

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

**Projeto de Extensão Universitária – Ciência da Computação - Estácio 2024 - Marcus Cesar Pereira Ferreira, Matrícula - 202102340918**

**Protótipo e linha de produção realizada na empresa FONART - ESPAÇO DE FONOAUDIOLOGIA APLICADA E SAÚDE INTEGRAL**

**Este projeto envolveu:**

Tema 2 Rad (rapid Applications Development)

Tema 3 Manipulação de Dados em Arquivos

Tema 4 Python Com Banco de Dados

Tema 5 Interface Gráfica Com Python

Tema 6 Aplicando Rad

## Repositório Github

<https://github.com/MarcusCCesar/FONART>

✓ **Ênfase nas páginas 9, 10, 68-83**

✓ **Código Python (principal) pronto, página 50 ou no repositório <https://github.com/MarcusCCesar/FONART>**

1

*Resumo do Projeto: Este sistema integra várias tecnologias e foi desenvolvido para a Extensão Universitária na Estácio 2024. Localmente, o é utilizado o sistema Python para integração com o Firebase do Google Cloud e o Cloud Firestore, armazenando dados de pacientes de fonoaudiologia.*

*A interface em Tkinter permite a coleta de informações como nome, idade e diagnóstico. O sistema também oferece funcionalidades como geração de relatórios em PDF e gráficos de análise, além de sincronizar automaticamente os dados locais e na nuvem. Posteriormente, avançamos para o formulário HTML e JavaScript para encaminhar os dados ao Cloud Firestore, que os valida e armazena.*

*O objetivo do formulário WEB <https://fonart-8cd50.web.app/> é coletar dados estatísticos de pacientes em tratamento fonoaudiológico. A coleta de informações como idade, diagnóstico, localidade e tratamento é essencial para entender as necessidades dos pacientes e aprimorar a atuação da equipe. O consentimento do paciente é obrigatório para a utilização dos dados, que são tratados com a máxima confidencialidade, visando à conformidade com a LGPD.*

*A análise do código apresenta os temas de aprendizagem com vários aspectos técnicos e acadêmicos. Vamos abordar cada um dos tópicos mencionados, detalhando como o código se alinha a eles:*

## TEMA 1

### 1. RAD (Rapid Application Development)

#### 1.1 Contextualização, conceitos, princípios, ferramentas e técnicas:

- O código utiliza Python como linguagem de desenvolvimento e integra com Firebase, o que demonstra uma aplicação prática de RAD com foco na agilidade. A escolha do Tkinter para a interface gráfica e bibliotecas como Pandas e Matplotlib também está alinhada com práticas de RAD.

#### 1.2 Fases da RAD:

- O sistema segue um ciclo iterativo, permitindo modificações rápidas na interface e funcionalidades (ex: atualização de dados no Firebase e no CSV).

#### 1.3 Quando aplicar e quando não aplicar RAD:

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

- O uso do RAD é apropriado aqui devido à natureza do projeto, que exige resposta rápida às necessidades dos usuários (profissionais de saúde).

## **1.4 Python e as ferramentas (frameworks):**

- O código utiliza Firebase para armazenamento em nuvem, Pandas para manipulação de dados, Tkinter para interface gráfica e ReportLab para geração de relatórios, cobrindo diversas ferramentas que facilitam o desenvolvimento ágil.

## **TEMA 2**

### **2. Manipulação de Dados em Arquivos**

#### **2.1 Manipulação de arquivos:**

- O código manipula arquivos CSV para armazenar e sincronizar dados de pacientes, o que é essencial para a persistência de dados.

2

#### **2.2 Manipulação de strings:**

- O código inclui validações de strings (ex: nome, diagnóstico) e manipulação ao criar registros no CSV, garantindo a integridade dos dados.

#### **2.3 Exceções na manipulação de arquivos e outras operações:**

- O uso de `try-except` para tratar exceções em operações de arquivo e banco de dados está presente, garantindo robustez ao código.

## **TEMA 4**

### **3. Python com Banco de Dados**

#### **3.1 Frameworks e bibliotecas para gerenciamento de banco de dados:**

- A integração com Firebase e Firestore é uma aplicação clara de gerenciamento de banco de dados.

#### **3.2 Conexão, acesso e criação de bancos de dados e tabelas:**

- O código contém funções que inicializam a conexão com o Firestore, demonstrando acesso e interação com o banco de dados.

#### **3.3 Inserção, remoção e atualização de registros em tabelas:**

- O código inclui funções para adicionar (inserir) dados ao Firestore, embora a remoção não esteja explicitamente implementada.

#### **3.4 Recuperação de registros em tabelas:**

- A recuperação de dados do Firestore para gerar gráficos e relatórios é feita corretamente, mostrando como os dados são consultados.

## **TEMA 4**

### **4. Interface Gráfica com Python**

#### **4.1 Principais frameworks e bibliotecas para a GUI:**

- O Tkinter é utilizado como framework para a interface gráfica, o que é apropriado para o projeto.

#### **4.2 Adição de widgets e montagem da interface gráfica:**

- O código mostra a adição de vários widgets, como campos de entrada e botões, para interação com o usuário.

#### **4.3 Interface para inclusão de dados em uma tabela no banco de dados:**

- O sistema permite a coleta de dados dos pacientes e os salva no Firestore, o que atende a esta necessidade.

#### **4.4 Interface para localização, alteração e exclusão de dados em tabela:**

- Embora a inclusão esteja presente, a localização e exclusão de dados não são explicitamente abordadas no trecho do código fornecido, contudo, esta função foi centralizada visando a privacidade e segurança dos dados sensíveis, no console do firebase para o administrador do sistema.

**TEMA 6**

**5. Aplicando RAD**

**5.1 Etapas para tratamento dos requisitos de um sistema:**

- O sistema demonstra uma abordagem iterativa, com validação de dados e feedback ao usuário, que são essenciais em RAD.

**5.2 Modelagens de negócios e de dados da RAD:**

- O uso de um banco de dados para armazenar dados dos pacientes reflete uma modelagem de dados apropriada.

**5.3 Design de interface com o usuário na RAD:**

- A interface é simples e funcional, permitindo a coleta de dados de forma eficiente, o que é fundamental em RAD.

**5.4 Aplicação RAD implementada em Python:**

- O projeto como um todo é uma aplicação prática dos conceitos de RAD, usando Python para desenvolvimento rápido e eficaz.

3

**Considerações Finais**

O código fornecido atende, em sua maior dimensão, as exigências acadêmicas e técnicas mencionadas.

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

## Sumário

<b>DESAFIOS:</b> .....	11
<b>INÍCIO DO PROJETO PYTHON</b> .....	15
<b>Instalação do Python no Windows 11</b> .....	15
<b>Criamos a Pasta para o Projeto:</b> .....	15
<b>Criação e Ativação um Ambiente Virtual:</b> .....	15
<b>Instalação das Bibliotecas Necessárias:</b> .....	15
Desenvolvimento inicial da estrutura do projeto em Python.....	16
Instalação de Bibliotecas.....	17
Avanços das Bibliotecas iniciais do projeto rascunho .....	17
<b>Criação dos Arquivos do Projeto</b> .....	18
Montagem no editor de texto como o Visual Studio Code.....	19
<b>Verificamos o Ambiente Virtual e Executamos o Sistema</b> .....	19
Árvore inicial do Python (rascunho).....	20
Ativação do ambiente virtual (rascunho e testes).....	20
<b>Criar a Pasta para Relatórios:</b> .....	21
Testes - Prompt de comando .....	22
<b>Verificar a Saída:</b> .....	22
<b>Geração e Verificação do Relatório</b> .....	22
<b>Abrir o Relatório:</b> .....	22
<b>Conteúdo do Relatório:</b> .....	22
<b>Garantindo a Conformidade com a LGPD</b> .....	23
<b>Conclusão</b> .....	23
<b>DADOS DIRETAMENTE NO SISTEMA</b> .....	23
<b>Passo 1</b> .....	23
<b>Inserir os Dados dos Pacientes Usando Tkinter</b> .....	23
<b>Código atualizado para permitir que o fonoaudiólogo insira dados usando a interface gráfica.</b> .....	23
<b>Passo 2:</b> .....	25
<b>Salvar os Dados em um Arquivo CSV</b> .....	25
<b>Passo 3:</b> .....	25
<b>Interface Gráfica para Inserir Dados</b> .....	25

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

<b>Passo 4:</b> .....	26
<b>Executar o Sistema</b> .....	26
<b>Explicação</b> .....	26
<b>CONCLUSÃO:</b> .....	26
<b>Como funciona:</b> .....	26
<b>Salvar os dados dos pacientes:</b> .....	26
<b>Atualizar dados:</b> .....	26
<b>Explicação mais detalhada do que acontece no código:</b> .....	27
<b>Trecho relevante para salvar os dados:</b> .....	27
<b>Explicação:</b> .....	27
<b>Fluxo para o fonoaudiólogo:</b> .....	27
<b>O código atualizado.</b> .....	28
<b>Alterações Implementadas:</b> .....	30
<b>Aviso LGPD:</b> .....	30
<b>Confirmação do Consentimento:</b> .....	30
<b>Alerta de Privacidade</b> .....	30
<b>UPDATE 2.0</b> .....	30
Atalho para área de trabalho do usuário .....	30
Etapas detalhadas para criar esse atalho no Windows: .....	31
Criar um arquivo .bat (Windows).....	31
Abra o Bloco de Notas ou Visual Studio Code .....	31
Colocar um ícone no arquivo .bat .....	31
Executar o sistema com um clique .....	31
Evidências das pastas: .....	32
Pastas iniciais (rascunho) .....	32
Pasta iniciais do projeto com os relatórios gerados pelo Python.....	34
<b>Relatórios gerados automaticamente do Excel e Gráficos</b> .....	34
<b>Relatórios gerados automaticamente do PDF com anonimização dos nomes, visando respeitar a privacidade desde a concepção <i>Privacy by Design</i> - LGP ....</b>	34
<b>SISTEMA E MELHORIAS:</b> .....	34
<b>PYTHON AVANÇADO - PARA O CONSULTÓRIO FONART.</b> .....	37

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

<b>VERSÃO 2.0 .....</b>	37
<b>Evidências técnicas do Projeto de Extensão: Criando o banco de dados google cloud .....</b>	42
<b>Projeto no Google Cloud .....</b>	42
Ativando o Firestore .....	43
Ativação dos recursos .....	44
Testes e experimentações do Google Cloud (conhecendo as ferramentas e possibilidades) .....	45
<b>Instalar Bibliotecas Necessárias .....</b>	46
Criando relações de compartilhamento da Empresa do Projeto, com o desenvolvedor responsável pelo Projeto Fonart, de Extensão Universitária (Estácio). .....	47
Relações realizadas .....	47
Evidências do trabalho em equipe .....	48
Teste de todos os projetos (ambiente de testes, homologação e de produção)...	48
<b>Implantação do Banco de Dados – Firestore/Firebase – Google Cloud .....</b>	49
Console firebase – Google Cloud (início dos projetos para produção) .....	49
Criando relações do Banco de Dados .....	50
Testes e sucessos com as relações de sincronização do arquivo “pacientes.csv” .	50
Testes e sucessos com a relação do Firebase Hosting.....	51
<b>VERSÃO FINAL 3.0 FIREBASE_SETUP.PY .....</b>	51
<b>Telas do novo sistema pronto .....</b>	61
Tela com os arquivos do sistema principal local (firebase_setup.py) .....	62
Sistema app WEB (index.html) .....	62
Formulário de pesquisa de MELHORIAS NO SISTEMA (qualidade.html) .....	62
Firestore database .....	62
Hosting – Google Cloud .....	62
Visão da Árvore do projeto Python .....	62
<b>Sistemas Prontos e funcionais.....</b>	63
<b>Projeto Fonart – firebase_setup.py.....</b>	63
<b>Tela do sistema pronto no computador: .....</b>	63
<b>Graficos e Dados gerados para o Projeto .....</b>	64

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

<b>Relatórios com Anonimização de dados .....</b>	64
<b>PDF gerado com anonimização .....</b>	65
<b>Sincronização total do Projeto .....</b>	66
Páginas de internet montadas e geradas (espelho do sistema local) .....	66
<b>Fusão do sistema Web Firebase com o domínio oficial da empresa – fonart.com.br .....</b>	67
Relatório Restrito   FONART (DOMÍNIO).....	67
Código pronto Python do sistema Relatórios via App WEB .....	67
Index.html.....	67
<b>Web APP do Formulário de Pesquisa de melhorias (projeto).....</b>	71
Qualidade.html.....	71
<b>Ambiente do Centro de MELHORIAS DO SISTEMA (solicitações, melhorias, desenvolvimento etc.).....</b>	72
Centro de Pesquisa de Melhorias — Projeto Fonart (fonart-8cd50.web.app).....	72
<b>Sincronização com EmailJS e Firebase .....</b>	73
<b>CIENCIA DA COMPUTAÇÃO - O PROJETO DE EXTENSÃO UNIVERSITARIA – ESTÁCIO DE SÁ .....</b>	73
<b>PLANO DE APRENDIZAGEM VISANDO MELHORIAS NA EMPRESA FONART - ESPAÇO FONOAUDIOLOGIA APLICADA E SAÚDE INTEGRAL .....</b>	73
1. Código e nome da disciplina: .....	73
<b>DGT0235 DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES EM PYTHON .....</b>	73
2. Área temática: .....	73
3. Natureza: .....	74
4. Competências a serem trabalhadas: .....	74
5. Ementa .....	74
6. Objetivos.....	74
7. OBJETIVOS SOCIOCOMUNITÁRIOS DA EMPRESA FONART.....	74
Instalação do Apps da Web - Web App Fonart.....	75
9. Justificativa:.....	76
10. Procedimentos de ensino-aprendizagem: .....	77
11. Os Estudos dos conteúdos digitais da disciplina; .....	78
12. Procedimentos de avaliação: .....	78

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

<b>RESULTADOS.....</b>	79
<b>1. RAD (Rapid Application Development):.....</b>	79
- Aplicação do RAD: .....	79
- Cumprimento:.....	79
- Ferramentas RAD Compatíveis com Python: .....	79
- Cumprimento:.....	79
<b>2. Manipulação de Dados em Arquivos:.....</b>	79
- Manipulação de Arquivos CSV:.....	79
- Cumprimento:.....	79
- Exceções:.....	79
- Cumprimento:.....	80
<b>3. Python com Banco de Dados: .....</b>	80
- Integração com o Firestore: .....	80
O código conecta-se ao Firebase, sincronizando dados entre o CSV local e a nuvem, gerenciando a adição e atualização de registros no Firestore.....	80
- Cumprimento:.....	80
- Gerenciamento de Registros: .....	80
- Cumprimento:.....	80
<b>4. Interface Gráfica com Python:.....</b>	80
- Tkinter para GUI:.....	80
- Cumprimento:.....	80
- Widget de Inserção e Edição de Dados: .....	80
- Cumprimento:.....	80
<b>5. Relatórios e Geração de Gráficos: .....</b>	81
- Relatórios em PDF: .....	81
- Cumprimento:.....	81
- Geração de Gráficos: .....	81
- Cumprimento:.....	81
<b>6. Aplicando RAD em Python:.....</b>	81
- Tratamento de Requisitos e Modelagem de Negócios:.....	81
- Cumprimento:.....	81

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

- Design de Interface com o Usuário: .....	81
- Cumprimento:.....	81
<b>7. Competências Técnicas e Socioemocionais: .....</b>	<b>81</b>
- Competências Técnicas:*	81
- Cumprimento:.....	82
- Competências Socioemocionais:.....	82
- Cumprimento:.....	82
<b>Conclusão da Análise técnica para avaliação acadêmica de nossos estudos:** .....</b>	<b>82</b>
<b>Pontos fortes: .....</b>	<b>82</b>
- O código é modular e bem-organizado.....	82
<b>Possíveis Melhorias (discussões com a empresária):.....</b>	<b>82</b>
<b>INTEGRAÇÃO DO PROJETO .....</b>	<b>82</b>
Resumo da Integração Esperada .....	83
<b>1. Sincronização de Dados Entre Firebase Hosting e Python.....</b>	<b>83</b>
<b>2. Autonomia Entre Interfaces (Tkinter e Web) .....</b>	<b>83</b>
<b>3. Sincronização com CSV Local no Python .....</b>	<b>83</b>
<b>4. Conformidade com o Projeto Acadêmico .....</b>	<b>84</b>
<b>Recomendação – Pesquisa de Melhorias de Sistema FONART. ....</b>	<b>84</b>
<b>Conclusão Geral .....</b>	<b>84</b>
<b>O Projeto e Integração entre a Aplicação Web e o Script Python .....</b>	<b>84</b>
A. Fluxo de Dados Centralizado via Firestore.....	84
B. Segurança e Conformidade com a LGPD .....	85
C. Sincronização Centralizada e Eficiente.....	85
D. Conclusão do desafio .....	85
<b>DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES EM PYTHON o TEMA 6, MÓDULO 4 ...</b>	<b>86</b>
<b>Objetivo do Módulo 4 (Esquematizar uma aplicação RAD em Python):.....</b>	<b>86</b>
<b>Objetivo do meu projeto com a empresa FONART: .....</b>	<b>87</b>
<b>Comparação de Estrutura .....</b>	<b>87</b>
<b>1. Prototipação e Iteração:.....</b>	<b>87</b>
- Módulo 4:.....	87
- Projeto Acadêmico Realizado: .....	87

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

<b>2. Metodologia RAD (Rápida): .....</b>	87
- Módulo 4:.....	87
- Projeto Acadêmico Realizado: .....	87
<b>3. Uso de Python e Bibliotecas: .....</b>	87
- Módulo 4:.....	87
- Projeto Acadêmico Realizado: .....	88
<b>4. Requisitos Funcionais: .....</b>	88
- Módulo 4:.....	88
- Projeto Acadêmico Realizado: .....	88
<b>5. Interface com o Usuário: .....</b>	88
- Módulo 4:.....	88
- Projeto Acadêmico Realizado: .....	88
<b>APLICAÇÃO DOS CONCEITOS DO MÓDULO 4 NO PROJETO.....</b>	88
- Implementação em Python:.....	88
- RAD como metodologia: .....	88
- Prototipagem e feedback do usuário: .....	89
<b>ÊNFASE - REFERÊNCIA ACADÊMICA.....</b>	89
Concluo e Referencio o Módulo 4 ao longo do Projeto FONART da seguinte maneira: .....	89
O que foi alterado ou adicionado no Projeto Acadêmico com a FONART:.....	89
EmailJS:.....	89
Códigos EmailJS:.....	89
Benefícios:.....	89
Erros encontrados e solucionados página (qualidade.html) após o comando “firebase deploy” .....	90
<b>PASTAS FINALIZADAS .....</b>	93
<b>PASSO A PASSO PARA USO DO SISTEMA FONART .....</b>	95
Interface do Sistema .....	95
<b>ÁRVORE DO PROJETO_FONART/ .....</b>	96
Arquivo README.md .....	96

## Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

### **DESAFIOS:**

Projeto de Extensão Universitária - Estácio 2024 - Marcus Cesar Pereira Ferreira - Matrícula - [202102340918@alunos.estacio.br](mailto:202102340918@alunos.estacio.br)

### **SISTEMA DE COLETA DE DADOS PARA ESTATÍSTICAS EM PATOLOGIAS FONOAUDIOLÓGICAS**

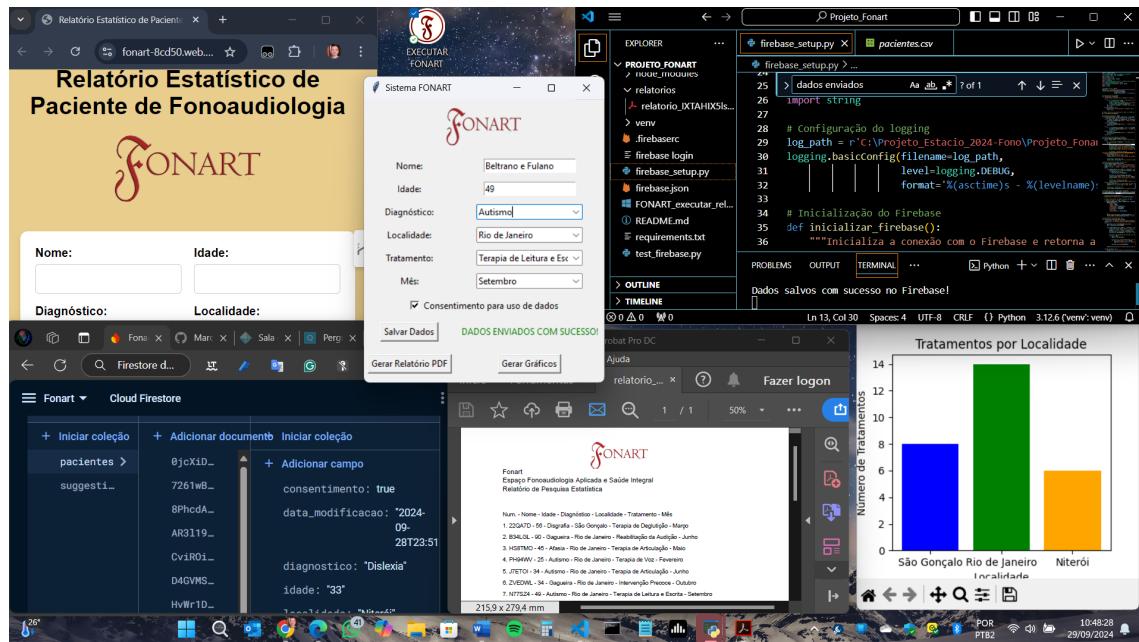
#### **Resumo do Projeto:**

Este sistema integra várias tecnologias e foi desenvolvido para a Extensão Universitária na Estácio 2024. Localmente, utiliza Python para integração com o Firebase do Google Cloud e o Cloud Firestore, armazenando dados de pacientes de fonoaudiologia.

A interface em Tkinter permite a coleta de informações como nome, idade e diagnóstico. O sistema também oferece funcionalidades como geração de relatórios em PDF e gráficos de análise, além de sincronizar automaticamente os dados locais e na nuvem. O formulário utiliza HTML e JavaScript para encaminhar os dados ao Cloud Firestore, que os valida e armazena.

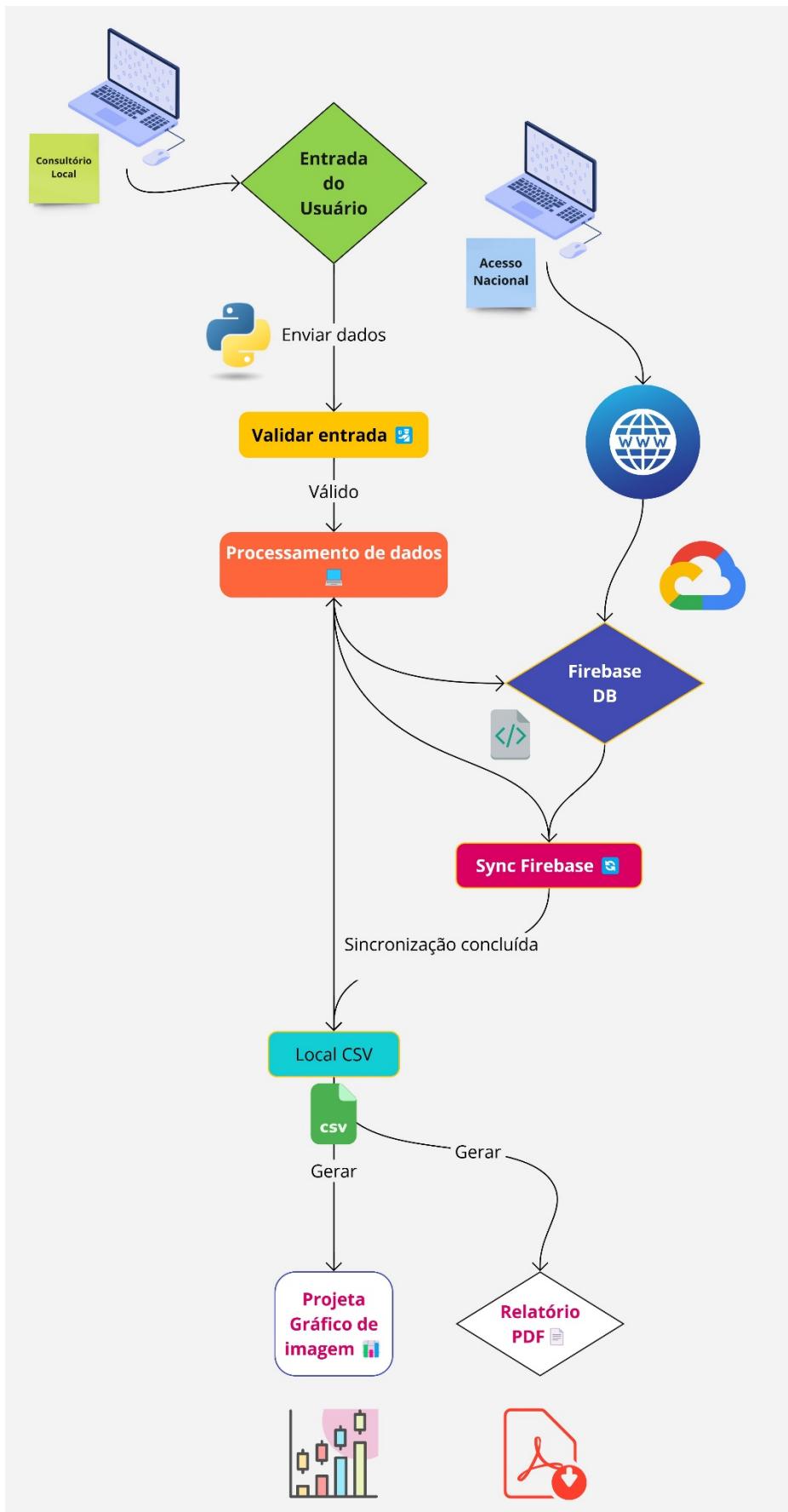
O objetivo do formulário WEB <https://fonart-8cd50.web.app/> é coletar dados estatísticos de pacientes em tratamento fonoaudiológico. A coleta de informações como idade, diagnóstico, localidade e tratamento é essencial para entender as necessidades dos pacientes e aprimorar a atuação da equipe. O consentimento do paciente é obrigatório para a utilização dos dados, que são tratados com a máxima confidencialidade, visando à conformidade com a LGPD.

## Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia



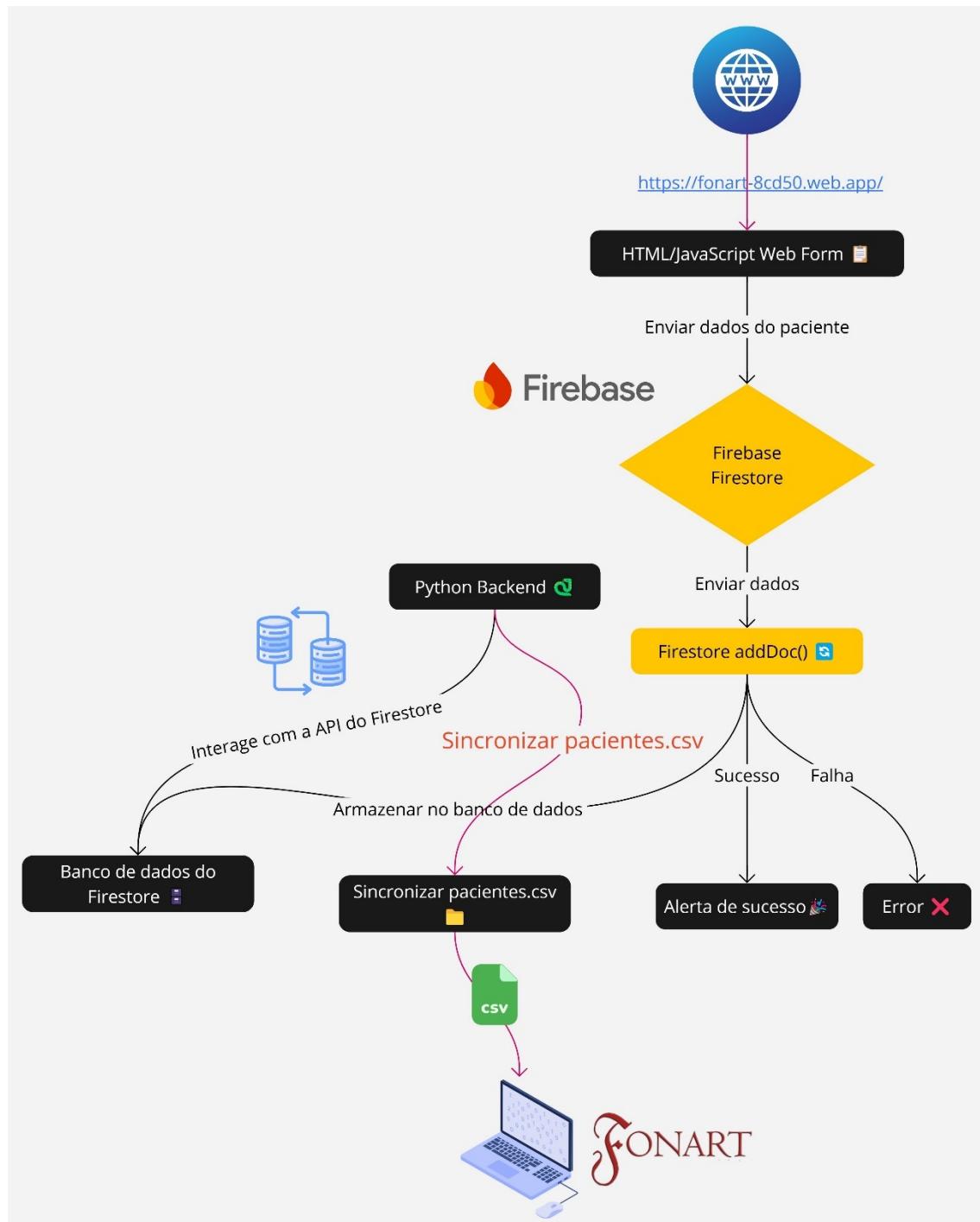
Tela com as soluções prontas do projeto de extensão universitária

## Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia



## Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

Conforme o tema 6, módulo 4, foi utilizado o Diagrama Python (local e Firebase) para demonstrar o fluxo de soluções relativas à sincronização local e em nuvem 1



Conforme o tema 6, módulo 4, foi utilizado o diagrama de conexão e sincronização de dados no Firebase/Firestore, visa o sucesso das integrações do Banco de dados e atualização do arquivo “pacientes.csv”, responsável pelos dados confidenciais em ambiente local do consultório 2

## INÍCIO DO PROJETO PYTHON

### Instalação do Python no Windows 11

1. Verificar a Instalação python –version
2. Criação do Ambiente de Desenvolvimento

### Criamos a Pasta para o Projeto:

3. Prompt de Comando.
4. pasta do projeto (no exemplo, C:\Projeto\_Estacio\_2024-Fono):
5. cd C:\Projeto\_Estacio\_2024-Fono
6. Nova pasta para o projeto  
mkdir consultorio\_fono  
cd consultorio\_fono

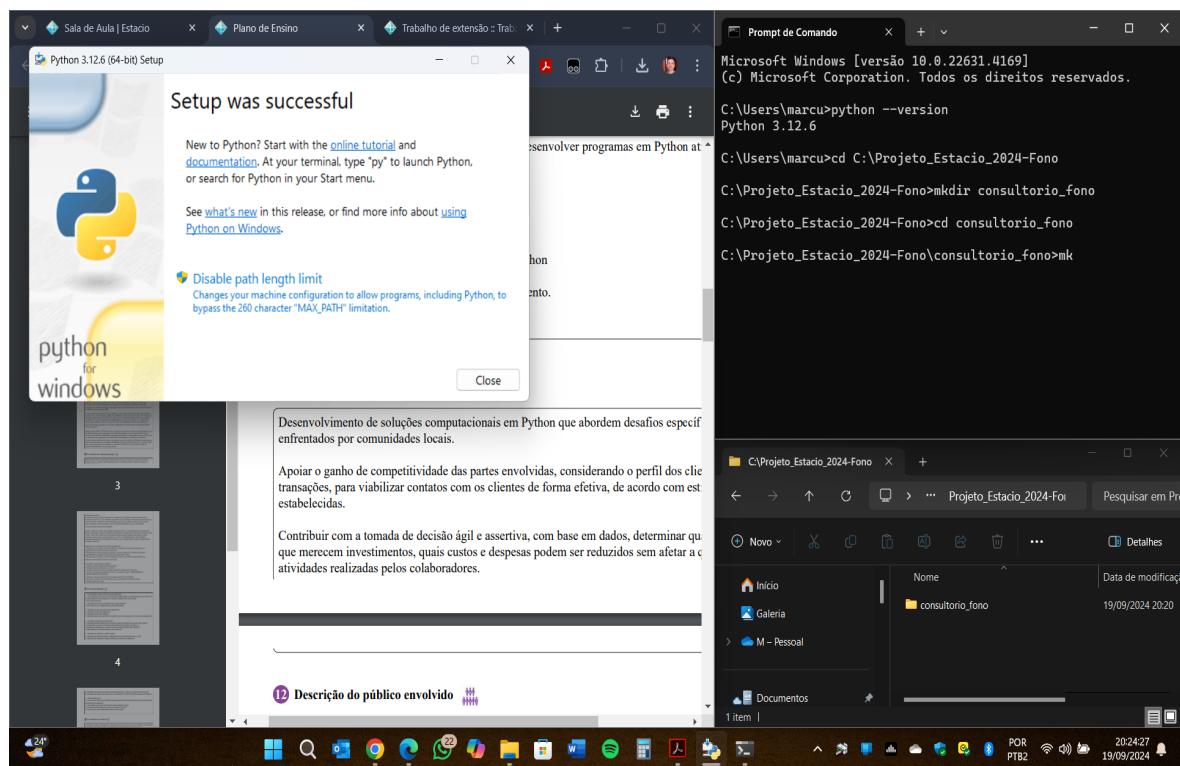
### Criação e Ativação um Ambiente Virtual:

7. No Prompt de Comando, criamos um ambiente virtual para o projeto:  
python -m venv env
8. Ativamos o ambiente virtual:  
env\Scripts\activate

### Instalação das Bibliotecas Necessárias:

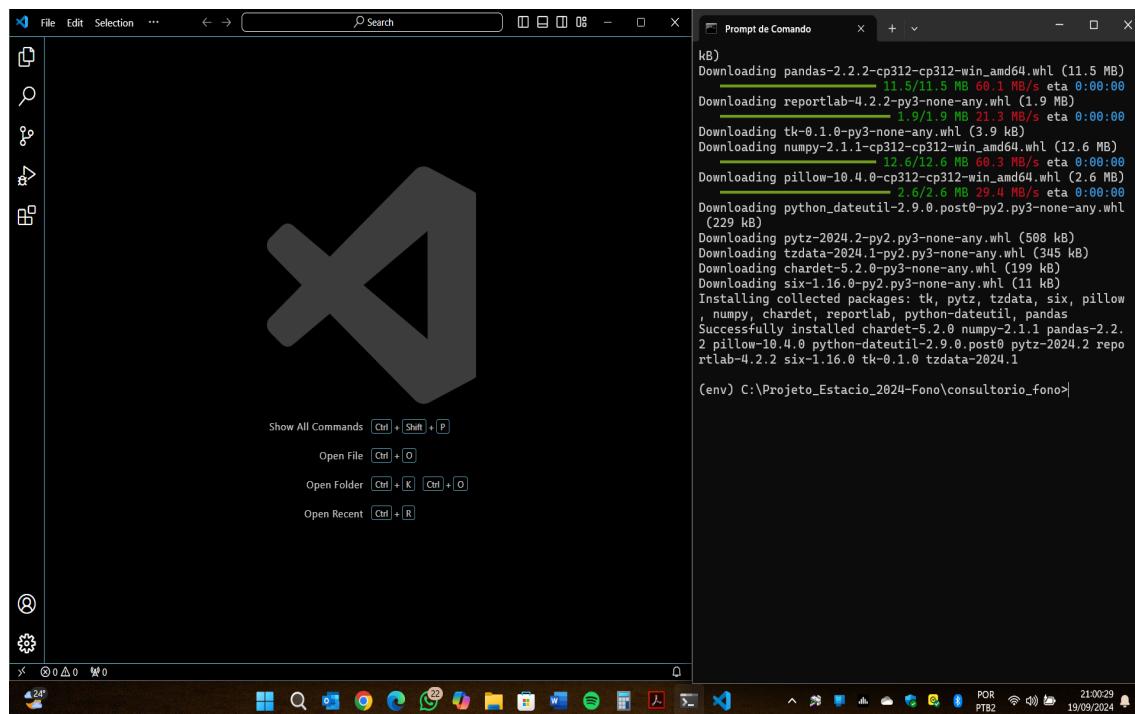
9. Ambiente virtual ativo, instalamos as bibliotecas Pandas, ReportLab e Tkinter:  
pip install pandas reportlab tk

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia



## Desenvolvimento inicial da estrutura do projeto em Python.

Incluindo suas pastas e instâncias, em notebooks com o sistema W11 mais avançado da empresa FONART, visando elaborar um campo estatístico fundamental para estudos técnicos em fonoaudiologia. O histórico da empresa era realizado manualmente e com notebooks inferiores, utilizados apenas para acesso a textos de laudos. Atenção fundamental à privacidade e anonimização de dados, visando respeitar a legislação vigente e demais protocolos de segurança.



# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

## Instalação de Bibliotecas

Plano de Ensino x Trabalho de extensão :: Trab... x + - □ ×

Plano de Aprendizagem

1 Código e nome da disciplina DGT0235 DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES EM PYTHON

2 Natureza Extensão

3 Carga horária semestral

4 Carga horária semanal

5 Perfil docente

O professor/tutor deve ser graduado em Ciência da Computação, Sistemas de Informação, Engenharia de Computação, ou afins. É desejável a PósGraduação Stricto Sensu (Mestrado e/ou Doutorado) na área do curso ou áreas afins.

É desejável que o professor/tutor possua experiência profissional na área de desenvolvimento, com experiência em Python, além de conhecimentos e habilidades teórico-práticos, capacidade de comunicação, interação e fluência digital para utilizar ferramentas necessárias ao desenvolvimento do processo de ensino-aprendizagem (SGC, SAVA, BÁO e SIA). Importante, também, o conhecimento do Projeto Pedagógico dos Cursos que a disciplina faz parte na Matriz Curricular.

É necessário que o professor/tutor domine as metodologias ativas inerentes à educação por competências, em especial a aprendizagem baseada em projetos e ferramentas digitais que tornem o processo mais interativo. Além disto, é estimula que o professor/tutor estimule o autoconhecimento e autoupardizagem entre seus alunos.

6 Área temática

Prompt de Comando x + - □ ×

```
(59 kB)
Collecting python-dateutil>=2.8.2 (from pandas)
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting pytz>=2020.1 (from pandas)
  Downloading pytz-2024.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas)
  Downloading tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting pillow>=9.0.0 (from reportlab)
  Downloading pillow-10.4.0-cp312-cp312-win_amd64.whl.metadata (9.3 kB)
Collecting chardet (from reportlab)
  Downloading chardet-5.2.0-py3-none-any.whl.metadata (3.4 kB)
Collecting six>=1.5 (from python-dateutil>=2.8.2->pandas)
  Downloading six-1.16.0-py2.py3-none-any.whl.metadata (1.8 kB)
Downloading pandas-2.2.2-cp312-cp312-win_amd64.whl (11.5 MB)
  11.5/11.5 MB 60.1 MB/s eta 0:00:00
Downloading reportlab-4.2.2-py3-none-any.whl (1.9 MB)
  1.9/1.9 MB 21.3 MB/s eta 0:00:00
Downloading tk-0.1.0-py3-none-any.whl (3.9 kB)
Downloading numpy-2.1.1-cp312-cp312-win_amd64.whl (12.6 MB)
  12.6/12.6 MB 60.3 MB/s eta 0:00:00
Downloading pillow-10.4.0-cp312-cp312-win_amd64.whl (2.6 MB)
  2.6/2.6 MB 29.4 MB/s eta 0:00:00
Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl
(229 kB)
Downloading pytz-2024.2-py2.py3-none-any.whl (508 kB)
Downloading tzdata-2024.1-py2.py3-none-any.whl (345 kB)
Downloading chardet-5.2.0-py3-none-any.whl (199 kB)
Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: tk, pytz, tzdata, six, pillow, numpy, chardet, reportlab, python-dateutil, pandas
Successfully installed chardet-5.2.0 numpy-2.1.1 pandas-2.2.2 pillow-10.4.0 python-dateutil-2.9.0.post0 pytz-2024.2 reportlab-4.2.2 six-1.16.0 tk-0.1.0 tzdata-2024.1
(env) C:\Projeto_Estacio_2024-Fono\consultorio_fono>
```

20:53:21 PTB2 19/09/2024

17

## Avanços das Bibliotecas iniciais do projeto rascunho

File Edit Selection ... ← → Search

Prompt de Comando x + - □ ×

```
(229 kB)
Downloading pandas-2.2.2-cp312-cp312-win_amd64.whl (11.5 MB)
  11.5/11.5 MB 60.1 MB/s eta 0:00:00
Downloading reportlab-4.2.2-py3-none-any.whl (1.9 MB)
  1.9/1.9 MB 21.3 MB/s eta 0:00:00
Downloading tk-0.1.0-py3-none-any.whl (3.9 kB)
Downloading numpy-2.1.1-cp312-cp312-win_amd64.whl (12.6 MB)
  12.6/12.6 MB 60.3 MB/s eta 0:00:00
Downloading pillow-10.4.0-cp312-cp312-win_amd64.whl (2.6 MB)
  2.6/2.6 MB 29.4 MB/s eta 0:00:00
Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl
(229 kB)
Downloading pytz-2024.2-py2.py3-none-any.whl (508 kB)
Downloading tzdata-2024.1-py2.py3-none-any.whl (345 kB)
Downloading chardet-5.2.0-py3-none-any.whl (199 kB)
Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: tk, pytz, tzdata, six, pillow, numpy, chardet, reportlab, python-dateutil, pandas
Successfully installed chardet-5.2.0 numpy-2.1.1 pandas-2.2.2 pillow-10.4.0 python-dateutil-2.9.0.post0 pytz-2024.2 reportlab-4.2.2 six-1.16.0 tk-0.1.0 tzdata-2024.1
(env) C:\Projeto_Estacio_2024-Fono\consultorio_fono>
```

21:00:29 PTB2 19/09/2024

## Criação dos Arquivos do Projeto

- 10.** Criar o Arquivo Principal (consultorio\_fono.py):
- 11.** Abrimos editor de texto como o Visual Studio Code.
- 12.** Criamos um arquivo chamado consultorio\_fono.py dentro da pasta do projeto.

```
import pandas as pd
from reportlab.lib.pagesizes import A4
from reportlab.pdfgen import canvas
from tkinter import Tk, Label, Button, filedialog

# Dados fictícios - Em um sistema real, seriam carregados de um banco de dados ou interface
pacientes = [
    {'nome': 'Carlos Silva', 'idade': 40, 'diagnostico': 'Disfonia', 'localidade': 'Rio de Janeiro', 'tratamento': 'Terapia vocal', 'consentimento': 1},
    {'nome': 'Ana Souza', 'idade': 30, 'diagnostico': 'Dislalia', 'localidade': 'São Paulo', 'tratamento': 'Terapia fonoaudiológica', 'consentimento': 1},
    # Mais pacientes podem ser adicionados aqui
]

# Função para gerar relatório estatístico
def gerar_relatorio_estatistico():
    df = pd.DataFrame(pacientes)

    # Filtra apenas os pacientes que consentiram
    df_consentidos = df[df['consentimento'] == 1]

    # Gera estatísticas anonimizadas
    estatisticas = {
        'Faixa Etária': df_consentidos['idade'].value_counts(bins=[0, 18, 30, 50, 100], sort=False),
        'Diagnósticos': df_consentidos['diagnostico'].value_counts(),
        'Localidades': df_consentidos['localidade'].value_counts(),
        'Tratamentos': df_consentidos['tratamento'].value_counts()
    }

    # Cria o PDF
    pdf_file = "relatorios/relatorio_estatistico.pdf"
    pdf = canvas.Canvas(pdf_file, pagesize=A4)
    pdf.setTitle("Relatório Estatístico - Pacientes Fonoaudiologia")

    pdf.drawString(100, 800, "Relatório Estatístico de Pacientes")
    pdf.drawString(100, 780, "Dados Gerados em Conformidade com a LGPD")

    # Adiciona as estatísticas no PDF
    y = 750
    for chave, valor in estatisticas.items():
        pdf.drawString(100, y, f'{chave}:')
        y -= 20
        for item, contagem in valor.items():
            pdf.drawString(120, y, f'{item}: {contagem}')
            y -= 20
        y -= 10

    # Salva o PDF
    pdf.save()
    print(f'Relatório gerado em: {pdf_file}')

# Função para abrir a interface gráfica
def abrir_interface():
    root = Tk()
    root.title("Sistema de Fonoaudiologia")

    Label(root, text="Sistema de Fonoaudiologia", font=("Arial", 16)).pack(pady=20)

    Button(root, text="Gerar Relatório Estatístico", command=gerar_relatorio_estatistico).pack(pady=10)

    root.mainloop()
```

Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
if __name__ == "__main__":
    abrir_interface()
```

The screenshot shows a Windows desktop environment with two main windows open:

- File Explorer:** Shows a folder structure under "CONSULTORIO\_FONO". The "consultorio\_fono.py" file is selected.
- Prompt de Comando (Terminal):** Displays the output of a pip install command for various Python packages, including pandas, reportlab, tk, numpy, pillow, python-dateutil, pytz, tzdata, six, chardet, and reportlab. It also shows the activation of a virtual environment named "env" and the path "C:\Projeto\_Estacio\_2024-Fono\consultorio\_fono".

The code editor window contains the following Python script:

```
consultorio_fono.py

# Importando bibliotecas necessárias
1 import pandas as pd
2 from reportlab.lib.pagesizes import A4
3 from reportlab.pdfgen import canvas
4 from tkinter import Tk, Label, Button, filedialog
5
6 # Dados fictícios - Em um sistema real, seriam carregados de um banco de
7 pacientes = [
8     {'nome': 'Carlos Silva', 'idade': 40, 'diagnostico': 'Disfonia', 'localidade': 'Belo Horizonte'},
9     {'nome': 'Ana Souza', 'idade': 30, 'diagnostico': 'Disália', 'localidade': 'São Paulo'},
10    # Mais pacientes podem ser adicionados aqui
11 ]
12
13 # Função para gerar relatório estatístico
14 def gerar_relatorio_estatistico():
15     df = pd.DataFrame(pacientes)
16
17     # Filtra apenas os pacientes que consentiram
18     df_consentidos = df[df['consentimento'] == 1]
19
20     # Gera estatísticas anonimizadas
21     estatisticas = {
22         'Faixa Etária': df_consentidos['idade'].value_counts(bins=[0, 18, 30, 45, 60, 75, 90]),
23         'Diagnósticos': df_consentidos['diagnostico'].value_counts(),
24         'Localidades': df_consentidos['localidade'].value_counts(),
25         'Tratamentos': df_consentidos['tratamento'].value_counts()
26     }
27
28     # Cria o PDF
29     pdf_file = "relatorios/relatorio_estatistico.pdf"
30     pdf = canvas.Canvas(pdf_file, pagesize=A4)
31     pdf.setTitle("Relatório Estatístico - Pacientes Fonoaudiologia")
32
33     pdf.drawString(100, 800, "Relatório Estatístico de Pacientes")
34     pdf.drawString(100, 780, "Dados Gerados em Conformidade com a LGPD")
35
36     # Adiciona as estatísticas no PDF
37     y = 750
```

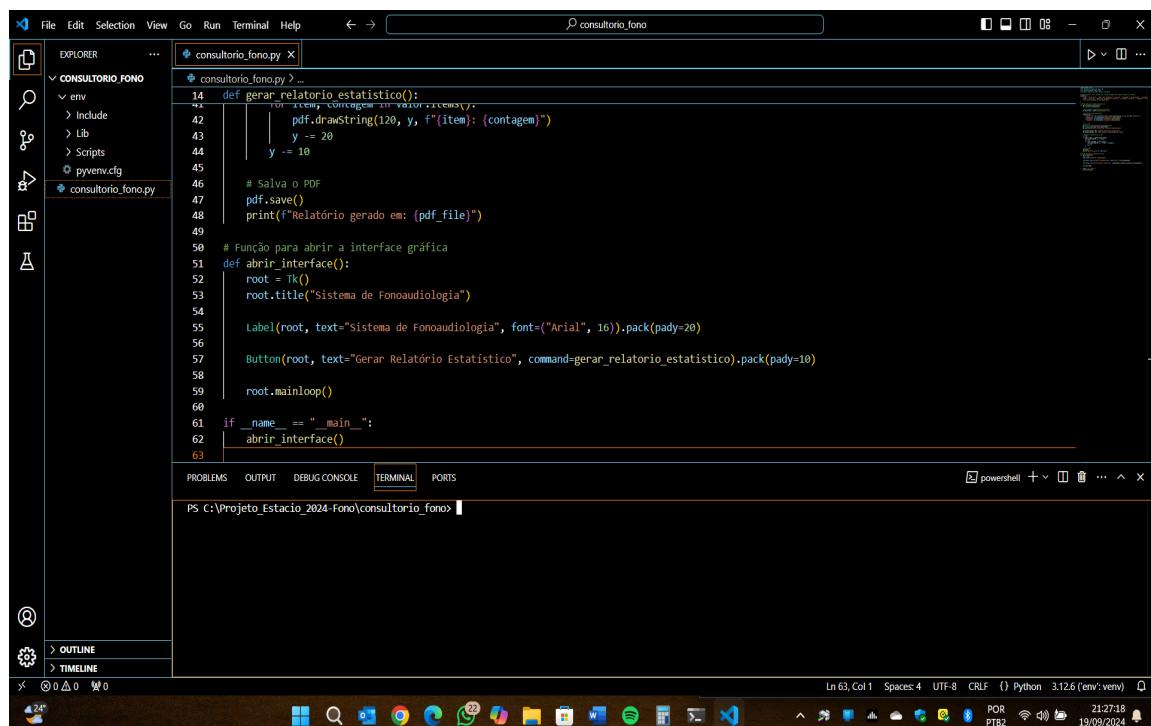
19

## Montagem no editor de texto como o Visual Studio Code

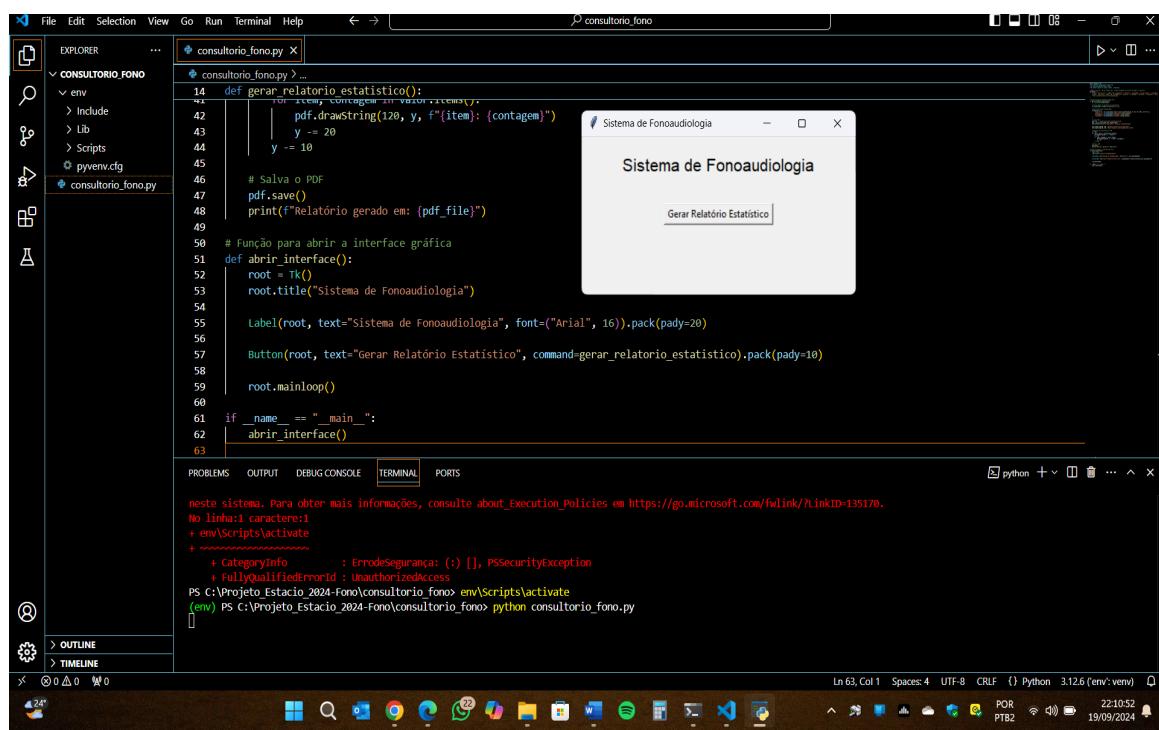
**Verificamos o Ambiente Virtual e Executamos o Sistema**

- 13.** Ativamos o ambiente virtual no VS Code:
  - 14.** Abrimos um terminal dentro do VS Code indo em Terminal > New Terminal (Novo Terminal).
  - 15.** No terminal aberto, ativamos o ambiente virtual:  
env\Scripts\activate

Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia



## Árvore inicial do Python (rascunho)



Ativação do ambiente virtual (rascunho e testes)

**16.** Ativamos o ambiente virtual, após liberarmos as restrições no Power Shell para continuar com o desenvolvimento do sistema.

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

## Criar a Pasta para Relatórios:

- 17.** No Prompt de Comando, dentro da pasta do projeto, criamos uma pasta para armazenar os relatórios:  
mkdir relatorios

The screenshot shows the Visual Studio Code interface. In the Explorer sidebar, a new folder named 'relatorios' is being created under the 'CONSULTORIO\_FONO' project. The terminal at the bottom shows the command 'mkdir relatorios' being run in the directory 'C:\Projeto\_Estacio\_2024-Fono\consultorio\_fono'. A green circle with the number '21' is in the top right corner.

The screenshot shows the Visual Studio Code interface again. The terminal at the bottom shows the command 'python consultorio\_fono.py' being run in the directory 'C:\Projeto\_Estacio\_2024-Fono\consultorio\_fono'. A Python application window titled 'Sistema de Fonoaudiologia' is open, displaying the message 'Gerar Relatório Estatístico'. The status bar at the bottom right shows the Python extension version '3.12.6 (env: venv)'.

## Testes - Prompt de comando

- 18.** No Prompt de Comando, estando na pasta do projeto, executei o seguinte comando:

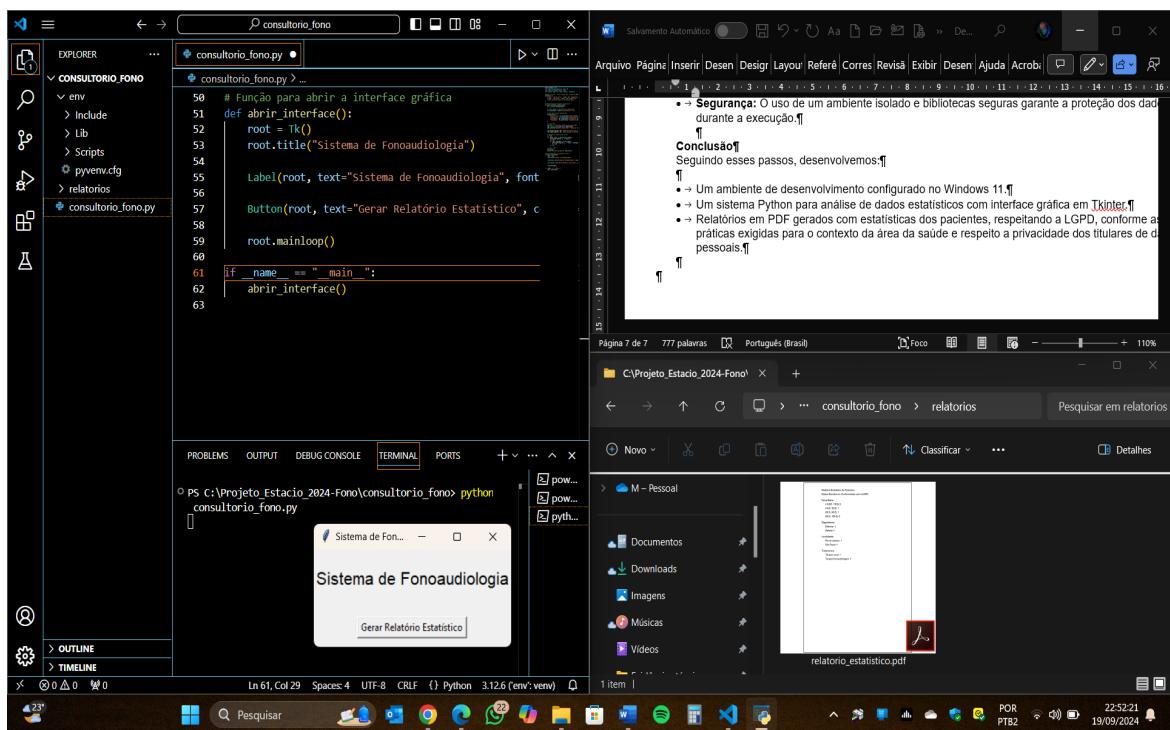
```
python consultorio_fono.py
```

- 19.** Este recurso criou uma janela com a interface gráfica do Tkinter. Possibilita clicar no botão para gerar o relatório.

### Verificar a Saída:

22

- 20.** Após clicar no botão "Gerar Relatório Estatístico", o sistema criará um arquivo PDF na pasta relatórios com as estatísticas dos pacientes consentidos.



## Geração e Verificação do Relatório

### Abrir o Relatório:

- 21.** Conferimos até a pasta relatórios criados no projeto.  
**22.** Foi aberto o arquivo relatorio\_estatistico.pdf para visualizar o relatório com as estatísticas dos pacientes.

### Conteúdo do Relatório:

- 23.** O relatório conterá estatísticas anonimizadas, como faixa etária, diagnósticos, localidades e tratamentos dos pacientes que deram consentimento, respeitando a LGPD.

## Garantindo a Conformidade com a LGPD

Este sistema respeita a LGPD das seguintes formas:

- **Consentimento:** Apenas os dados dos pacientes que deram consentimento são usados na geração de relatórios.
- **Anonimização:** Os relatórios são gerados com dados agregados, sem exposição de informações pessoais identificáveis (como nomes ou informações sensíveis).
- **Minimização de Dados:** Somente dados necessários para análise estatística são incluídos.
- **Segurança:** O uso de um ambiente isolado e bibliotecas seguras garante a proteção dos dados durante a execução.

23

## Conclusão

Seguindo esses passos, desenvolvemos:

- Um ambiente de desenvolvimento configurado no Windows 11.
- Um sistema Python para análise de dados estatísticos com interface gráfica em Tkinter.
- Relatórios em PDF gerados com estatísticas dos pacientes, respeitando a LGPD, conforme as práticas exigidas para o contexto da área da saúde e respeito à privacidade dos titulares de dados pessoais.

## DADOS DIRETAMENTE NO SISTEMA

Para permitir que o fonoaudiólogo insira os dados diretamente no sistema, podemos criar uma interface gráfica usando o **Tkinter** que permita a entrada de informações sobre os pacientes. Isso vai simplificar a coleta de dados e a geração de relatórios.

### Passo 1

#### Inserir os Dados dos Pacientes Usando Tkinter

1. **Atualizamos o arquivo `consultorio_fono.py`** para incluir uma interface de inserção de dados.
2. **O fonoaudiólogo poderá inserir os dados** como nome, idade, diagnóstico, localidade e consentimento por meio da interface gráfica.
3. **Os dados inseridos serão salvos em um arquivo CSV** ou diretamente processados para gerar relatórios.

**Código atualizado para permitir que o fonoaudiólogo insira dados usando a interface gráfica.**

```
import pandas as pd
from reportlab.lib.pagesizes import A4
```

## Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```

from reportlab.pdfgen import canvas
import tkinter as tk
from tkinter import messagebox
import csv

# Função para salvar os dados inseridos no CSV
def salvar_dados(nome, idade, diagnostico, localidade, tratamento, consentimento):
    try:
        with open('pacientes.csv', 'a', newline='', encoding='utf-8') as file:
            writer = csv.writer(file)
            writer.writerow([nome, idade, diagnostico, localidade, tratamento, consentimento])
        messagebox.showinfo("Sucesso", "Dados do paciente salvos com sucesso!")
    except Exception as e:
        messagebox.showerror("Erro", f"Erro ao salvar os dados: {e}")

# Função para gerar relatório estatístico
def gerar_relatorio_estatistico():
    try:
        df = pd.read_csv('pacientes.csv', names=['nome', 'idade', 'diagnostico', 'localidade', 'tratamento', 'consentimento'])
        df_consistentes = df[df['consentimento'] == '1']

        estatisticas = {
            'Faixa Etária': df_consistentes['idade'].value_counts(bins=[0, 18, 30, 50, 100], sort=False),
            'Diagnósticos': df_consistentes['diagnostico'].value_counts(),
            'Localidades': df_consistentes['localidade'].value_counts(),
            'Tratamentos': df_consistentes['tratamento'].value_counts()
        }

        pdf_file = "relatorios/relatorio_estatistico.pdf"
        pdf = canvas.Canvas(pdf_file, pagesize=A4)
        pdf.setTitle("Relatório Estatístico - Pacientes Fonoaudiologia")

        pdf.drawString(100, 800, "Relatório Estatístico de Pacientes")
        pdf.drawString(100, 780, "Dados Gerados em Conformidade com a LGPD")

        y = 750
        for chave, valor in estatisticas.items():
            pdf.drawString(100, y, f"{chave}:")
            y -= 20
            for item, contagem in valor.items():
                pdf.drawString(120, y, f"{item}: {contagem}")
                y -= 20
            y -= 10

        pdf.save()
        messagebox.showinfo("Sucesso", f"Relatório gerado em: {pdf_file}")
    except Exception as e:
        messagebox.showerror("Erro", f"Erro ao gerar o relatório: {e}")

# Função para abrir a janela de inserção de dados
def abrir_janela():
    janela = tk.Tk()
    janela.title("Inserir Dados do Paciente")

    tk.Label(janela, text="Nome").grid(row=0)
    tk.Label(janela, text="Idade").grid(row=1)
    tk.Label(janela, text="Diagnóstico").grid(row=2)
    tk.Label(janela, text="Localidade").grid(row=3)
    tk.Label(janela, text="Tratamento").grid(row=4)
    tk.Label(janela, text="Consentimento (1 = Sim, 0 = Não)").grid(row=5)

    nome_entry = tk.Entry(janela)
    idade_entry = tk.Entry(janela)
    diagnostico_entry = tk.Entry(janela)
    localidade_entry = tk.Entry(janela)
    tratamento_entry = tk.Entry(janela)
    consentimento_entry = tk.Entry(janela)

    nome_entry.grid(row=0, column=1)
    idade_entry.grid(row=1, column=1)

```

## Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
diagnostico_entry.grid(row=2, column=1)
localidade_entry.grid(row=3, column=1)
tratamento_entry.grid(row=4, column=1)
consentimento_entry.grid(row=5, column=1)

def inserir_dados():
    nome = nome_entry.get()
    idade = idade_entry.get()
    diagnostico = diagnostico_entry.get()
    localidade = localidade_entry.get()
    tratamento = tratamento_entry.get()
    consentimento = consentimento_entry.get()

    if nome and idade and diagnostico and localidade and tratamento and consentimento:
        salvar_dados(nome, idade, diagnostico, localidade, tratamento, consentimento)
        nome_entry.delete(0, tk.END)
        idade_entry.delete(0, tk.END)
        diagnostico_entry.delete(0, tk.END)
        localidade_entry.delete(0, tk.END)
        tratamento_entry.delete(0, tk.END)
        consentimento_entry.delete(0, tk.END)
    else:
        messagebox.showwarning("Atenção", "Preencha todos os campos.")

tk.Button(janela, text="Salvar Dados", command=inserir_dados).grid(row=6, column=1, pady=10)
tk.Button(janela, text="Gerar Relatório", command=gerar_relatorio_estatistico).grid(row=7, column=1, pady=10)

janela.mainloop()

# Executa a janela
abrir_janela()
```

25

### Passo 2:

#### Salvar os Dados em um Arquivo CSV

Os dados dos pacientes inseridos pelo fonoaudiólogo são salvos em um arquivo chamado pacientes.csv. Esse arquivo pode ser usado posteriormente para gerar o relatório.

### Passo 3:

#### Interface Gráfica para Inserir Dados

Ao rodar o sistema, uma janela gráfica será aberta, permitindo ao fonoaudiólogo inserir os dados dos pacientes diretamente no sistema.

- **Campos:** Nome, Idade, Diagnóstico, Localidade, Tratamento e Consentimento.
- **Botões:**
  - "Salvar Dados": Armazena os dados no arquivo CSV.
  - "Gerar Relatório": Cria um relatório PDF com as estatísticas.

**Passo 4:**

**Executar o Sistema**

**1. Abrimos o terminal e navegue até a pasta do projeto:**

```
cd C:\Projeto_Estacio_2024-Fono\consultorio_fono
```

**2. Ativamos o ambiente virtual:**

```
env\Scripts\activate
```

26

**3. Executamos o sistema:**

```
python consultorio_fono.py
```

Agora, o fonoaudiólogo terá uma interface fácil de usar para inserir os dados e gerar relatórios.

**Explicação**

- **Tkinter** é usado para criar uma interface gráfica simples.
- Os dados inseridos são salvos em um arquivo CSV para facilitar o gerenciamento.
- O sistema também gera um relatório PDF com as estatísticas dos pacientes que deram consentimento, conforme requerido pela LGPD.

**CONCLUSÃO:**

no sistema atualizado, o arquivo **pacientes.csv** será gerado automaticamente assim que o fonoaudiólogo inserir os dados dos pacientes e clicar no botão "Salvar" na interface gráfica. Se o arquivo **pacientes.csv** já existir, ele será atualizado com as novas informações; caso contrário, o arquivo será criado na mesma pasta do projeto.

**Como funciona:**

**Salvar os dados dos pacientes:**

Quando o fonoaudiólogo insere os dados e clica no botão "Salvar", o sistema verifica se o arquivo **pacientes.csv** existe. Se não existir, ele cria o arquivo e grava os dados.

**Atualizar dados:**

Se o arquivo já existe, os novos dados são adicionados ao final do arquivo sem apagar os dados anteriores.

**Explicação mais detalhada do que acontece no código:**

**Trecho relevante para salvar os dados:**

```
python
Copiar código
# Função que salva os dados no arquivo pacientes.csv
def salvar_dados(nome, idade, diagnostico, localidade, tratamento, consentimento):
    # Dicionário com os dados do paciente
    paciente = {
        'nome': nome,
        'idade': idade,
        'diagnostico': diagnostico,
        'localidade': localidade,
        'tratamento': tratamento,
        'consentimento': consentimento
    }

    # Verifica se o arquivo pacientes.csv já existe
    try:
        df = pd.read_csv('pacientes.csv') # Carrega o arquivo se existir
        df = df.append(paciente, ignore_index=True) # Adiciona o novo paciente
    except FileNotFoundError:
        df = pd.DataFrame([paciente]) # Cria o arquivo se ele não existir

    # Salva os dados no arquivo CSV
    df.to_csv('pacientes.csv', index=False)
    print("Dados salvos com sucesso!")
```

27

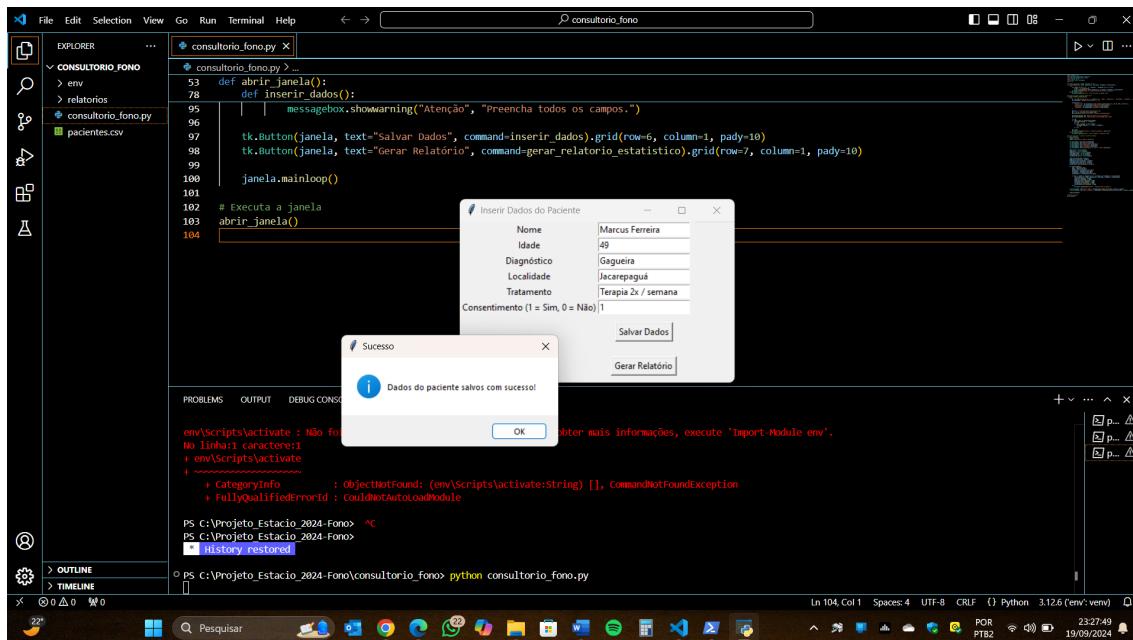
**Explicação:**

- Tentativa de leitura:** O código tenta carregar o arquivo **pacientes.csv** usando `pd.read_csv('pacientes.csv')`. Se o arquivo existir, ele será carregado e os novos dados serão adicionados ao arquivo.
- Criação do arquivo:** Se o arquivo não existir (ocorre um **FileNotFoundException**), ele cria um novo DataFrame com os dados do paciente e salva esse arquivo como **pacientes.csv**.
- Salvamento:** O comando `df.to_csv('pacientes.csv', index=False)` garante que os dados sejam gravados no arquivo CSV.

**Fluxo para o fonoaudiólogo:**

- O fonoaudiólogo abre o sistema.
- Insere os dados dos pacientes (nome, idade, diagnóstico, localidade, tratamento, consentimento).
- Clica no botão "Salvar".
- Os dados são armazenados no arquivo **pacientes.csv** na pasta do projeto.

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia



28

## O código atualizado.

includindo uma mensagem sobre a LGPD e um aviso para garantir que os pacientes concordem com o tratamento de seus dados:

```
python
Copiar código
import pandas as pd
from reportlab.lib.pagesizes import A4
from reportlab.pdfgen import canvas
import tkinter as tk
from tkinter import messagebox
import csv

# Função para salvar os dados inseridos no CSV
def salvar_dados(nome, idade, diagnostico, localidade, tratamento, consentimento):
    try:
        with open('pacientes.csv', 'a', newline='', encoding='utf-8') as file:
            writer = csv.writer(file)
            writer.writerow([nome, idade, diagnostico, localidade, tratamento, consentimento])
        messagebox.showinfo("Sucesso", "Dados do paciente salvos com sucesso!")
    except Exception as e:
        messagebox.showerror("Erro", f"Erro ao salvar os dados: {e}")

# Função para gerar relatório estatístico
def gerar_relatorio_estatistico():
    try:
        df = pd.read_csv('pacientes.csv', names=['nome', 'idade', 'diagnostico', 'localidade', 'tratamento', 'consentimento'])
        df_considentes = df[df['consentimento'] == '1']

        estatisticas = {
            'Faixa Etária': df_considentes['idade'].value_counts(bins=[0, 18, 30, 50, 100], sort=False),
            'Diagnósticos': df_considentes['diagnostico'].value_counts(),
            'Localidades': df_considentes['localidade'].value_counts(),
            'Tratamentos': df_considentes['tratamento'].value_counts()
        }

        pdf_file = "relatorios/relatorio_estatistico.pdf"
        pdf = canvas.Canvas(pdf_file, pagesize=A4)
        pdf.setTitle("Relatório Estatístico - Pacientes Fonoaudiologia")

        pdf.drawString(100, 800, "Relatório Estatístico de Pacientes")
```

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
pdf.drawString(100, 780, "Dados Gerados em Conformidade com a LGPD")

y = 750
for chave, valor in estatisticas.items():
    pdf.drawString(100, y, f'{chave}:')
    y -= 20
for item, contagem in valor.items():
    pdf.drawString(120, y, f'{item}: {contagem}')
    y -= 20
y -= 10

pdf.save()
messagebox.showinfo("Sucesso", f"Relatório gerado em: {pdf_file}")
except Exception as e:
    messagebox.showerror("Erro", f"Erro ao gerar o relatório: {e}")

# Função para abrir a janela de inserção de dados
def abrir_janela():
    janela = tk.Tk()
    janela.title("Inserir Dados do Paciente")

    # Mensagem sobre LGPD
    messagebox.showinfo("Aviso LGPD", "Os dados coletados serão utilizados para fins de tratamento e relatórios estatísticos. Você consente com o armazenamento e uso de seus dados?")

    tk.Label(janela, text="Nome").grid(row=0)
    tk.Label(janela, text="Idade").grid(row=1)
    tk.Label(janela, text="Diagnóstico").grid(row=2)
    tk.Label(janela, text="Localidade").grid(row=3)
    tk.Label(janela, text="Tratamento").grid(row=4)
    tk.Label(janela, text="Consentimento (1 = Sim, 0 = Não)").grid(row=5)

    nome_entry = tk.Entry(janela)
    idade_entry = tk.Entry(janela)
    diagnostico_entry = tk.Entry(janela)
    localidade_entry = tk.Entry(janela)
    tratamento_entry = tk.Entry(janela)
    consentimento_entry = tk.Entry(janela)

    nome_entry.grid(row=0, column=1)
    idade_entry.grid(row=1, column=1)
    diagnostico_entry.grid(row=2, column=1)
    localidade_entry.grid(row=3, column=1)
    tratamento_entry.grid(row=4, column=1)
    consentimento_entry.grid(row=5, column=1)

def inserir_dados():
    nome = nome_entry.get()
    idade = idade_entry.get()
    diagnostico = diagnostico_entry.get()
    localidade = localidade_entry.get()
    tratamento = tratamento_entry.get()
    consentimento = consentimento_entry.get()

    if nome and idade and diagnostico and localidade and tratamento and consentimento:
        salvar_dados(nome, idade, diagnostico, localidade, tratamento, consentimento)
        nome_entry.delete(0, tk.END)
        idade_entry.delete(0, tk.END)
        diagnostico_entry.delete(0, tk.END)
        localidade_entry.delete(0, tk.END)
        tratamento_entry.delete(0, tk.END)
        consentimento_entry.delete(0, tk.END)
    else:
        messagebox.showwarning("Atenção", "Preencha todos os campos.")

tk.Button(janela, text="Salvar Dados", command=inserir_dados).grid(row=6, column=1, pady=10)
tk.Button(janela, text="Gerar Relatório", command=gerar_relatorio_estatistico).grid(row=7, column=1, pady=10)

janela.mainloop()
```

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
# Executa a janela  
abrir_janela()
```

## Alterações Implementadas:

### Aviso LGPD:

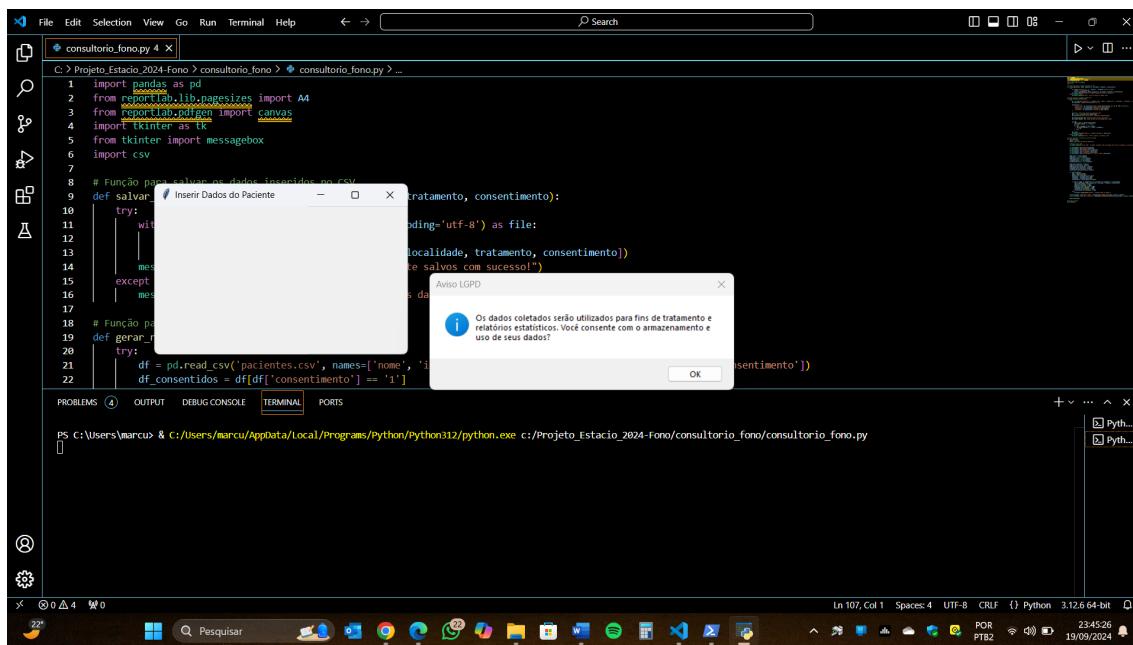
Adicionei uma mensagem inicial informando sobre a coleta de dados e a necessidade de consentimento.

### Confirmação do Consentimento:

30

A estrutura do código já usa o campo de consentimento, que deve ser preenchido pelo usuário no formulário.

Essa implementação visa garantir que o sistema respeite as diretrizes da LGPD, informando os usuários sobre como seus dados serão utilizados.



## Alerta de Privacidade

### UPDATE 2.0

#### Atalho para área de trabalho do usuário

Para que o usuário final tenha um atalho direto que execute o sistema gráfico do projeto em qualquer sistema operacional, especialmente no Windows 11, desenvolvemos um atalho que abra o ambiente gráfico diretamente, sem a necessidade de utilizar o terminal.

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

Etapas detalhadas para criar esse atalho no Windows:

## Criar um arquivo .bat (Windows)

No Windows, você pode usar um arquivo .bat para automatizar a execução do seu script Python. Siga os passos abaixo:

### Abra o Bloco de Notas ou Visual Studio Code

Utilizamos o Visual Studio Code - Inserimos o seguinte conteúdo no arquivo, que executará o Python e o seu script diretamente:

```
bat
@echo off
cd C:\Users\marcu\AppData\Local\Programs\Python\Python312
python.exe c:/Projeto_Estacio_2024-Fono/Projeto_Fonart/sistema_fonart.py
pause
```

Salve esse arquivo com a extensão .bat, por exemplo: iniciar\_sistema\_fonart.bat.

## Colocar um ícone no arquivo .bat

Transformaremos o arquivo .bat em um atalho com um ícone mais amigável. Para isso:

Clique com o botão direito no arquivo .bat que você criou e escolha "Criar atalho".

Clique com o botão direito no atalho criado e vá em "Propriedades".

Na aba "Atalho", clique em "Alterar ícone...".

Escolha um ícone desejado (você pode usar o logotipo do sistema localizado em

**Em nosso exemplo para o projeto de extensão universitário da Estácio de Sá, utilizamos as pastas no NOTEBOOK da Empresária, da empresa FONART**

C:\Projeto\_Estacio\_2024-Fono\Projeto\_Fonart\assets\fonart4.png, convertendo para .ico antes).

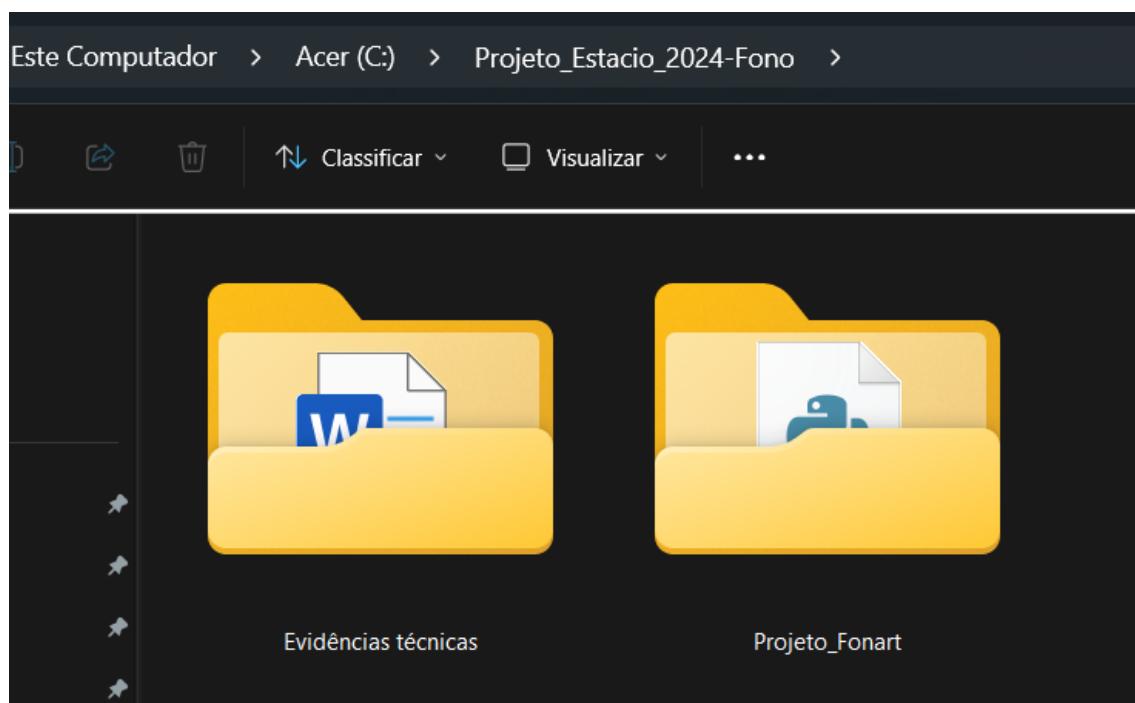
(Utilizamos qualquer site gratuito para conversão das extensões e gravamos na pasta "assets")

## Executar o sistema com um clique

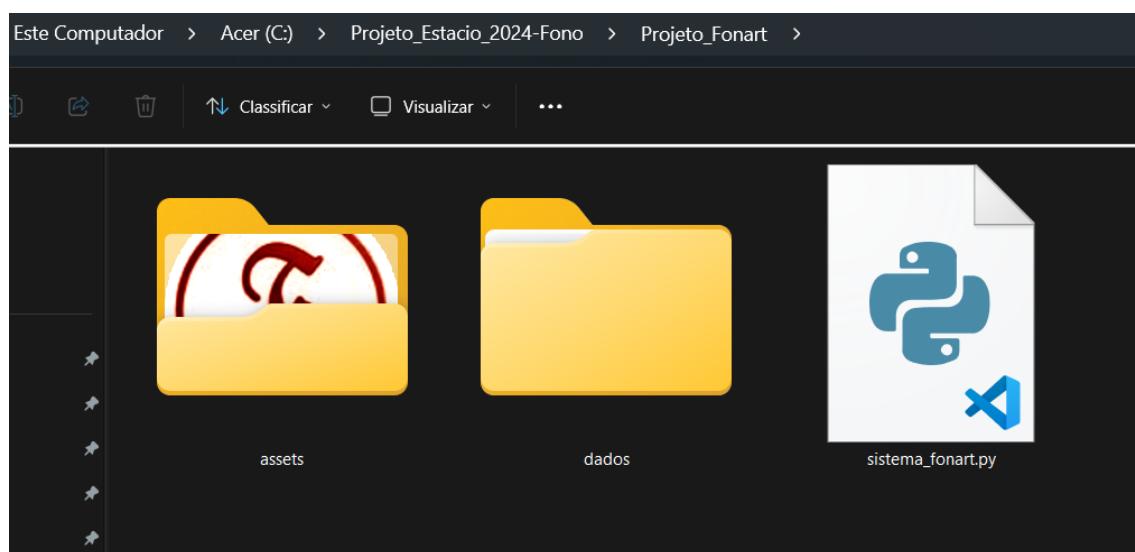
Agora, o usuário poderá clicar diretamente no atalho, que abrirá a interface gráfica do sistema sem a necessidade de lidar com o terminal ou comandos. Isso torna o processo muito mais simples e acessível.

## Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

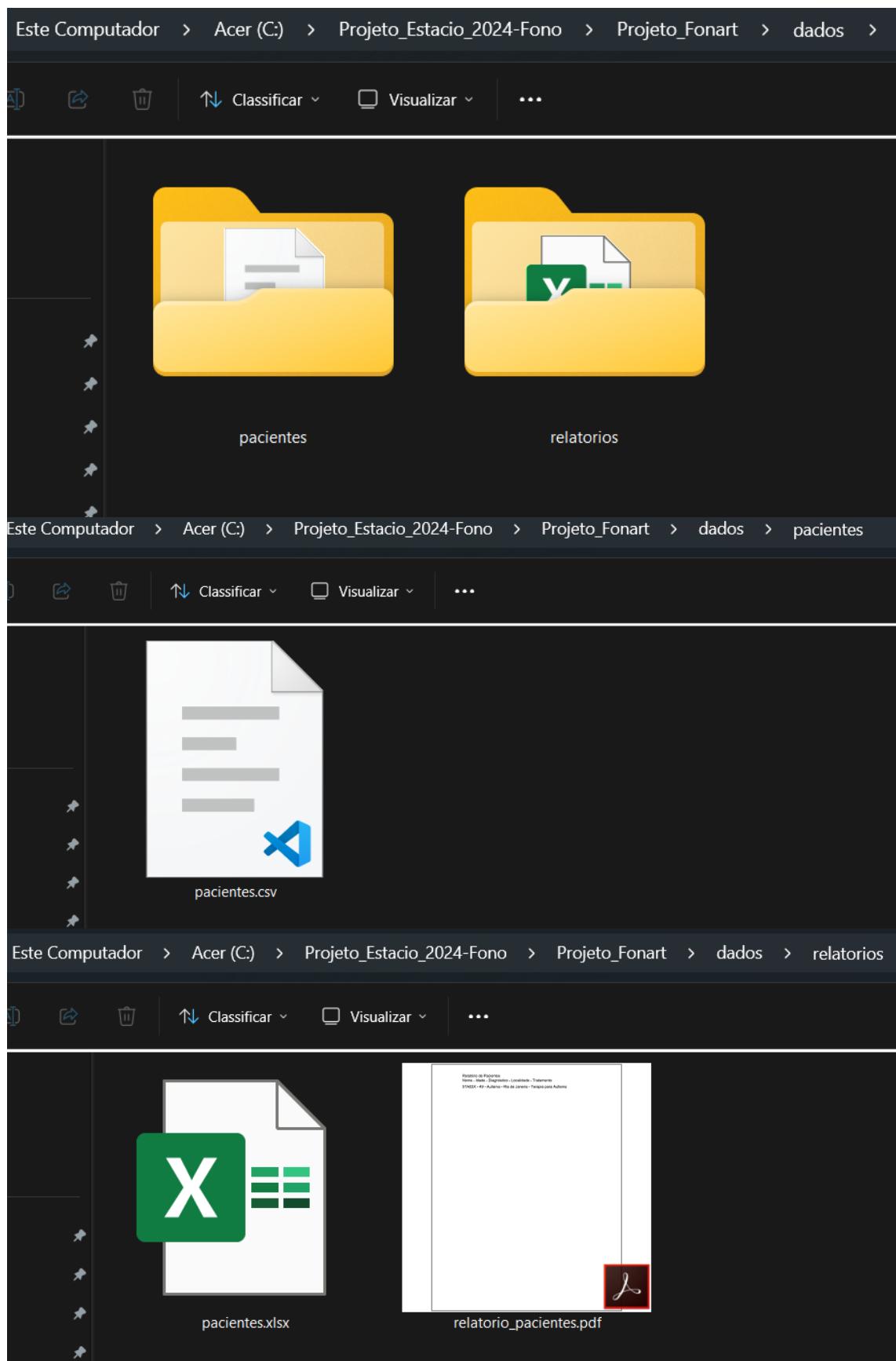
Evidências das pastas:



Pastas iniciais (rascunho)



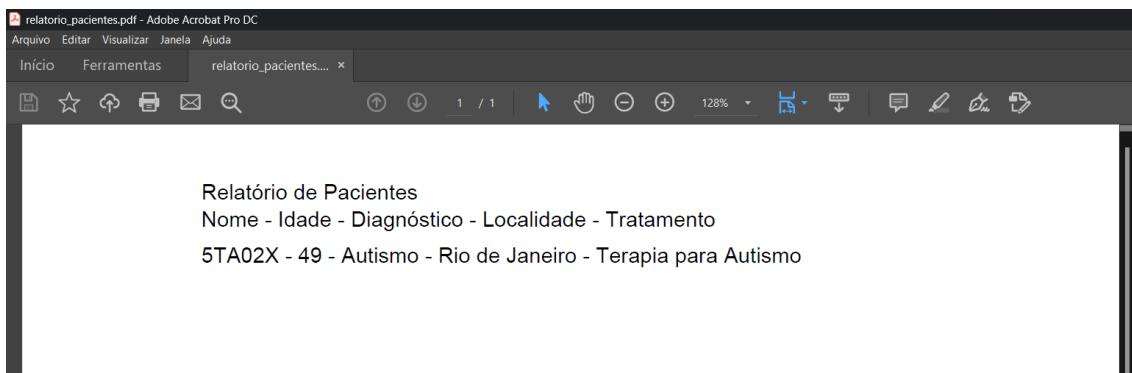
## Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia



# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

Pasta iniciais do projeto com os relatórios gerados pelo Python

## Relatórios gerados automaticamente – PDF e Gráficos



34

**Relatórios gerados automaticamente do PDF com anonimização dos nomes, visando respeitar a privacidade desde a concepção *Privacy by Design* - LGP**

## SISTEMA E MELHORIAS:

```
import subprocess
import sys

# Função para instalar pacotes se não estiverem instalados
def instalar_pacotes(pacotes):
    for pacote in pacotes:
        try:
            __import__(pacote)
        except ImportError:
            print(f"Instalando {pacote}...")
            subprocess.check_call([sys.executable, "-m", "pip", "install", pacote])

# Lista de pacotes necessários
pacotes_necessarios = ['pandas', 'reportlab', 'xlsxwriter']

# Chama a função para verificar e instalar pacotes
instalar_pacotes(pacotes_necessarios)

import pandas as pd
from reportlab.lib.pagesizes import A4
from reportlab.pdfgen import canvas
import tkinter as tk
from tkinter import messagebox
import csv
import os

# Função para salvar os dados inseridos no CSV
def salvar_dados(nome, idade, diagnostico, localidade, tratamento, consentimento):
    try:
        with open('pacientes.csv', 'a', newline='', encoding='utf-8') as file:
            writer = csv.writer(file)
            writer.writerow([nome, idade, diagnostico, localidade, tratamento, consentimento])
        messagebox.showinfo("Sucesso", "Dados do paciente salvos com sucesso!")
    except Exception as e:
        messagebox.showerror("Erro", f"Erro ao salvar os dados: {e}")

# Função para gerar relatório estatístico
def gerar_relatorio_estatistico():
```

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
try:  
    if not os.path.exists('relatorios'):  
        os.makedirs('relatorios')  
  
    df = pd.read_csv('pacientes.csv', names=['nome', 'idade', 'diagnostico', 'localidade', 'tratamento', 'consentimento'])  
    df['idade'] = pd.to_numeric(df['idade'], errors='coerce')  
    df_consentidos = df[df['consentimento'] == '1']  
  
    estatisticas = {  
        'Faixa Etária': df_consentidos['idade'].value_counts(bins=[0, 18, 30, 50, 100], sort=False),  
        'Diagnósticos': df_consentidos['diagnostico'].value_counts(),  
        'Localidades': df_consentidos['localidade'].value_counts(),  
        'Tratamentos': df_consentidos['tratamento'].value_counts()  
    }  
  
    pdf_file = "relatorios/relatorio_estatistico.pdf"  
    pdf = canvas.Canvas(pdf_file, pagesize=A4)  
    pdf.setTitle("Relatório Estatístico - Pacientes Fonoaudiologia")  
  
    pdf.drawString(100, 800, "Relatório Estatístico de Pacientes")  
    pdf.drawString(100, 780, "Dados Gerados em Conformidade com a LGPD")  
  
    y = 750  
    for chave, valor in estatisticas.items():  
        pdf.drawString(100, y, f"{chave}:")  
        y -= 20  
        for item, contagem in valor.items():  
            pdf.drawString(120, y, f"{item}: {contagem}")  
            y -= 20  
        y -= 10  
  
    pdf.save()  
    messagebox.showinfo("Sucesso", f"Relatório gerado em: {pdf_file}")  
except Exception as e:  
    messagebox.showerror("Erro", f"Erro ao gerar o relatório: {e}")  
  
# Função para gerar arquivo Excel com gráficos  
def gerar_excel_com_graficos():  
    try:  
        if not os.path.isfile('pacientes.csv'):  
            messagebox.showerror("Erro", "O arquivo 'pacientes.csv' não foi encontrado.")  
            return  
  
        df = pd.read_csv('pacientes.csv', names=['nome', 'idade', 'diagnostico', 'localidade', 'tratamento', 'consentimento'])  
        df['idade'] = pd.to_numeric(df['idade'], errors='coerce')  
        df_consentidos = df[df['consentimento'] == '1']  
  
        os.makedirs('relatorios', exist_ok=True)  
        excel_file = 'relatorios/relatorio_estatistico.xlsx'  
  
        with pd.ExcelWriter(excel_file, engine='xlsxwriter') as writer:  
            df_consentidos.to_excel(writer, sheet_name='Dados', index=False)  
  
            # Gráfico de faixa etária  
            faixa_etaria = pd.cut(df_consentidos['idade'], bins=[0, 18, 30, 50, 100])  
            faixa_etaria_count = faixa_etaria.value_counts()  
  
            worksheet = writer.sheets['Dados']  
            worksheet.write(len(df_consentidos) + 2, 0, "Faixa Etária")  
            worksheet.write(len(df_consentidos) + 3, 0, "Contagem")  
  
            for index, (faixa, count) in enumerate(faixa_etaria_count.items()):  
                worksheet.write(len(df_consentidos) + 4 + index, 0, str(faixa))  
                worksheet.write(len(df_consentidos) + 4 + index, 1, count)  
            # Criando gráfico  
            chart = writer.book.add_chart({'type': 'column'})  
            chart.add_series({  
                'name': 'Contagem por Faixa Etária',  
                'categories': f'Dados!$A${len(df_consentidos) + 4}:$A${len(df_consentidos) + 4 + len(faixa_etaria_count) - 1}',  
                'values': f'Dados!$B${len(df_consentidos) + 4}:$B${len(df_consentidos) + 4 + len(faixa_etaria_count) - 1}',  
            })  
    
```

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
})
worksheet.insert_chart('D2', chart)

messagebox.showinfo("Sucesso", f"Arquivo Excel gerado em: {excel_file}")
except Exception as e:
    messagebox.showerror("Erro", f"Erro ao gerar o arquivo Excel: {e}")

# Função para realizar a pesquisa universal
def pesquisar_paciente(termo):
    try:
        df = pd.read_csv('pacientes.csv', names=['nome', 'idade', 'diagnostico', 'localidade', 'tratamento', 'consentimento'])
        resultados = df[df.apply(lambda row: row.astype(str).str.contains(termo, case=False).any(), axis=1)]
    except Exception as e:
        messagebox.showerror("Erro", f"Erro ao gerar o arquivo Excel: {e}")

    if not resultados.empty:
        resultado_texto = "\n".join([f'{row["nome"]}', f'{row["idade"]}', f'{row["diagnostico"]}', f'{row["localidade"]}', f'{row["tratamento"]}' for idx, row in resultados.iterrows()])
        messagebox.showinfo("Resultados da Pesquisa", resultado_texto)
    else:
        messagebox.showinfo("Resultados da Pesquisa", "Nenhum paciente encontrado.")
    except Exception as e:
        messagebox.showerror("Erro", f"Erro na pesquisa: {e}")

# Função para abrir a janela de inserção de dados
def abrir_janela():
    if messagebox.askyesno("Consentimento", "Você está ciente de que deve proteger a privacidade do paciente de acordo com a LGPD?"):
        janela = tk.Tk()
        janela.title("Sistema de Pacientes - Fonoaudiologia")

        tk.Label(janela, text="Nome").grid(row=0)
        tk.Label(janela, text="Idade").grid(row=1)
        tk.Label(janela, text="Diagnóstico").grid(row=2)
        tk.Label(janela, text="Localidade").grid(row=3)
        tk.Label(janela, text="Tratamento").grid(row=4)
        tk.Label(janela, text="Consentimento (1 = Sim, 0 = Não)").grid(row=5)

        nome_entry = tk.Entry(janela)
        idade_entry = tk.Entry(janela)
        diagnostico_entry = tk.Entry(janela)
        localidade_entry = tk.Entry(janela)
        tratamento_entry = tk.Entry(janela)
        consentimento_entry = tk.Entry(janela)

        nome_entry.grid(row=0, column=1)
        idade_entry.grid(row=1, column=1)
        diagnostico_entry.grid(row=2, column=1)
        localidade_entry.grid(row=3, column=1)
        tratamento_entry.grid(row=4, column=1)
        consentimento_entry.grid(row=5, column=1)

        def inserir_dados():
            nome = nome_entry.get()
            idade = idade_entry.get()
            diagnostico = diagnostico_entry.get()
            localidade = localidade_entry.get()
            tratamento = tratamento_entry.get()
            consentimento = consentimento_entry.get()

            if nome and idade and diagnostico and localidade and tratamento and consentimento:
                salvar_dados(nome, idade, diagnostico, localidade, tratamento, consentimento)
                nome_entry.delete(0, tk.END)
                idade_entry.delete(0, tk.END)
                diagnostico_entry.delete(0, tk.END)
                localidade_entry.delete(0, tk.END)
                tratamento_entry.delete(0, tk.END)
                consentimento_entry.delete(0, tk.END)
            else:
                messagebox.showwarning("Atenção", "Preencha todos os campos.")

        tk.Button(janela, text="Salvar Dados", command=inserir_dados).grid(row=6, column=1, pady=10)
```

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
tk.Button(janela, text="Gerar Relatório", command=gerar_relatorio_estatistico).grid(row=7, column=1, pady=10)
tk.Button(janela, text="Gerar Excel", command=gerar_excel_com_graficos).grid(row=8, column=1, pady=10)

tk.Label(janela, text="Pesquisa Universal").grid(row=9)
termo_pesquisa_entry = tk.Entry(janela)
termo_pesquisa_entry.grid(row=10, column=1)
    tk.Button(janela, text="Pesquisar", command=lambda: pesquisar_paciente(termo_pesquisa_entry.get())).grid(row=11, column=1, pady=10)

janela.mainloop()

# Chamando a função para abrir a janela
abrir_janela()
```

37

## PYTHON AVANÇADO - PARA O CONSULTÓRIO FONART.

### VERSÃO 2.0

Projeto de Extensão Universitária - Estácio 2024 - Marcus Cesar Pereira Ferreira - Matrícula - 202102340918@alunos.estacio.br

Resumo do Projeto: Este sistema integra várias tecnologias e foi desenvolvido para a Extensão Universitária na Estácio 2024. Localmente, utiliza Python para integração com o Firebase do Google Cloud e o Cloud Firestore, armazenando dados de pacientes de fonoaudiologia.

A interface em Tkinter permite a coleta de informações como nome, idade e diagnóstico. O sistema também oferece funcionalidades como geração de relatórios em PDF e gráficos de análise, além de sincronizar automaticamente os dados locais e na nuvem. O formulário utiliza HTML e JavaScript para encaminhar os dados ao Cloud Firestore, que os valida e armazena.

O objetivo do formulário WEB <https://fonart-8cd50.web.app/> é coletar dados estatísticos de pacientes em tratamento fonoaudiológico. A coleta de informações como idade, diagnóstico, localidade e tratamento é essencial para entender as necessidades dos pacientes e aprimorar a atuação da equipe. O consentimento do paciente é obrigatório para a utilização dos dados, que são tratados com a máxima confidencialidade, visando à conformidade com a LGPD.

```
"""
# Importações necessárias para o sistema
import pandas as pd
import csv
import firebase_admin
from firebase_admin import credentials, firestore
import tkinter as tk
from tkinter import ttk, messagebox
from PIL import Image, ImageTk
import os
import threading
import matplotlib.pyplot as plt # Importação do matplotlib para gráficos
from reportlab.pdfgen import canvas
from reportlab.lib.pagesizes import letter
import logging
import random
import string
```

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
# Configuração do logging
log_path = r'C:\Projeto_Estacio_2024-Fono\Projeto_Fonart\logs\system.log'
logging.basicConfig(filename=log_path,
                    level=logging.DEBUG,
                    format='%(asctime)s - %(levelname)s - %(message)s')

# Inicialização do Firebase
def inicializar_firebase():
    """Inicia a conexão com o Firebase e retorna a instância do Firestore."""
    try:
        # Verifica se o Firebase já foi inicializado
        if not firebase_admin._apps: # Verifica se não há aplicativos já inicializados
            cred = credentials.Certificate("C:\\Projeto_Estacio_2024-Fono\\Projeto_Fonart\\dados\\Firebase\\fonart-8cd50.firebaseio-adminsdk-8qau1-3dd6e0939e.json")
            firebase_admin.initialize_app(cred)
        db = firestore.client() # Conexão com o Firestore
        return db
    except Exception as e:
        messagebox.showerror("Erro", f"Falha ao inicializar o Firebase: {e}")
        logging.error(f"Erro ao inicializar o Firebase: {e}") # Log de erro
        return None

# Função para validar os dados
def validar_dados(nome, idade, diagnostico, localidade, tratamento, mes, consentimento):
    """Valida os dados de entrada antes de salvar."""
    if not (nome and idade and diagnostico and localidade and tratamento and mes):
        messagebox.showwarning("Aviso", "Todos os campos devem ser preenchidos!")
        return False
    if not consentimento:
        messagebox.showwarning("Aviso", "É necessário o consentimento para uso de dados!")
        return False
    try:
        if not isinstance(int(idade), int) or int(idade) < 0:
            messagebox.showwarning("Aviso", "A idade deve ser um número válido!")
            return False
    except ValueError:
        messagebox.showwarning("Aviso", "A idade deve ser um número válido!")
        return False
    return True

# Função para salvar dados no Firebase
def salvar_dados_firebase(dados_paciente, status_label):
    try:
        db = inicializar_firebase() # Inicializa o Firebase e conecta ao Firestore
        if db:
            db.collection('pacientes').add(dados_paciente) # Adiciona os dados à coleção 'pacientes'
            print("Dados salvos com sucesso no Firebase!")
            logging.info(f"Dados salvos no Firebase: {dados_paciente}") # Log de dados salvos
            status_label.config(text="Dados salvos com sucesso!", fg="green") # Atualiza o label de status
    except Exception as e:
        print(f"Erro ao salvar dados: {e}")
        logging.error(f"Erro ao salvar dados no Firebase: {e}") # Log de erro
        status_label.config(text="Erro ao salvar dados!", fg="red") # Atualiza o label de status

# Função para salvar os dados no CSV local
def salvar_dados_csv(dados_paciente):
    """Salva os dados em um arquivo CSV chamado pacientes.csv."""
    arquivo_csv = 'C:\\Projeto_Estacio_2024-Fono\\Projeto_Fonart\\dados\\pacientes\\pacientes.csv'
    # Verifica se o arquivo já existe
    arquivo_existe = os.path.isfile(arquivo_csv)
    # Define o cabeçalho e os dados a serem salvos
    campos = ['nome', 'idade', 'diagnostico', 'localidade', 'tratamento', 'mes', 'consentimento']
    try:
        with open(arquivo_csv, mode='a', newline='', encoding='utf-8') as arquivo:
            escritor_csv = csv.DictWriter(arquivo, fieldnames=campos)
            if not arquivo_existe:
                escritor_csv.writeheader() # Se o arquivo não existir, escreve o cabeçalho
            # Escreve os dados do paciente no CSV
            escritor_csv.writerow(dados_paciente)
            logging.info(f"Dados salvos no CSV: {dados_paciente}") # Registro de dados salvos
            print(f"Dados salvos com sucesso no arquivo {arquivo_csv}!")
    except Exception as e:
        logging.error(f"Erro ao salvar dados no CSV: {e}") # Registro de erro
```

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
print(f"Erro ao salvar dados no CSV: {e}")
# Função para salvar os dados após validação
def salvar_dados(db, nome, idade, diagnostico, localidade, tratamento, mes, consentimento, status_label):
    """Salva os dados se forem válidos."""
    if validar_dados(nome, idade, diagnostico, localidade, tratamento, mes, consentimento):
        dados_paciente = {
            "nome": nome,
            "idade": idade,
            "diagnostico": diagnostico,
            "localidade": localidade,
            "tratamento": tratamento,
            "mes": mes,
            "consentimento": consentimento
        }
        print(f"Salvando dados: {dados_paciente}") # Log detalhado
        salvar_dados_firebase(dados_paciente, status_label) # Salva no Firebase
        salvar_dados_csv(dados_paciente) # Salva no CSV
# Sincronização Firestore para CSV
def sincronizar_firestore_para_csv(db):
    """Sincroniza os dados do Firestore com o arquivo CSV."""
    caminho_csv = "C:/Projeto_Estacio_2024-Fono/Projeto_Fonart/dados/pacientes/pacientes.csv"
    try:
        pacientes_ref = db.collection('pacientes')
        docs = pacientes_ref.stream()
        with open(caminho_csv, mode='w', newline='', encoding='utf-8') as arquivo_csv:
            campos = ["nome", "idade", "diagnostico", "localidade", "tratamento", "mes", "consentimento"]
            escritor_csv = csv.DictWriter(arquivo_csv, fieldnames=campos)
            escritor_csv.writeheader() # Escreve o cabeçalho
            # Escreve cada documento do Firestore no CSV
            for doc in docs:
                escritor_csv.writerow(doc.to_dict())
        print("Sincronização do Firestore para o CSV concluída!")
    except Exception as e:
        print(f"Erro ao sincronizar Firestore para CSV: {e}")
# Sincronização CSV para Firestore
def sincronizar_csv_para_firestore(db):
    """Sincroniza os dados do CSV com o Firestore, resolvendo conflitos de data de modificação."""
    caminho_csv = "C:/Projeto_Estacio_2024-Fono/Projeto_Fonart/dados/pacientes/pacientes.csv"
    try:
        with open(caminho_csv, mode='r', encoding='utf-8') as arquivo_csv:
            leitor_csv = csv.DictReader(arquivo_csv)
            for linha in leitor_csv:
                # Verifica se o documento já existe
                doc_ref = db.collection('pacientes').where('nome', '==', linha['nome']).get() # Usando argumentos nomeados
                if doc_ref:
                    # Compara datas de modificação
                    for doc in doc_ref:
                        if linha.get('data_modificacao', '') > doc.to_dict().get('data_modificacao', ''):
                            db.collection('pacientes').document(doc.id).set(linha) # Atualiza Firestore
                else:
                    db.collection('pacientes').add(linha) # Adiciona novo documento
        print("Sincronização do CSV para o Firestore concluída!")
    except Exception as e:
        print(f"Erro ao sincronizar CSV para Firestore: {e}")
# Função para iniciar a sincronização em uma thread
def iniciar_sincronizacao(db):
    sincronizar_firestore_para_csv(db)
    sincronizar_csv_para_firestore(db)
# Função para mostrar gráfico
def mostrar_grafico():
    # Lendo dados do CSV
    csv_path = 'C:/Projeto_Estacio_2024-Fono/Projeto_Fonart/dados/pacientes/pacientes.csv'
    df = pd.read_csv(csv_path)
    # Agrupar os dados do CSV por localidade e contar o número de tratamentos
    tratamentos_por_localidade_csv = df['localidade'].value_counts()
    # Consultar dados do Firebase
    tratamentos_por_localidade_firebase = {}
    pacientes_ref = db.collection('pacientes')
    for doc in pacientes_ref.stream():
```

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
data = doc.to_dict()
localidade = data.get('localidade')
if localidade:
    tratamentos_por_localidade_firebase[localidade] = tratamentos_por_localidade_firebase.get(localidade, 0) + 1
# Juntando dados do CSV e do Firebase
localidades = list(set(tratamentos_por_localidade_csv.index).union(set(tratamentos_por_localidade_firebase.keys())))
tratamentos_por_localidade = [
    tratamentos_por_localidade_csv.get(localidade, 0) + tratamentos_por_localidade_firebase.get(localidade, 0)
    for localidade in localidades
]
# Criando o gráfico de barras
plt.figure(figsize=(8, 6))
plt.bar(localidades, tratamentos_por_localidade, color=['blue', 'green', 'orange'])
plt.title("Tratamentos por Localidade")
plt.xlabel("Localidade")
plt.ylabel("Número de Tratamentos")
# Exibir o gráfico
plt.show()
# Define o diretório base do projeto
diretorio_base = r"C:\Projeto_Estacio_2024-Fono\Projeto_Fonart"
# Função para gerar relatório PDF
def gerar_relatorio_pdf():
    print("Iniciando geração do relatório PDF...")
    """
    Gera um relatório em PDF com os dados dos pacientes.
    Anonimiza o nome dos pacientes para proteção de dados, conforme a LGPD.
    Inclui o logotipo da empresa centralizado no cabeçalho e evita cortes.
    Também adiciona as informações da empresa no rodapé do relatório, com margens consistentes.
    Agora, inclui um espaçamento extra entre o título e os dados.
    Adiciona numeração automática aos dados dos pacientes.
    """
    try:
        if not messagebox.askyesno("Atenção", "Você está prestes a acessar dados pessoais - (Nome será anonimizado). Deseja continuar?"):
            return
        arquivo = os.path.join(diretorio_base, 'dados', 'pacientes', 'pacientes.csv')
        if not os.path.exists(arquivo):
            logging.warning("Tentativa de gerar relatório sem dados disponíveis.")
            messagebox.showwarning("Aviso", "Nenhum dado para gerar o relatório.")
            return
        df = pd.read_csv(arquivo)
        pdf_file = os.path.join(diretorio_base, 'dados', 'relatórios', 'relatorio_pacientes.pdf')
        c = canvas.Canvas(pdf_file, pagesize=letter)
        # Adiciona o logotipo centralizado
        logotipo_path = r"C:\Projeto_Estacio_2024-Fono\Projeto_Fonart\assets\fonart.matriz.jpg"
        logotipo_width = 100
        logotipo_height = 46
        x_position = (letter[0] - logotipo_width) / 2
        y_position = 720
        c.drawImage(logotipo_path, x_position, y_position, width=logotipo_width, height=logotipo_height)
        title = "Fonart - Espaço Fonoaudiologia Aplicada e Saúde Integral - Relatório de Pesquisa Estatística"
        title_x = 100
        title_y = 710
        for line in title.split(" - "):
            c.drawString(title_x, title_y, line)
            title_y -= 15
        title_y -= 30
        c.setFont("Helvetica", 10) # Define o tamanho da fonte aqui
        c.drawString(100, title_y, "Num. - Nome - Idade - Diagnóstico - Localidade - Tratamento - Mês")
        linha_atual = title_y - 20
        for index, row in df.iterrows():
            if linha_atual < 60: # Checa se vai ultrapassar o rodapé
                adicionar_rodape(c) # Adiciona rodapé antes de mudar de página
                c.showPage() # Inicia uma nova página
                linha_atual = 750 # Reinicia a posição vertical
                # Adiciona o logotipo novamente na nova página
                c.drawImage(logotipo_path, x_position, y_position, width=logotipo_width, height=logotipo_height)
                # Redefine o título e a numeração
                for line in title.split(" - "):
                    c.drawString(title_x, title_y, line)
                    title_y -= 15
                title_y -= 30
            adicionar_rodape(c) # Adiciona rodapé antes de mudar de página
            c.showPage() # Inicia uma nova página
            linha_atual = 750 # Reinicia a posição vertical
            # Adiciona o logotipo novamente na nova página
            c.drawImage(logotipo_path, x_position, y_position, width=logotipo_width, height=logotipo_height)
            # Redefine o título e a numeração
            for line in title.split(" - "):
                c.drawString(title_x, title_y, line)
                title_y -= 15
            title_y -= 30
    except Exception as e:
        logging.error(f"Erro ao gerar o relatório: {str(e)}")
```

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
title_y -= 15
title_y -= 30
c.drawString(100, title_y, "Num. - Nome - Idade - Diagnóstico - Localidade - Tratamento, Mês")
linha_atual = title_y - 20
numero = index + 1
nome_anonimo = ''.join(random.choices(string.ascii_uppercase + string.digits, k=6))
c.drawString(100, linha_atual, f'{numero}. {nome_anonimo} - {row["idade"]} - {row["diagnosticos"]} - {row["localidade"]} - {row["tratamento"]} - {row["mes"]}')
linha_atual -= 20 # Move para a próxima linha
# Adiciona rodapé na última página
adicionar_rodape(c)
c.save()
logging.info(f'Relatório PDF gerado com sucesso: {pdf_file}')
messagebox.showinfo("Sucesso", f'Relatório gerado: {pdf_file}')
# Após gerar o PDF, abrir o arquivo
abrir_arquivo(pdf_file)
except Exception as e:
    logging.error(f'Erro ao gerar relatório PDF: {e}')
    messagebox.showerror("Erro", "Falha ao gerar o relatório PDF.")
def abrir_arquivo(caminho):
    try:
        os.startfile(caminho) # Abre o arquivo no visualizador padrão
    except Exception as e:
        logging.error(f'Erro ao abrir o arquivo: {e}')
        messagebox.showerror("Erro", "Falha ao abrir o arquivo PDF.")
def adicionar_rodape(c):
    """ Adiciona o rodapé e numeração da página no PDF. """
    c.setFont("Helvetica", 8) # Define o tamanho da fonte para o rodapé
    rodape_texto1 = "FONART - Maxxi Business Freguesia"
    rodape_texto2 = "Estr. dos Três Rios, 1097 - Sala 530, Freguesia de Jacarepaguá, 22745-004"
    rodape_texto3 = "E-mail: fonartrio@gmail.com - Tel.: 21 9 7282-2546"
    rodape_texto4 = "Relatório gerado via tecnologia Python, Extensão Universitária - Estácio 2024 - Marcus Cesar Ferreira"
    c.drawString(100, 50, rodape_texto1)
    c.drawString(100, 40, rodape_texto2)
    c.drawString(100, 30, rodape_texto3)
    c.drawString(100, 20, rodape_texto4)
    # Adiciona numeração da página
    c.drawString(500, 20, f'Página {c.getPageNumber()}')
# Criação da interface gráfica
def criar_janela_principal(db):
    """Cria a janela principal da aplicação."""
    janela = tk.Tk()
    janela.title("Sistema FONART")
    # Carrega o logotipo da empresa
    with Image.open(r"C:\Projeto_Estacio_2024-Fono\Projeto_Fonart\assets\fonart.matriz.png") as imagem_original:
        imagem_redimensionada = imagem_original.resize((100, 50))
        imagem_logo = ImageTk.PhotoImage(imagem_redimensionada)
    label_logo = tk.Label(janela, image=imagem_logo)
    label_logo.grid(row=0, column=0, columnspan=2, padx=10, pady=10)
    # Label de status
    status_label = tk.Label(janela, text="", font=("Arial", 10), bg=janela.cget("bg"), padx=10, pady=5)
    status_label.grid(row=9, columnspan=2, padx=10, pady=10)
    # Campos de entrada
    tk.Label(janela, text="Nome:").grid(row=1, column=0, padx=5, pady=5)
    entry_nome = tk.Entry(janela)
    entry_nome.grid(row=1, column=1, padx=5, pady=5)
    tk.Label(janela, text="Idade:").grid(row=2, column=0, padx=5, pady=5)
    entry_idade = tk.Entry(janela)
    entry_idade.grid(row=2, column=1, padx=5, pady=5)
    tk.Label(janela, text="Diagnóstico:").grid(row=3, column=0, padx=5, pady=5)
    diagnosticos = ["Disartria", "Afasia", "Dislexia", "Disgrafia", "Disfagia", "Gagueira", "Autismo"]
    combobox_diagnostico = ttk.Combobox(janela, values=diagnosticos)
    combobox_diagnostico.grid(row=3, column=1, padx=5, pady=5)
    # Listagens para localidade, tratamento e meses
    localidades = ["Rio de Janeiro", "Niterói", "São Gonçalo"]
    tratamentos = ["Terapia de Linguagem", "Terapia de Voz", "Terapia de Deglutição",
                  "Reabilitação da Audição", "Terapia de Articulação", "Intervenção Precoce",
                  "Terapia de Leitura e Escrita"]
    meses = ["Janeiro", "Fevereiro", "Março", "Abril", "Maio", "Junho", "Julho", "Agosto",
             "Setembro", "Outubro", "Novembro", "Dezembro"]
```

## Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```

tk.Label(janela, text="Localidade:").grid(row=4, column=0, padx=5, pady=5)
combobox_localidade = ttk.Combobox(janela, values=localidades)
combobox_localidade.grid(row=4, column=1, padx=5, pady=5)
tk.Label(janela, text="Tratamento:").grid(row=5, column=0, padx=5, pady=5)
combobox_tratamento = ttk.Combobox(janela, values=tratamentos)
combobox_tratamento.grid(row=5, column=1, padx=5, pady=5)
tk.Label(janela, text="Mês:").grid(row=6, column=0, padx=5, pady=5)
combobox_mes = ttk.Combobox(janela, values=meses)
combobox_mes.grid(row=6, column=1, padx=5, pady=5)
# Checkbutton para consentimento
consentimento_var = tk.BooleanVar() # Variável para o consentimento
check_consentimento = tk.Checkbutton(janela, text="Consentimento para uso de dados", variable=consentimento_var)
check_consentimento.grid(row=7, columnspan=2, padx=5, pady=5)
# Botão para salvar os dados
btn_salvar = tk.Button(janela, text="Salvar Dados", command=lambda: salvar_dados(
    db, entry_nome.get(), entry_idade.get(), combobox_diagnostico.get(),
    combobox_localidade.get(), combobox_tratamento.get(), combobox_mes.get(),
    consentimento_var.get(), status_label
))
btn_salvar.grid(row=8, columnspan=1, padx=5, pady=5)
# Label para mostrar a mensagem de sucesso
status_label = tk.Label(janela, text="", fg="green") # Mensagem de sucesso em verde
status_label.grid(row=8, column=1, padx=5, pady=5) # Coluna 2 ao lado do botão
# Botão para gerar relatório PDF
btn_gerar_relatorio = tk.Button(janela, text="Gerar Relatório PDF", command=gerar_relatorio_pdf)
btn_gerar_relatorio.grid(row=9, column=0, padx=5, pady=5) # Posicionado na linha 9, coluna 0
# Botão para gerar gráficos
btn_graficos = tk.Button(janela, text="Gerar Gráficos", command=mostrar_grafico)
btn_graficos.grid(row=9, column=1, padx=5, pady=5)
# Iniciar a sincronização em uma thread
threading.Thread(target=iniciar_sincronizacao, args=(db,), daemon=True).start()
janela.mainloop() # Executa a interface gráfica
if __name__ == "__main__":
    db = inicializar_firebase() # Inicializa o Firebase
    criar_janela_principal(db) # Cria a janela principal

```

### Evidências técnicas do Projeto de Extensão: Criando o banco de dados google cloud

#### Projeto no Google Cloud

- Acesso Google Cloud Console.
- Novo projeto.

Nome do projeto

FONART statistic

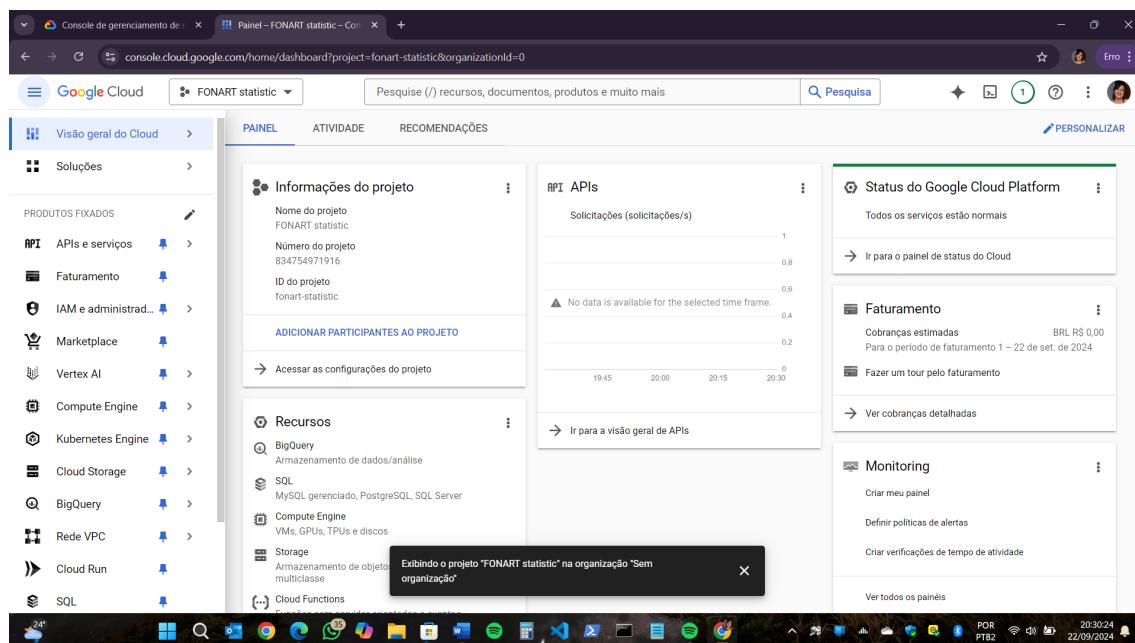
Número do projeto

8347549XXXX

ID do projeto

fonart-statistic

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia



43

## Ativando o Firestore

The screenshot shows the process of creating a new Firestore database. Step 1: 'Selecionar o modo Firestore' (Select the Firestore mode) - It says to choose between native or Datastore. Step 2: 'Configurar o banco de dados' (Configure the database) - It asks to select the mode: 'Modo nativo (recomendado)' (Native mode (recommended)) is selected. On the right, there's a 'Resumo de preços' (Price summary) for the 'NÍVEL GRATUITO' (Free tier) which states: 'Este banco de dados tem uma cota de nível gratuito. Cada projeto só pode ter um banco de dados chamado "(padrão)", qualificado para a cota de nível gratuito. Quando essa cota é esgotada, a cobrança é feita com base nas operações, no armazenamento e no uso da rede'. A 'COMPARAR MODOS' (Compare modes) button is also visible. At the bottom, there are 'CONTINUAR' (Continue) and 'CRIAR BANCO DE DADOS' (Create database) buttons.

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

## Ativação dos recursos

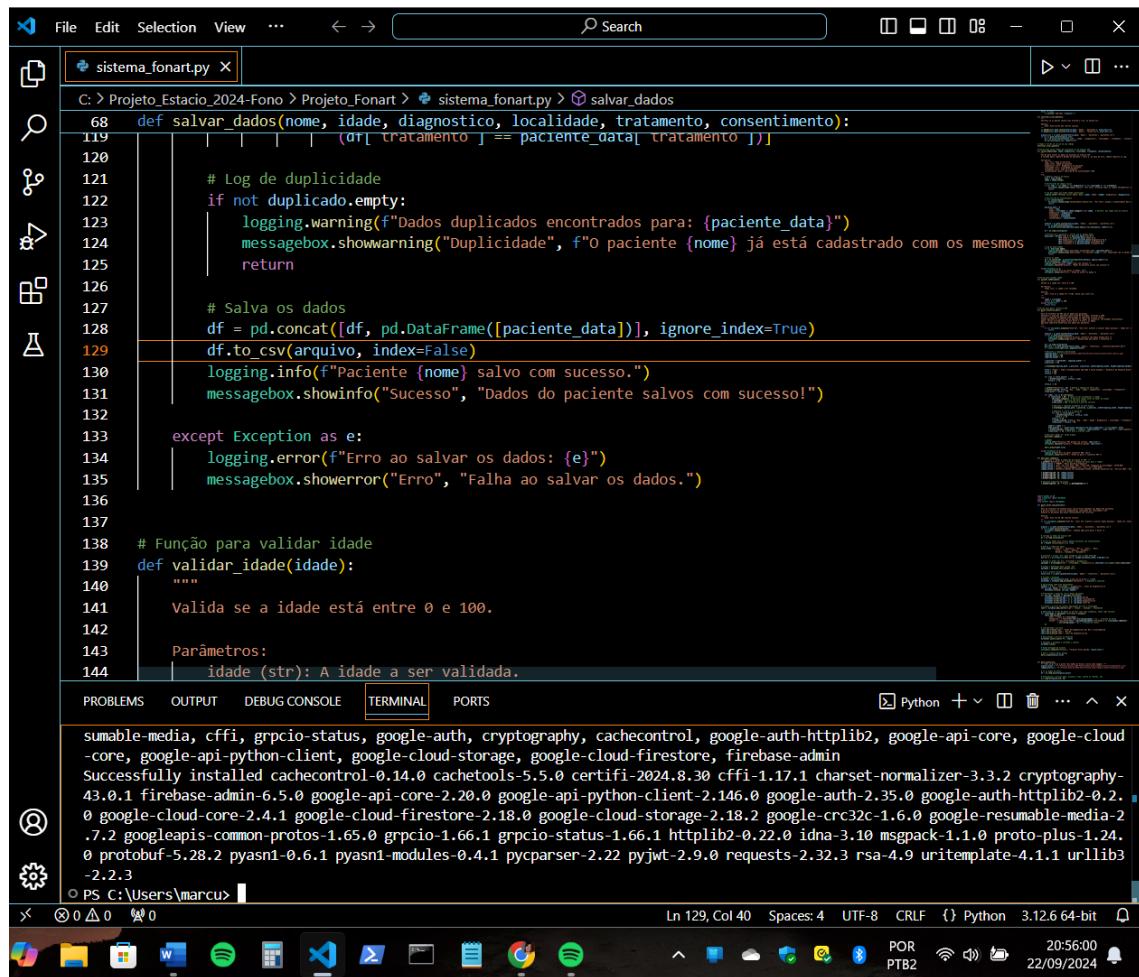
The screenshot shows the Google Cloud Firestore console interface. The left sidebar has a tree view with 'Firestore' selected, under which 'Banco de dados' is expanded, showing options like 'Firestore Studio', 'Índices', 'Importação/exportação', etc. The main area shows a large dashed cloud icon with the text 'Seu banco de dados está pronto, basta adicionar dados.' Below it are two buttons: 'INICIAR COLEÇÃO' and 'Leia o guia de início rápido'. The top navigation bar includes 'Google Cloud', 'FONART statistic', 'Firestore', 'Pesquisa', and other standard browser controls. The bottom taskbar shows various application icons and system status.

## Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

The screenshot shows a browser window with the Google Cloud Console URL: https://console.cloud.google.com/. The title bar indicates the current tab is 'API Google Cloud Firestore API'. The main content area displays the 'Google Cloud Firestore API' page. At the top, there's a promotional banner for a free trial with \$300 credit. Below it, the product logo and name 'Google Cloud Firestore API' are shown, along with its sub-type 'Google Enterprise API'. A brief description states: 'Easily store and sync app data at global scale'. Two buttons are present: 'ATIVAR' (Activate) and 'TESTAR ESTA API' (Test this API). Below these buttons, three navigation links are visible: 'VISÃO GERAL' (General View), 'DOCUMENTAÇÃO' (Documentation), and 'PRODUTOS RELACIONADOS' (Related Products). A green circular badge in the top right corner contains the number '45'. The bottom of the screenshot shows a portion of the Windows taskbar with various icons and system status information.

Testes e experimentações do Google Cloud (conhecendo as ferramentas e possibilidades)

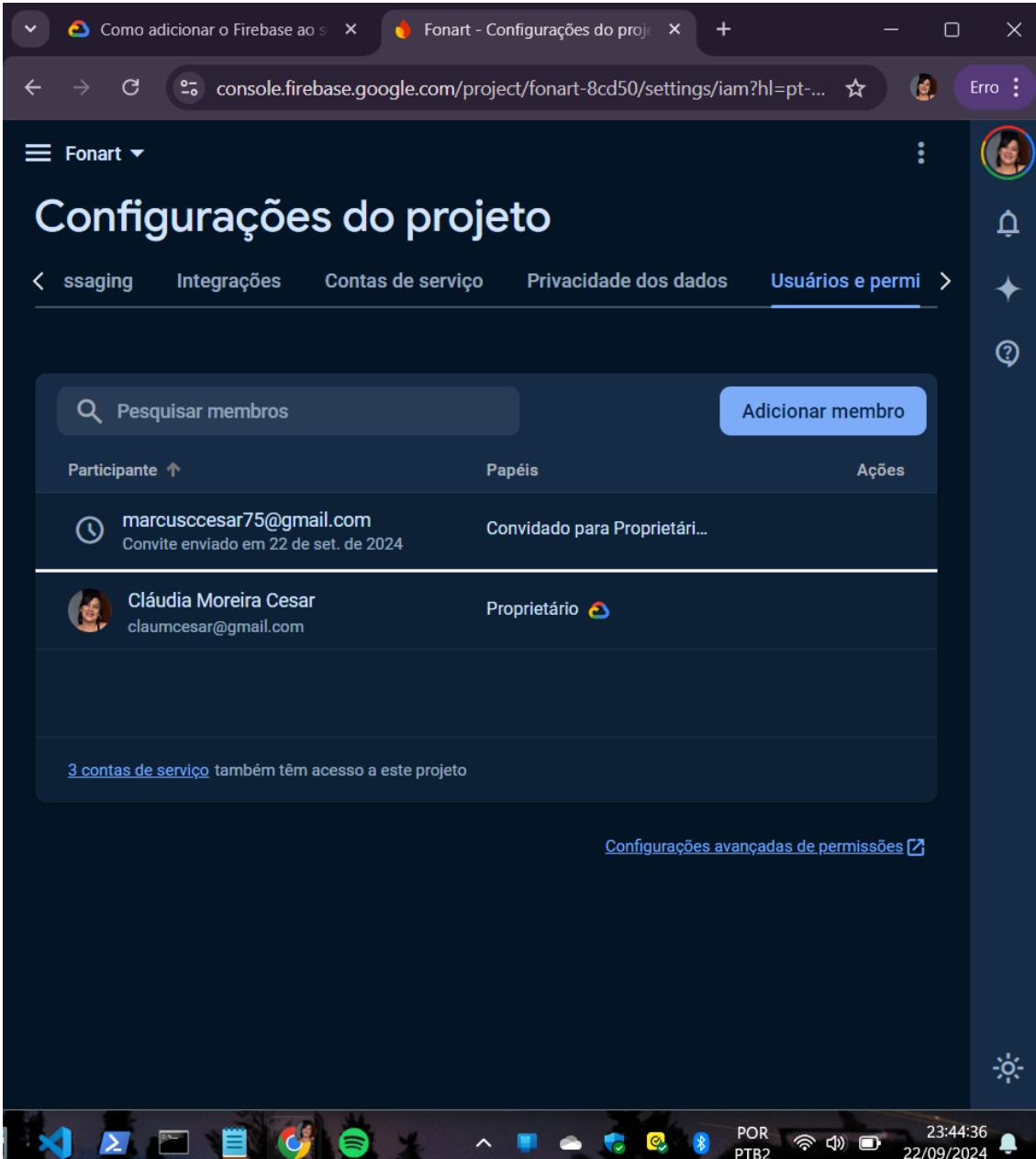
# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia



```
File Edit Selection View ... ← → Search
sistema_fonart.py
C: > Projeto_Estacio_2024-Fono > Projeto_Fonart > sistema_fonart.py > salvar_dados
68 def salvar_dados(nome, idade, diagnostico, localidade, tratamento, consentimento):
119     | | | | (df[tratamento] == paciente_data[tratamento]))
120
121     # Log de duplicidade
122     if not duplicado.empty:
123         logging.warning(f"Dados duplicados encontrados para: {paciente_data}")
124         messagebox.showwarning("Duplicidade", f"O paciente {nome} já está cadastrado com os mesmos")
125         return
126
127     # Salva os dados
128     df = pd.concat([df, pd.DataFrame([paciente_data])], ignore_index=True)
129     df.to_csv(arquivo, index=False)
130     logging.info(f"Patiente {nome} salvo com sucesso.")
131     messagebox.showinfo("Sucesso", "Dados do paciente salvos com sucesso!")
132
133 except Exception as e:
134     logging.error(f"Erro ao salvar os dados: {e}")
135     messagebox.showerror("Erro", "Falha ao salvar os dados.")
136
137
138 # Função para validar idade
139 def validar_idade(idade):
140     """
141         Valida se a idade está entre 0 e 100.
142
143     Parâmetros:
144         idade (str): A idade a ser validada.
145
146     """
147     if not isinstance(idade, str) or not idade.isdigit():
148         raise ValueError("A idade deve ser uma string contendo apenas dígitos.")
149
150     idade_int = int(idade)
151
152     if not 0 <= idade_int <= 100:
153         raise ValueError("A idade deve estar entre 0 e 100, inclusive.")
154
155     return True
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
427
428
429
429
430
431
432
433
434
435
436
437
437
438
439
439
440
441
442
443
444
445
446
447
447
448
449
449
450
451
452
453
454
455
456
457
457
458
459
459
460
461
462
463
463
464
465
465
466
467
467
468
468
469
469
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
599
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1
```

## Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

Criando relações de compartilhamento da Empresa do Projeto, com o desenvolvedor responsável pelo Projeto Fonart, de Extensão Universitária (Estácio).



The screenshot shows the 'Configurações do projeto' (Project Settings) page in the Firebase console. The 'Usuários e permissões' (Users & permissions) tab is selected. A table lists project members:

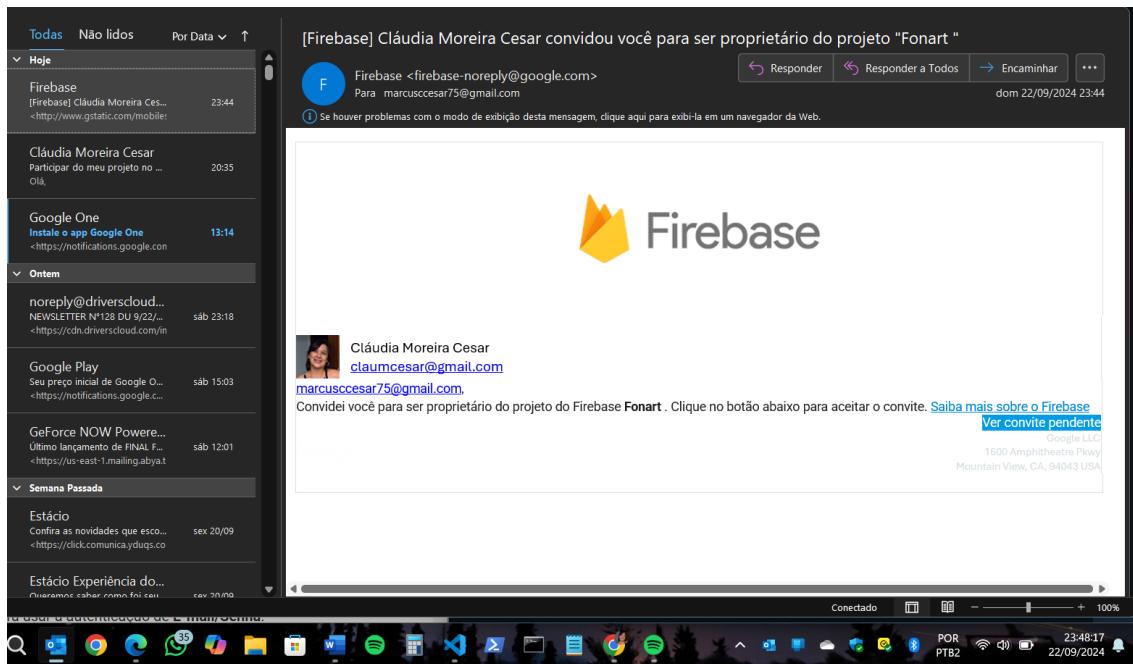
Participante	Papéis	Ações
marcusccesar75@gmail.com Convite enviado em 22 de set. de 2024	Convidado para Proprietário...	
Cláudia Moreira Cesar claucesar@gmail.com	Proprietário	

Below the table, it says '3 contas de serviço também têm acesso a este projeto'. At the bottom right of the main area, there is a link 'Configurações avançadas de permissões'.

The browser address bar shows 'console.firebaseio.google.com/project/fonart-8cd50/settings/iam?hl=pt-BR'. The status bar at the bottom of the screen shows various icons and the date '22/09/2024'.

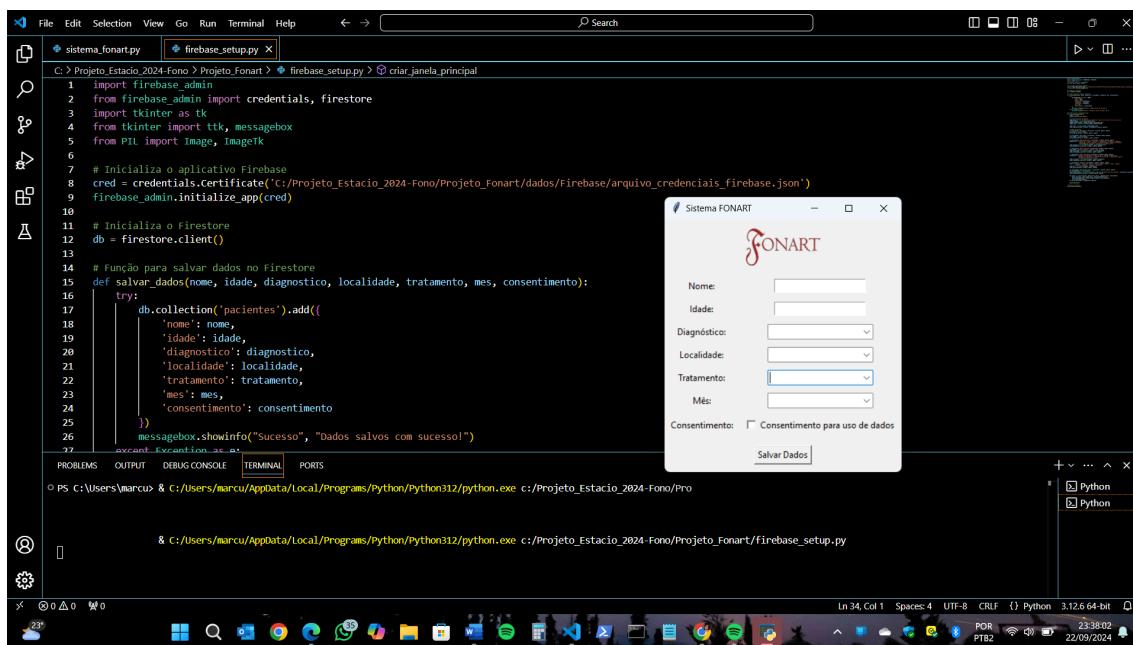
Relações realizadas

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia



48

## Evidências do trabalho em equipe

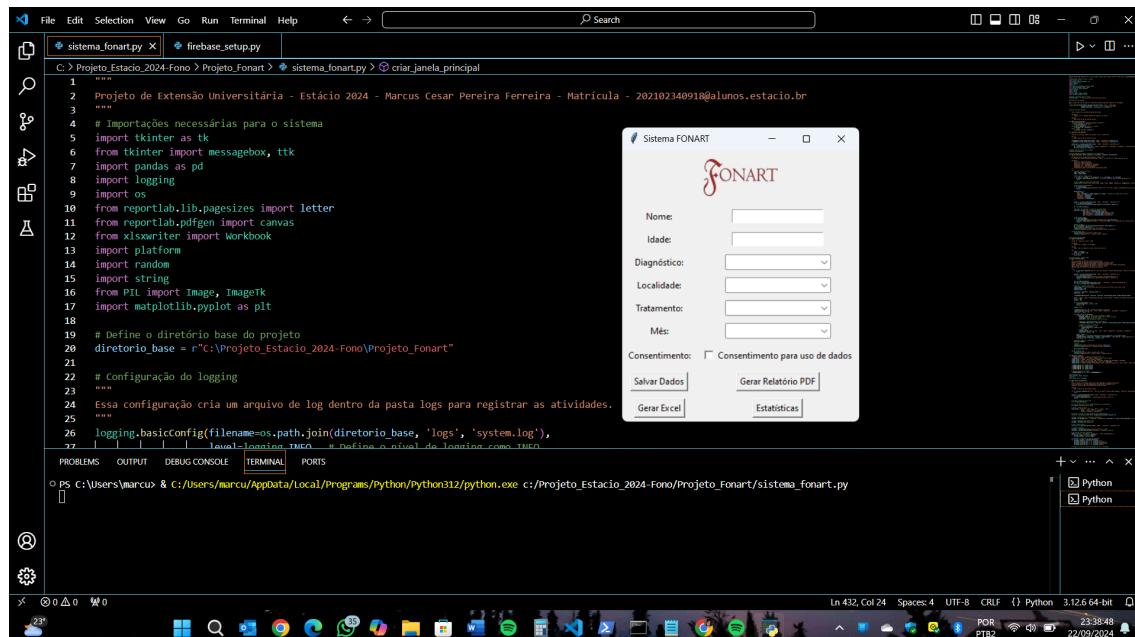


## Teste de todos os projetos (ambiente de testes, homologação e de produção)

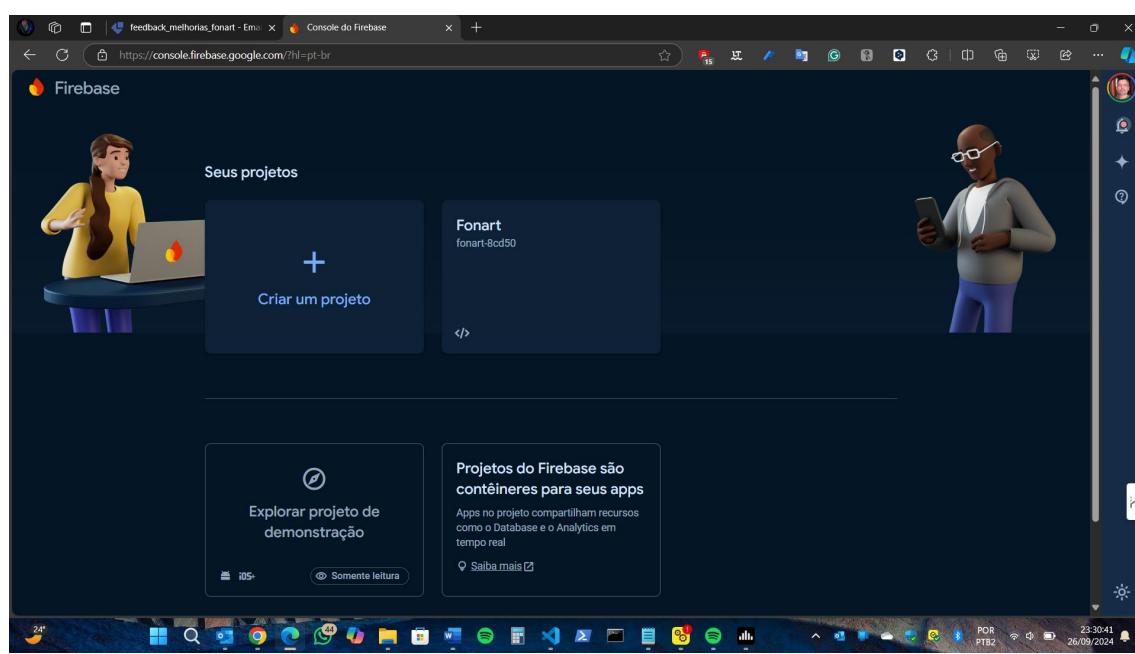
**Sistema\_fonart.py** (testes) com todas as configurações solicitadas e relatórios, contudo, sem integração com firebase.

**Sistema firebase\_setup.py** (produção em desenvolvimento para inserir relatórios) com integração e relação com o BD com sistemas de sincronização e APP Web.

## Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

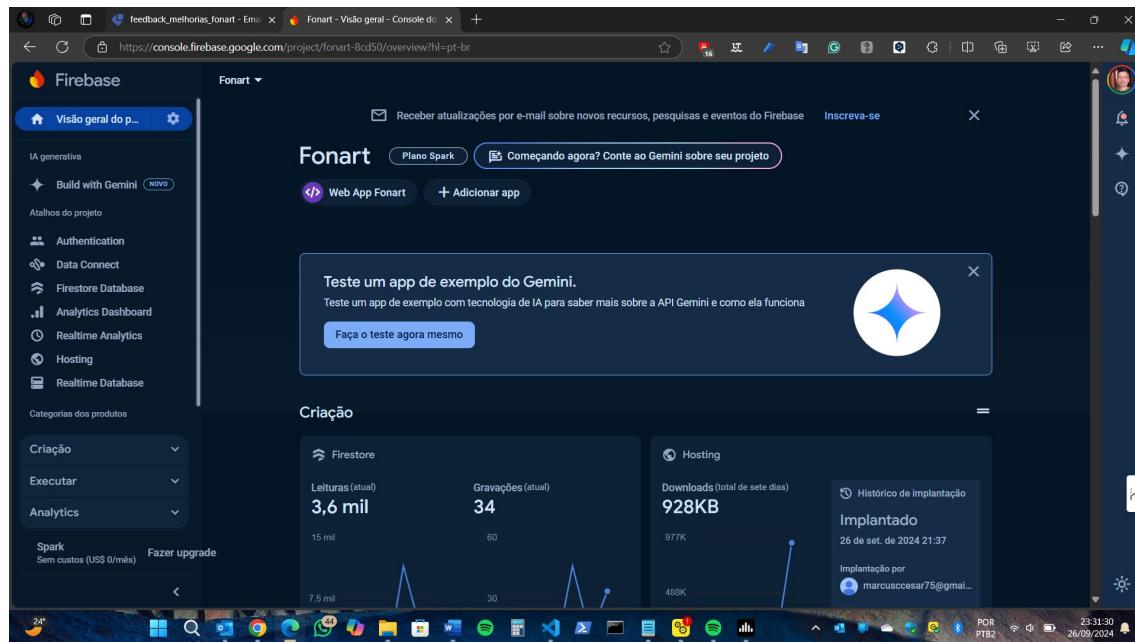


### Implantação do Banco de Dados – Firestore/Firebase – Google Cloud

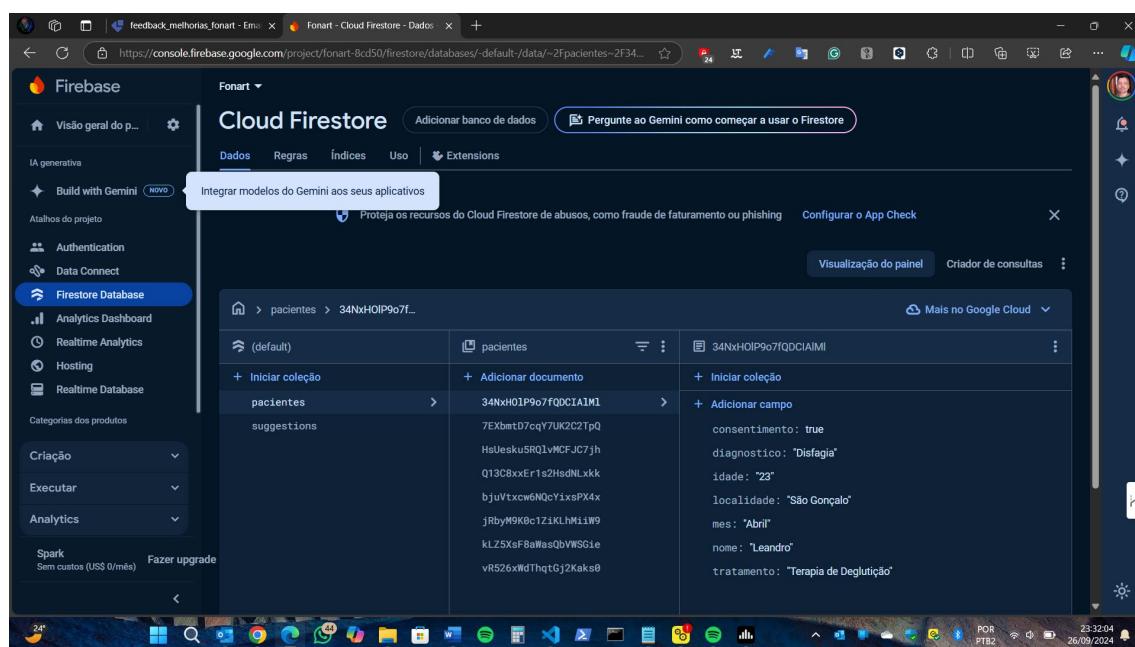


Console firebase – Google Cloud (início dos projetos para produção)

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia



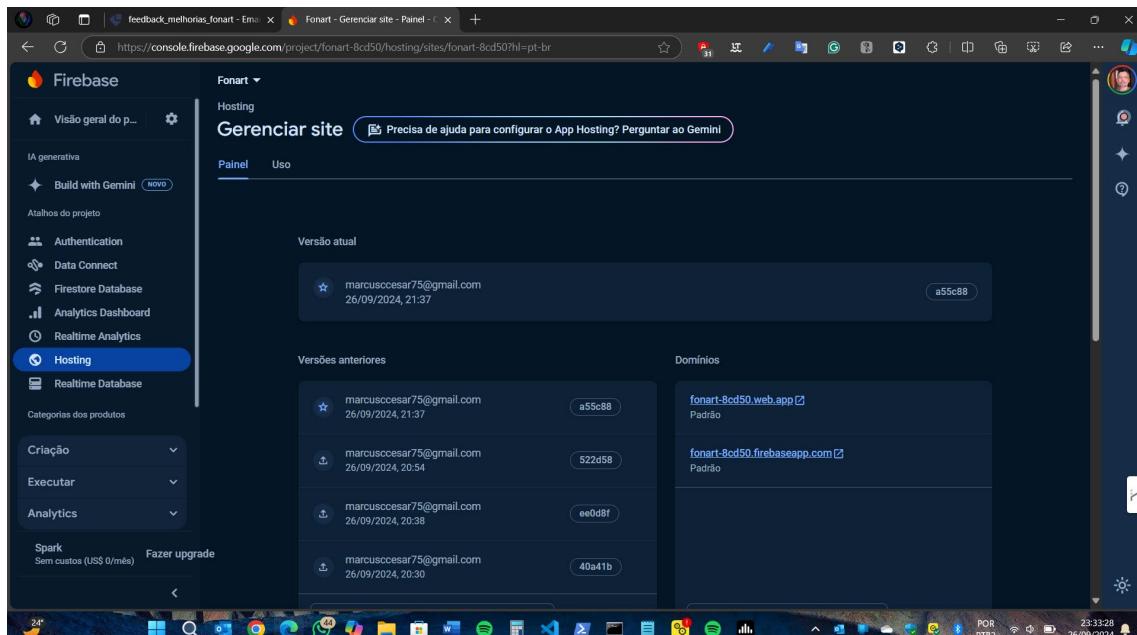
## Criando relações do Banco de Dados



Testes e sucessos com as relações de sincronização do arquivo “pacientes.csv”

Pasta Local, Web app e questionários online realizados pelo Firestore (páginas de internet) e com relacionamento com o domínio do site oficial da empresa [www.fonart.com.br](http://www.fonart.com.br) e EmailJS.

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia



Testes e sucessos com a relação do Firebase Hosting

## VERSÃO FINAL 3.0 FIREBASE\_SETUP.PY

\*\*\*\*\*

**Projeto de Extensão Universitária - Estácio 2024 - Marcus Cesar Pereira Ferreira - Matrícula - 202102340918@alunos.estacio.br**

**Resumo do Projeto:** Este sistema integra várias tecnologias e foi desenvolvido para a Extensão Universitária na Estácio 2024. Localmente, utiliza Python para integração com o Firebase do Google Cloud e o Cloud Firestore, armazenando dados de pacientes de fonoaudiologia.

A interface em Tkinter permite a coleta de informações como nome, idade e diagnóstico. O sistema também oferece funcionalidades como geração de relatórios em PDF e gráficos de análise, além de sincronizar automaticamente os dados locais e na nuvem. O formulário utiliza HTML e JavaScript para encaminhar os dados ao Cloud Firestore, que os valida e armazena.

O objetivo do formulário WEB <https://fonart-8cd50.web.app/> é coletar dados estatísticos de pacientes em tratamento fonoaudiológico. A coleta de informações como idade, diagnóstico, localidade e tratamento é essencial para entender as necessidades dos pacientes e aprimorar a atuação da equipe. O consentimento do paciente é obrigatório para a utilização dos dados, que são tratados com a máxima confidencialidade, visando à conformidade com a LGPD.

\*\*\*\*\*

**# Importações necessárias para o sistema**

```
import os
import csv
import logging
```

```
from datetime import datetime
from google.cloud import firestore
import firebase_admin
from firebase_admin import credentials,firestore
from tkinter import ttk, messagebox
import pandas as pd
import tkinter as tk
from PIL import Image, ImageTk
import threading
import matplotlib.pyplot as plt # Importação do matplotlib para gráficos
from reportlab.pdfgen import canvas
from reportlab.lib.pagesizes import letter
import random
import string

# Configuração do logging
log_path = r'C:\Projeto_Estacio_2024-Fono\Projeto_Fonart\logs\system.log'
logging.basicConfig(filename=log_path,
                    level=logging.DEBUG,
                    format='%(asctime)s - %(levelname)s - %(message)s')

# Inicialização do Firebase
def inicializar_firebase():
    """Inicia a conexão com o Firebase e retorna a instância do Firestore."""
    try:
        if not firebase_admin._apps: # Verifica se o Firebase já foi inicializado
            # Configuração do Firebase
            cred = credentials.Certificate(r"C:\Projeto_Estacio_2024-Fono\Projeto_Fonart\dados\Firebase\fonart-8cd50-firebase-adminsdk-8qauidb13e0addd.json")
            firebase_admin.initialize_app(cred)
            db = firestore.client() # Conexão com o Firestore
            return db
    except Exception as e:
        print(f"Erro ao inicializar o Firebase: {e}")
        return None

# Chama a função para inicializar e obter o cliente do Firestore
db = inicializar_firebase()

# Função para validar os dados
def validar_dados(nome, idade, diagnostico, localidade, tratamento, mes, consentimento):
    """Valida os dados de entrada antes de salvar."""
    if not (nome and idade and diagnostico and localidade and tratamento and mes):
        messagebox.showwarning("Aviso", "Todos os campos devem ser preenchidos!")
        return False
```

```
if not consentimento:  
    messagebox.showwarning("Aviso", "É necessário o consentimento para uso de  
dados!")  
    return False  
try:  
    if not isinstance(int(idade), int) or int(idade) < 0:  
        messagebox.showwarning("Aviso", "A idade deve ser um número válido!")  
        return False  
    except ValueError:  
        messagebox.showwarning("Aviso", "A idade deve ser um número válido!")  
        return False  
    return True  
  
# Função para salvar dados no Firebase  
def salvar_dados_firebase(dados_paciente):  
    global status_label # Torna o status_label acessível dentro da função  
    try:  
        db = inicializar_firebase() # Inicializa o Firebase e conecta ao Firestore  
        if db:  
            dados_paciente['data_modificacao'] = datetime.now().isoformat()  
            db.collection('pacientes').add(dados_paciente)  
            # Lógica para salvar os dados no Firebase...  
            print("Dados salvos com sucesso no Firebase!")  
            logging.info(f"Dados salvos no Firebase: {dados_paciente}")  
  
            # Atualiza o status de sucesso  
            status_label.config(text="DADOS ENVIADOS COM SUCESSO!", fg="green")  
        except Exception as e:  
            print(f"ERRO AO SALVAR DADOS: {e}")  
            logging.error(f"Erro ao salvar dados no Firebase: {e}")  
  
            # Atualiza o status de erro  
            status_label.config(text="Erro ao enviar os dados.", fg="red")  
  
    # Função para salvar os dados no CSV local  
    def salvar_dados_csv(dados_paciente):  
        global status_label # Torna o status_label acessível dentro da função  
        arquivo_csv = r'C:\Projeto_Estacio_2024-  
Fono\Projeto_Fonart\dados\pacientes\pacientes.csv'  
        campos = ['nome', 'idade', 'diagnostico', 'localidade', 'tratamento', 'mes',  
'consentimento', 'data_modificacao']  
        try:  
            with open(arquivo_csv, mode='a', newline='', encoding='utf-8') as arquivo:  
                escritor_csv = csv.DictWriter(arquivo, fieldnames=campos)  
                if arquivo.tell() == 0:  
                    escritor_csv.writeheader()
```

```
escritor_csv.writerow(dados_paciente)
logging.info(f'Dados salvos no CSV: {dados_paciente}')
print(f'Dados salvos com sucesso no arquivo {arquivo_csv}!')

# Atualiza o status de sucesso
status_label.config(text="Dados salvos no CSV com sucesso!", fg="green")
except Exception as e:
    logging.error(f'Erro ao salvar dados no CSV: {e}')
    print(f'Erro ao salvar dados no CSV: {e}')

# Atualiza o status de erro
status_label.config(text="Erro ao salvar os dados no CSV.", fg="red")

# Função para salvar os dados após validação
def salvar_dados(db, nome, idade, diagnostico, localidade, tratamento, mes,
consentimento, data_modificacao):
    data_modificacao = datetime.now() # Corrige o uso do datetime

    # Exibe a data de modificação
    print(f"Data de Modificação: {data_modificacao}")

    # Lógica para salvar os dados no banco de dados
    """Salva os dados se forem válidos."""
    if validar_dados(nome, idade, diagnostico, localidade, tratamento, mes,
consentimento):
        dados_paciente = {
            "nome": nome,
            "idade": idade,
            "diagnostico": diagnostico,
            "localidade": localidade,
            "tratamento": tratamento,
            "mes": mes,
            "consentimento": consentimento
        }
        print(f"Salvando dados: {dados_paciente}") # Log detalhado
        salvar_dados_firebase(dados_paciente) # Salva no Firebase

# Sincronização Firestore para CSV
def sincronizar_firestore_para_csv(db):
    """Sincroniza os dados do Firestore com o arquivo CSV."""
    caminho_csv = "C:/Projeto_Estacio_2024-Fono/Projeto_Fonart/dados/pacientes/pacientes.csv"
    try:
        pacientes_ref = db.collection('pacientes')
        docs = pacientes_ref.stream()
```

```
with open(caminho_csv, mode='w', newline='', encoding='utf-8') as arquivo_csv:  
    # Adicionar 'data_modificacao' à lista de campos  
    campos = ["nome", "idade", "diagnostico", "localidade", "tratamento", "mes",  
    "consentimento", "data_modificacao"]  
    escritor_csv = csv.DictWriter(arquivo_csv, fieldnames=campos)  
    escritor_csv.writeheader() # Escreve o cabeçalho  
  
    # Escreve cada documento do Firestore no CSV  
    for doc in docs:  
        dados = doc.to_dict()  
  
        # Verifica se 'data_modificacao' está presente no documento, senão insere  
        # uma data padrão  
        if 'data_modificacao' not in dados:  
            dados['data_modificacao'] = "Não disponível" # Você pode mudar essa  
            # linha para uma data padrão, se preferir  
  
        escritor_csv.writerow(dados)  
  
    print("Sincronização do Firestore para o CSV concluída!")  
  
except Exception as e:  
    print(f"Erro ao sincronizar Firestore para CSV: {e}")  
  
# Sincronização CSV para Firestore  
def sincronizar_csv_para_firestore(db):  
    """Sincroniza os dados do CSV com o Firestore, resolvendo conflitos de data de  
    modificação."""  
    caminho_csv = "C:/Projeto_Estacio_2024-  
    Fono/Projeto_Fonart/dados/pacientes/pacientes.csv"  
    try:  
        with open(caminho_csv, mode='r', encoding='utf-8') as arquivo_csv:  
            leitor_csv = csv.DictReader(arquivo_csv)  
            for linha in leitor_csv:  
  
                # Verifica se o documento já existe  
                doc_ref = db.collection('pacientes').where('nome', '==', linha['nome']).get() #  
                Usando argumentos nomeados  
  
                if doc_ref:  
  
                    # Compara datas de modificação  
                    for doc in doc_ref:  
                        if linha.get('data_modificacao', '') > doc.to_dict().get('data_modificacao',  
":'): # Atualiza  
                            db.collection('pacientes').document(doc.id).set(linha) # Atualiza  
                            # Firestore
```

```
else:  
    db.collection('pacientes').add(linha) # Adiciona novo documento  
    print("Sincronização do CSV para o Firestore concluída!")  
except Exception as e:  
    print(f"Erro ao sincronizar CSV para Firestore: {e}")  
  
# Função para iniciar a sincronização em uma thread  
def iniciar_sincronizacao(db):  
    sincronizar_firestore_para_csv(db)  
    sincronizar_csv_para_firestore(db)  
  
# Função para mostrar gráfico  
def mostrar_grafico():  
  
    # Lendo dados do CSV  
    csv_path = 'C:/Projeto_Estacio_2024-Fono/Projeto_Fonart/dados/pacientes/pacientes.csv'  
    df = pd.read_csv(csv_path)  
  
    # Agrupar os dados do CSV por localidade e contar o número de tratamentos  
    tratamentos_por_localidade_csv = df['localidade'].value_counts()  
  
    # Consultar dados do Firebase  
    tratamentos_por_localidade_firestore = {}  
    pacientes_ref = db.collection('pacientes')  
    for doc in pacientes_ref.stream():  
        data = doc.to_dict()  
        localidade = data.get('localidade')  
        if localidade:  
            tratamentos_por_localidade_firestore[localidade] = tratamentos_por_localidade_firestore.get(localidade, 0) + 1  
  
    # Juntando dados do CSV e do Firebase  
    localidades = list(set(tratamentos_por_localidade_csv.index).union(set(tratamentos_por_localidade_firestore.keys())))  
    tratamentos_por_localidade = [  
        tratamentos_por_localidade_csv.get(localidade, 0) +  
        tratamentos_por_localidade_firestore.get(localidade, 0)  
        for localidade in localidades  
    ]  
  
    # Criando o gráfico de barras  
    plt.figure(figsize=(8, 6))  
    plt.bar(localidades, tratamentos_por_localidade, color=['blue', 'green', 'orange'])  
    plt.title("Tratamentos por Localidade")  
    plt.xlabel("Localidade")
```

```
plt.ylabel("Número de Tratamentos")

# Exibir o gráfico
plt.show()

# Define o diretório base do projeto
diretorio_base = r"C:\Projeto_Estacio_2024-Fono\Projeto_Fonart"

# Função para gerar relatório PDF
def gerar_relatorio_pdf():
    print("Iniciando geração do relatório PDF...")
    """
    Gera um relatório em PDF com os dados dos pacientes.
    Anonimiza o nome dos pacientes para proteção de dados, conforme a LGPD.
    Inclui o logotipo da empresa centralizado no cabeçalho e evita cortes.
    Também adiciona as informações da empresa no rodapé do relatório, com margens consistentes.
    Agora, inclui um espaçamento extra entre o título e os dados.
    Adiciona numeração automática aos dados dos pacientes.
    """
    try:
        if not messagebox.askyesno("Atenção", "Você está prestes a acessar dados pessoais - (Nome será anonimizado). Deseja continuar?"):
            return
        arquivo = os.path.join(diretorio_base, 'dados', 'pacientes', 'pacientes.csv')
        if not os.path.exists(arquivo):
            logging.warning("Tentativa de gerar relatório sem dados disponíveis.")
            messagebox.showwarning("Aviso", "Nenhum dado para gerar o relatório.")
            return
        df = pd.read_csv(arquivo)
        pdf_file      =      os.path.join(diretorio_base,      'dados',      'relatorios',
        'relatorio_pacientes.pdf')
        c = canvas.Canvas(pdf_file, pagesize=letter)

        # Adiciona o logotipo centralizado
        logotipo_path          =          r"C:\Projeto_Estacio_2024-
Fono\Projeto_Fonart\assets\fonart.matriz.jpg"
        logotipo_width = 100
        logotipo_height = 46
        x_position = (letter[0] - logotipo_width) / 2
        y_position = 720
        c.drawImage(logotipo_path,  x_position,  y_position,  width=logotipo_width,
height=logotipo_height)
        title = "Fonart - Espaço Fonoaudiologia Aplicada e Saúde Integral - Relatório de
Pesquisa Estatística"
        title_x = 100
        title_y = 710
```

```
for line in title.split(" - "):
    c.drawString(title_x, title_y, line)
    title_y -= 15
title_y -= 30
c.setFont("Helvetica", 10) # Define o tamanho da fonte aqui
c.drawString(100, title_y, "Num. - Nome - Idade - Diagnóstico - Localidade -
Tratamento - Mês")
linha_atual = title_y - 20
for index, row in df.iterrows():
    if linha_atual < 60: # Checa se vai ultrapassar o rodapé
        adicionar_rodape(c) # Adiciona rodapé antes de mudar de página
        c.showPage() # Inicia uma nova página
        linha_atual = 750 # Reinicia a posição vertical

    # Adiciona o logotipo novamente na nova página
    c.drawImage(logotipo_path, x_position, y_position, width=logotipo_width,
height=logotipo_height)

    # Redefine o título e a numeração
    for line in title.split(" - "):
        c.drawString(title_x, title_y, line)
        title_y -= 15
    title_y -= 30
    c.drawString(100, title_y, "Num. - Nome - Idade - Diagnóstico - Localidade -
Tratamento, Mês")
    linha_atual = title_y - 20
    numero = index + 1
    nome_anonimo = ".join(random.choices(string.ascii_uppercase + string.digits,
k=6))
    c.drawString(100, linha_atual, f"{numero}. {nome_anonimo} - {row['idade']} -
{row['diagnostico']} - {row['localidade']} - {row['tratamento']} - {row['mes']}")"
    linha_atual -= 20 # Move para a próxima linha

    # Adiciona rodapé na última página
    adicionar_rodape(c)
    c.save()
logging.info(f"Relatório PDF gerado com sucesso: {pdf_file}")
messagebox.showinfo("Sucesso", f"Relatório gerado: {pdf_file}")

# Após gerar o PDF, abrir o arquivo
abrir_arquivo(pdf_file)
except Exception as e:
    logging.error(f"Erro ao gerar relatório PDF: {e}")
    messagebox.showerror("Erro", "Falha ao gerar o relatório PDF.")
def abrir_arquivo(caminho):
    try:
        os.startfile(caminho) # Abre o arquivo no visualizador padrão
```

```
except Exception as e:  
    logging.error(f"Erro ao abrir o arquivo: {e}")  
    messagebox.showerror("Erro", "Falha ao abrir o arquivo PDF.")  
  
def adicionar_rodape(c):  
    """ Adiciona o rodapé e numeração da página no PDF. """  
    c.setFont("Helvetica", 8) # Define o tamanho da fonte para o rodapé  
    rodape_texto1 = "FONART - Maxxi Business Freguesia"  
    rodape_texto2 = "Estr. dos Três Rios, 1097 - Sala 530, Freguesia de Jacarepaguá,  
    22745-004"  
    rodape_texto3 = "E-mail: fonartrio@gmail.com - Tel.: 21 9 7282-2546"  
    rodape_texto4 = "Relatório gerado via tecnologia Python, Extensão Universitária -  
    Estácio 2024 - Marcus Cesar Ferreira"  
    c.drawString(100, 50, rodape_texto1)  
    c.drawString(100, 40, rodape_texto2)  
    c.drawString(100, 30, rodape_texto3)  
    c.drawString(100, 20, rodape_texto4)  
  
    # Adiciona numeração da página  
    c.drawString(500, 20, f"Página {c.getPageNumber()}")  
  
# Criação da interface gráfica  
def criar_janela_principal(db):  
    global status_label # Declara que irá usar a variável global  
    """Cria a janela principal da aplicação."""  
    janela = tk.Tk()  
    janela.title("Sistema FONART")  
  
    # Carrega o logotipo da empresa  
    with Image.open(r"C:\Projeto_Estacio_2024-  
    Fono\Projeto_Fonart\assets\fonart.matriz.png") as imagem_original:  
        imagem_redimensionada = imagem_original.resize((100, 50))  
        imagem_logo = ImageTk.PhotoImage(imagem_redimensionada)  
        label_logo = tk.Label(janela, image=imagem_logo)  
        label_logo.grid(row=0, column=0, columnspan=2, padx=10, pady=10)  
  
    # Label de status  
    status_label = tk.Label(janela, text="", font=("Arial", 10), bg=janela.cget("bg"),  
    padx=10, pady=5)  
    status_label.grid(row=9, columnspan=2, padx=10, pady=10)  
  
    # Campos de entrada  
    tk.Label(janela, text="Nome:").grid(row=1, column=0, padx=5, pady=5)  
    entry_nome = tk.Entry(janela)  
    entry_nome.grid(row=1, column=1, padx=5, pady=5)  
    tk.Label(janela, text="Idade:").grid(row=2, column=0, padx=5, pady=5)  
    entry_idade = tk.Entry(janela)  
    entry_idade.grid(row=2, column=1, padx=5, pady=5)
```

## Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
tk.Label(janela, text="Diagnóstico:").grid(row=3, column=0, padx=5, pady=5)
diagnosticos = ["Disartria", "Afasia", "Dislexia", "Disgrafia", "Disfagia", "Gagueira",
"Autismo"]
combobox_diagnostico = ttk.Combobox(janela, values=diagnosticos)
combobox_diagnostico.grid(row=3, column=1, padx=5, pady=5)

# Listagens para localidade, tratamento e meses
localidades = ["Rio de Janeiro", "Niterói", "São Gonçalo"]
tratamentos = ["Terapia de Linguagem", "Terapia de Voz", "Terapia de Deglutição",
    "Reabilitação da Audição", "Terapia de Articulação", "Intervenção Precoce",
    "Terapia de Leitura e Escrita"]
meses = ["Janeiro", "Fevereiro", "Março", "Abril", "Maio", "Junho", "Julho",
"Agosto",
    "Setembro", "Outubro", "Novembro", "Dezembro"]
tk.Label(janela, text="Localidade:").grid(row=4, column=0, padx=5, pady=5)
combobox_localidade = ttk.Combobox(janela, values=localidades)
combobox_localidade.grid(row=4, column=1, padx=5, pady=5)
tk.Label(janela, text="Tratamento:").grid(row=5, column=0, padx=5, pady=5)
combobox_tratamento = ttk.Combobox(janela, values=tratamentos)
combobox_tratamento.grid(row=5, column=1, padx=5, pady=5)
tk.Label(janela, text="Mês:").grid(row=6, column=0, padx=5, pady=5)
combobox_mes = ttk.Combobox(janela, values=meses)
combobox_mes.grid(row=6, column=1, padx=5, pady=5)

# Checkbutton para consentimento
consentimento_var = tk.BooleanVar() # Variável para o consentimento
check_consentimento = tk.Checkbutton(janela, text="Consentimento para uso de
dados", variable=consentimento_var)
check_consentimento.grid(row=7, columnspan=2, padx=5, pady=5)

# Botão para salvar os dados
btn_salvar = tk.Button(
    janela,
    text="Salvar Dados",
    command=lambda: salvar_dados(
        db, # Certifique-se de passar o db aqui
        entry_nome.get(),
        entry_idade.get(),
        combobox_diagnostico.get(),
        combobox_localidade.get(),
        combobox_tratamento.get(),
        combobox_mes.get(),
        consentimento_var.get(),
        datetime.now() # Adiciona a data de modificação
    )
)
btn_salvar.grid(row=8, columnspan=1, padx=5, pady=5)
```

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
# Label para mostrar a mensagem de sucesso
status_label = tk.Label(janela, text="", fg="green") # Mensagem de sucesso em
verde
status_label.grid(row=8, column=1, padx=5, pady=5) # Coluna 2 ao lado do botão

# Botão para gerar relatório PDF
btn_gerar_relatorio = tk.Button(janela, text="Gerar Relatório PDF",
command=gerar_relatorio_pdf)
btn_gerar_relatorio.grid(row=9, column=0, padx=5, pady=5) # Posicionado na linha
9, coluna 0

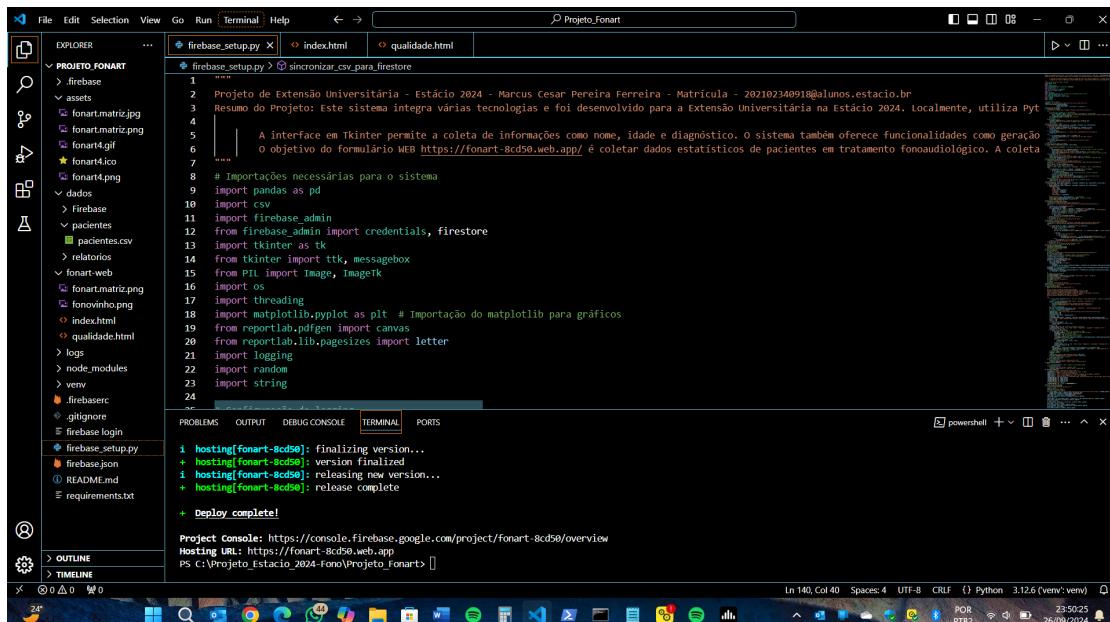
# Botão para gerar gráficos
btn_graficos = tk.Button(janela, text="Gerar Gráficos", command=mostrar_grafico)
btn_graficos.grid(row=9, column=1, padx=5, pady=5)

# Iniciar a sincronização em uma thread
threading.Thread(target=iniciar_sincronizacao, args=(db,), daemon=True).start()
janela.mainloop() # Executa a interface gráfica

if __name__ == "__main__":
    db = inicializar_firebase() # Inicializa o Firebase
    criar_janela_principal(db) # Cria a janela principal
```

Agora, o sistema estará disponível na web, sincronizado com o Firestore e com geração de relatórios e gráficos, conforme suas necessidades.

## Telas do novo sistema pronto



# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

Tela com os arquivos do sistema principal local (firebase\_setup.py)

Sistema app WEB (index.html)

Formulário de pesquisa de MELHORIAS NO SISTEMA (qualidade.html)

Firestore database

Hosting – Google Cloud

The screenshot shows a code editor interface with a sidebar containing project files and a main panel displaying the content of a Python script. The sidebar includes icons for Explorer, Search, Open, and Outline/Timeline. The main panel shows the following code:

```
1 """
2 Projeto de E
3 Resumo do Pr
4 |
5 | A in
6 | O ob
7 """
8 # Importações
9 import pandas
10 import csv
11 import firebase
12 from firebase import *
13 import tkinter
14 from tkinter import *
15 from PIL import *
16 import os
17 import thread
18 import matplotlib
19 from reportlab import *
20 from reportlab.pdfgen import *
21 import logging
22 import random
23 import string
24
25
i hosting[fonart-80
+ hosting[fonart-80
i hosting[fonart-80
+ hosting[fonart-80
+
+ Deploy complete!
```

Below the code, there are tabs for PROBLEMS and OUTPUT, and a message indicating the deployment is complete. The bottom status bar shows the project path: PS C:\Projeto\_Estac.

62

## Visão da Árvore do projeto Python

Instalada e pronta do sistema com as devidas instalações das bibliotecas e upload via firebase deploy das páginas WEB.

Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

## Sistemas Prontos e funcionais

```
firebase_setup.py x index.html qualidade.html
firebase_setup.py > sincronizar_csv_para_firestore

1 """
2 Projeto de Extensão Universitária - Estácio 2024 - Marcus Cesar Pereira Ferreira - Matrícula - 202102340918@alunos.estacio.br
3 Resumo do Projeto: Este sistema integra várias tecnologias e foi desenvolvido para a Extensão Universitária na Estácio 2024. Localmente, utiliza Pyt
4
5 A interface em Tkinter permite a coleta de informações como nome, idade e diagnóstico. O sistema também oferece funcionalidades como geração
6 O objetivo do formulário WEB https://fonart-8cd50.web.app/ é coletar dados estatísticos de pacientes em tratamento fonoaudiológico. A coleta
7 """
8 # Importações necessárias para o sistema
9 import pandas as pd
10 import csv
11 import firebase_admin
12 from firebase_admin import credentials, firestore
13 import tkinter as tk
14 from tkinter import ttk, messagebox
15 from PIL import Image, ImageTk
16 import os
17 import threading
18 import matplotlib.pyplot as plt # Importação do matplotlib para gráficos
19 from reportlab.pdfgen import canvas
20 from reportlab.lib.pagesizes import letter
21 import logging
22 import random
23 import string
24
25 # Configuração do logging
26 log_path = r'C:\Projeto_Estacio_2024-Fono\Projeto_Fonart\logs\system.log'
27 logging.basicConfig(filename=log_path,
28                     level=logging.DEBUG,
29                     format='%(asctime)s - %(levelname)s - %(message)s')
30 # Inicialização do Firebase
31 def inicializar_firebase():
32     """Inicia a conexão com o Firebase e retorna a instância do Firestore."""
33     try:
34         # Verifica se o Firebase já foi inicializado
35         if not firebase_admin._apps: # Verifica se não há aplicativos já inicializados
36             cred = credentials.Certificate('C:\Projeto_Estacio_2024-Fono\Projeto_Fonart\dados\Firebase\fonart-8cd50-firebase-adminsdk-aqui-3dd
37             firebase_admin.initialize_app(cred)
```

63

Projeto Fonart – firebase\_setup.py

C:\Projeto\_Estacio\_2024-Fono\Projeto\_Fonart\firebase\_setup.py

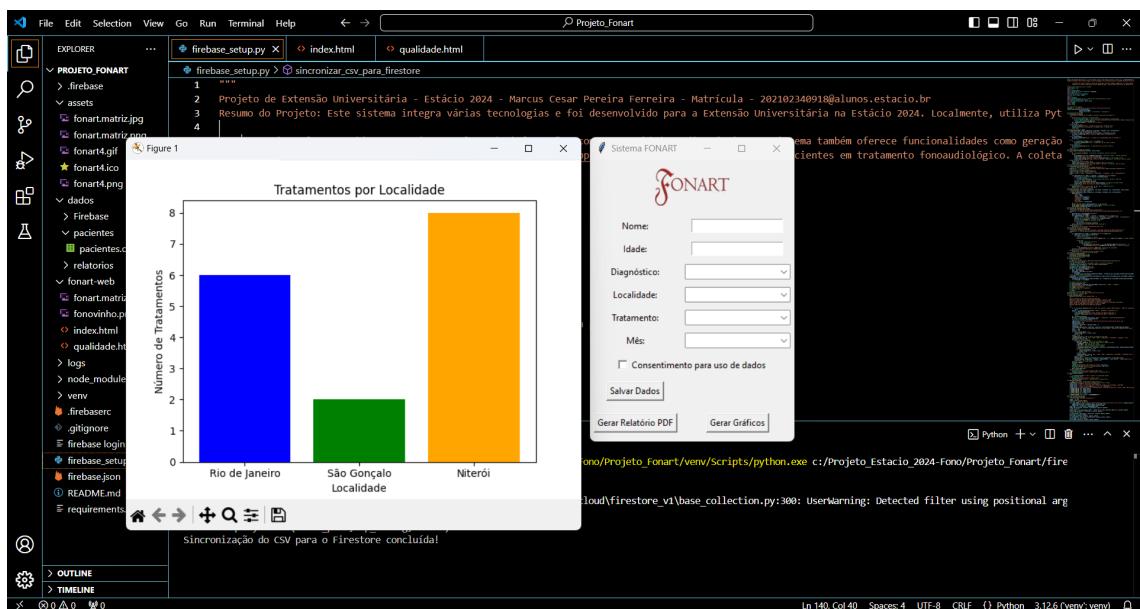
```
1 """
2 Projeto de Extensão Universitária - Estácio 2024 - Marcus Cesar Pereira Ferreira - Matrícula - 202102340919@alunos.estacio.br
3 Resumo do Projeto: Este sistema integra várias tecnologias e foi desenvolvido para a Extensão Universitária na Estácio 2024. Localmente, utiliza Pyt
4 
5 A interface em Tkinter permite a coleta de informações co
6 O objetivo do formulário WEB https://fonart-8cd50.web.app
7 """
8 # Importações necessárias para o sistema
9 import pandas as pd
10 import csv
11 import firebase_admin
12 from firebase_admin import credentials, firestore
13 import tkinter as tk
14 from tkinter import ttk, messagebox
15 from PIL import Image, ImageTk
16 import os
17 import threading
18 import matplotlib.pyplot as plt # Importação do matplotlib para
19 from reportlab.pdfgen import canvas
20 from reportlab.lib.pagesizes import letter
21 import logging
22 import random
23 import string
24

PS C:\Projeto_Estacio_2024-Fono\Projeto_Fonart> & c:/Projeto_Estacio_2024-Fono/Projeto_Fonart/venv/Scripts/python.exe c:/Projeto_Estacio_2024-Fono/Projeto_Fonart/fire
base setup.py
sincronização do Firestore para o CSV concluída!
C:\Projeto_Estacio_2024-Fono\Projeto_Fonart\venv\lib\site-packages\google\cloud\firestore_v1\base_collection.py:300: UserWarning: Detected filter using positional arg
uments. Prefer using the 'filter' keyword argument instead.
    return query.where(field_path, op_string, value)
Sincronização do CSV para o Firestore concluída!
```

## Tela do sistema pronto no computador:

Visa gerar os dados, relatórios PDF e Gráficos, com sincronização com o Firestore ([Fonart - Cloud Firestore - Dados - Console do Firebase \(google.com\)](#)), e atualização do arquivo de dados “pacientes.csv” C:\Projeto\_Estacio\_2024-Fono\Projeto Fonart\dados\relatorios\pacientes.xlsx

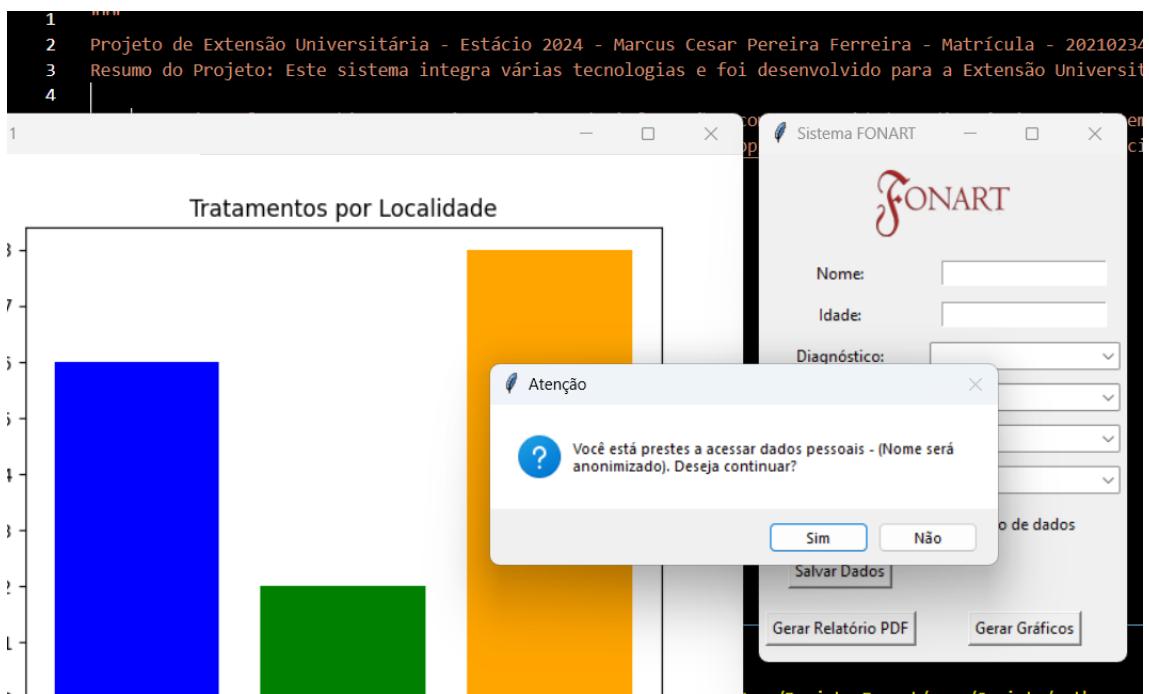
# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia



64

## Graficos e Dados gerados para o Projeto

Dados sendo gerados pelo sistema - C:\Projeto\_Estacio\_2024-Fono\Projeto\_Fonart\dados\relatorios\grafico.png



## Relatórios com Anonimização de dados

Geração de relatório PDF com anonimização do nome - C:\Projeto\_Estacio\_2024-Fono\Projeto\_Fonart\dados\relatorios\relatorio\_pacientes.pdf

## Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia



Fonart  
Espaço Fonoaudiologia Aplicada e Saúde Integral  
Relatório de Pesquisa Estatística

Num. - Nome - Idade - Diagnóstico - Localidade - Tratamento - Mês

1. 6J9X89 - 23 - Disfagia - São Gonçalo - Terapia de Deglutição - Abril
2. YXEJ41 - 34 - Disgrafia - Rio de Janeiro - Intervenção Precoce - Março
3. N4JLGM - 45 - Afasia - Niterói - Terapia de Voz - Janeiro
4. DAC4XY - 55 - Afasia - Niterói - Terapia de Leitura e Escrita - Julho
5. WDL9OA - 34 - Dislexia - Rio de Janeiro - Terapia de Voz - Novembro
6. W2B90W - 34 - Afasia - Niterói - Terapia de Deglutição - Novembro
7. 2HZX28 - 24 - Afasia - Niterói - Terapia de Voz - Março
8. YW4DXH - 56 - Dislexia - Rio de Janeiro - Terapia de Leitura e Escrita - Dezembro

FONART - Maxxi Business Freguesia  
Estr. dos Trés Rios, 1097 - Sala 530, Freguesia de Jacarepaguá, 22745-004  
E-mail: fonartrio@gmail.com - Tel.: 21 9 7282-2546  
Relatório gerado via tecnologia Python, Extensão Universitária - Estácio 2024 - Marcus Cesar Ferreira

Página 1

65

### PDF gerado com anonimização

PDF gerado pelo sistema com a base de dados atualizável do pacientes.csv, com layout solicitado, cabeçalho, rodapé e logotipo. C:\Projeto\_Estacio\_2024-Fono\Projeto\_Fonart\dados\relatorios\relatorio\_pacientes.pdf

Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

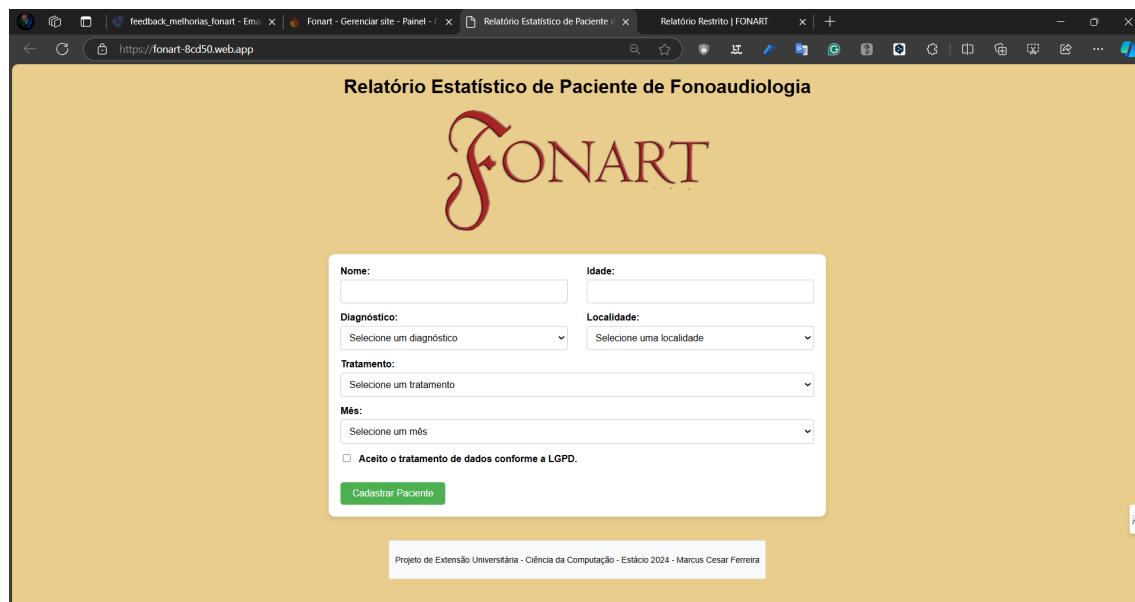
```
❖ firebase_setup.py | ❖ index.html | ❖ qualidade.html | 
fonart-web > ❖ index.html > {} "index.html" > ✎ html > ✎ head > ✎ script > ✎ document.addEventListener('DOMContentLoaded') callback > ✎ idadeInput.addEventListener('input') callback
1  <!DOCTYPE html>
2  <html lang="pt-BR">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Relatório Estatístico do Paciente de Fonoaudiologia</title>
7
8      <link rel="stylesheet" href="styles.css">
9
10     <script type="module">
11         import { initializeApp } from "https://www.gstatic.com/firebasejs/9.20.0/firebase-app.js";
12         import { getFirestore, collection, addDoc } from "https://www.gstatic.com/firebasejs/9.20.0.firebaseio_firestore.js";
13
14         const firebaseConfig = {
15             apiKey: "AIzaSyD0_td7R5bSLXg_25FNjf7Tj8YZINC5t4",
16             authDomain: "fonart-8cd50.firebaseio.com",
17             projectId: "fonart-8cd50",
18             storageBucket: "fonart-8cd50.appspot.com",
19             messagingSenderId: "359305470538",
20             appId: "1:359305470538:web:080940a6afe1df0c5fcf3",
21             measurementId: "G-H6Z37MJL7D"
22         };
23
24         const app = initializeApp(firebaseConfig);
25         const db = getFirestore(app);
26
27         document.addEventListener('DOMContentLoaded', function () {
28             const idadeInput = document.getElementById('idade');
29             idadeInput.addEventListener('input', function () {
30                 this.value = this.value.replace(/\B(\d{3})\B/g, ',');
31                 if (this.value.length > 3) {
32                     this.value = this.value.slice(0, 3);
33                 }
34                 const idade = parseInt(this.value, 10);
35                 if (idade > 100) {
36                     this.value = '';
37                 } else if (idade < 0) {

```

66

## Sincronização total do Projeto

Página index.html “espelhando” o sistema local firebase\_setup.py para o APP WEB, firestore hosting. Do Sistema local C:\Projeto\_Estacio\_2024-Fono\Projeto\_Fonart\fonart-web\index.html para WEB - [Relatório Estatístico de Paciente de Fonoaudiologia \(fonart-8cd50.web.app\)](#)



Páginas de internet montadas e geradas (espelho do sistema local)

Relatório Estatístico de Paciente de Fonoaudiologia (fonart-8cd50.web.app)  
(Firebase)

## Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

9 7282-2546 [fonart.rio](#)

FONART Relatório Restrito Cláudia Cesar Contatos

Relatório Estatístico de Paciente de Fonoaudiologia

**FONART**

Nome: \_\_\_\_\_ Idade: \_\_\_\_\_

Diagnóstico: \_\_\_\_\_ Localidade: \_\_\_\_\_

Tratamento: \_\_\_\_\_

Mês: \_\_\_\_\_

Aceito o tratamento de dados conforme a LGPD.

Cadastrar Paciente

Projeto de Extensão Universitária - Ciência da Computação - Estácio 2024 - Marcus Cesar Ferreira

**Fusão do sistema Web Firebase com o domínio oficial da empresa – fonart.com.br**

[Relatório Restrito | FONART \(DOMÍNIO\)](#)

Código pronto Python do sistema Relatórios via App WEB

### Index.html

Tela com o sistema inserido no site [fonart.com.br](http://fonart.com.br) em plataforma WIX, compondo página, com link do “Relatório” através da inserção do link [html fonart-8cd50.web.app](http://fonart-8cd50.web.app)

Código Python

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Relatório Estatístico de Paciente de Fonoaudiologia</title>

  <link rel="stylesheet" href="styles.css">

<script type="module">
  import { initializeApp } from "https://www.gstatic.com/firebasejs/9.20.0/firebase-app.js";
  import { getFirestore, collection, addDoc } from "https://www.gstatic.com/firebasejs/9.20.0.firebaseio-firestore.js";

  const firebaseConfig = {
    apiKey: "AlzaSyDo_td7R5bSLZXg_25FNjf7Tj8yZINc5T4",
    authDomain: "fonart-8cd50.firebaseioapp.com",
    projectId: "fonart-8cd50",
    storageBucket: "fonart-8cd50.appspot.com",
    messagingSenderId: "359305470538",
    appId: "1:359305470538:web:080940a6afe1df09c5fc3",
    measurementId: "G-H6Z37MJL7D"
  };

```

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
const app = initializeApp(firebaseConfig);
const db = getFirestore(app);

document.addEventListener('DOMContentLoaded', function () {
  const idadeInput = document.getElementById('idade');
  idadeInput.addEventListener('input', function () {
    this.value = this.value.replace(/\[^0-9]/g, '');
    if (this.value.length > 3) {
      this.value = this.value.slice(0, 3);
    }
    const idade = parseInt(this.value, 10);
    if (idade > 100) {
      this.value = '';
    } else if (idade < 0) {
      this.value = 0;
    }
  });
});

document.getElementById('pacienteForm').addEventListener('submit', async function (e) {
  e.preventDefault();
  const nome = document.getElementById('nome').value;
  const idade = idadeInput.value;
  const diagnostico = document.getElementById('diagnostico').value;
  const localidade = document.getElementById('localidade').value;
  const tratamento = document.getElementById('tratamento').value;
  const mes = document.getElementById('mes').value;
  const consentimento = document.getElementById('consentimento').checked ? "Sim" : "Não";

  if (!document.getElementById('consentimento').checked) {
    alert("É necessário dar consentimento para continuar.");
    return;
  }

  try {
    const docRef = await addDoc(collection(db, "pacientes"), {
      nome,
      idade,
      diagnostico,
      localidade,
      tratamento,
      mes,
      consentimento
    });
    alert("Paciente cadastrado com sucesso! ID: " + docRef.id);
  } catch (error) {
    console.error("Erro ao adicionar documento: ", error);
    alert("Erro ao cadastrar paciente. Tente novamente.");
  }
});
});
</script>
<style>
/* Projeto de Extensão Universitária - Estácio 2024 - Marcus Cesar Pereira Ferreira - Matrícula - 202102340918@alunos.estacio.br
Resumo do Projeto: O sistema integra várias tecnologias e foi desenvolvido para a Extensão Universitária na Estácio 2024.
Localmente, utiliza Python firebase_setup.py para integração com o Firebase do Google Cloud e o Cloud Firestore,
armazenando dados de pacientes de fonoaudiologia.
A interface em Tkinter permite a coleta de informações como nome, idade e diagnóstico. O sistema também oferece
funcionalidades como geração de relatórios em PDF e gráficos de análise, além de sincronizar automaticamente os dados locais e na
nuvem.
Este formulário utiliza HTML e JavaScript para encaminhar os dados ao Cloud Firestore, que os valida e armazena. O objetivo
do formulário WEB https://fonart-8cd50.web.app/ é coletar dados estatísticos de pacientes em tratamento fonoaudiológico.
A coleta de informações como idade, diagnóstico, localidade e tratamento é essencial para entender as necessidades dos
pacientes e aprimorar a atuação da equipe. O consentimento do paciente é obrigatório para a utilização dos dados, que são tratados
com a máxima confidencialidade, visando à conformidade com a LGPD.*/
```

```
/* Configuração do layout com Flexbox */
body {
  font-family: Arial, sans-serif;
  background-color: #E9CD8D;
  margin: 0;
```

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
padding: 0;
display: flex;
flex-direction: column;
align-items: center;
}

h1 {
  text-align: center;
  margin-top: 20px;
}

img.logo {
  max-width: 40%; /* Para responsividade */
  height: auto; /* Mantém a proporção da imagem */
  margin-bottom: 20px; /* Espaçamento abaixo do logotipo */
}

form {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between;
  max-width: 800px;
  margin: 20px;
  background-color: white;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

.form-group {
  display: flex;
  flex-direction: column;
  margin-bottom: 15px;
  width: 48%; /* Campos lado a lado */
}

.form-group.full-width {
  width: 100%;
}

label {
  margin-bottom: 5px;
  font-weight: bold;
}

input, select {
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
  font-size: 16px;
}

input[type="checkbox"] {
  width: auto;
  margin-right: 10px;
}

button {
  padding: 10px 20px;
  background-color: #4CAF50;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  font-size: 16px;
  margin-top: 10px;
}

button:hover {
  background-color: #45a049;
```

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
}
```

```
footer {  
    margin-top: 20px;  
    padding: 10px;  
    background-color: #f8f9fa;  
    text-align: center;  
    border-top: 1px solid #dee2e6;  
    max-width: 800px;  
}  
  
footer p {  
    font-size: 14px;  
}  
/style>  
</head>  
<body>  
    <h1>Relatório Estatístico de Paciente de Fonoaudiologia</h1>  
      
  
    <form id="pacienteForm">  
        <div class="form-group">  
            <label for="nome">Nome:</label>  
            <input type="text" id="nome" name="nome" required>  
        </div>  
        <div class="form-group">  
            <label for="idade">Idade:</label>  
            <input type="number" id="idade" name="idade" min="0" max="100" required>  
        </div>  
        <div class="form-group">  
            <label for="diagnostico">Diagnóstico:</label>  
            <select id="diagnostico" name="diagnostico" required>  
                <option value="" disabled selected>Selecione um diagnóstico</option>  
                <option value="Disartria">Disartria</option>  
                <option value="Afasia">Afasia</option>  
                <option value="Dislexia">Dislexia</option>  
                <option value="Disgrafia">Disgrafia</option>  
                <option value="Disfagia">Disfagia</option>  
                <option value="Gagueira">Gagueira</option>  
                <option value="Autismo">Autismo</option>  
            </select>  
        </div>  
        <div class="form-group">  
            <label for="localidade">Localidade:</label>  
            <select id="localidade" name="localidade" required>  
                <option value="" disabled selected>Selecione uma localidade</option>  
                <option value="Rio de Janeiro">Rio de Janeiro</option>  
                <option value="Niterói">Niterói</option>  
                <option value="São Gonçalo">São Gonçalo</option>  
            </select>  
        </div>  
        <div class="form-group full-width">  
            <label for="tratamento">Tratamento:</label>  
            <select id="tratamento" name="tratamento" required>  
                <option value="" disabled selected>Selecione um tratamento</option>  
                <option value="Terapia de Linguagem">Terapia de Linguagem</option>  
                <option value="Terapia de Voz">Terapia de Voz</option>  
                <option value="Terapia de Deglutição">Terapia de Deglutição</option>  
                <option value="Reabilitação da Audição">Reabilitação da Audição</option>  
                <option value="Terapia de Articulação">Terapia de Articulação</option>  
                <option value="Intervenção Precoce">Intervenção Precoce</option>  
                <option value="Terapia de Leitura e Escrita">Terapia de Leitura e Escrita</option>  
            </select>  
        </div>  
        <div class="form-group full-width">  
            <label for="mes">Mês:</label>  
            <select id="mes" name="mes" required>  
                <option value="" disabled selected>Selecione um mês</option>  
                <option value="Janeiro">Janeiro</option>  
                <option value="Fevereiro">Fevereiro</option>  
            </select>  
        </div>
```

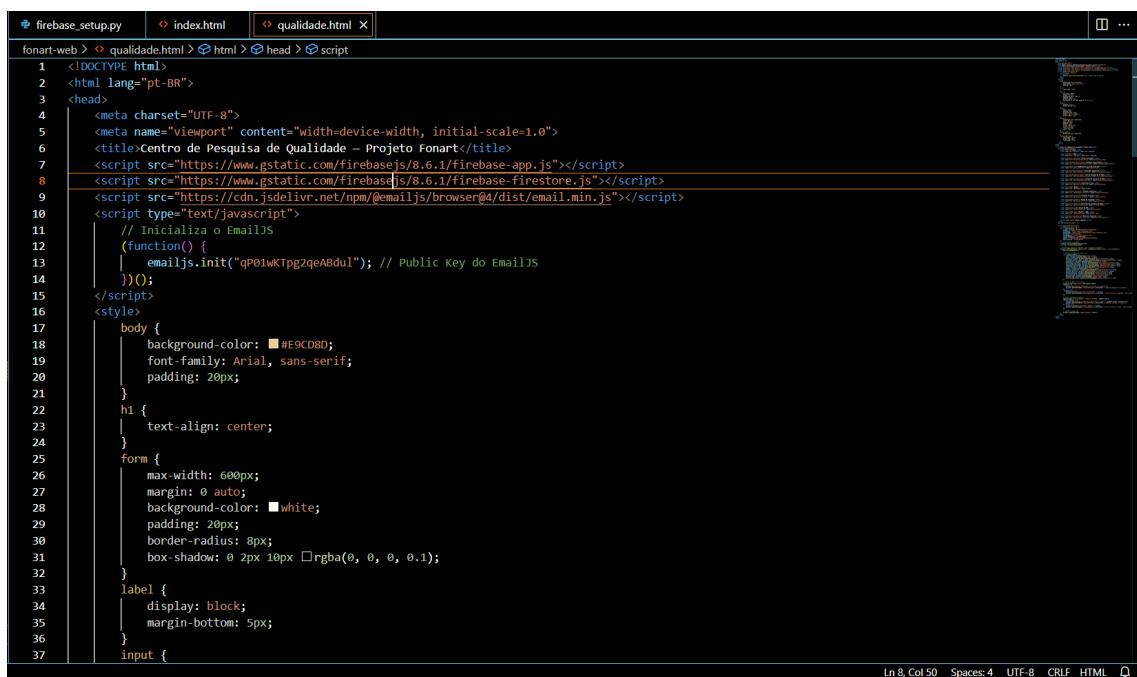
# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
<option value="Março">Março</option>
<option value="Abril">Abril</option>
<option value="Maio">Maio</option>
<option value="Junho">Junho</option>
<option value="Julho">Julho</option>
<option value="Agosto">Agosto</option>
<option value="Setembro">Setembro</option>
<option value="Outubro">Outubro</option>
<option value="Novembro">Novembro</option>
<option value="Dezembro">Dezembro</option>
</select>
</div>
<div class="form-group full-width">
    <label for="consentimento">
        <input type="checkbox" id="consentimento" name="consentimento" required>
        Aceito o tratamento de dados conforme a LGPD.
    </label>
</div>
<button type="submit">Cadastrar Paciente</button>
</form>

<footer>
    <p>Projeto de Extensão Universitária - Ciência da Computação - Estácio 2024 - Marcus Cesar Ferreira</p>
</footer>
</body>
</html>
```

71

## Web APP do Formulário de Pesquisa de melhorias (projeto)



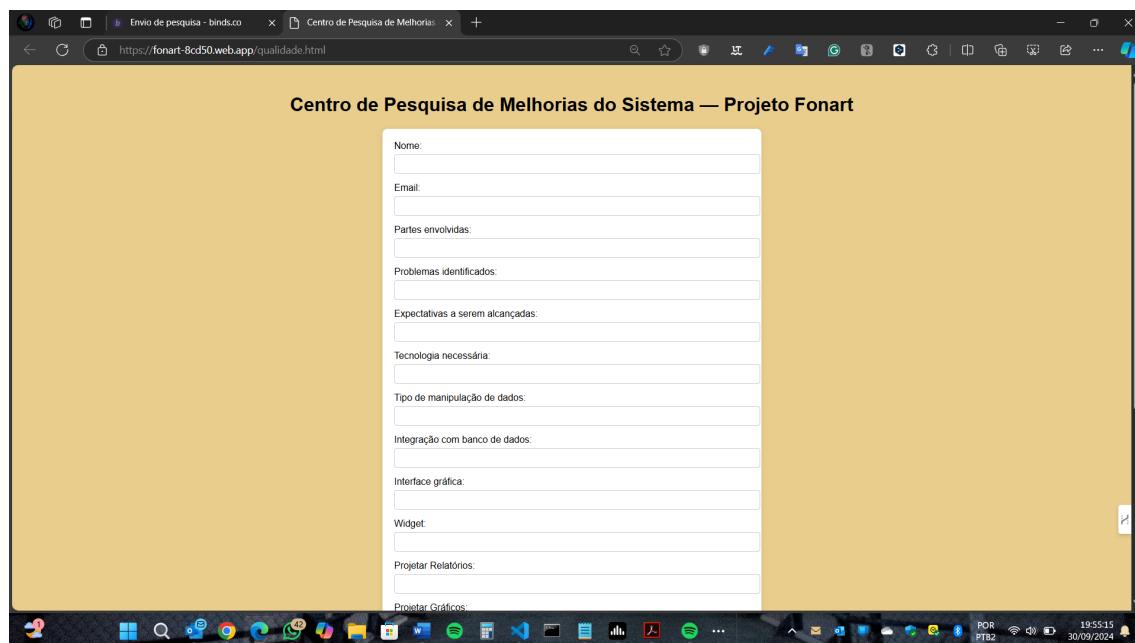
```
firebase_setup.py | index.html | qualidade.html
fonart-web > qualidade.html > html > head > script
1  <!DOCTYPE html>
2  <html lang="pt-BR">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Centro de Pesquisa de Qualidade – Projeto Fonart</title>
7      <script src="https://www.gstatic.com/firebasejs/8.6.1.firebaseio-app.js"></script>
8      <script src="https://www.gstatic.com/firebasejs/8.6.1/firebase-firestore.js"></script>
9      <script src="https://cdn.jsdelivr.net/npm/@emailjs/browser@4/dist/email.min.js"></script>
10     <script type="text/javascript">
11         // Inicializa o EmailJS
12         (function() {
13             |   emailjs.init("qp01wKtpg2qeABdul"); // Public Key do EmailJS
14         })();
15     </script>
16     <style>
17         body {
18             background-color: #E9CD8D;
19             font-family: Arial, sans-serif;
20             padding: 20px;
21         }
22         h1 {
23             text-align: center;
24         }
25         form {
26             max-width: 600px;
27             margin: 0 auto;
28             background-color: white;
29             padding: 20px;
30             border-radius: 8px;
31             box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
32         }
33         label {
34             display: block;
35             margin-bottom: 5px;
36         }
37         input {
```

Ln 8, Col 50 Spaces: 4 UTF-8 CRLF HTML □

### Qualidade.html

C:\Projeto\_Estacio\_2024-Fono\Projeto\_Fonart\fonart-web\qualidade.html

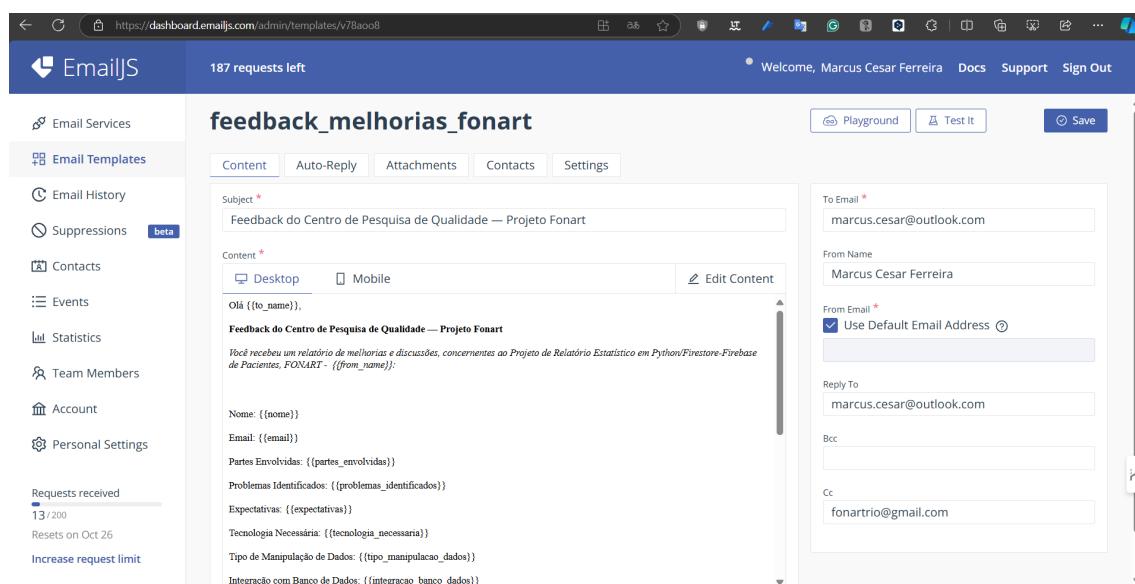
## Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia



**Ambiente do Centro de MELHORIAS DO SISTEMA (solicitações, melhorias, desenvolvimento etc.).**

[Centro de Pesquisa de Melhorias — Projeto Fonart \(fonart-8cd50.web.app\)](https://fonart-8cd50.web.app/qualidade.html)

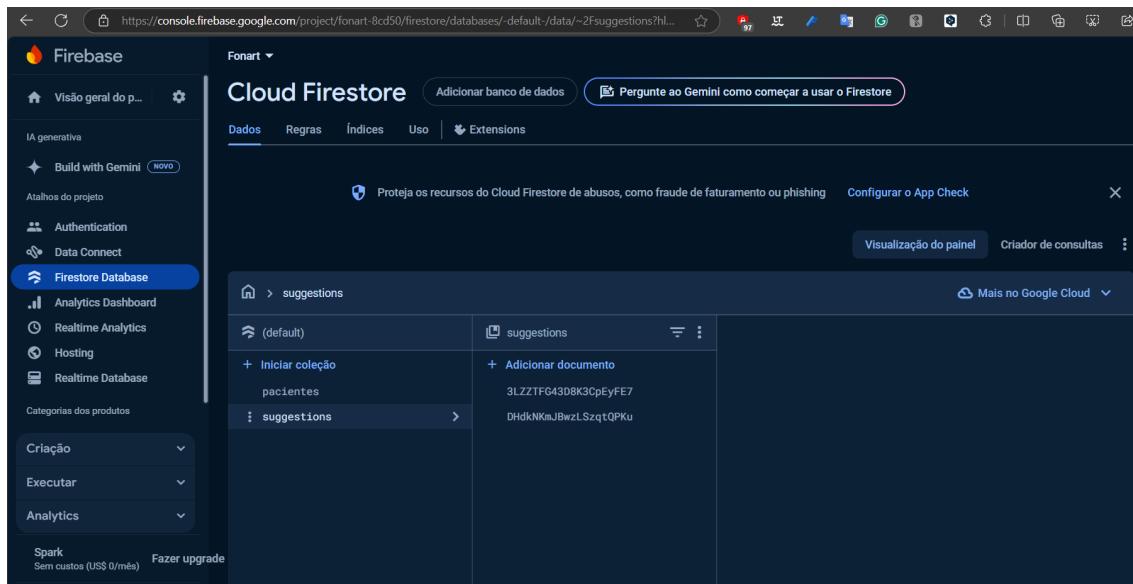
**Ambiente responsável por recolher informações de melhorias, tecnologias solicitadas, críticas do sistema, necessidades administrativas e tecnológicas para o sistema de Relatórios em atendimentos fonoaudiológicos. – Desenvolvido no ambiente WEB do Firestore, com envio automático de dados para o Firebase - [Fonart - Cloud Firestore - Dados - Console do Firebase \(google.com\)](#), simultaneamente, encaminha um e-mail para o desenvolvedor Marcus Cesar Ferreira (Estácio). Utilizando Tecnologia EmailJS, visando replicar os dados com o [feedback melhorias fonart - EmailJS](#)**



# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

## Sincronização com EmailJS e Firebase

**Painel de desenvolvimento do questionário EmailJS, retroalimentado pelo formulário Firestore Cloud Google.**



The screenshot shows the Firebase Cloud Firestore interface. On the left sidebar, 'Firestore Database' is selected under the 'Cloud Firestore' section. The main area displays the 'suggestions' collection, which contains two documents: '3LZZTFG43D8K3CpEyFE7' and 'DHdKNKmJBwzLSzqtQPKu'. A green circle in the top right corner contains the number '73'.

Este é o Painel Fonart - Cloud Firestore - Dados - Console do Firebase (google cloud), com dois centros de captação de dados:

- 1. Relatório de Pacientes:** Sincroniza os dados com o arquivo local no computador, localizado em "C:\Projeto\_Estacio\_2024-Fono\Projeto\_Fonart\dados\pacientes\pacientes.csv", e, com o Relatório Estatístico acessível via o Web app **Relatório Estatístico de Pacientes de Fonoaudiologia** (fonart-8cd50.web.app) - (coleção "pacientes").
- 2. Formulário de Melhorias:** Disponível no Web app **Centro de Pesquisa de Melhorias do Sistema — Projeto Fonart** (fonart-8cd50.web.app/qualidade.html) e integrado ao serviço de e-mails **EmailJS** (coleção "suggestions").

## CIENCIA DA COMPUTAÇÃO - O PROJETO DE EXTENSÃO UNIVERSITARIA – ESTÁCIO DE SÁ

### PLANO DE APRENDIZAGEM VISANDO MELHORIAS NA EMPRESA FONART - ESPAÇO FONOAUDIOLOGIA APLICADA E SAÚDE INTEGRAL

1. Código e nome da disciplina:

### DGT0235 DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES EM PYTHON

2. Área temática:

Em atendimento à Resolução CNE/CES nº 7, de 18 de dezembro de 2018, que Estabelece as Diretrizes para a Extensão na Educação Superior Brasileira e dá outras providências, a área temática priorizada neste Plano é tecnologia, produção e trabalho.

### **3. Natureza:**

Extensão:

### **4. Competências a serem trabalhadas:**

Com base na proposta institucional para a formação do egresso e as competências gerais e específicas desenvolvidas no curso, previstas em seu PPC, e em consonância com a Resolução CNE/CES nº 7, de 18 de dezembro de 2018, as competências que serão trabalhadas neste componente serão prioritariamente:

Competências técnicas: Desenvolvimento de soluções computacionais em Python, conhecimento em Desenvolvimento Rápido de Aplicações (RAD) e familiaridade com frameworks de RAD compatíveis com Python para criação de interfaces de usuário rápidas e interativas. Competências socioemocionais: trabalho em equipe; capacidade de lidar com conflitos; o aprendizado contínuo e colaborativo; e a liderança para a execução de ações nos diversos ambientes organizacionais.

### **5. Ementa**

RAD (Rapid Applications Development); Manipulação de dados em arquivos; Python com banco de dados; Interface gráfica com python; Aplicando RAD.

### **6. Objetivos**

- a) Desenvolver a capacidade do aluno de entender e desenvolver programas em Python através do modelo RAD, acelerando o desenvolvimento.
- b) Conhecer o conceito de script em Python
- c) Desenvolver programas estruturados em Python.
- d) Desenvolver programas orientados a objetos em Python
- e) Aplicar o aprendizado em várias áreas de conhecimento.
- f) Desenvolver aplicações seguindo o modelo RAD.

### **7. OBJETIVOS SOCIOCOMUNITÁRIOS DA EMPRESA FONART**

Desenvolvimento de soluções computacionais em Python que abordem desafios específicos enfrentados por uma clínica particular de Fonoaudiologia, chamada FONART ESPAÇO FONOAUDIOLOGIA APLICADA E SAÚDE INTEGRAL CNPJ 19.064.473/0001-80, localizado na Freguesia do Jacarepaguá, sendo responsável por apresentar e elaborar relatórios estatísticos anonimizados, para análise de dados de diagnósticos, tratamentos por regiões, para redes públicas e particulares no Rio de Janeiro, Niterói e São Gonçalo.

Apoiamos o ganho de competitividade das partes envolvidas, considerando o perfil dos clientes da FONART e das suas transações, para viabilizar contatos com os clientes de forma efetiva, de acordo com estratégias estabelecidas.

O código Python e demais soluções de produtividade, deve contribuir com a tomada de decisão ágil e assertiva, com base em dados, determinar quais são as áreas que merecem investimentos, neste caso, o relatório de estatísticas de pacientes, para estudos mais automatizado e íntegro, para substituir a manipulação manual de dados, por papel ou word, visando apresentar um ambiente informático moderno, acessível em computador local e internet para usuários do sistema, ou seja, proporcionar o acesso facilitado e tecnológico, para apresentar relatório moderno para os profissionais de organismos públicos e privados, minimizando todos os custos e despesas que podem ser reduzidos sem afetar a qualidade das atividades realizadas pelos colaboradores.

Neste item, esta solução apresenta **investimento de um computador moderno, montagem e modelam da solução**, conforme a necessidade identificada do sistema Python, instalado localmente neste computador, um **site de internet da empresa** compreendendo um link para o sistema WEB nos seguintes endereços:

<https://www.fonart.com.br/relatorio-restrito>;

<https://www.fonart.com.br/>

<https://fonart-8cd50.web.app/> (WEB app do Firestore/Firebase), este, apresentam o formulário que envia os dados para o Firestore - Firebase no Cloud Google.

Foi utilizado localmente o arquivo em:

C:\Projeto\_Estacio\_2024-Fono\Projeto\_Fonart\firebase\_setup.py;

C:\Projeto\_Estacio\_2024-

Fono\Projeto\_Fonart\dados\relatorios\relatorio\_pacientes.pdf;

C:\Projeto\_Estacio\_2024-Fono\Projeto\_Fonart\dados\relatorios\pacientes.xlsx;

C:\Projeto\_Estacio\_2024-Fono\Projeto\_Fonart\dados\relatorios\grafico.png;

C:\Projeto\_Estacio\_2024-Fono\Projeto\_Fonart\dados\pacientes\pacientes.csv;

C:\Projeto\_Estacio\_2024-Fono\Projeto\_Fonart\README.md;

C:\Projeto\_Estacio\_2024-Fono\Projeto\_Fonart\dados\Firebase\fonart-8cd50-firebase-adminsdk-8quaui-3dd6e0939e.json;

C:\Projeto\_Estacio\_2024-Fono\Projeto\_Fonart\logs\system.log;

C:\Projeto\_Estacio\_2024-Fono\Projeto\_Fonart\fonart-web\index.html; e

Configurações para sincronização com o Firestore/Firebase compreendendo os endereços em:

<https://console.firebaseio.google.com/project/fonart-8cd50/firestore/databases/-default-/data/2FpacientesNxHOIXXXXXXXXXP9o7fhl=pt-br> e

### **Instalação do Apps da Web - Web App Fonart.**

For Firebase JS SDK v7.20.0 and later, measurementId is optional

```
const firebaseConfig = {
```

## Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
apiKey: "AlzaSyDo_td7R5bSLZXg_25Fxxxxxxxxxxxxx",
authDomain: "fonart-8cd50.firebaseio.com",
databaseURL: "https://fonart-8cd50-default-rtdb.firebaseio.com",
projectId: "fonart-8cd50",
storageBucket: "fonart-8cd50.appspot.com",
messagingSenderId: "3593xxxxxxxx",
appId: "1:359305470538:web:080xxxxxxxxxxxxxx",
measurementId: "G-H6Z37MJL7D"
}; e Conas de Serviço Python import firebase_admin
from firebase_admin import credentials
cred = credentials.Certificate("path/to/serviceAccountKey.json")
firebase_admin.initialize_app(cred) .
```

76

**As despesas com o projeto para a empresa FONART foram reduzidas sem afetar a qualidade das atividades realizadas pelos colaboradores.**

### **8. Descrição do público envolvido:**

O público externo à IES, que chamaremos de partes envolvidas, e implicado na ação proposta é composto por privadas, (empresa, clínica de fonoaudiologia **FONART ESPAÇO FONOAUDIOLOGIA APlicada E SAÚDE INTEGRAL**), compreendendo um microempreendedor individual, sócia da empresa, membro da comunidade local, no entorno da instituição Polo de ensino à distância da Universidade Estácio de Sá, em Freguesia, Jacarepaguá - RJ.

### **9. Justificativa:**

De acordo com os artigos 3º e 6º do Capítulo I da Resolução CNE/CES nº 7, de 18 de dezembro de 2018, a Extensão na Educação Superior Brasileira ao integrar a matriz curricular e à organização de pesquisa, promove, em um processo interdisciplinar, a formação integral do aluno, através da aprendizagem por projetos, que estabelece um diálogo construtivo e transformador com diferentes setores da sociedade brasileira e internacional. Esse componente na formação do aluno justificasse pela importância de promover a atuação da comunidade acadêmica e técnica, a partir das demandas socio comunitárias onde se encontra a IES, para o enfrentamento das questões da sociedade brasileira, inclusive por meio do desenvolvimento econômico, social e cultural.

A atividade extensionista busca promover a transformação digital por meio da integração interdisciplinar entre teoria e prática, contextualizada à realidade das partes envolvidas e guiada por princípios socioemocionais. Essa abordagem busca promover um diálogo amistoso e mutuamente benéfico, visando o desenvolvimento conjunto de soluções que atendam às necessidades de todas as partes envolvidas.

## Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

Do ponto de vista acadêmico, para que seja possível desenvolver esta atividade, será necessário ampla articulação entre teoria e prática através de uma visão interdisciplinar, dialogando com várias disciplinas. Por exemplo: Paradigmas de Linguagem de Programação em Python, Pensamento Computacional, Estrutura de Dados, Introdução à Segurança da Informação e Desenvolvimento Web em HTML5, CSS, Javascript e PHP.

No que se refere à relevância social, criamos soluções inovadoras para problemas sociais em áreas como, saúde. Por exemplo, desenvolver aplicativos para promover os relatórios oficiais para a melhor visão das deficiências locais ou regionais, propagar soluções e mitigar riscos na área da saúde fonoaudiológica e conscientização sobre o horizonte dos eventos das patologias. Essas aplicações proporcionaram impacto positivo na sociedade, melhorando a vida das pessoas e abordando questões sociais relevantes.

No que se refere a aplicação do projeto em empresas locais, neste caso, **Clínica de Fonoaudiologia FONART**, a atividade possibilitará às organizações ao seu redor, e alunos, a análise das informações de seus clientes, **principalmente pacientes com privacidade**, facilitando a tomada de decisões mais ágeis e precisas. Além disso, foi possível avaliar as necessidades da empresa e clientes, a satisfação dos clientes, e da empresa, via formulário de qualidade o que permitiu a entrega de produtos mais alinhados com suas expectativas, com a solução proposta acima.

### **10. Procedimentos de ensino-aprendizagem:**

Esta disciplina adota a metodologia de aprendizagem baseada em projetos, construídos de forma dialógica com a sociedade (representada pela parte envolvida) de forma a atender aos objetivos citados anteriormente. O estudo das teorias e práticas previstas na disciplina para a aptidão de desenvolvimento desta atividade extensionista, serão realizadas em paralelo com as atividades extensionistas para alcançar os objetivos técnicos e objetivos sociocomunitários previstos neste Plano de Aprendizagem. Dessa forma viabilizasse o desenvolvimento concomitante das habilidades técnicas e da atividade extensionista prevista para esta disciplina.

### **ETAPAS DO DESENVOLVIMENTO REALIZADO NESTA ATIVIDADE EXTENSIONISTA PARA A EMPRESA FONART - ESPAÇO FONOAUDIOLOGIA APlicada E SAÚDE INTEGRAL:**

- I. Diagnóstico e teorização, conforme detalhado neste Roteiro de Extensão e realização do Laboratório de Extensão da Sala de Aula Virtual.
- II. Planejamento e desenvolvimento do projeto, conforme detalhado no neste Roteiro de Extensão.
- III. Encerramento do Projeto.

### **Realização das atividades de extensão na empresa - FONART - ESPAÇO FONOAUDIOLOGIA APlicada E SAÚDE INTEGRAL:**

## **11. Os Estudos dos conteúdos digitais da disciplina;**

Nosso código precisava cumprir os Temas de aprendizagem, ou seja, o código e as soluções propostas devem cumprir neste Projeto que foi criado para a empresa **FONART**

- **ESPAÇO FONOAUDIOLOGIA APLICADA E SAÚDE INTEGRAL**, nesta proposta, criamos o PYTHON, com O CÓDIGO QUE FORNECEREMOS PARA A ANÁLISE AQUI. ESTE CÓDIGO, QUE CRIAMOS, DEVE E PRECISA CUMPRIR TODAS AS EXIGÊNCIAS ACADÊMICAS E TÉCNICAS E DISPONIBILIZAREMOS APÓS O ENTENDIMENTO DESTA ANÁLISE. O CÓDIGO CONFORME O QUE DECLARAMOS ACIMA ABRANGE:

78

### **1. RAD (RAPID APPLICATIONS DEVELOPMENT):**

1.1 A CONTEXTUALIZAÇÃO, OS CONCEITOS, PRINCÍPIOS, AS FERRAMENTAS E TÉCNICAS DA METODOLOGIA DE DESENVOLVIMENTO RÁPIDO DE SOFTWARE

1.2 AS FASES DA RAD

1.3 QUANDO APLICAR E QUANDO NÃO APLICAR RAD

1.4 O PYTHON E AS FERRAMENTAS (FRAMEWORK)

### **2. MANIPULAÇÃO DE DADOS EM ARQUIVOS:**

2.1 MANIPULAÇÃO DE ARQUIVOS

2.2 MANIPULAÇÃO DE STRINGS

2.3 DESCREVER AS EXCEÇÕES NA MANIPULAÇÃO DE ARQUIVOS E OUTRAS OPERAÇÕES

### **3. PYTHON COM BANCO DE DADOS:**

3.1 FRAMEWORKS E BIBLIOTECAS PARA GERENCIAMENTO DE BANCO DE DADOS

3.2 CONEXÃO, ACESSO E CRIAÇÃO DE BANCOS DE DADOS E TABELAS

3.3 INSERÇÃO, REMOÇÃO E ATUALIZAÇÃO DE REGISTROS EM TABELAS

3.4 RECUPERAÇÃO DE REGISTROS EM TABELAS

### **4. INTERFACE GRÁFICA COM PYTHON:**

4.1 PRINCIPAIS FRAMEWORKS E BIBLIOTECAS NECESSÁRIAS PARA A GUI

4.2 ADIÇÃO DE WIDGETS E MONTAGEM DA INTERFACE GRÁFICA

4.3 INTERFACE PARA INCLUSÃO DE DADOS EM UMA TABELA NO BANCO DE DADOS

4.4 INTERFACE PARA LOCALIZAÇÃO, ALTERAÇÃO E EXCLUSÃO DE DADOS EM TABELA

### **5. APLICANDO RAD:**

5.1 AS ETAPAS PARA TRATAMENTO DOS REQUISITOS DE UM SISTEMA NA METODOLOGIA DE DESENVOLVIMENTO:

5.2 AS MODELAGENS DE NEGÓCIOS E DE DADOS DA RAD

5.3 O DESIGN DE INTERFACE COM O USUÁRIO NA RAD

5.4 APLICAÇÃO RAD IMPLEMENTADA EM PYTHON

## **12. Procedimentos de avaliação:**

Contemplar as competências desenvolvidas por meio da realização neste Atividade de Extensão.

**Os critérios utilizados para a avaliação dos alunos na Atividade de Extensão serão:**

Linguagem adequada, correção ortográfica, clareza e objetividade; autoavaliação discente com relato das experiências vivenciadas durante a participação na atividade de extensão.

---

---

## RESULTADOS

79

### **1. RAD (Rapid Application Development):**

#### **- Aplicação do RAD:**

O código segue o princípio de desenvolvimento rápido. Ele usa bibliotecas e frameworks existentes (Tkinter para a interface gráfica, ReportLab para relatórios em PDF, Firebase para o banco de dados) e combina essas ferramentas em um sistema integrado.

#### **- Cumprimento:**

O projeto adere ao modelo RAD, entregando funcionalidades rapidamente por meio de componentes reutilizáveis e um processo iterativo.

#### **- Ferramentas RAD Compatíveis com Python:**

O uso de Tkinter, Firestore (Firebase) e manipulação de arquivos CSV permite um desenvolvimento ágil e de fácil modificação.

#### **- Cumprimento:**

O código utiliza ferramentas e frameworks compatíveis com Python, otimizando o processo de desenvolvimento rápido.

### **2. Manipulação de Dados em Arquivos:**

#### **- Manipulação de Arquivos CSV:**

O sistema salvo os dados dos pacientes em um arquivo CSV local e oferece a funcionalidade de sincronização com o banco de dados Firebase, o que demonstra o uso eficaz de arquivos para armazenamento de dados temporários.

#### **- Cumprimento:**

O código manipula arquivos de forma eficaz, criando, atualizando e validando a estrutura CSV conforme exigido.

#### **- Exceções:**

O código trata exceções ao manipular arquivos e ao realizar operações no Firebase. O uso de logs para registrar erros garante a detecção e tratamento de falhas.

**- Cumprimento:**

A manipulação de exceções está implementada corretamente, principalmente no que se refere à manipulação de arquivos e operações críticas.

**3. Python com Banco de Dados:**

**- Integração com o Firestore:**

O código conecta-se ao Firebase, sincronizando dados entre o CSV local e a nuvem, gerenciando a adição e atualização de registros no Firestore.

80

**- Cumprimento:**

A conexão, inserção e recuperação de dados no Firestore estão presentes e funcionam corretamente. A sincronização bidirecional também está implementada, atendendo às expectativas.

**- Gerenciamento de Registros:**

O sistema adiciona novos registros e verifica a existência de registros duplicados (com base no nome). Isso é feito antes de inserir dados no Firestore.

**- Cumprimento:**

O sistema gerencia registros de forma eficiente, evitando duplicações e permitindo atualizações consistentes.

**4. Interface Gráfica com Python:**

**- Tkinter para GUI:**

A interface gráfica desenvolvida com Tkinter permite a inserção de dados pelos usuários de forma intuitiva. A interface contém campos de texto, comboboxes, e um checkbox para consentimento.

**- Cumprimento:**

A interface gráfica cumpre os requisitos, permitindo a coleta e manipulação de dados conforme esperado.

**- Widget de Inserção e Edição de Dados:**

A interface fornece campos de inserção e uma estrutura lógica para inclusão de dados no sistema. Também oferece feedback visual ao usuário, indicando sucesso ou erro ao salvar os dados.

**- Cumprimento:**

O sistema GUI inclui as funcionalidades exigidas para inserção e validação de dados.

## 5. Relatórios e Geração de Gráficos:

### - Relatórios em PDF:

A funcionalidade de geração de relatórios em PDF está implementada corretamente, utilizando a biblioteca ReportLab. O código anonimiza os nomes dos pacientes para conformidade com a LGPD e inclui o logotipo da empresa no cabeçalho.

### - Cumprimento:

O código gera relatórios conforme as exigências acadêmicas e legais, incluindo anonimização de dados e logotipo da empresa.

### - Geração de Gráficos:

O código usa o Matplotlib para gerar gráficos que mostram os dados de tratamentos por localidade, o que atende às necessidades estatísticas descritas.

### - Cumprimento:

A funcionalidade de geração de gráficos está presente e atende às necessidades do projeto.

## 6. Aplicando RAD em Python:

### - Tratamento de Requisitos e Modelagem de Negócios:

O código demonstra claramente o tratamento dos requisitos de coleta de dados, manipulação, geração de relatórios e sincronização com um banco de dados em nuvem.

### - Cumprimento:

O sistema implementa os requisitos do negócio de forma ágil e eficiente, utilizando a metodologia RAD para prototipagem rápida e entregas contínuas.

### - Design de Interface com o Usuário:

A interface gráfica foi projetada para ser funcional e clara, com os campos de entrada organizados de maneira lógica, garantindo uma boa experiência ao usuário.

### - Cumprimento:

O design da interface é simples e direto, com funcionalidades bem implementadas para inserção e gerenciamento de dados.

## 7. Competências Técnicas e Socioemocionais:

### - Competências Técnicas:\*

O sistema demonstra o uso de Python para o desenvolvimento de soluções computacionais eficientes. O uso de frameworks como Tkinter, Firebase, e ReportLab está alinhado com as expectativas técnicas.

**- Cumprimento:**

O sistema está alinhado com as competências técnicas exigidas pela disciplina.

**- Competências Socioemocionais:**

O código foi projetado para facilitar o trabalho em equipe e melhorar a comunicação entre a equipe da clínica **FONART** e seus clientes. A anonimização dos dados mostra sensibilidade às questões de privacidade.

**- Cumprimento:**

As competências socioemocionais também estão contempladas, principalmente em relação à conformidade com a LGPD e respeito à privacidade dos pacientes.

**Conclusão da Análise técnica para avaliação acadêmica de nossos estudos:\*\***

O código nos parece apresentar integralmente todas as exigências acadêmicas e técnicas descritas no Plano de Aprendizagem. As principais funcionalidades, como a coleta de dados, integração com o Firebase, geração de relatórios em PDF e gráficos, além da interface gráfica, estão corretamente implementadas.

**Pontos fortes:**

- O código é modular e bem-organizado.
- Funcionalidades críticas, como a sincronização de dados e o tratamento de exceções, estão bem implementadas.
- A conformidade com a LGPD e a geração de relatórios anonimizados estão claramente presentes.

**Possíveis Melhorias (discussões com a empresária):**

- Poderia haver uma melhoria no layout da interface gráfica, tornando-a ainda mais amigável e talvez mais harmonizada visualmente.

---

## INTEGRAÇÃO DO PROJETO

**Integração do Projeto Python com segregação de ambientes, visando Estrutura HTML/JavaScript hospedado no \*\*Firebase Hosting\*\* do Formulário internet via Firebase – Web App Fonart.**

## Resumo da Integração Esperada

O projeto Python (`firebase\_setup.py`) possui a função de gerenciar o envio e sincronização de dados entre o **arquivo local CSV** e o **Firestore**. Ele coleta dados de pacientes por meio de uma interface gráfica em **Tkinter** e salva localmente e na nuvem (Firestore).

O código HTML/JavaScript hospedado no **Firebase Hosting** também coleta dados de pacientes e envia diretamente ao **Firestore**. Dessa forma, ele funciona como uma interface web, oferecendo uma alternativa ao uso da interface desktop em Python.

83

### Verificando a Integração:

#### 1. Sincronização de Dados Entre Firebase Hosting e Python

- A **integração entre a aplicação web e o Firestore é completa**. Tanto o código web quanto o código Python estão apontando para o mesmo projeto Firebase, o que significa que todos os dados inseridos via formulário web estarão automaticamente acessíveis pelo projeto Python.
- A função `addDoc` no HTML está enviando dados para a coleção `pacientes` no **Firestore**. O script Python (`firebase\_setup.py`) também está configurado para se conectar ao Firestore e sincronizar os dados da mesma coleção.

#### 2. Autonomia Entre Interfaces (Tkinter e Web)

- O código web e o script Python atuam de forma autônoma, ou seja, o usuário pode usar qualquer uma das interfaces (web ou Tkinter) para inserir ou consultar dados, e os dois sistemas irão interagir com o **mesmo banco de dados no Firestore**.
- Isso garante que qualquer dado inserido via a aplicação web estará automaticamente disponível para o script Python e vice-versa, sem necessidade de duplicação ou problemas de sincronização.

#### 3. Sincronização com CSV Local no Python

- No entanto, **os dados inseridos através da interface web não são automaticamente sincronizados com o arquivo CSV local gerado pelo Python**. Isso ocorre porque a funcionalidade de salvar no CSV está implementada apenas no script Python.
- **Recomendamos evitar projetos de sincronização entre a aplicação web e o arquivo CSV local**, porque seria preciso implementar uma solução para baixar os dados do Firestore (incluindo aqueles inseridos pelo formulário web) e atualizá-los no CSV local pelo Python, **causando insegurança no acesso aos dados e descumpriria a LGPD 13.709**, por consequência, poderia confrontar com a “concepção da privacidade por design”, significa, a **Privacy by Design**, visa adotarmos uma postura proativa, em vez de reativa,

ou seja, antecipar e se prevenir de incidentes de privacidade antes que eles possam ocorrer e não tentar remediar depois de ocorrido.

#### 4. Conformidade com o Projeto Acadêmico

- A aplicação web está integrada corretamente ao Firebase e coleta dados, conforme especificado no projeto acadêmico. Ela complementa o sistema Python, oferecendo uma interface alternativa de coleta de dados. No entanto, a integração completa com o CSV local depende de uma ação do script Python.

84

#### Recomendação – Pesquisa de Melhorias de Sistema FONART.

As recomendações, soluções, desafios e recomendações foram realizadas através das informações fornecidas no formulário de qualidade disponível no site: [Centro de Pesquisa de Melhorias do Sistema — Projeto Fonart \(fonart-8cd50.web.app\)](#)

#### Conclusão Geral

O código HTML hospedado no Firebase Hosting está integrado ao projeto Python no que diz respeito ao compartilhamento de dados via Firestore.

#### O Projeto e Integração entre a Aplicação Web e o Script Python

O projeto, inclui tanto uma interface web hospedada no Firebase Hosting quanto o script Python **firebase\_setup.py**, utiliza o **Cloud Firestore** como ponto central para armazenamento e sincronização de dados entre as interfaces web e local.

Analisamos a proposta, os problemas atuais da Clínica, suas limitações e solicitações da empresa **FONART**, visando viabilizar uma arquitetura para englobar os requisitos acadêmicos e técnicos, e, discutimos o motivo pelo qual **não é necessária uma sincronização direta entre o formulário web e o arquivo CSV local**.

##### A. Fluxo de Dados Centralizado via Firestore

A interface web desenvolvida em **HTML e JavaScript** está configurada para enviar diretamente os dados dos pacientes para o **Cloud Firestore**. Este banco de dados na nuvem serve como ponto central onde todas as informações são armazenadas e gerenciadas de forma eficiente.

Por outro lado, o script Python **firebase\_setup.py** também está configurado para interagir com o **Firestore**. Quando o sistema local é ativado, ele baixa automaticamente

os dados do Firestore e os sincroniza com o arquivo **CSV local**, garantindo que as informações sensíveis sejam armazenadas em um arquivo controlado localmente.

### B. Segurança e Conformidade com a LGPD

Ao adotar essa abordagem, garantimos a **segurança dos dados sensíveis**, conforme os requisitos da **LGPD (Lei Geral de Proteção de Dados)**. O arquivo pacientes.csv, que contém informações confidenciais dos pacientes, **não é exposto diretamente na aplicação web**. Ele é atualizado apenas no ambiente local, onde é possível aplicar controles de segurança mais rígidos, como o uso de criptografia, controle de acesso, e backup.

85

Além disso, o formulário web exige que o usuário marque o **consentimento** antes de submeter os dados, o que está em conformidade com os princípios da LGPD.

### C. Sincronização Centralizada e Eficiente

Como tanto a interface web quanto o sistema local utilizam o **Firestore** como base de dados central, a sincronização dos dados é feita de maneira fluida e automática. O Firestore age como intermediário entre os dados inseridos na interface web e os dados locais.

A sincronização entre a aplicação local e o arquivo CSV acontece quando o script Python é executado. Assim, o CSV local é atualizado **com os dados mais recentes do Firestore**, que incluem informações enviadas tanto pela interface web quanto diretamente pelo script Python. Esse modelo de sincronização centralizada é altamente eficiente, pois:

- **Evita duplicações** ou inconsistências nos dados, já que todos os registros passam primeiro pelo Firestore.
- **Reduz a complexidade** do sistema, não havendo necessidade de uma sincronização direta e paralela entre o formulário web e o arquivo CSV local.

### D. Conclusão do desafio

Com base na análise das funcionalidades e integração do projeto, **APROVADA** pela Sócia **Cláudia Moreira Cesar**, da empresa **FONART ESPACO FONOAUDIOLOGIA APLICADA E SAUDE INTEGRAL**, com a razão social CLAUDIA MOREIRA CESAR, com o CNPJ 19.064.473/0001-80 e tem sua sede localizada na Estrada dos Três Rios, 1097 - Freguesia (Jacarepaguá), Rio de Janeiro - RJ, 22.745-004, com foco principal de atuação de Atividades de fonoaudiologia, de acordo com o código CNAE Q-8650-0/06, pode-se afirmar que:

- **A sincronização entre a aplicação web e o arquivo CSV local não é necessária.**  
Isso ocorre porque o Firestore já centraliza todos os dados, permitindo que o script Python faça a sincronização com o CSV local de forma automática e segura.

- **A arquitetura atual é eficiente e segura**, garantindo que os dados dos pacientes estejam protegidos e em conformidade com a LGPD.
- **O projeto cumpre os requisitos técnicos e acadêmicos** ao integrar múltiplas tecnologias (HTML, JavaScript, Python, Firebase) para coletar, armazenar e sincronizar dados de forma eficiente, preservando a confidencialidade e integridade das informações.

Esse fluxo de trabalho, em que o Firestore centraliza os dados, permite que tanto a interface web quanto o script local operem de forma harmônica, sem a necessidade de duplicar funcionalidades ou criar integrações desnecessárias. A solução está bem implementada e otimizada para o contexto proposto.

---

## DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES EM PYTHON O TEMA 6, MÓDULO 4

Este projeto acadêmico, de extensão universitária realizado por Marcus Cesar Ferreira proporcionou a comparação com o modelo proposto, conforme a disciplina **DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES EM PYTHON** o **TEMA 6, MÓDULO 4**.

Podemos analisar se ambos seguem uma estrutura similar e aplicam conceitos correspondentes ao objetivo de desenvolver uma aplicação em Python com foco em metodologias de Desenvolvimento Rápido de Aplicações (RAD).

### Objetivo do Módulo 4 (Esquematizar uma aplicação RAD em Python):

O Módulo 4 orienta a implementação de um sistema simples de cadastro de notas de alunos, destacando a importância de prototipação, feedback dos usuários, e iterações rápidas. A metodologia RAD é apresentada como uma abordagem para desenvolver sistemas de baixa complexidade com entregas rápidas. A escolha do Python, especialmente utilizando bibliotecas como Tkinter para interface gráfica e Pandas para manipulação de dados, é fundamentada pela sua simplicidade, facilidade de uso e ampla aceitação no mercado.

### **Objetivo do meu projeto com a empresa FONART:**

O projeto, também utiliza Python como base para desenvolver um sistema, que integra coleta e armazenamento de dados com interface gráfica (Tkinter) e sincronização com banco de dados (Firebase). O sistema é voltado para o gerenciamento de dados de pacientes de fonoaudiologia, incluindo a geração de relatórios em PDF e gráficos de análise. Assim como no **Módulo 4**, o foco está na interação entre usuário e sistema, entregando resultados tangíveis (dados e relatórios) para melhorar a experiência do usuário.

87

### **Comparação de Estrutura**

#### **1. Prototipação e Iteração:**

##### **- Módulo 4:**

A entrega de protótipos é um ponto central. O sistema simples de cadastro de notas permite aos usuários testarem e fornecerem feedback para melhorias rápidas.

##### **- Projeto Acadêmico Realizado:**

Embora eu tenha realizado uma estrutura mais robusta, a utilização do Tkinter e da integração com o Firebase permite uma rápida visualização dos resultados e ajustes conforme necessários. O uso de prototipação também é aplicável, já que trabalha com uma interface gráfica e coleta de dados que são refinados ao longo do tempo.

#### **2. Metodologia RAD (Rápida):**

##### **- Módulo 4:**

RAD foca em sistemas de baixa complexidade que podem ser rapidamente desenvolvidos. A metodologia é destacada pelo ciclo rápido de desenvolvimento e feedback.

##### **- Projeto Acadêmico Realizado:**

O projeto inclui aspectos de iteração rápida ao combinar coleta de dados, geração de relatórios e visualização de gráficos, tudo dentro de um ciclo que pode ser rapidamente testado e aprimorado. A sincronização automática com a nuvem (Firebase) também oferece uma resposta ágil às necessidades dos usuários.

#### **3. Uso de Python e Bibliotecas:**

##### **- Módulo 4:**

Python é escolhido pela simplicidade e robustez, com destaque para Tkinter e Pandas, oferecendo interfaces gráficas e manipulação de dados.

### - **Projeto Acadêmico Realizado:**

Assim como no Módulo 4, utilizei Python com Tkinter para a interface gráfica, além de bibliotecas adicionais como o Firebase para armazenamento de dados, ampliando o escopo do projeto para além da manipulação local, o que acrescenta complexidade, mas mantém a simplicidade no desenvolvimento.

### **4. Requisitos Funcionais:**

#### - **Módulo 4:**

O sistema deve permitir o cadastro de alunos, cálculo de médias e visualização de suas situações (aprovado/reprovado).

### - **Projeto Acadêmico Realizado:**

Embora o foco seja diferente (dados de pacientes e relatórios), o sistema que desenvolvi para **FONART** também se estrutura em torno de requisitos funcionais claros: coleta de dados, validação, sincronização com o banco de dados, e geração de relatórios e gráficos.

### **5. Interface com o Usuário:**

#### - **Módulo 4:**

A interface gráfica usa Tkinter para facilitar a entrada e visualização de dados.

### - **Projeto Acadêmico Realizado:**

Utilizo Tkinter para a interface gráfica, mantendo uma abordagem amigável e interativa para a entrada e visualização dos dados, alinhando-se à proposta do Módulo 4.

## **APLICAÇÃO DOS CONCEITOS DO MÓDULO 4 NO PROJETO**

Conforme o que foi proposto no **Módulo 4**, o trabalho **saiu do modelo de protótipo para a produtividade**, está alinhado aos objetivos da disciplina com base em vários pontos:

#### - Implementação em Python:

Assim como no Módulo 4, o projeto utiliza Python e foca em uma interface gráfica com Tkinter, o que cumpre os requisitos de implementação prática discutidos no módulo.

#### - **RAD como metodologia:**

O desenvolvimento do sistema em Python, com foco em entregas rápidas e ajustáveis, pode ser comparado à metodologia RAD, conforme discutido no Módulo 4. A abordagem rápida e iterativa é aplicável tanto para sistemas de cadastro de notas quanto para o seu sistema de coleta de dados de pacientes.

- **Prototipagem e feedback do usuário:**

Embora o contexto seja diferente (alunos vs. pacientes), a interação entre o sistema e o usuário e a possibilidade de melhorias a partir do feedback são princípios aplicáveis nos dois casos. No projeto desenvolvido para atender a extensão e as expectativas da **FONART**, isso é visível na capacidade de gerar relatórios e sincronizar dados de maneira eficiente, permitindo ajustes conforme o feedback dos usuários.

## ÊNFASE - REFERÊNCIA ACADÊMICA

### Concluo e Referencio o Módulo 4 ao longo do Projeto FONART da seguinte maneira:

"Conforme discutido no tema 6, Módulo 4 da disciplina, a metodologia RAD aplicada ao desenvolvimento de sistemas em Python permite a entrega rápida de funcionalidades básicas, possibilitando o ajuste contínuo através de prototipação e feedback do usuário. A escolha do Python, conforme o Módulo 4, é uma decisão natural devido à simplicidade da linguagem, ampla aceitação e a disponibilidade de pacotes que facilitam o desenvolvimento de interfaces gráficas (Tkinter) e manipulação de dados (Pandas). Esta abordagem foi implementada neste projeto, que utiliza Python para desenvolver uma interface gráfica interativa, similar ao sistema de cadastro de notas, com a adição de funcionalidades de armazenamento de dados no Firebase e geração de relatórios."

### O que foi alterado ou adicionado no Projeto Acadêmico com a FONART:

#### EmailJS:

- Adicionamos o script do EmailJS no HTML.
- Na função de envio do formulário, agora usamos a API emailjs.send() para enviar um e-mail com as informações do formulário.
- Configurei o **USER\_ID**, **SERVICE\_ID**, e **TEMPLATE\_ID** que são fornecidos no painel de controle do EmailJS após a criação de uma conta.

#### Códigos EmailJS:

- **USER\_ID**: ID no painel do EmailJS.
- **SERVICE\_ID**: Refere-se ao serviço de e-mail que configuramos no EmailJS, ou seja, OUTLOOK.
- **TEMPLATE\_ID**: Este ID será gerado quando configura um template de e-mail no EmailJS. O template deve incluir variáveis como {{userName}}, {{interfaceFeedback}}, etc.

#### Benefícios:

- **Firebase**: Continua armazenando os dados de forma centralizada para análise.

## Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

- **Email:** Agora você receberá as sugestões dos usuários diretamente no seu e-mail.

Seguindo este procedimento, realizamos com sucesso tanto as respostas armazenadas no Firebase quanto um e-mail automático com as respostas dos usuários!

### Erros encontrados e solucionados página (qualidade.html) após o comando “firebase deploy”

(<https://fonart-8cd50.web.app/qualidade.html>):

(C:\Projeto\_Estacio\_2024-Fono\Projeto\_Fonart\fonart-web\qualidade.html)

Erros encontrados, o sistema (formulário de qualidade) encaminhava corretamente para o firebase, contudo, não nos encaminhava os dados para o e-mail cadastrado para a melhoria do projeto **FONART**. Não existia identificação, de que era necessário atualizar a linha do comando, para utilizar a versão mais recente do SDK. Modificamos o código HTML para usar o novo CDN, disponível. Por exemplo:

#### Antes:

Html...  
<script src="https://cdn.jsdelivr.net/npm/emailjs-com@2.6.4/dist/email.min.js"></script>

#### Substituído para:

<script src="https://cdn.jsdelivr.net/npm/@emailjs/browser@4/dist/email.min.js"></script>

---

#### Código final: QUALIDADE.HTML

Local: C:\Projeto\_Estacio\_2024-Fono\Projeto\_Fonart\fonart-web\qualidade.html

WEB: <https://fonart-8cd50.web.app/qualidade.html>

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Centro de Pesquisa de Qualidade — Projeto Fonart</title>
<script src="https://www.gstatic.com/firebasejs/8.6.1.firebaseio-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/8.6.1/firebase-firebase.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@emailjs/browser@4/dist/email.min.js"></script>
<script type="text/javascript">
// Inicializa o EmailJS
(function() {
  emailjs.init("qP01wKTpg2qeABdul"); // Public Key do EmailJS
})();
</script>
<style>
body {
  background-color: #E9CD8D;
  font-family: Arial, sans-serif;
  padding: 20px;
}
h1 {
```

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
text-align: center;
}
form {
    max-width: 600px;
    margin: 0 auto;
    background-color: white;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
}
label {
    display: block;
    margin-bottom: 5px;
}
input {
    width: 100%;
    padding: 8px;
    margin-bottom: 15px;
    border: 1px solid #ccc;
    border-radius: 4px;
}
button {
    background-color: #4CAF50;
    color: white;
    padding: 10px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    font-size: 16px;
    width: 100%;
}
button:hover {
    background-color: #45a049;
}
#responseMessage {
    text-align: center;
    margin-top: 20px;
    font-weight: bold;
}
}
</style>
</head>
<body>
    <h1>Centro de Pesquisa de Qualidade — Projeto Fonart</h1>
    <form id="suggestionForm">
        <label for="name">Nome:</label>
        <input type="text" id="name" name="nome" required>

        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>

        <label for="partes_envolvidas">Partes envolvidas:</label>
        <input type="text" id="partes_envolvidas" name="partes_envolvidas">

        <label for="problemas_identificados">Problemas identificados:</label>
        <input type="text" id="problemas_identificados" name="problemas_identificados">

        <label for="expectativas">Expectativas a serem alcançadas:</label>
        <input type="text" id="expectativas" name="expectativas">

        <label for="tecnologia_necessaria">Tecnologia necessária:</label>
        <input type="text" id="tecnologia_necessaria" name="tecnologia_necessaria">
```

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
<label for="tipo_manipulacao_dados">Tipo de manipulação de dados:</label>
<input type="text" id="tipo_manipulacao_dados" name="tipo_manipulacao_dados">

<label for="integracao_banco_dados">Integração com banco de dados:</label>
<input type="text" id="integracao_banco_dados" name="integracao_banco_dados">

<label for="interface_grafica">Interface gráfica:</label>
<input type="text" id="interface_grafica" name="interface_grafica">

<label for="widget">Widget:</label>
<input type="text" id="widget" name="widget">

<label for="projetar_relatorios">Projetar Relatórios:</label>
<input type="text" id="projetar_relatorios" name="projetar_relatorios">

<label for="projetar_graficos">Projetar Gráficos:</label>
<input type="text" id="projetar_graficos" name="projetar_graficos">

<label for="design_interface">Design de Interface:</label>
<input type="text" id="design_interface" name="design_interface">

<label for="sincronizacao_bd">Sincronização com banco de dados:</label>
<input type="text" id="sincronizacao_bd" name="sincronizacao_bd">

<label for="aplicacao_web">Aplicação WEB:</label>
<input type="text" id="aplicacao_web" name="aplicacao_web">

<label for="atender_lgpd">Atender à LGPD:</label>
<input type="text" id="atender_lgpd" name="atender_lgpd">

<label for="possiveis_melhorias">Possíveis melhorias:</label>
<input type="text" id="possiveis_melhorias" name="possiveis_melhorias">

    <button type="submit">Enviar Sugestão</button>
</form>
<div id="responseMessage"></div>

<script type="text/javascript">
// Configuração do Firebase
const firebaseConfig = {
  apiKey: "AlzaSyDo_td7R5bSLZXg_25FNjf7Tj8yZINc5T4",
  authDomain: "fonart-8cd50.firebaseio.com",
  databaseURL: "https://fonart-8cd50-default.firebaseio.com",
  projectId: "fonart-8cd50",
  storageBucket: "fonart-8cd50.appspot.com",
  messagingSenderId: "359305470538",
  appId: "1:359305470538:web:080940a6afe1df09c5fcf3",
  measurementId: "G-H6Z37MJL7D"
};
// Inicializando o Firebase
firebase.initializeApp(firebaseConfig);
const db = firebase.firestore();

// Função para enviar as sugestões para o Firestore e EmailJS
document.getElementById('suggestionForm').addEventListener('submit', function(event) {
  event.preventDefault();

  // Capturar os dados do formulário
  const suggestionData = {
    nome: document.getElementById('name').value,
```

# Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia

```
email: document.getElementById('email').value,
partes_envolvidas: document.getElementById('partes_envolvidas').value,
problemas_identificados: document.getElementById('problemas_identificados').value,
expectativas: document.getElementById('expectativas').value,
tecnologia_necessaria: document.getElementById('tecnologia_necessaria').value,
tipo_manipulacao_dados: document.getElementById('tipo_manipulacao_dados').value,
integracao_banco_dados: document.getElementById('integracao_banco_dados').value,
interface_grafica: document.getElementById('interface_grafica').value,
widget: document.getElementById('widget').value,
projetar_relatorios: document.getElementById('projetar_relatorios').value,
projetar_graficos: document.getElementById('projetar_graficos').value,
design_interface: document.getElementById('design_interface').value,
sincronizacao_bd: document.getElementById('sincronizacao_bd').value,
aplicacao_web: document.getElementById('aplicacao_web').value,
atender_lgpd: document.getElementById('atender_lgpd').value,
possiveis_melhorias: document.getElementById('possiveis_melhorias').value
};

// Salvar os dados no Firestore
db.collection("suggestions").add(suggestionData)
.then(() => {
  console.log("Sugestão armazenada com sucesso no Firestore!");
  document.getElementById('responseMessage').textContent = "Sugestão enviada com sucesso!";
})
.catch((error) => {
  console.error("Erro ao salvar no Firestore: ", error);
  document.getElementById('responseMessage').textContent = "Erro ao enviar a sugestão. Tente novamente.";
});

// Enviar o e-mail via EmailJS
emailjs.send("service_v58s0wn", "template_hdthm96", suggestionData)
.then((response) => {
  console.log("Sugestão enviada com sucesso para o e-mail!", response.status, response.text);
  document.getElementById('responseMessage').textContent = "Sugestão enviada com sucesso!";
}, (error) => {
  console.error("Erro ao enviar o e-mail: ", error);
  document.getElementById('responseMessage').textContent = "Erro ao enviar o e-mail. Tente novamente.";
});

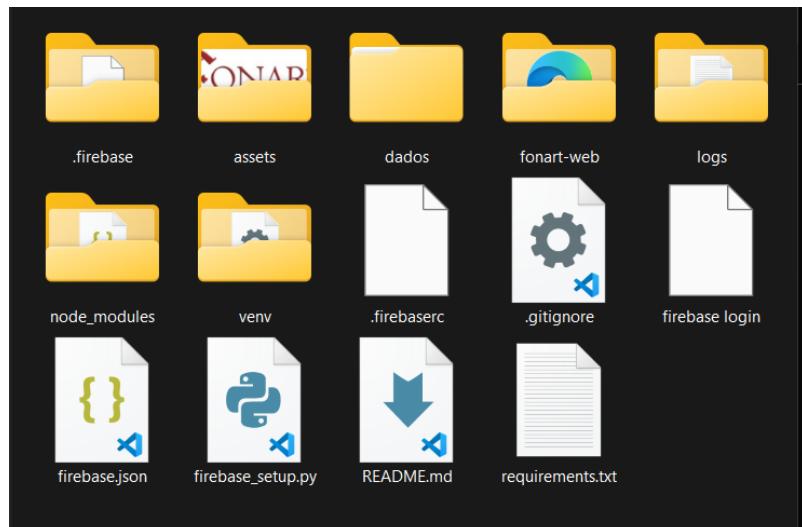
// Limpar o formulário
document.getElementById('suggestionForm').reset();
});

</script>
</body>
</html>
```

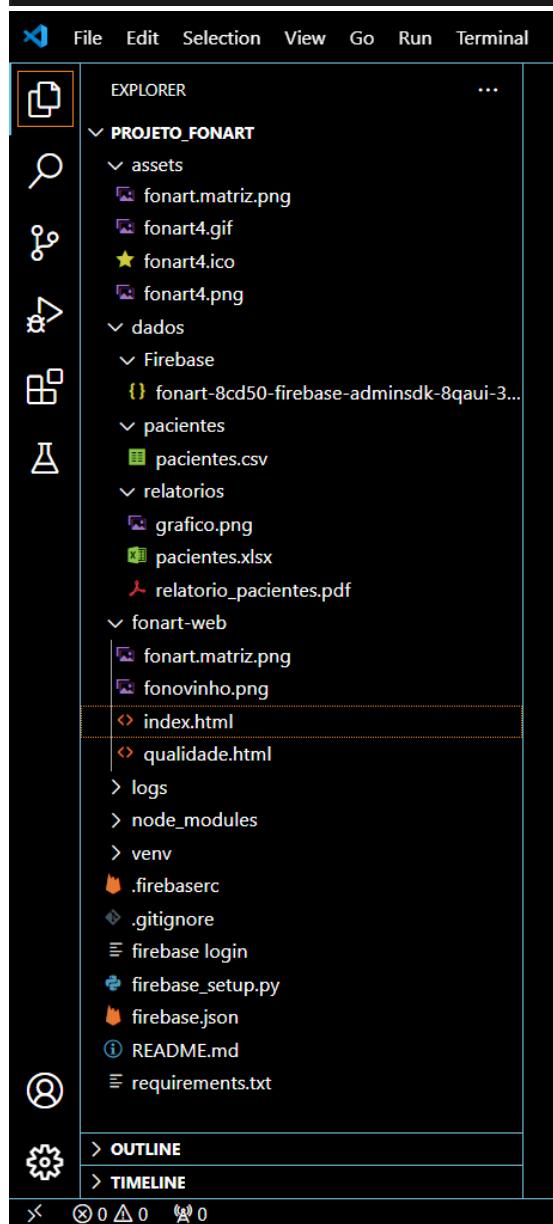
## PASTAS FINALIZADAS

Imagen das pastas atuais do sistema e da Estrutura Python

## Extensão Universitária – Python - Empresa Fonart: Projeto Piloto em Dados e Estatísticas de Casos Clínicos em Fonoaudiologia



94



# PASSO A PASSO PARA USO DO SISTEMA FONART

## Interface do Sistema

Após a inicialização, a janela principal do sistema será exibida.

### 1. Cadastro de Pacientes

#### 1.1. Preencha as Informações do Paciente

Na janela principal, você verá diversos campos que precisam ser preenchidos:

- **Nome:** Digite o nome completo do paciente.
- **Idade:** Informe a idade do paciente (somente números).
- **Diagnóstico:** Escolha o diagnóstico do paciente no menu suspenso (por exemplo: Disartria, Afasia, Autismo).
- **Localidade:** Escolha a cidade de onde o paciente é.
- **Tratamento:** Selecione o tipo de tratamento do paciente (por exemplo: Terapia de Linguagem, Terapia de Voz).
- **Mês:** Escolha o mês em que o paciente iniciou o tratamento.
- **Consentimento para uso de dados:** Marque a caixa de seleção se o paciente deu o consentimento para uso de dados.

95

#### 1.2. Salvar os Dados

- Após preencher todos os campos, clique no botão “**Salvar Dados**”.
- O sistema fará a validação automática das informações e, se tudo estiver correto, os dados serão salvos no **Firebase** (banco de dados na nuvem) e também localmente, em um arquivo CSV.
- Caso algum campo não esteja preenchido corretamente, uma mensagem de aviso será exibida, solicitando que corrija os erros.

---

## 2. Geração de Relatórios

### 2.1. Gerar Relatório em PDF

- Após salvar os dados de pacientes, você pode gerar um relatório estatístico.
- Para isso, clique no botão “**Gerar Relatório**” na janela principal.
- O sistema gera um relatório PDF que contém informações anonimizadas sobre os pacientes (para proteção de dados conforme a LGPD).
- Após a geração do relatório, o PDF será salvo na pasta “**relatorios**” do projeto, e o sistema abrirá automaticamente o arquivo para visualização.

---

## 3. Sincronização de Dados

### 3.1. Sincronizar Dados entre CSV Local e Firebase

- O sistema permite sincronizar os dados salvos localmente (CSV) com o banco de dados na nuvem (Firebase).
- O sistema automaticamente realiza a sincronização dos dados.

- O sistema sincroniza automaticamente os dados entre os dois locais, garantindo que tanto o Firebase quanto o arquivo CSV estejam atualizados.

## 4. Análise de Dados

### 4.1. Visualizar Gráficos

- O sistema também permite gerar gráficos baseados nas informações dos pacientes.
- Para visualizar gráficos de análise, como o número de tratamentos por localidade, clique no botão “**Mostrar Gráfico**”.
- Um gráfico será exibido na tela com base nos dados coletados.

96

# ÁRVORE DO PROJETO\_FONART/

## Arquivo README.md

```
PROJETO_FONART/
├── assets/
│   ├── fonart4.gif
│   ├── fonart4.ico
│   └── fonart4.png
├── dados/
│   ├── Firebase/
│   │   └── fonart-8cd50-firebase-admin.json # Credenciais para o Firebase
│   ├── pacientes/
│   │   └── pacientes.csv
│   └── relatorios/
│       ├── grafico.png
│       ├── pacientes.xlsx
│       └── relatorio_pacientes.pdf
├── fonart-web/
│   ├── fonart.matriç.png
│   ├── fonovinho.png
│   ├── index.html
│   └── qualidade.html
├── logs/
│   └── system.log
├── node_modules/
└── venv/
    ├── .firebaserc
    ├── .gitignore
    ├── firebase_login
    ├── firebase_setup.py
    ├── firebase.json
    └── README.md
    └── requirements.txt

# Arquivos de mídia utilizados no projeto
# Animação para a interface ou relatórios
# Ícone da aplicação ou site
# Logotipo da empresa utilizado na interface

# Arquivo CSV contendo os dados dos pacientes

# Gráfico gerado a partir dos dados de pacientes
# Dados dos pacientes em formato Excel
# Relatório em PDF gerado a partir dos dados
# Arquivos para a interface web (se aplicável)
# Logotipo utilizado na versão web
# Imagem utilizada na interface web
# Página inicial da interface web
# Página relacionada à qualidade do serviço ou produto

# Arquivo de log com registros de eventos do sistema
# Dependências do projeto (Node.js)
# Ambiente virtual do Python
# Arquivo de configuração do Firebase CLI
# Arquivo para ignorar determinados arquivos no Git
# Script ou arquivo de configuração para login no Firebase
# Script de configuração do Firebase
# Arquivo de configuração do Firebase
# Arquivo de documentação do projeto
# Lista de dependências do Python necessárias ao projeto
```