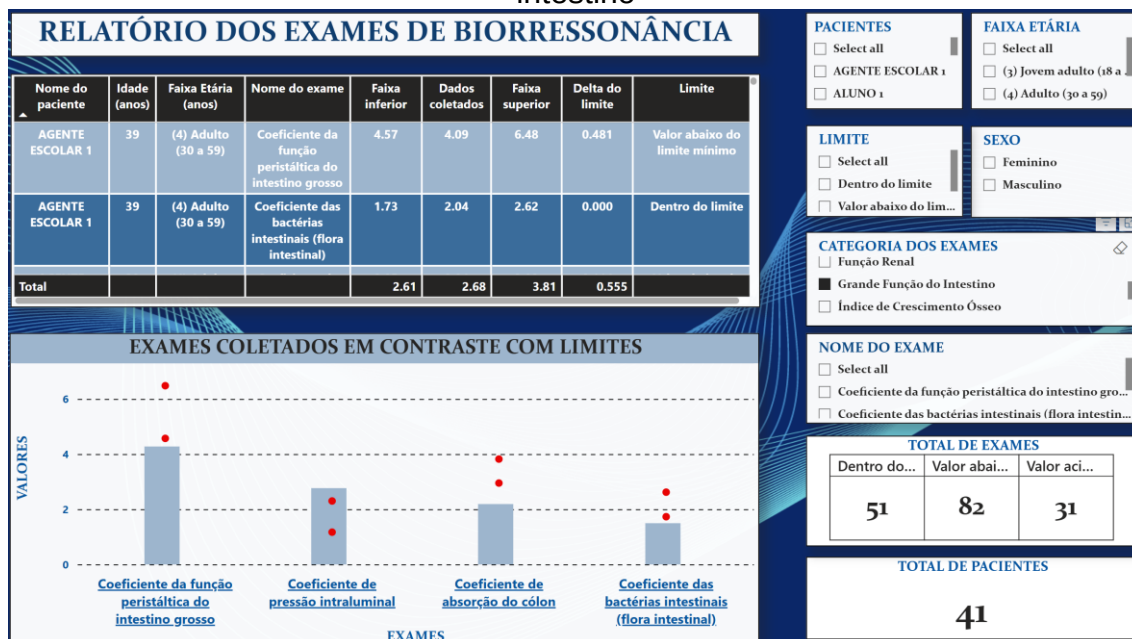
Fonte: Captura de tela de autoria própria

- **Categoria grande função do intestino**

Os resultados desses exames também são preocupantes: nenhum dos exames coletados pelo aparelho de biorressonância apresentou média dentro do limite saudável.

Os resultados da Figura 35 mostram que 69% dos exames estão fora dos parâmetros considerados saudáveis, apontando para possíveis disfunções intestinais.

Figura 35 - Painel com dados dos Exames da categoria da Grande função do intestino



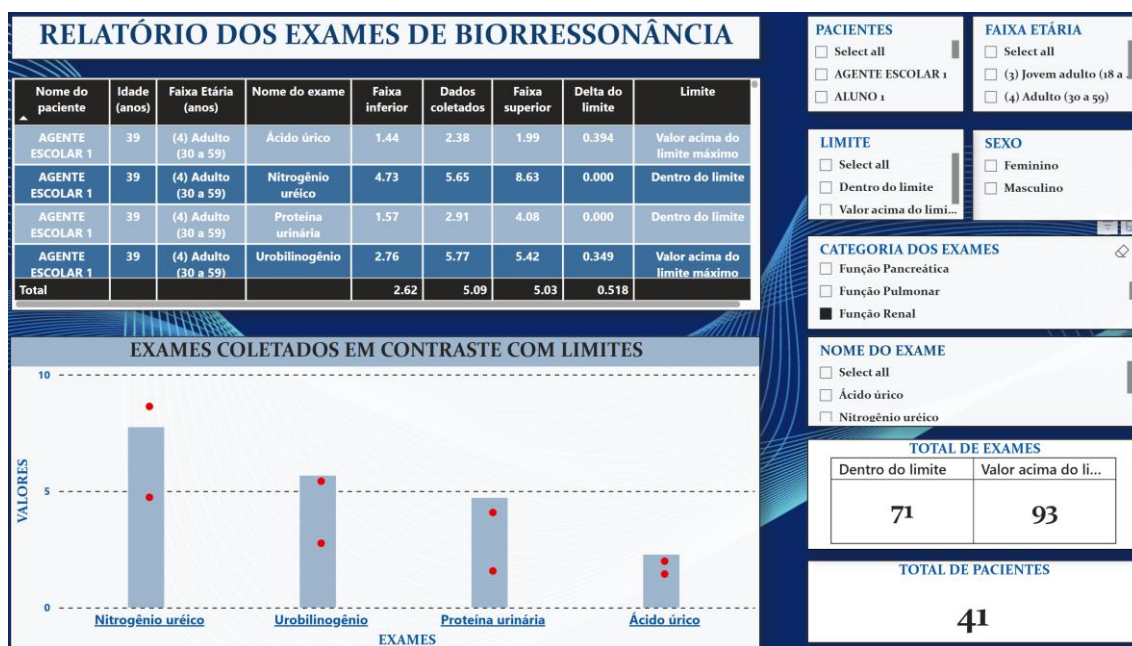
Fonte: Captura de tela de autoria própria.

- **Categoria função renal**

A avaliação dos exames dessa categoria, mostrada na Figura 36, indica que apenas o exame de nitrogênio uréico apresentou média dentro do limite saudável, enquanto os exames de nível de proteína urinária, ácido úrico e urobilinogênio possuem médias acima do limite saudável.

Além disso, 57% dos exames renais estão fora da faixa de normalidade, reforçando a necessidade de atenção para essa função.

Figura 36 - Painel com dados dos Exames da categoria da Função Renal



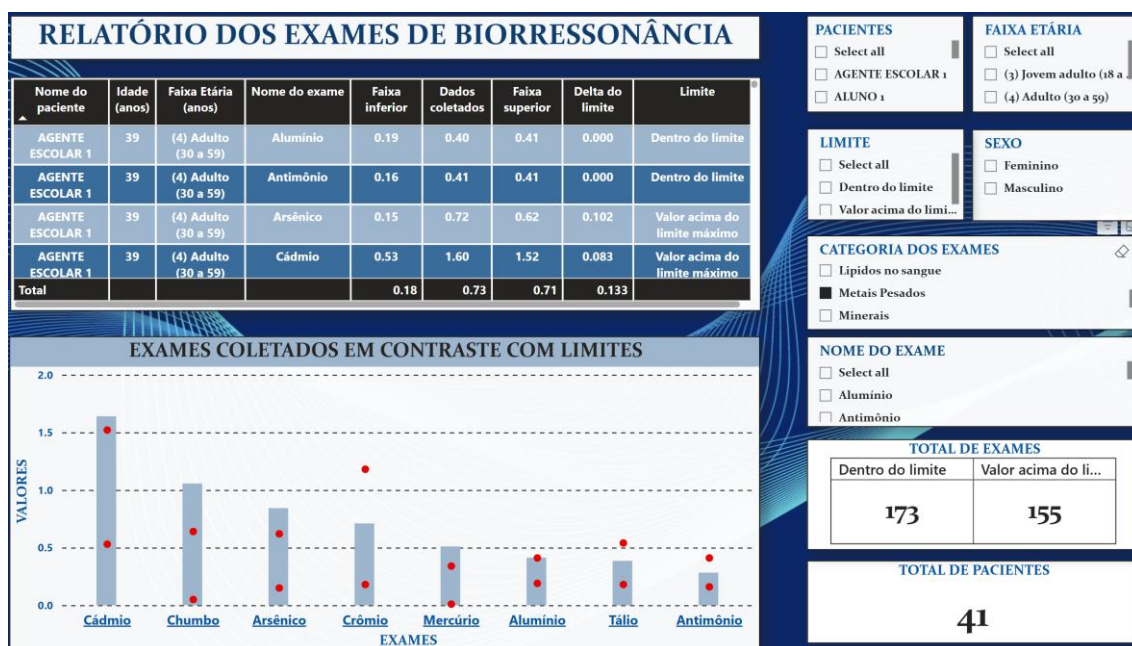
Fonte: Captura de tela de autoria própria.

- **Categoria metais pesados**

A avaliação da intoxicação por metais pesados, apresentada na Figura 37, mostra médias fora dos limites em exames que necessitam de atenção mais pontual. Os elementos mais críticos foram:

- Arsênio, cádmio: média fora do limite da normalidade

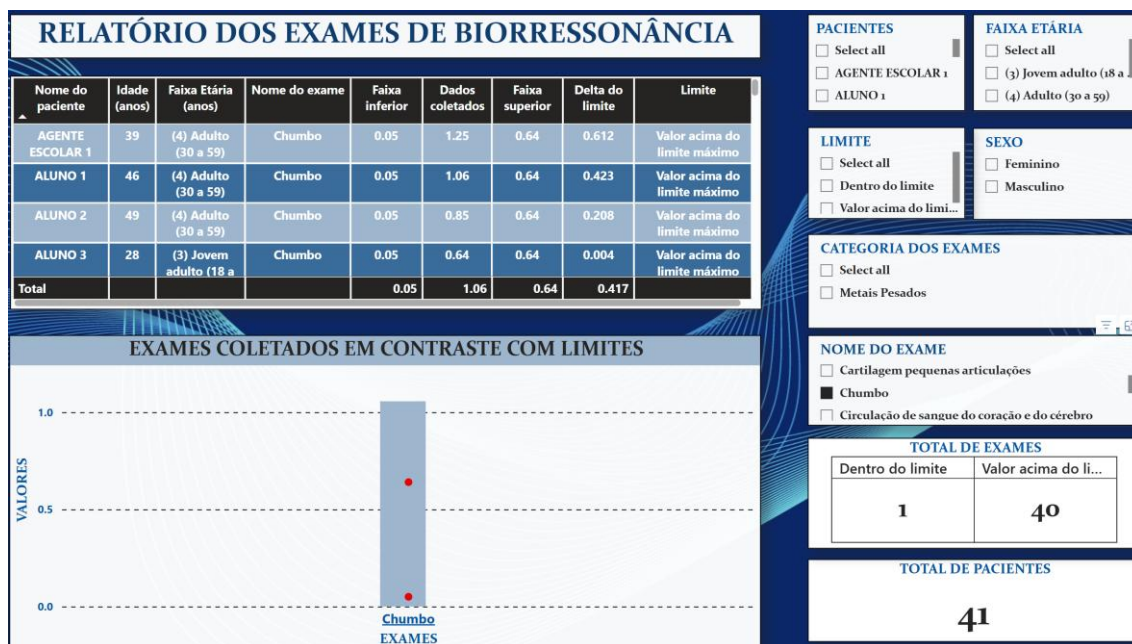
Figura 37 - Painel com dados dos Exames da categoria da metais pesados



Fonte: Captura de tela de autoria própria.

- Chumbo: apenas 1 paciente apresentou níveis dentro da normalidade, conforme Figura 38.

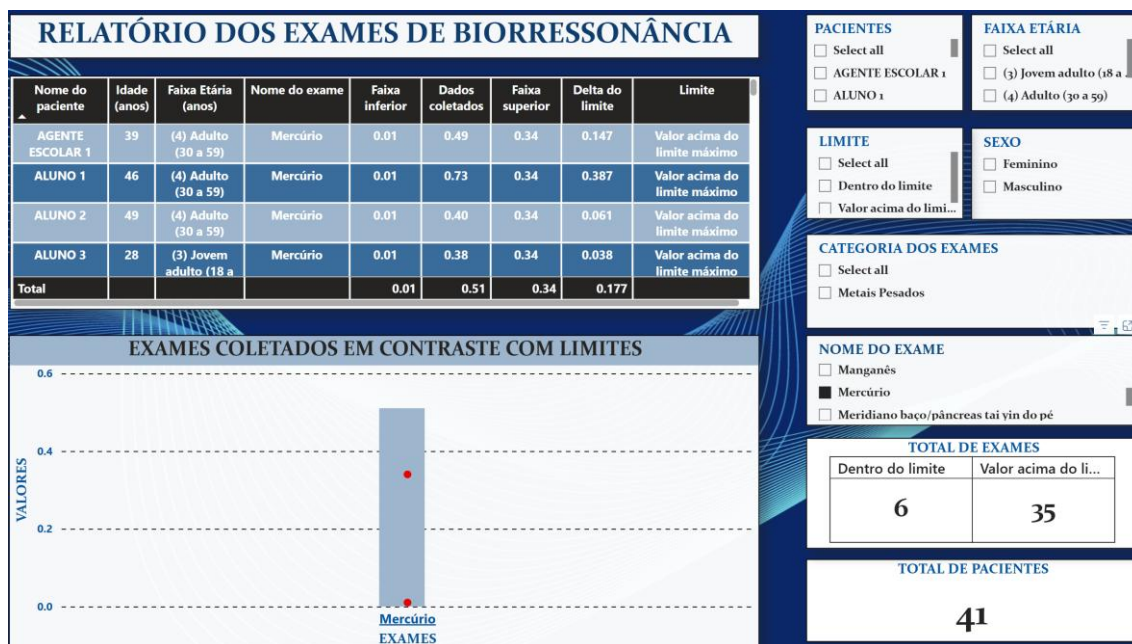
Figura 38 - Painel com dados dos Exame "Chumbo"



Fonte: Captura de tela de autoria própria.

- Mercúrio: 6 pacientes com exames normais, o que ainda representa um número baixo, conforme Figura 39.

Figura 39 - Painel com dados dos Exame “Mercúrio”

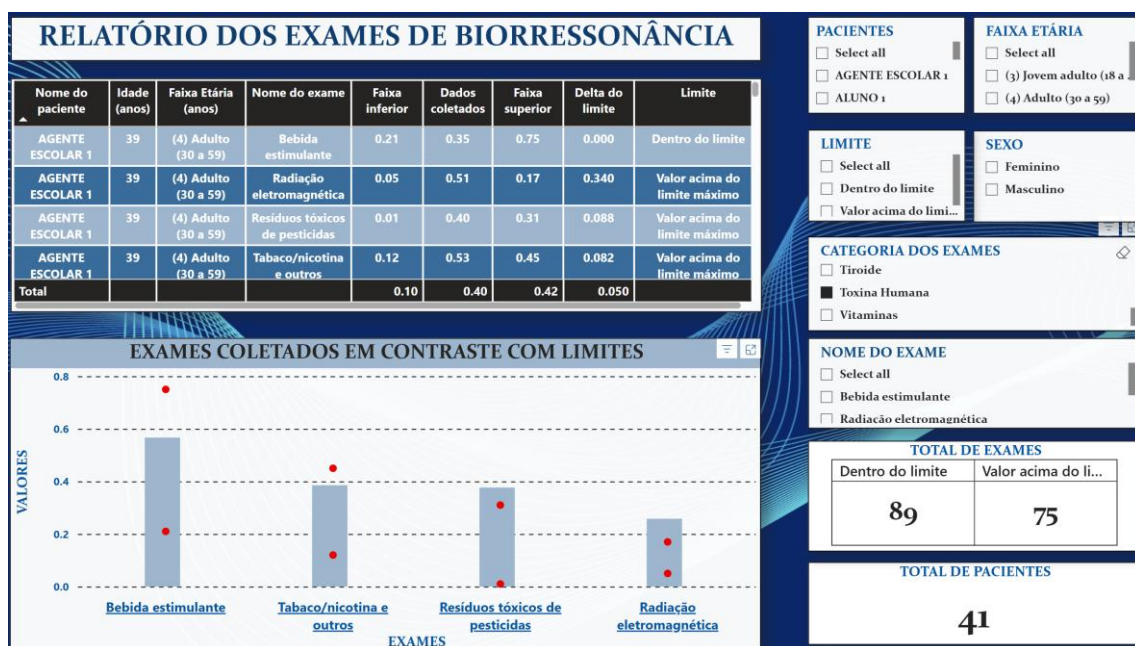


Fonte: Captura de tela de autoria própria

- **Categoria toxinas humanas**

Os exames relacionados a presença de toxinas (Figura 40), indicam médias fora dos limites saudáveis para radiação eletromagnética e resíduos de pesticidas, sugerindo que existe uma possível exposição dos pacientes a agentes nocivos que podem estar associados a hábitos de vida e alimentação contaminada.

Figura 40 - Painel com dados dos Exames da categoria da toxina humana



Fonte: Captura de tela de autoria própria.

5. CONCLUSÃO

A implementação da solução trouxe avanços significativos na análise dos exames gerados pelo equipamento de biorressonância magnética. Antes, a interpretação detalhada dos dados clínicos era limitada pela imensa quantidade de informações - 234 exames por paciente - o que tornava o processo manual extremamente trabalhoso e propenso a erros. Com a nova abordagem, essa etapa tornou-se mais eficiente, confiável e acessível aos usuários do dispositivo.

O uso da linguagem de programação Python aliado ao Power BI demonstrou ser uma solução eficaz para a automatização de processos manuais, especialmente aqueles repetitivos e de alta demanda operacional. Essa combinação permitiu análises segmentadas de populações específicas com mais agilidade e precisão. Além disso, a automação da coleta e organização dos dados eliminou etapas redundantes, reduziu significativamente a necessidade de intervenção humana e otimizou o gerenciamento de grandes volumes de informações, ampliando a capacidade de interpretação dos resultados de forma estruturada e eficiente.

Além disso, a utilização de dashboards personalizados tornou a visualização gráfica dos dados mais eficiente, facilitando a análise dos resultados e promovendo uma interação mais clara e acessível com o paciente. Os gráficos desenvolvidos permitem que o indivíduo compreenda intuitivamente seu estado de saúde com base nos resultados obtidos, destacando de forma clara os pontos que requerem maior atenção ou intervenção. Essa abordagem contribui para um acompanhamento mais eficaz e para uma tomada de decisão mais fundamentada.

Com o sistema desenvolvido, tornou-se possível identificar pontos críticos nos resultados da população analisada. Os dados revelam padrões recorrentes, como sobrepeso, dificuldades na digestão de gorduras e alto consumo de alimentos industrializados com potenciais efeitos tóxicos à saúde. Essas informações não apenas fortalecem a gestão da saúde ocupacional, como também promovem o bem-estar e a qualidade de vida dos pacientes, ao oferecer bases concretas para a adoção de práticas alimentares mais saudáveis e a melhoria dos hábitos de vida.

5.1. Trabalhos futuros

Para trabalhos futuros, recomenda-se o desenvolvimento de mecanismos que possibilitem a coleta e análise de dados históricos dos pacientes. Isso permitirá a realização de exames periódicos, viabilizando comparações ao longo do tempo que possam evidenciar a evolução do estado de saúde dos indivíduos e os efeitos das intervenções implementadas. Dessa forma, será possível monitorar e quantificar os resultados das ações voltadas à melhoria da saúde dos pacientes.

Além disso, recomenda-se a criação de uma estrutura organizada para o armazenamento e a categorização dos exames. Considerando que, nesta etapa, os dados analisados correspondem apenas ao quadro de colaboradores da UNIP, a expansão do projeto para outras populações exigirá cuidados adicionais para evitar a mistura indevida de informações, garantindo a integridade dos dados e a segmentação adequada das análises.

REFERÊNCIAS

ALURA. Introdução ao PyCharm. [S.l.], 2023. Disponível em: <https://www.alura.com.br/apostila-python-orientacao-a-objetos/introducao-ao-pycharm>. Acesso em: 10 jan. 2025.

ASIMOV ACADEMY. Como automatizar Excel com Python usando OpenPyXL. Disponível em: <https://hub.asimov.academy/tutorial/como-automatizar-excel-com-python-usando-openpyxl/>. Acesso em: 10 jan. 2025.

AWARI. Tabula: aprenda Python de forma fácil e rápida. Disponível em: <https://awari.com.br/tabula-aprenda-python-de-forma-facil-e-rapida/>. Acesso em: 10 jan. 2025.

BANSAL, Ankit. Power BI Semantic Models to Enhance Data Analytics and Decision-Making. [S.l.]: IAEME Publications, [s.d.], 2023. Disponível em: https://iaeme.com/MasterAdmin/Journal_uploads/IJM/VOLUME_14_ISSUE_5/IJM_14_05_012.pdf. Acesso em: 16 nov. 2024.

BARBOSA FILHO, Luiz Henrique. Lidando com datas no Python: as bibliotecas time e datetime. Rio de Janeiro: Análise Macro, 2022. Disponível em: <https://analisemacro.com.br/datascience/python/lidando-com-datas-no-python-as-bibliotecas-time-e-datetime/>. Acesso em: 10 jan. 2025.

BBC. Empresas estão ganhando dinheiro com seu DNA. BBC News Brasil, [S.l.], 7 maio 2019. Disponível em: <https://www.bbc.com/portuguese/vert-cap-47926294>. Acesso em: 6 set. 2024.

BHARDWAJ, V. A Systematic Review of Robotic Process Automation in Business Operations: Contemporary Trends and Insights. Journal of Intelligent Systems and Control, v. 2, 2023. Disponível em: https://www.acadlore.com/article/JISC/2023_2_3/jisc020304. Acesso em: 6 set. 2024.

BOTCITY. 6 reasons to consider using Python RPA in your automations. BotCity Blog, 5 maio 2023. Disponível em: <https://blog.botcity.dev/2023/05/05/6-reasons-consider-using-python-rpa-in-your-automations/>. Acesso em: 6 set. 2024.

BRASIL. Ministério do Esporte. Lei Geral de Proteção de Dados (LGPD). Disponível em: <https://www.gov.br/esporte/pt-br/aceso-a-informacao/lgpd>. Acesso em: 9 set. 2024.

CHAUDHURI, S.; DAYAL, U.; NARASAYYA, V. Business Intelligence: A survey. ACM Computing Surveys (CSUR), v. 43, n. 3, p. 1–26, 2011. Disponível em: <https://dl.acm.org/doi/abs/10.1145/1978542.1978562>. Acesso em: 9 set. 2024.

DAVENPORT, T.; RONANKI, R. Artificial Intelligence for the Real World. Harvard Business Review, v. 96, n. 1, p. 108–116, 2018. Disponível em: <https://hbr.org/2018/01/artificial-intelligence-for-the-real-world>. Acesso em: 9 set. 2024.

DIGITAL INNOVATION ONE. Executando tarefas simultâneas em Python com a biblioteca Threading. Disponível em: <https://www.dio.me/articles/executando-tarefas-simultaneas-em-python-com-a-biblioteca-threading>. Acesso em: 10 jan. 2025.

GITHUB. Google API Python Client. Disponível em: <https://github.com/googleapis/google-api-python-client>. Acesso em: 10 jan. 2025.

GUERRA, Kainã Dias; CAMARGO, Hélio Luis; ROSA, Igor Rian. Ferramentas de RPA na automação de processos. 2022. Trabalho de Conclusão de Curso – Faculdade de Tecnologia Deputado Ary Fossen, Jundiaí, 2022. Disponível em: <https://ric.cps.sp.gov.br/handle/123456789/9518>. Acesso em: 6 set. 2024.

HARVE. NumPy Python: o que é, vantagens e tutorial inicial. Disponível em: <https://harve.com.br/blog/programacao-python-blog/numpy-python-o-que-e-vantagens-e-tutorial-inicial/>. Acesso em: 10 jan. 2025.

HARVE. Pandas Python: vantagens e como começar. Disponível em: <https://harve.com.br/blog/programacao-python-blog/pandas-python-vantagens-e-como-comecar/>. Acesso em: 10 jan. 2025.

HASHTAG TREINAMENTOS. Tkinter no Python: como criar interfaces gráficas. Disponível em: <https://www.hashtagtreinamentos.com/tkinter-no-python>. Acesso em: 10 jan. 2025.

HORITA, Angela. Exames de imagem para diagnóstico médico. 2023. Trabalho de Conclusão de Curso – Faculdade de Tecnologia de Bauru, 2023. Disponível em: <https://ric.cps.sp.gov.br/handle/123456789/15358>. Acesso em: 6 set. 2024.

HUMPHRIES, Jeff. O que é Python? [S.l.]: [s.n.], 2020. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=eZmtBAAAQBAJ&oi=fnd&pg=PA5>. Acesso em: 6 set. 2024.

KATIIYAR, Nirvikar et al. Robotic Process Automation (RPA) in Business Operations: Opportunities and Implementation Strategies. Educational Administration: Theory and Practice, v. 30, n. 1, 2024. Disponível em: <https://kuey.net/index.php/kuey/article/view/5987>. Acesso em: 6 set. 2024.

KRISHNAN, Vijay; BHARANIDHARAN, S.; KRISHNAMOORTHY, G. Research Data Analysis with Power BI. [S.l.]: INFLIBNET, [s.d.]. Disponível em: <https://www.scribd.com/document/724694750/24>. Acesso em: 16 nov. 2024.

LI, Y.; YANG, C.; YANG, F. Application of Business Intelligence Based on Big Data in E-commerce Data Evaluation. Heliyon, v. 10, n. 6, e147998, 2024. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2405844024147998>. Acesso em: 16 nov. 2024.

MARTINS, Leandro. O segmento da medicina diagnóstica no Brasil, [S.l.], v. 16, n. 3, 2014. Disponível em: <https://revistas.pucsp.br/RFCMS/article/view/20736>. Acesso em: 6 set. 2024.

MEND. Python Import: mastering the advanced features. Mend.io, 2023. Disponível em: <https://www.mend.io/blog/python-import-mastering-the-advanced-features/>. Acesso em: 10 jan. 2025.

MESQUITA, T. Power BI: 10 razões pelas quais você deve utilizá-lo. 26 dez. 2022. Disponível em: <https://thiagoanalytics.com.br/power-bi-10-razoes-pelas-quais-voce-deve-utiliza-lo/>. Acesso em: 16 nov. 2024.

MICROSOFT. Microsoft named a Leader in Gartner's 2020 Magic Quadrant for Analytics and BI Platforms. Power BI Blog, 13 fev. 2020. Disponível em: <https://powerbi.microsoft.com/pt-br/blog/microsoft-named-a-leader-in-gartners-2020-magic-quadrant-for-analytics-and-bi-platforms/>. Acesso em: 16 nov. 2024.

MICROSOFT. Power BI – Visualize seus dados e compartilhe insights.

Microsoft Power Platform, 16 nov. 2024. Disponível em:

<https://www.microsoft.com/pt-br/power-platform/products/power-bi#Use-cases>.

Acesso em: 16 nov. 2024.

MIRANDA, A. Inovação na gestão de projetos de construção: escalando a pirâmide. LinkedIn, [S.l.], 2021. Disponível em:

<https://www.linkedin.com/pulse/inova%C3%A7%C3%A3o-na-gest%C3%A3o-de-projetos-constru%C3%A7%C3%A3o-escalando-pir%C3%A2mide-/>. Acesso em: 6 set. 2024.

MORAES, C. H. V. de et al. Using machine learning for cognitive Robotic Process Automation (RPA). ResearchGate, 2020. Disponível em:

https://www.researchgate.net/publication/342964390_Using_machine_learning_for_cognitive_Robotic_Process_Automation_RPA. Acesso em: 6 set. 2024.

NEGASH, S. Business Intelligence. Communications of the Association for Information Systems, v. 13, n. 1, p. 177–195, 2004. Disponível em:

<https://aisel.aisnet.org/cais/vol13/iss1/15/>. Acesso em: 9 set. 2024.

PRTI DIGITAL. Automação com Robô (RPA): como surgiu, o que é e no que vai se transformar. PRTI Digital, [S.l.], 15 jul. 2020. Disponível em:

<https://www.prtidigital.com.br/automacao-com-robo-rpa-como-surgiu-o-que-e-e-no-que-vai-se-transformar/>. Acesso em: 6 set. 2024.

PREFEITURA DA CIDADE DO RIO DE JANEIRO. Guia de desenvolvedores: visão geral da infraestrutura. Portal de Dados Abertos do Rio de Janeiro, [s.d.].

Disponível em: <https://docs.dados.rio/guia-desenvolvedores/visao-geral-infra/>. Acesso em: 6 set. 2024.

PRODANOV, C. C.; FREITAS, E. C. Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico. 2. ed. Novo Hamburgo: Feevale, 2013. Disponível em:

https://books.google.com.br/books?hl=p&id=zUDsAQAAQBAJ&printsec=frontcover&utm_source=chatgpt.com#v=onepage&q&f=false. Acesso em: 10 jan. 2025.

PYTHON SOFTWARE FOUNDATION. What is Python? Executive Summary. [S.l.], 2023a. Disponível em: <https://www.python.org/doc/essays/blurb/>. Acesso em: 10 jan. 2025.

RAGHUPATHI, W.; RAGHUPATHI, V. Big data analytics in healthcare: promise and potential. *Health Information Science and Systems*, v. 2, n. 3, p. 1–10, 2014. Disponível em: <https://link.springer.com/article/10.1186/2047-2501-2-3>. Acesso em: 16 nov. 2024.

RAMOS, Vinícius. A biblioteca OS do Python. [S.l.], 2023. Disponível em: <https://pythonacademy.com.br/blog/a-biblioteca-os-do-python>. Acesso em: 10 jan. 2025.

RESTUM, Guilherme. O que é o Power Query e por que é importante no Power BI?. [S.l.], 2023. Disponível em: <https://www.hashtagtreinamentos.com/o-que-e-power-query-powerbi>. Acesso em: 10 jan. 2025.

ROSA, Felipe Pimentel. Ferramentas de Python para RPA: explorando bibliotecas e comandos úteis. Medium, 2023. Disponível em: <https://felipepimentelrosa.medium.com/ferramentas-de-python-para-rpa-explorando-bibliotecas-e-comandos-%C3%BAtteis-48c4bbe1471>. Acesso em: 10 jan. 2025.

SANTOS, F. V.; CASTRO, J. B. Criação de um repositório utilizando o Google Drive. *Revista SAJEBTT*, Universidade Federal do Acre, v. 1, n. 1, 2021. Disponível em: <https://periodicos.ufac.br/index.php/SAJEBTT/article/view/5603>. Acesso em: 9 set. 2024.

SHARDA, Ramesh; DELEN, Dursun; TURBAN, Efraim. *Business Intelligence, Analytics, and Data Science: A Managerial Perspective*. 2017. Disponível em: https://www.researchgate.net/publication/318456888_Business_Intelligence_Analytics_and_Data_Science_A_Managerial_Perspective. Acesso em: 16 nov. 2024.

SILVA, Marcos. Hierarquia DIKW e Capital Humano. [S.l.], 2017. Disponível em: <https://anais.suceg.ufsc.br/index.php/suceg/article/view/20>. Acesso em: 6 set. 2024.

VAZ, Tiago. Parecer técnico COREN-DF nº 09/2020 para realização de biorressonância em consultório privado de enfermagem. [S.l.], 2020. Disponível em: <https://www.coren-df.gov.br/parecer-tecnico-coren-df-no-09-2021/> Acesso em: 16 nov. 2024.

VILLELA, L. C. S. Extraíndo dados de arquivos PDF com Python. Disponível em: <https://lcsvillela.com/extraindo-dados-arquivos-pdf.html>. Acesso em: 10 jan. 2025.

APÊNDICE A – DOWNLOAD e UPLOAD

```
# Configurando as credenciais
client_id = "877316879847-6js0cko1fo6r4r7dbrmjmqmhrv8bj7s.apps.googleusercontent.com"
client_secret = "G0CSPX-0ISYye0ceBrpLCkbFKIj4FaDt_Nw"
#Variavel associada ao link do google drive, campo para o download da lista vazia
FileId = '1-cZ20DzAu4xBcN9p6e8X8vcbFnmVTRqqlNb3537dmWs'
#Variavel associada ao link da pasta para salvar o documento novo
TargetfolderId = '14JhnJm-kfXm-YXa80p-0I5oR_BQemw6R'
#Variavel associada ao link do google drive, campo para o upload da lista completa
TargetFileId = '1w9QQXrRc7Ftu1rSo_bXUvnnS99ooQW63SbYEwTVdipY'

# Definição do escopo da API do google drive
SCOPES = ['https://www.googleapis.com/auth/drive.readonly']
SCOPES_UPLOAD = ['https://www.googleapis.com/auth/drive']

#Função para criar fluxo de autenticação OAuth 2.0 e acessar a API do google, gerando acesso para interagir com o
#google drive
def generate_tokens(client_id, client_secret):
    flow = InstalledAppFlow.from_client_config(
        client_config={
            "web": {
                "client_id": client_id,
                "client_secret": client_secret,
                "redirect_uris": ["urn:ietf:wg:oauth:2.0:oob"],
                "auth_uri": "https://accounts.google.com/o/oauth2/auth",
                "token_uri": "https://accounts.google.com/o/oauth2/token",
            }
        },
        scopes=SCOPES
    )
    # Cria e inicia a sequencia de cliques (thread) ao abrir o navegador
    browser_thread = threading.Thread(target=open_browser, args=(flow,))
    browser_thread.start()

    # Executa o servidor local
    flow.run_local_server(port=0)
    return flow.credentials

#função de verificação de autenticação das credenciais de acesso, caso elas forem expiradas, renova utilizando o token
# de atualização.
def authenticate_with_token(credentials):
    if not credentials.valid:
        if credentials.expired and credentials.refresh_token:
            credentials.refresh(Request())
    return credentials
```

```

#função para gerar Tokens de upload dos documentos no Google Drive
def generate_tokens_upload(client_id, client_secret):
    flow = InstalledAppFlow.from_client_config(
        client_config={
            "web": {
                "client_id": client_id,
                "client_secret": client_secret,
                "redirect_uris": ["urn:ietf:wg:oauth:2.0:oob"],
                "auth_uri": "https://accounts.google.com/o/oauth2/auth",
                "token_uri": "https://accounts.google.com/o/oauth2/token",
            }
        },
        scopes=SCOPES_UPLOAD
    )
    # Inicia a sequencia de cliques (thread) para abrir o navegador
    browser_thread = threading.Thread(target=open_browser, args=(flow,))
    browser_thread.start()
    # Executa o servidor local
    flow.run_local_server(port=0)
    return flow.credentials.token, flow.credentials.refresh_token

# Função para autenticar o Usuario com o Token da API
def authenticate_with_token_upload(token):
    creds = Credentials.from_authorized_user_info(token)
    if not creds.valid:
        if creds.expired and creds.refresh_token:
            creds.refresh(Request())
        return creds

```



```

#função para automatizar cliques em paralelo (Threading)
def open_browser(flow):
    # Definição da quantidade de tempo do intervalo 1
    intervalo = 3
    # Definição da quantidade de tempo do intervalo 2
    intervalo2 = 1
    print("Iniciando sequência de teclas...")

    # Definição da quantidade de tempo para garantir que o navegador foi aberto
    time.sleep(3)

    # Localiza a janela do navegador que foi aberta pelo comando flow.run_local_server
    janela = None
    for window in gw.getWindowsWithTitle("Google OAuth 2.0"):
        janela = window
        # Seleciona a primeira janela encontrada com o título "Google OAuth 2.0"
        break

    if janela:
        # Coloca a janela do navegador no foco
        janela.activate()

    # Sequência de teclas para primeira página
    time.sleep(intervalo)
    pyautogui.press('tab')
    time.sleep(intervalo2)
    pyautogui.press('tab')
    time.sleep(intervalo2)
    pyautogui.press('enter')
    time.sleep(intervalo)

    # Sequência de teclas para segunda página
    pyautogui.press('tab')
    time.sleep(intervalo2)
    pyautogui.press('tab')
    time.sleep(intervalo2)
    pyautogui.press('enter')
    time.sleep(intervalo2)
    pyautogui.press('tab')
    time.sleep(intervalo2)
    pyautogui.press('tab')
    time.sleep(intervalo2)
    pyautogui.press('tab')
    time.sleep(intervalo2)
    pyautogui.press('enter')
    time.sleep(intervalo)

    # Sequência de teclas para terceira página
    pyautogui.press('tab')
    time.sleep(intervalo2)
    pyautogui.press('tab')
    time.sleep(intervalo2)
    pyautogui.press('tab')
    time.sleep(intervalo2)
    pyautogui.press('tab')
    time.sleep(intervalo2)
    pyautogui.press('tab')
    time.sleep(intervalo2)
    pyautogui.press('tab')
    time.sleep(intervalo2)
    pyautogui.press('tab')
    time.sleep(intervalo2)
    pyautogui.press('tab')
    time.sleep(intervalo2)
    pyautogui.press('enter')
    time.sleep(intervalo)

    # Após o último 'enter', fecha a aba do navegador
    pyautogui.hotkey('ctrl', 'w')
    print("threading funcionou")

```

APÊNDICE B – FUNCIONAMENTO – PROGRAMA PRINCIPAL

```
# Criação de um marcador de paginas para a leitura do arquivo pdf
paginas = [1, 7, 10, 12, 14, 16, 18, 20, 22, 27, 30, 32, 34, 36, 44, 47, 51, 53, 55, 58, 61, 64, 67, 70, 72, 76, 79, 83, 86, 91, 94, 96]
paginas2 = [1, 7, 10, 12, 14, 16, 20, 22, 27, 30, 32, 34, 55, 58, 61, 64, 70, 72, 76, 79, 83, 86, 91, 94, 96]

# Nomeando colunas que serão preenchidas
colunas_preenchimento = ['Nome', 'Sexo', 'Idade', 'Valores', 'Nome do Exame', 'Categoria', 'Faixa inferior', 'Faixa superior']

if __name__ == "__main__":
    # Cria um arquivo excel temporario quando é iniciado o programa
    temp_excel_path = "temp_downloaded_file.xlsx"
    DestinationFilepath = select_save_location()

    # checa se o botão 'cancel' foi selecionado e para o programa em caso positivo
    if DestinationFilepath is None:
        print("No Excel file location selected. Exiting.")
        messagebox.showinfo(title="Encerrado", message="Nenhum local de arquivo foi selecionado.")
        exit()

    # Continua o programa
    else:
        credentials = generate_tokens(client_id, client_secret)
        download_google_sheets_as_excel(FileId, credentials, temp_excel_path)
        convert_excel_to_xlsx(temp_excel_path, DestinationFilepath)
        os.remove(temp_excel_path)

# Chama a função para selecionar a pasta contendo os arquivos PDF
caminho_pasta = selecionar_pasta()

# Se o usuário clicar em "Cancelar", encerrará o programa
if caminho_pasta is None:
    print("Nenhuma pasta selecionada. O programa será encerrado.")
    messagebox.showinfo(title="Encerrado", message="Nenhuma pasta foi selecionada. O programa será encerrado.")
    exit()

# Chama a função para selecionar o arquivo Excel
caminho_xlsx = DestinationFilepath

# Verifica se a pasta e o arquivo Excel foram selecionados
if caminho_pasta and caminho_xlsx:
    processar_todos_pdfs(caminho_pasta, caminho_xlsx)

# Se o usuário não selecionar, encerra o programa
else:
    print("Nenhuma pasta ou arquivo Excel selecionado.")

# inicia a função principal do programa
if __name__ == "__main__":
    select_file()
```

APÊNDICE C – FUNÇÕES DO PROGRAMA

```
# função para definir o local de salvamento do documento baixado
def select_save_location():
    # decisão para continuar ou encerrar o programa
    if not iniciar_ou_encerrar():
        return None

    # criação de uma janela para decisão do local de salvamento do documento
    root = tk.Tk()
    root.withdraw()
    root.lift()
    root.attributes('-topmost', True)
    root.after_idle(root.attributes, *args: '-topmost', True)

    file_path = filedialog.asksaveasfilename(
        defaultextension=".xlsx",
        filetypes=[("Excel files", "*.xlsx"), ("All files", "*.*")],
        title="Selecione a localização para salvar o documento em Excel"
    )

    # checa se o botão 'cancel' foi selecionado e para o programa em caso positivo
    if file_path == "":
        print("No file selected. Exiting.")
        root.quit()
        return None

    # Checa se o arquivo existe, se não existe, gera um arquivo novo
    if file_path:
        if not os.path.exists(file_path):
            wb = Workbook()
            wb.save(file_path)
            print(f"Novo arquivo Excel criado em: {file_path}")

    return file_path
```

```
# função para abrir uma interface para o usuário escolher o local de download
def iniciar_ou_encerrar():
    root = tk.Tk()
    root.withdraw()
    root.lift()
    root.attributes('-topmost', True)
    root.after_idle(root.attributes, *args: '-topmost', True)

    # Criação de uma janela com as opções "Iniciar Programa" e "Encerrar Programa"
    resposta = tk.messagebox.askquestion(
        title="Leitura do exame dos pacientes", message="Deseja iniciar o programa?",
        icon='question'
    )

    # Se a resposta for 'Não', fecha o loop e finaliza o programa
    if resposta == 'no':
        print("Programa encerrado.")
        root.quit()
        return False

    # Se a resposta for 'sim', continua o programa
    return True
```

```
# Função para abrir a tela de seleção de pasta e retornar o caminho
def selecionar_pasta():
    root = tk.Tk()
    root.withdraw()
    caminho_pasta = filedialog.askdirectory(
        title="Selecione a pasta com arquivos PDF"
    )

    # Se o usuário clicar em "Cancelar", encerrará o programa
    if caminho_pasta == "":
        return None

    return caminho_pasta
```

```
# Função principal para processar todos os arquivos PDF de uma pasta
def processar_todos_pdfs(caminho_pasta, caminho_xlsx):
    df_xlsx_vazio = carregar_xlsx_vazio(caminho_xlsx)
    # Cria um DataFrame vazio para armazenar os dados concatenados
    df_final = pd.DataFrame(columns=df_xlsx_vazio.columns)
    # Itera sobre todos os arquivos na pasta selecionada
    for arquivo in os.listdir(caminho_pasta):
        if arquivo.endswith(".pdf"):
            caminho_pdf = os.path.join(caminho_pasta, arquivo)
            print(f"Processando PDF: {caminho_pdf}")
            # Processa o PDF e adiciona os dados ao DataFrame final
            df_xlsx_vazio_atualizado = processar_pdf(caminho_pdf, caminho_xlsx, df_xlsx_vazio)
            df_final = pd.concat([df_final.dropna(axis=1, how='all'), df_xlsx_vazio_atualizado.dropna
                                (axis=1, how='all')], ignore_index=True)
    # Salva o DataFrame final como arquivo Excel
    df_final.to_excel(caminho_xlsx, index=False, engine='openpyxl')
    print(f"Excel final salvo em: {caminho_xlsx}")
```

```
# Função para analisar o arquivo PDF selecionado
def processar_pdf(caminho_pdf, caminho_xlsx, df_xlsx_vazio):
    # Checa se as colunas esperadas existem no DataFrame (com minúsculas e sem espaços extras)
    if not all(col in df_xlsx_vazio.columns for col in ['nome', 'sexo', 'valores']):
        print("As colunas esperadas não estão presentes no Excel.")
        print(f"Colunas disponíveis no Excel: {df_xlsx_vazio.columns}")
        return df_xlsx_vazio
    try:
        nome_paciente = os.path.basename(caminho_pdf).replace(_old: ".pdf", _new: "")
        print(f"Processando arquivo: {caminho_pdf}")
        # Criando um DataFrame para salvar o nome do paciente
        info_paciente = [nome_paciente] * len(df_xlsx_vazio)
        df_xlsx_vazio['nome'] = info_paciente
        dataframes = {}
        # Processa o PDF para preencher as colunas "Sexo" e "Valores"
        for i in paginas:
            if i == 1:
                search_words = ['Sexo', 'Idade']
                result = extract_data(caminho_pdf, search_words)
                if result:
                    # Preenche a coluna "Sexo" e "Idade" para todas as linhas
                    if 'Sexo' in result:
                        df_xlsx_vazio['sexo'] = result['Sexo']
                    if 'Idade' in result:
                        df_xlsx_vazio['idade'] = result['Idade']
                lista_tabelas = tabula.read_pdf(caminho_pdf, pages=i)
                if lista_tabelas:
                    primeira_tabela = lista_tabelas[0]
                    # Criação de lógica para concatenar todos os DataFrames
                    nome_df = f"df_{i}"
                    df = pd.DataFrame(primeira_tabela)
                    # Padroniza o título dos eixos da tabela
                    df = df.set_axis(['item de teste', 'Faixa normal', 'valores', 'Resultado'], axis=1)
```

```

# Transforma todas as strings da tabela em valores nulos
if i in paginas2:
    linha_para_nan = 1
    df.loc[linha_para_nan] = np.nan
# Deleta as colunas que não serão usadas
df = df.drop(df.columns[[0, 1, 3]], axis=1, errors="ignore")
# Retira todos os valores nulos da tabela
df_vazio = df.dropna(how='all')
# Adiciona o DataFrame ao dicionário
dataframes[nome_df] = df_vazio

# Concatena o DataFrame
df_concatenado = pd.concat(dataframes.values(), ignore_index=True)
# Preenche a coluna "valores" com os dados de "valores" extraídos do PDF
if 'valores' in df_concatenado.columns:
    df_xlsx_vazio['valores'] = df_concatenado['valores'].head(len(df_xlsx_vazio)).values
# Retorna o DataFrame atualizado
return df_xlsx_vazio
except Exception as e:
    print(f"Ocorreu um erro ao ler o arquivo PDF {caminho_pdf}: {e}")
return df_xlsx_vazio

```

```

# função para extrair os caracteres que representam o sexo e a idade dos pacientes do PDF
def extract_data(pdf_path, search_words):
    document = fitz.open(pdf_path)
    page = document.load_page(0)
    # Obtém o texto da página
    text = page.get_text()
    # Dicionário para armazenar os resultados
    extracted_data = {}

    # Procura as palavras escolhidas na primeira página e salva esses dados (Sexo, Idade)
    for word in search_words:
        if word in text:
            word_index = text.find(word)
            # Calcula o início da extração após a palavra-chave
            start_index = word_index + len(word)
            # Extrai os próximos 9 caracteres relacionado ao sexo
            subsequent_text = text[start_index:start_index + 11]
            # Armazena o valor extraído
            extracted_data[word] = subsequent_text.strip()

    # Fecha o documento
    document.close()
    return extracted_data

```

APÊNDICE D – EXAMES

Categoria dos exames	Exame
Ácido graxo	Acido araquidonico
Ácido graxo	Acido linoleico
Ácido graxo	Alfa acido linoleico
Ácido graxo	Gama acido linoleito
Açúcar no sangue	Coeficiente de açúcar na urina
Açúcar no sangue	Coeficiente de açúcar na sangue
Açúcar no sangue	Coeficiente de secreção de insulina
Alérgenos	Alergia a corante de tintas de cabelo
Alérgenos	Alergia a acessórios de metal
Alérgenos	Índice alergia marisco
Alérgenos	Alergia a tintas e vernizes
Alérgenos	Índice alergia proteína do leite
Alérgenos	Índice de alergia a fumos
Alérgenos	Índice de alergia a poeira
Alérgenos	Fibra química
Alérgenos	Índice alergia de contato
Alérgenos	Índice de alergia a antibióticos
Alérgenos	Índice de alergia a medicamentos
Alérgenos	Índice de alergia a álcool
Alérgenos	Índice de alergia ao pólen
Aminoácidos	Triptofano
Aminoácidos	Valina
Aminoácidos	Isoleucina
Aminoácidos	Leucina
Aminoácidos	Lisina

Aminoácidos	Treonina
Aminoácidos	Arginina
Aminoácidos	Fenilalanina
Aminoácidos	Histidina
Aminoácidos	Metionina
Avaliação física básica	Falta de água
Avaliação física básica	Capacidade cerebral
Avaliação física básica	Capacidade de reação física
Avaliação física básica	Ph
Avaliação física básica	Hipoxia
Cardiovascular e cerebrovascular	Consumo de oxigênio do miocárdio
Cardiovascular e cerebrovascular	Demanda de sangue do miocárdio
Cardiovascular e cerebrovascular	Elasticidade vascular
Cardiovascular e cerebrovascular	Gordura no sangue
Cardiovascular e cerebrovascular	Resistência vascular
Cardiovascular e cerebrovascular	Volume de perfusão do sangue do miocárdio
Cardiovascular e cerebrovascular	Volume sistólico
Cardiovascular e cerebrovascular	Cristais de colesterol
Cardiovascular e cerebrovascular	Elasticidade da artéria coronária
Cardiovascular e cerebrovascular	Elasticidade dos vasos sanguíneos do cérebro
Cardiovascular e cerebrovascular	Força de bombeamento efetiva do ventrículo esquerdo
Cardiovascular e cerebrovascular	Impedância do bombeamento de sangue do ventrículo esquerdo
Cardiovascular e cerebrovascular	Pressão de perfusão da artéria coronária
Cardiovascular e cerebrovascular	Situação do fornecimento de sangue ao tecido cerebral

Cardiovascular e cerebrovascular	Viscosidade do sangue
Coenzima	Acido fólico
Coenzima	Acido pantotenico
Coenzima	Biotina
Coenzima	Coenzima Q10
Coenzima	Nicotinamida
Coenzima	Glutaciona
Colágeno	Articulação
Colágeno	Cabelo e pele
Colágeno	Circulação de sangue do coração e do cérebro
Colágeno	Dentes
Colágeno	Desintoxicação e metabolismo
Colágeno	Metabolismo da gordura
Colágeno	Sistema endocrino
Colágeno	Sistema nervoso
Colágeno	Sistema reprodutivo
Colágeno	Esqueleto
Colágeno	Estomago e intervalo intestinal
Colágeno	Olhos
Colágeno	Sistema imunológico
Colágeno	Tecido muscular
Densidade mineral óssea	Densidade óssea
Densidade mineral óssea	Grau de hiperplasia óssea
Densidade mineral óssea	Grau de osteoporose
Densidade mineral óssea	Perda de cálcio
Densidade mineral óssea	Coeficiente de osteoclasto

Função da vesícula biliar	Acido biliar do soro sanguineo
Função da vesícula biliar	Bilirrubina
Função da vesícula biliar	Bilirrubina total
Função da vesícula biliar	Fosfatase Alcalina (ALP)
Função da vesícula biliar	Globulina do soro sanguineo
Função do fígado	Teor de gordura no figado
Função do fígado	Função de desintoxicação
Função do fígado	Função de produção de energia
Função do fígado	Função de secreção de bilis
Função do fígado	Metabolismo de proteínas
Função gastrointestinal	Coeficiente de secreção de pepsina
Função gastrointestinal	Coeficiente das funções de absorção do intestino delgado
Função gastrointestinal	Coeficiente das funções de absorção gastrica
Função gastrointestinal	Coeficiente das funções peristalticos do intestino delgado
Função gastrointestinal	Coeficiente das funções peristalticas gastricas
Função pancreática	Glucagon
Função pancreática	Insulina
Função pancreática	Polipectideo pancreatico
Função pulmonar	Atividade pulmonar VC
Função pulmonar	Capacidade pulmonar total TLC
Função pulmonar	Resistencia das vias aereas RAM
Função pulmonar	Teor de oxigenio no sangue arterial PaCO2
Função renal	Acido urico
Função renal	Proteina urinaria

Função renal	Urobilinogeneo
Função renal	Nitrogenio ureico
Grande função do intestino	Coeficiente da função peristaltica do intestino grosso
Grande função do intestino	Coeficiente das bacterias intestinais
Grande função do intestino	Coeficiente de absorção do cólon
Grande função do intestino	Coeficiente de pressão intraluminal
Índice de crescimento ósseo	Cartilagem de pequenas articulações
Índice de crescimento ósseo	Cartilagem de grandes articulações
Índice de crescimento ósseo	Fosfatase alcalina ossea
Índice de crescimento ósseo	Osteocalcina
Índice de crescimento ósseo	Linha epifisária
Lipídeos no sangue	Lipoproteína de alta densidade
Lipídeos no sangue	Triglicerídeos (TG)
Lipídeos no sangue	Viscosidade gordurosa do sangue
Lipídeos no sangue	Lipoproteína de baixa densidade
Lipídeos no sangue	Colesterol total (TC)
Lipídeos no sangue	Complexo imunológico circulatório (CIC)
Lipídeos no sangue	Gordura neutra (MB)
Metais pesados	Alumínio
Metais pesados	Antimônio
Metais pesados	Cromio
Metais pesados	Arsênico
Metais pesados	Cádmio
Metais pesados	Chumbos
Metais pesados	Mercurio
Metais pesados	Tálio

Minerais	Boro
Minerais	Calcio
Minerais	Cobalto
Minerais	Cobre
Minerais	Estanho
Minerais	Estroncio
Minerais	Fluor
Minerais	Fósforo
Minerais	Ferro
Minerais	Potassio
Minerais	Zinco
Minerais	Iodo
Minerais	Magnesio
Minerais	Manganes
Minerais	Molibdenio
Minerais	Niquel
Minerais	Selenio
Minerais	Silicio
Minerais	Vanadio
Obesidade	Coeficiente de anomalia hipotalamo nucleo
Obesidade	Coeficiente de hiperinsulina
Obesidade	Anormalidade tecido adiposo
Obesidade	Coeficiente de conteudo anormal de triglicerideos
Obesidade	Coeficiente de metabolismo anormal de lipidios
Olhos	Afrouxamento e queda

Olhos	Atividades das células dos olhos
Olhos	Bolsas sob os olhos
Olhos	Colágeno das rugas nos olhos
Olhos	Edema
Olhos	Fadiga visual
Olhos	Obstrução linfática
Olhos	Índice de olheiras (pigmentação)
Ossos e fator reumatóide	Coefficiente de osteoporose
Ossos e fator reumatóide	Calcificação coluna lombar
Ossos e fator reumatóide	Calcificação coluna cervical
Ossos e fator reumatóide	Coefficiente de hiperplasia óssea
Ossos e fator reumatóide	Coefficiente de reumatismo
Pele	Índice de colágeno na pele
Pele	Índice de dilatação dos vasos sanguíneos da pele
Pele	Índice de elasticidade da pele
Pele	Índice de hidratação da pele
Pele	Índice de imunidade da pele
Pele	Índice de melanina na pele
Pele	Índice de oleosidade na pele
Pele	Índice de queratínicos da pele
Pele	Índice dos radicais livres da pele
Pele	Perda de hidratação da pele
Pulso do coração e cérebro	Coefficiente da onda de pulso k
Pulso do coração e cérebro	Pressão do oxigênio do sangue cerebrovascular
Pulso do coração e cérebro	Pulso (SV)

Pulso do coração e cérebro	Saturação do oxigenio do sangue cerebrovascular
Pulso do coração e cérebro	Indice de acidente vascular cerebral
Pulso do coração e cérebro	Resistencia periferica do coração
Pulso do coração e cérebro	Volume do oxigenio do sangue cerebrovascular
Sistema endócrino	Indice de secreção da glandula pineal
Sistema endócrino	Indice de secreção da paratireoide
Sistema endócrino	Indice de secreção da pituitária
Sistema endócrino	Indice de secreção da tireoide
Sistema endócrino	Indice de secreção da glandula suprarenal
Sistema endócrino	Indice de secreção do timo
Sistema endócrino	Indice gonodal
Sistema imunológico	Indice de imunidade das amígdalas
Sistema imunológico	Indice de imunidade gastrointestinal
Sistema imunológico	Indice de imunoglobulina
Sistema imunológico	Indice de linfonodolo
Sistema imunológico	Indice do baço
Sistema imunológico	Indice do timo
Sistema imunológico	Indice da medula ossea
Sistema imunológico	Indice de imunidade mucosa
Sistema imunológico	Indice de imunidade do trato respiratório
Sistema nervoso	Arterioesclerose cerebral
Sistema nervoso	Condição das função neurológicas
Sistema nervoso	Fornecimento de sangue ao cérebro
Sistema nervoso	Indicador de depressão
Sistema nervoso	Indicador de memoria

Tireoide	Anticorpos antitireoglobulina
Tireoide	Tiroglobulina
Tireoide	Tiroxina livre (T4)
Tireoide	Triiodotironina
Toxina humana	Radiação eletromagnetica
Toxina humana	Bebida estimulante
Toxina humana	Tabaco /nicotina
Toxina humana	Residuos toxicos de pesticida
Vitaminas	Vitamina B1
Vitaminas	Vitamina B12
Vitaminas	Vitamina B3
Vitaminas	Vitamina B6
Vitaminas	Vitamina D3
Vitaminas	Vitamina K
Vitaminas	Vitamina A
Vitaminas	Vitamina B2
Vitaminas	Vitamina C
Vitaminas	Vitamina E