



Centro Universitário

Medição e Monitoramento do Peso e da Pressão Arterial

Projeto Integrador

Rafhael de Oliveira Martins

Centro Universitário IESB – Brasília – DF – Brazil

Abstract. *Obesity affects people of all ages and from all social groups in developed and developing countries, reaching 650 million people worldwide. Hypotension occurs when blood pressure drops to the point of causing symptoms such as dizziness and fainting. Too low blood pressure can cause organ damage, a process called shock. Hypertension, on the other hand, occurs when the blood pressure on the artery walls is very strong and is above normal limits for age. Most of the time this is noticed when measuring pressure. The measurement of blood pressure is provided by the values of the cardiac cycle, evaluated from the end of one heartbeat to the end of the next cardiac cycle, observing a systole and a diastolic phase. From this perspective, the idea of this project is to build a Desktop Application using the Python Programming Language, and with the help of a blood pressure meter, and a body scale, you can fulfill your needs and register your pressure levels and weight and view the daily monitoring on an interactive panel, thus providing a more effective and controlled monitoring of your blood pressure and weight levels.*

Keywords: *Database, Data Science, Obesity, Blood Pressure, Python, PyInstaller, SQL.*

Resumo. *A obesidade afeta pessoas de todas as idades e de todos os grupos sociais nos países desenvolvidos e em desenvolvimento, alcançando 650 milhões de pessoas em todo o mundo. A hipotensão arterial ocorre quando a pressão arterial cai a ponto de provocar sintomas como tonturas e desmaios. Uma pressão arterial muito baixa pode provocar danos a órgãos, um processo chamado choque. Já na hipertensão arterial ocorre quando a pressão que o sangue*

faz na parede das artérias é muito forte e fica acima dos limites considerados normais para a idade. Na maioria das vezes isto só é percebido quando se mede a pressão. A aferição da pressão arterial é determinada pelos valores do ciclo cardíaco, sendo avaliado desde o fim de um batimento cardíaco até o fim do seguinte ciclo cardíaco, observada a fase da sístole e a fase da diástole. Por esta perspectiva, a ideia deste projeto é realizar a construção de um Aplicativo Desktop através da Linguagem de Programação Python, e com o auxílio de medidor de pressão arterial, e uma balança corporal, um indivíduo possa realizar suas medições e cadastrar seus níveis de pressão e peso e visualizar em um painel interativo o monitoramento diário, proporcionando assim, um monitoramento mais efetivo e controlado de seus níveis de pressão arterial e de seu peso.

Palavras-chave: Banco de dados, Data Science, Obesidade, Pressão Arterial, Python, PyInstaller, SQL.

Sumário

1	Concepção do Projeto	4
2	Pesquisa	5
2.1	Obesidade	5
2.2	Pressão Arterial (PA)	6
2.2.1	Hipotensão Arterial	7
2.2.2	Hipertensão Arterial	7
3	Montagem do Projeto	9
3.1	Descrição das Atividades	12
3.2	Programação	14
3.2.1	Bibliotecas necessárias	14
3.2.2	Janela do Medidor e Banco de Dados	15
3.2.3	Criação do Aplicativo Desktop	19
3.2.4	Primeiro teste	23
3.2.5	Erro e a solução	24
3.2.6	Funcionamento do aplicativo	26
3.2.7	Criação do atalho	27
3.2.8	Análise Exploratória dos dados	29
4	Avaliação dos Resultados	31
5	Conclusões	32
5.1	Relato da Experiência	32

1. Concepção do Projeto

As taxas de obesidade quase triplicaram desde 1975 e aumentaram quase cinco vezes entre crianças e adolescentes. Todo dia 04/03 de cada ano, tem-se o Dia Mundial da Obesidade, que afeta pessoas de todas as idades e de todos os grupos sociais nos países desenvolvidos e em desenvolvimento, alcançando 650 milhões de pessoas em todo o mundo [Biblioteca Virtual em Saúde].

O Dia Mundial da Obesidade incentiva soluções práticas para ajudar as pessoas a alcançar e manter um peso saudável, realizar tratamento adequado e reverter a crise da obesidade. Há muito a fazer, incluindo restringir a comercialização de alimentos e bebidas com alto teor de gorduras, açúcar e sal para crianças; tributar bebidas açucaradas e proporcionar melhor acesso a alimentos saudáveis; abrir espaço para caminhadas, ciclismo e recreação seguros nas cidades; ensinar às crianças hábitos saudáveis desde a infância.

Hipotensão arterial é o termo popular conhecido como “pressão baixa” – quando a medida da pressão arterial está abaixo de 9 por 6. O coração tem função de bomba do sangue que circula em todo o organismo, levando oxigênio e nutrientes, e a pressão nada mais é do que a força com a qual o sangue atinge as paredes das artérias que compõem o sistema vascular [Luiz].

O equilíbrio da “pressão” é bastante delicado; se elevada demais (hipertensão) ou baixa demais (hipotensão), o paciente pode apresentar sintomas. Ambos os estados podem ser assintomáticos (não causam sintomas). Doença muito comum no mundo todo, a hipotensão deve ser tratada por toda a vida, acompanhada de medidas higiênico-dietéticas com mudanças de hábito e estilo de vida.

Em 2020 uma matéria da OPAS - Organização Pan-Américas da Saúde para o Dia Mundial da Hipertensão 2020, ocorrendo todo dia 17 de maio, foi criado com objetivo de alertar a população para a importância da doença. A OPAS diz que a hipertensão arterial afeta mais de 30% da população adulta em todo o mundo, ou seja, mais de um bilhão de pessoas. Ainda segundo a OPAS, este é o principal fator de risco para doenças cardiovasculares, especialmente doença coronariana e acidente vascular cerebral, mas também para doença renal crônica, insuficiência cardíaca, arritmia e demência.

No Brasil o dia 26 de abril têm-se também o Dia Nacional de Prevenção e Combate à Hipertensão Arterial, data para conscientizar a população sobre a importância do diagnóstico preventivo e do tratamento da doença, que mata mais de 10 milhões de pessoas por ano no mundo. Cerca de 30% dos brasileiros são hipertensos, aponta a Sociedade Brasileira de Cardiologia (SBC).

A Organização Mundial de Saúde afirma: a obesidade é um dos mais graves problemas de saúde que temos para enfrentar. Em 2025, a estimativa é de que 2,3 bilhões de adultos ao redor do mundo estejam acima do peso, sendo 700 milhões de indivíduos com obesidade, isto é, com um índice de massa corporal (IMC) acima de 30.

No Brasil, essa doença crônica aumentou 67,8% nos últimos treze anos, saindo de 11,8% em 2006 para 19,8% em 2018. Diante dessa prevalência, vale chamar a atenção que, de acordo com a Pesquisa de Vigilância de Fatores de Risco e Proteção para Doenças Crônicas

por Inquérito Telefônico (Vigitel), realizada pelo Ministério da Saúde, a obesidade voltou a crescer entre nós após uma breve trégua. Entre 2015 e 2017, ela ao menos se manteve estável em 18.9%.

Por esta perspectiva, a ideia deste projeto é realizar a construção de um Aplicativo Desktop através da Linguagem de Programação Python, e com o auxílio de medidor de pressão arterial, e uma balança corporal, um indivíduo possa realizar suas medições e cadastrar seus níveis de pressão e peso e visualizar em um painel interativo o monitoramento diário, proporcionando assim, um monitoramento mais efetivo e controlado de seus níveis de pressão arterial e de seu peso.

2. Pesquisa

2.1. Obesidade

É um dos principais fatores de risco para várias doenças não transmissíveis (DNTs), como diabetes tipo 2, doenças cardiovasculares, hipertensão, acidente vascular cerebral e várias formas de câncer. Pessoas com obesidade são constantemente envergonhadas e culpadas porque muitos – incluindo médicos, formuladores de políticas e outros – não entendem completamente a obesidade, que, como todas as doenças crônicas, tem causas profundas e complexas provenientes de fatores dietéticos, de estilo de vida, genéticos, psicológicos, socioculturais, econômicos e ambientais.

Pela definição da Organização Mundial da Saúde, obesidade é o excesso de gordura corporal, em quantidade que determine prejuízos à saúde. Uma pessoa é considerada obesa quando seu Índice de Massa Corporal (IMC) é maior ou igual a 30 kg/m² e a faixa de peso normal varia entre 18,5 e 24,9 kg/m².

Os indivíduos que possuem IMC entre 25 e 29,9 kg/m² são diagnosticados com sobrepeso e já podem ter alguns prejuízos com o excesso de gordura [Biblioteca Virtual em Saúde].

O tratamento inclui alimentação saudável com diminuição da ingestão de calorias e aumento da atividade física, podendo-se associar o uso de medicamentos. Em casos mais graves e refratários, pode ser indicado o tratamento cirúrgico.

Abaixo segue a tabela de IMC:

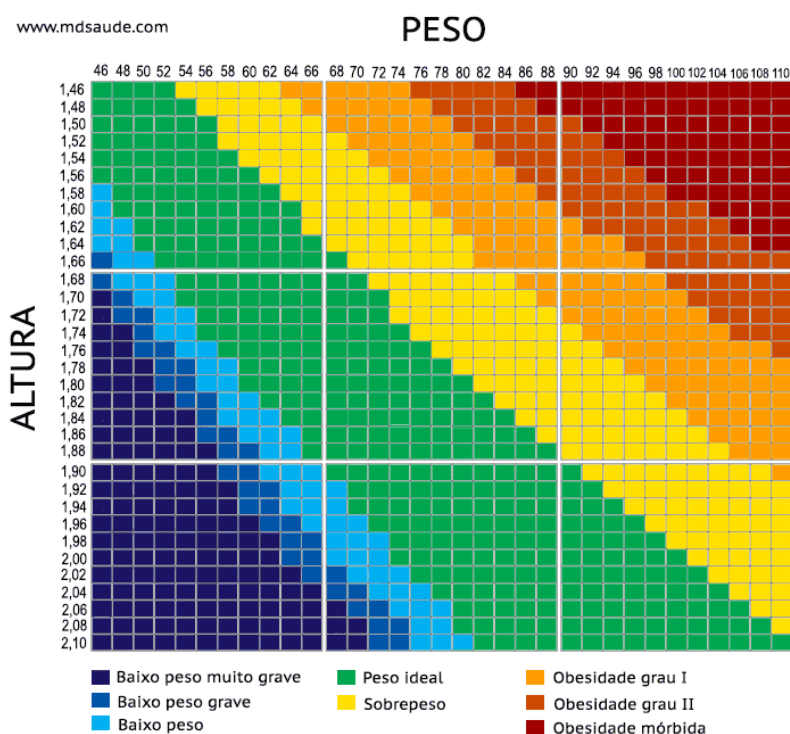


Figura 1. Tabela de IMC

2.2. Pressão Arterial (PA)

Refere-se á pressão exercida pelo sangue contra a parede das artérias. A pressão arterial bem como a de todo o sistema circulatório encontra-se normalmente um pouco acima da pressão atmosférica, sendo a diferença de pressões responsável por manter as artérias e demais vasos não colapsados. Em uma pessoa saudável, o valor da pressão pode variar continuamente, dependendo do stress, a emotividade ou se está fazendo atividade física.

A aferição da pressão arterial é determinada pelos valores do ciclo cardíaco, sendo avaliado desde o fim de um batimento cardíaco até o fim do seguinte ciclo cardíaco, observada a fase da sístole e a fase da diástole.

A fase da sístole, também conhecida como pressão arterial sistólica, ocorre no momento em que o coração bombeia seu conteúdo na aorta mediante contração do ventrículo esquerdo, encontra-se a válvula mitral fechada e a válvula aórtica aberta, quando a pressão ventricular esquerda é máxima, a pressão calculada a nível das artérias também é máxima.

Na fase da diástole, também conhecida como pressão arterial diastólica, ocorre imediatamente antes do próximo batimento cardíaco, com a válvula aórtica fechada e a válvula mitral aberta, o ventrículo esquerdo está em relaxamento e a receber o sangue das aurículas. Neste momento a pressão arterial nas artérias é baixa, no entanto esta pressão mínima ainda é consideravelmente superior à pressão presente do lado exterior da aorta e de todo o sistema arterial, sendo esta certamente maior do que a pressão atmosférica razão pela qual as artérias não colapsam nesta fase do ciclo.

2.2.1. Hipotensão Arterial

Ocorre quando a pressão arterial cai a ponto de provocar sintomas como tonturas e desmaios. Uma pressão arterial muito baixa pode causar danos a órgãos, um processo chamado choque. Quando a pressão arterial é muito baixa, não chega uma quantidade suficiente de sangue a todas as partes do corpo. Assim, as células não recebem oxigênio e nutrientes suficientes e os resíduos não são eliminados da forma adequada. Por isso, os órgãos e as células afetadas em seu interior começam a funcionar mal. A pressão arterial muito baixa pode ser fatal, já que pode provocar choque, pois a falta de fluxo sanguíneo causa danos aos órgãos [Levi D. Procter 2020].

Pessoas saudáveis com pressão arterial baixa, mas ainda dentro dos limites normais (quando medida em repouso), tendem a viver mais tempo do que pessoas com pressão arterial que esteja na extremidade alta do normal.

O corpo conta com vários mecanismos para normalizar a pressão arterial após ela aumentar ou diminuir durante as atividades normais, como exercício físico ou sono. Estes envolvem:

- Alteração do diâmetro das artérias pequenas (arteríolas) e, em menor grau, das veias.
- Alteração do volume de sangue bombeado do coração para o corpo (débito cardíaco).
- Alteração do volume de sangue nos vasos sanguíneos.
- Alteração da posição do corpo.

2.2.2. Hipertensão Arterial

Chamamos de pressão alta ou hipertensão arterial (HA) quando a pressão que o sangue faz na parede das artérias é muito forte e fica acima dos limites considerados normais para a idade. Na maioria das vezes isto só é percebido quando se mede a pressão.

A maior parte das vezes esta pressão alta não tem uma única causa bem definida e também não dá sintomas, por isso, é considerada um inimigo invisível. No entanto, sabemos que a doença é de natureza hereditária (atinge vários membros da mesma família), aumenta sua frequência com a idade, sedentarismo, ganho de peso e excesso de sal na comida. Confirmada a pressão alta, a pessoa deverá fazer acompanhamento médico pelo resto de sua vida para não ser prejudicada, sem mesmo saber que isso está ocorrendo [Prof^a. Dr^a. Cibele Isaac Saad Rodrigues].

Na hipertensão são cinco faixas de pressão arterial reconhecidas pela [Association 2021], são elas:

CATEGORIA DE PRESSÃO SANGÜÍNEA	SISTÓLICO mm Hg (número superior)		DIASTÓLICO mm Hg (número inferior)
NORMAL	MENOS DE 120	e	MENOS DE 80
ELEVADO	120 - 129	e	MENOS DE 80
ALTA PRESSÃO SANGÜÍNEA (HIPERTENSÃO) ESTÁGIO 1	130 - 139	ou	80-89
ALTA PRESSÃO SANGÜÍNEA (HIPERTENSÃO) ESTÁGIO 2	140 OU SUPERIOR	ou	90 OU SUPERIOR
CRISE HIPERTENSA (consulte seu médico imediatamente)	SUPERIOR A 180	e / ou	SUPERIOR A 120

Figura 2. Tabela de pressão arterial saudáveis e não saudáveis

Normal: Valores de pressão arterial de menos de 120/80 mm Hg são considerados dentro da faixa normal. Se seus resultados se enquadrarem nesta categoria, mantenha hábitos saudáveis para o coração, como seguir uma dieta balanceada e fazer exercícios regularmente.

Elevado: A pressão arterial elevada é quando as leituras variam consistentemente de 120-129 sistólica e menos de 80 mm Hg diastólica. Pessoas com pressão arterial elevada provavelmente desenvolverão pressão alta, a menos que sejam tomadas medidas para controlar a doença.

Hipertensão Estágio 1: É quando a pressão arterial varia consistentemente de 130-139 sistólica ou 80-89 mm Hg diastólica. Nesse estágio de pressão alta, os médicos provavelmente prescreverão mudanças no estilo de vida e podem considerar a adição de medicamentos para a pressão arterial com base no risco de doença cardiovascular aterosclerótica (ASCVD), como ataque cardíaco ou derrame.

Hipertensão Estágio 2: É quando a pressão arterial varia consistentemente em 140/90 mm Hg ou mais. Nesse estágio de hipertensão, os médicos provavelmente prescreverão uma combinação de medicamentos para a pressão arterial e mudanças no estilo de vida.

Crise de hipertensão: Este estágio da pressão alta requer atenção médica. Se suas leituras de pressão arterial repentinamente excederem 180/120 mm Hg, espere cinco minutos e teste sua pressão arterial novamente. Se suas leituras ainda estiverem excepcionalmente altas, entre em contato com seu médico imediatamente. Você pode estar passando por uma crise hipertensiva .

Se sua pressão arterial for superior a 180/120 mm Hg e você estiver experimentando sinais de possíveis danos aos órgãos, como dor no peito, falta de ar, dor nas costas, dormência / fraqueza, mudança na visão ou dificuldade para falar, não espere para ver se sua pressão diminui por conta própria.

Uma leitura elevada pode ou não ser acompanhada por um ou mais dos seguintes sintomas:

- Dor de cabeça severa

- Falta de ar
- Hemorragias nasais
- Ansiedade severa

Esteja preparado caso você foi diagnosticado com pressão alta, monitore sua pressão arterial e os medicamentos. Se possível durante uma emergência, ter esses registros com você pode fornecer informações valiosas para a equipe médica que fornece o tratamento.

3. Montagem do Projeto

Planejamento: Para este projeto foram definidos alguns pontos para um bom desenvolvimento, entre estes estão: Desenvolvimento da janela, criação do banco de dados, criação do aplicativo desktop no windows .exec, coleta de dados, análise de dados, criação do painel interativo.

Este projeto contou com os seguintes materiais abaixo descritos:

- 1 Balança Corporal Digital.
- 1 Aparelho de pressão arterial digital ou analógico de pulso ou braço.
- 1 Computador com conexão a internet.
- Editor de códigos.
- Linguagem de Programação Python.
- Google Colab, Jupyter Notebook ou Spyder.
- 1 Software de Self-Service Business Intelligence (BI).
- Banco de dados SQLite.
- 1 Voluntário para realizar o monitoramento.

Estes materiais foram os utilizados como também, sugestão de outros modelos de materiais e software a serem utilizados, ou seja, necessariamente não precisam ser os mesmos materiais e softwares, desde que façam o que é proposto para o projeto e que cada material realize o feito desejado, conforme descrição da utilidade dos materiais e porquê do seu uso a seguir.

Balança Corporal Digital: A balança foi utilizada com a finalidade de medição do peso corporal do participante/voluntário para a fase de coleta de dados. Para este foi utilizado um modelo parecido com o da figura abaixo.



Figura 3. Balança Corporal Digital

Aparelho de pressão arterial digital de pulso: O aparelho de pressão arterial foi utilizado com a finalidade da medição da pressão arterial sistólica, diastólica e pulsação do participante/voluntário para a fase da coleta de dados. Para tal, facilitando a fase de coleta de dados, foi escolhido o aparelho digital de pulso, por ser mais simples a utilização e ser independente de terceiros para a medição, proporcionando assim uma medição simples, intuitiva e independente, bastando o participante/voluntário colocar o aparelho no pulso, posicionar-se corretamente, pressionar o botão start/stop, e seguir a orientação do próprio aparelho em relação a altura ideal para a medição correta. Modelo utilizado segue na figura abaixo.



Figura 4. Aparelho de Pressão Arterial Digital de Pulso

Computador com acesso a internet: O computador foi utilizado com a finalidade de

resolução do projeto por completo, entre estes o registro dos dados no banco de dados, desenvolvimento e utilização do aplicativo desktop, como a análise e desenvolvimento do painel interativo dos dados, sem modelo e sistema operacional(OS) específico para o desenvolvimento deste projeto, bastando ser possível utilização dos softwares e atividades descritas.

Editor de Códigos: Para este projeto, o editor de códigos foi utilizado com a finalidade de facilitar a criação do aplicativo desktop. Para tal foi escolhido o editor de códigos Visual Studio Code, popularmente chamado de VS Code, por ser leve e ter a praticidade de instalação de extensões para qualquer linguagem de programação, podendo ser encontrado para download no site oficial: <https://code.visualstudio.com/>.

Linguagem de Programação Python: Neste projeto foi utilizado a linguagem de programação Python por ser uma linguagem de alto nível e por ter uma comunidade relativamente crescente a cada ano e por proporcionar o desenvolvimento em praticamente todas as etapas que envolveram programação para este projeto, ressaltando a utilização no painel interativo. Para o desenvolvimento foi utilizado a versão do Python 3.6.7, mas podendo ser utilizada qualquer versão do Python 3, podendo ser encontrado para download e informações no site oficial: <https://www.python.org/>.

Google Colab: Para a etapa de análise dos dados coletados, foi utilizado o Colaboratory ou Colab disponibilizado pelo Google, que permite escrever código Python no navegador, sem a necessidade de instalação local ou configuração, acesso gratuito a GPUs, e de fácil compartilhamento, podendo encontrar informações no site oficial: <https://colab.research.google.com>.

Banco de dados SQLite: Por ser um projeto com o intuito de desenvolvimento de um aplicativo desktop, foi utilizado o banco de dados SQLite por ser um banco de dados em arquivo, ou seja, é um banco de dados em SQL ou também conhecido como estruturado, em arquivo, não sendo necessária a instalação de software e nem servidor local ou remoto para utilização, e sem executar um processo SGBD separado, em português SGBD (Sistema de Gerenciamento de Banco de Dados) ou em inglês DBMS (Data Base Management System) é o sistema de software responsável pelo gerenciamento de um ou mais bancos de dados, sendo também um mecanismo de armazenamento seguro com transações ACID, em português (acrônimo de Atomicidade, Consistência, Isolamento e Durabilidade) ou em inglês (Atomicity, Consistency, Isolation, Durability) é um conjunto de propriedades de transação (sequência de operações) em banco de dados, e sendo também um software livre. O SQLite foi escolhido por essas características, sendo pequeno, rápido, independente, de alta confiabilidade e recursos completos, sendo possível encontrar mais informações no site oficial: <https://www.sqlite.org>.

Participante/Voluntário: Para este projeto ou a quem deseja executar tal, será necessário a escolha/encontro de um ou mais participantes/voluntários que queiram coletar e disponibilizar seus dados para análise, podendo este ser até mesmo o próprio desenvolvedor.

DBeaver: É um aplicativo de software cliente SQL e uma ferramenta de administração de banco de dados. Para bancos de dados relacionais, ele usa a interface de programação de aplicativo JDBC para interagir com os bancos de dados por meio de um driver JDBC, mais

informações podem ser encontradas no site oficial: <https://dbeaver.io/>.

3.1. Descrição das Atividades

Desenvolvimento da janela: Neste projeto foi desenvolvida uma janela/windows em que é possível o usuário inserir os valores de peso, pressão arterial sistólica, pressão arterial diastólica, como também um botão em que é possível fazer o cadastro das medidas realizadas no banco de dados, e um botão para fechar a janela, como pequenas caixa de diálogo para a interação com o voluntário para as etapas do processo de cadastros da medida e solicitação para fechar a janela.

Por não ter conseguido mais voluntários para o projeto, foi definido então uma única janela para este projeto, com as características informadas acima. Perante este detalhe, não será necessário para o momento criação de janela de login ou cadastro de medidas de acordo com cada voluntário, por possuir neste projeto apenas um voluntário.

Na criação de uma janela com linguagem Python foi utilizado a biblioteca Tkinter, onde está biblioteca é o pacote GUI (Graphical User Interface). Tk/Tcl tem sido parte integrante do Python. Ele fornece um kit de ferramentas de janela robusto e independente de plataforma, que está disponível para programadores Python usando o pacote tkinter e sua extensão, os módulos tkinter.tix e tkinter.ttk [Foundation 2021a], [Foundation 2021b] e [Claw20017 2021].

O programa em Tkinter mais simples possível tem três linhas. Este programa cumpre somente uma das quatro questões básicas na programação de GUIs, ele roda o event loop, que é um loop eterno, esperando que algo aconteça.

A primeira linha importa o módulo Tkinter e deixa-o disponível para uso. Perceba que a forma de importar (“from Tkinter import *”) significa que nós não queremos ter que usar a forma “Tkinter.” para especificar nada que quisermos utilizar.

A segunda linha cria uma janela toplevel (que pode ou não se tornar visível). Tecnicamente, o que esta linha faz é criar uma instância da classe “Tkinter.Tk”. Esta janela toplevel é o componente GUI de mais alto nível de qualquer aplicação de Tkinter. Por convenção, esta janela é normalmente chamada de “raiz”.

A terceira linha executa o método mainloop (isso é, o event loop) deste objeto raiz. Assim que roda, o mainloop espera que eventos aconteçam no objeto raiz. Se um evento ocorre, então ele é alimentado (o event handler é executado) e o loop continua rodando, esperando pelo próximo evento, ou até que aconteça um evento que “destrua” a raiz. Um evento destruidor pode ser o fechamento da janela pelo botão X de fechamento. Quando a raiz é destruída, a janela é fechada e o event loop cancelado.

Comportamento do programa: Ao rodar este programa, você verá (graças ao Tk) a janela toplevel automaticamente com os widgets para minimizar, maximizar e fechar a janela. Tente usá-los e você verá que eles realmente funcionam.

Clicando no widget “fechar” (o X em uma caixa, do lado direito da barra de título) será gerado um evento destruidor terminando o event loop principal, que no caso deste programa

é mainloop. E desde que não haja mais nada depois da linha “root.mainloop()”, como neste caso, o programa não faz mais nada e se encerra [Ferg].

```
1 from tkinter import *          # Modulo tkinter.
2 root = Tk()                    # Janela toplevel.
3 root.mainloop()                # Metodo mainloop.
```

Após execução deste código, será apresentado está janela, conforme imagem a seguir:

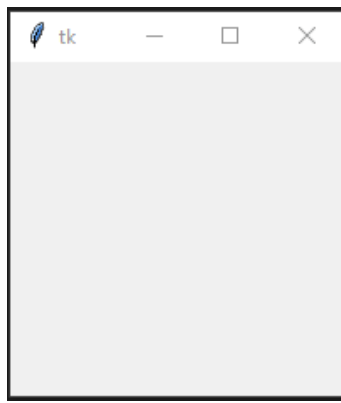


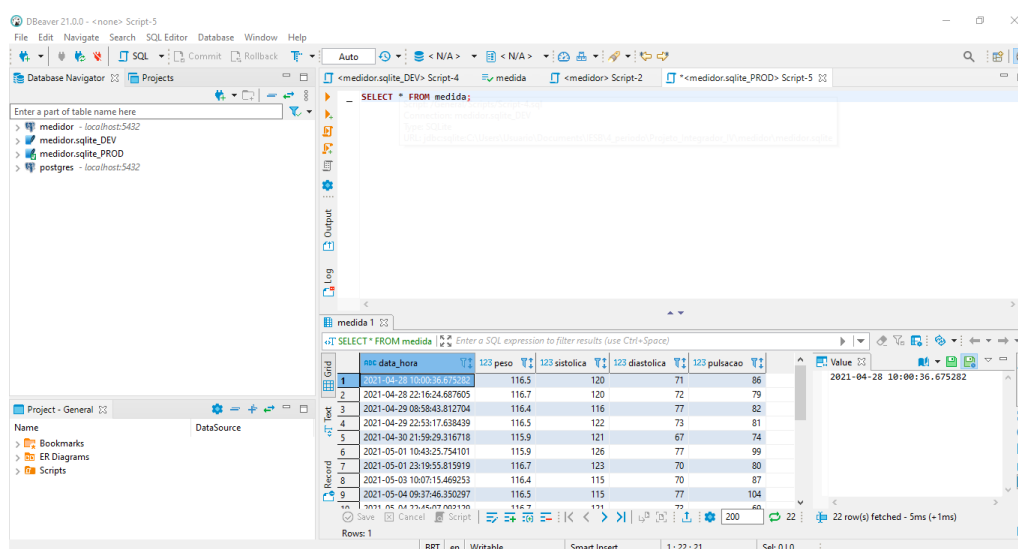
Figura 5. Janela simples com 3 linhas de código

Criação do Bando de Dados: No banco de dados, como já citado na descrição das atividades, para este projeto foi desenvolvido uma tabela para guardar os registros da medições de peso, pressão arterial sistólica, pressão arterial diastólica, pulsação, como também a data e hora do registro das medições. Para tal foi definido o banco de dados SQLite. Abaixo terá o código SQL de criação da tabela de medidas.

Para os campos e estrutura dos dados no banco, a data_hora foi definida como text, peso como real, sistolica como interger, diastolica como interger, pulsacao como interger também, e o nome da tabela como medida.

```
1 CREATE TABLE IF NOT EXISTS medida (
2     data_hora text,
3     peso real,
4     sistolica interger,
5     diastolica interger,
6     pulsacao interger)
```

Abaixo será demonstrado o aplicativo SGBD que é opcional, como também a tabela com os dados registrados:



	data_hora	peso	sístolica	diastólica	pulscacao
1	2021-04-28 10:00:36.675282	116.5	120	71	86
2	2021-04-28 22:16:24.687605	116.7	120	72	79
3	2021-04-29 08:58:43.812704	116.4	116	77	82
4	2021-04-29 22:53:17.638439	116.5	122	73	81
5	2021-04-30 21:59:29.316718	115.9	121	67	74
6	2021-05-01 10:43:25.754101	115.9	126	77	99
7	2021-05-01 23:18:55.815919	116.7	123	70	80
8	2021-05-03 10:07:15.469253	116.4	115	70	87
9	2021-05-04 09:37:46.350297	116.5	115	77	104

Figura 6. DBeaver e a tabela com alguns dos dados registrados

Criação do aplicativo desktop no windows .exec: Para a criação do aplicativo desktop no Sistema Operacional (OS) Windows, comumente chamado de executável com extensão .exec, foi utilizado a biblioteca PyInstaller.

As principais vantagens do PyInstaller sobre ferramentas semelhantes são que o PyInstaller funciona com Python 3.5—3.9, cria executáveis menores graças à compressão transparente, é totalmente multiplataforma e usa o suporte do sistema operacional para carregar as bibliotecas dinâmicas, garantindo assim total compatibilidade. Mais informações no site oficial: <http://www.pyinstaller.org/>.

Coleta de dados: Para a coleta de dados, foi definido a coleta em dois horários, a primeira medição no início do dia, entre 09:00 e 10:00 e a última no final do dia entre 21:00 e 23:00. Por não ter conseguido muitos voluntários, ficou definido também só um participante para coletar os dados de suas medições, em que este realiza as medições neste horário definido, abre o aplicativo desktop, e realiza a inserção dos valores para registro no banco de dados.

Análise de dados: Nesta etapa da análise, foi definido que seria feito uma Análise Exploratória dos Dados (EDA), em que nos permite o conhecimento dos dados alcançados por meio das medições, como também foi realizada nesta, algumas plotagens de gráficos para melhor visualização das medidas registrados pelo voluntário.

3.2. Programação

3.2.1. Bibliotecas necessárias

Antes de iniciar a programação do projeto, é preciso instalar todas as dependências do projeto, ou seja, todas as bibliotecas que serão utilizadas, e para isso, é necessário criar ou fazer o download de todos os arquivos do projeto, que estão disponibilizados no github: <https://github.com/rafaelom/medidor-pressao>. Logo após o download, abrir o editor de códigos e instalar as bibliotecas necessárias através do arquivo

.requirements.txt, para uma melhor visualização dos passos, basta seguir os passos abaixo, após o download de todos os arquivos e abertura do editor de códigos:

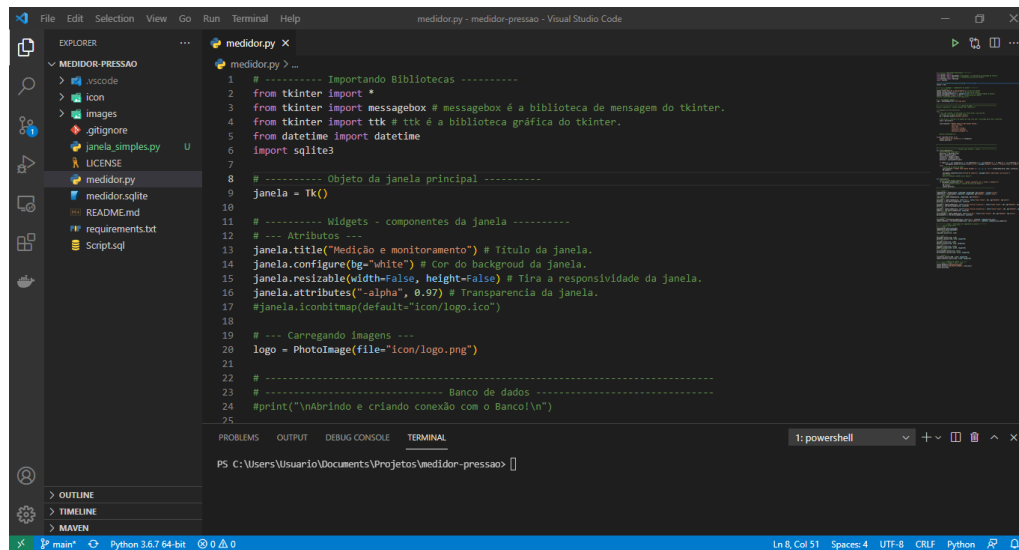


Figura 7. Passo 01 -Terminal no editor de códigos

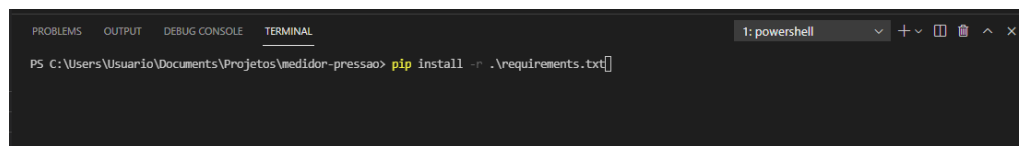


Figura 8. Passo 02 - Instalando as bibliotecas com o arquivo .requirements.txt

O código do passo 02 está a seguir:

```
1 # Passo 02 - Instalando Bibliotecas
2 pip install -r .\requirements.txt
```

3.2.2. Janela do Medidor e Banco de Dados

Abaixo será demonstrado o código de criação da janela principal, em que é a única para este projeto, pois só temos um voluntário para a coleta dos dados de seus níveis como também o código para criação do banco de dados e tabela para armazenar os dados, lembrando que como é um banco de dados em arquivo, é possível a criação do banco e tabela no próprio código de criação da janela [BR 2020], [eXcript 2018] e [Ribeiro 2019].

```
1 # ----- Importando Bibliotecas -----
2 from tkinter import *
3 from tkinter import messagebox # messagebox é a biblioteca de mensagem
  do tkinter.
4 from tkinter import ttk # ttk é a biblioteca gráfica do tkinter.
5 from datetime import datetime
6 import sqlite3
```

```

7
8 # ----- Objeto da janela principal -----
9 janela = Tk()
10
11 # ----- Widgets - componentes da janela -----
12 # --- Atributos ---
13 janela.title("Medição e monitoramento") # Título da janela.
14 janela.configure(bg="white") # Cor do background da janela.
15 janela.resizable(width=False, height=False) # Tira a responsividade da
    janela.
16 janela.attributes("-alpha", 0.97) # Transparencia da janela.
17 janela.iconbitmap(default="icon/logo.ico")
18
19 # --- Carregando imagens ---
20 logo = PhotoImage(file="icon/logo.png")
21
22 # ----- Banco de dados -----
23 #print("\nAbrindo e criando conexão com o Banco!\n")
24
25 # Tratamento de exceções (erros)
26 try:
27     # Faz uma conexão, se não puder ser feita haverá uma exceção.
28     # Conexão com o Banco de Dados SQLite.
29     bd = sqlite3.connect("medidor.sqlite")
30
31     # bd.cursor() retornará um objeto do tipo conn que é utilizado
    para fazer consultas.
32     conn = bd.cursor()
33
34     conn.execute("""CREATE TABLE IF NOT EXISTS medida (
35                     data_hora text,
36                     peso real,
37                     sistolica interger,
38                     diastolica interger,
39                     pulsacao interger)""")
40
41     #print("Conectado!\n")
42
43 except (sqlite3.Error) as e:
44     print("\nFalha de conexão!\n {}".format(e))
45     janela.destroy()
46
47 # ----- Funcoes para eventos - events -----
48 def click_cadastrar():
49     #print("CLICOU CADASTRAR!")
50     data_hora = datetime.now()
51     peso = pesoEntry.get()
52     sistolica = pasEntry.get()
53     diastolica = padEntry.get()
54     pulsacao = pulsacaoEntry.get()
55
56     if (peso == "" and sistolica == "" and diastolica == "" and
        pulsacao == "" or peso == "" and sistolica == "" or diastolica ==
        "" and pulsacao == "" or peso == "" or sistolica == "" or
        diastolica == "" or pulsacao == ""):
57         messagebox.showerror(title="Cadastro Erro",

```



```

message="Campo(s) vazio(s). Preencha todos os campos!")
58 else:
59     # Inserção no banco de dados
60     conn.execute("""INSERT INTO medida VALUES('{}', {}, {}, {},
        {}, {})""".format(data_hora, peso, sistolica, diastolica, pulsacao))
61     bd.commit()
62
63     messagebox.showinfo(title="Informação Cadastro",
        message="Dados cadastrados com Sucesso!")
64     #bd.close()
65     #print("Fechando conexão com o banco!")
66
67 def click_sair():
68     #print("CLICOU FECHAR JANELA!")
69     if messagebox.askokcancel("Sair", "Deseja realmente sair e fechar
        o medidor?"):
70         #print("Fechando janela e conexão do banco de dados!")
71         bd.close()
72         janela.destroy()
73 #-----
74
75 # --- Frames ---
76 ladoEsquerdo = Frame(janela, width=300, height=400, bg="#329542",
        relief="raise")
77 ladoDireito = Frame(janela, width=497, height=400, bg="#63A355",
        relief="raise")
78 # --- Imagem Logo ---
79 logoLabel = Label(ladoEsquerdo, image=logo, bg="#329542")
80 # --- peso ---
81 pesoLabel = Label(ladoDireito, text="Peso:", font=("Indie Flower",
        20), bg="#63A355", fg="white")
82 pesoEntry = ttk.Entry(ladoDireito, width=20)
83 # --- pas ---
84 pasLabel = Label(ladoDireito, text="Pressão Arterial Sistólica:",
        font=("Indie Flower", 20), bg="#63A355", fg="white")
85 pasEntry = ttk.Entry(ladoDireito, width=20)
86 # --- pad ---
87 padLabel = Label(ladoDireito, text="Pressão Arterial Diastólica:",
        font=("Indie Flower", 20), bg="#63A355", fg="white")
88 padEntry = ttk.Entry(ladoDireito, width=20)
89 # --- pulsacao ---
90 pulsacaoLabel = Label(ladoDireito, text="Pulsação:", font=("Indie
        Flower", 20), bg="#63A355", fg="white")
91 pulsacaoEntry = ttk.Entry(ladoDireito, width=20)
92
93 # --- botoes ---
94 sairButton = ttk.Button(ladoDireito, text="Sair", width=20,
        command=click_sair)
95 cadastrarButton = ttk.Button(ladoDireito, text="Cadastrar", width=20,
        command=click_cadastrar)
96
97 # ----- Layout - Gereciador de componetes da janela -----
98 # ----- Ineri Componentes -----
99 # --- frames ---
100 ladoEsquerdo.pack(side=LEFT)
101 ladoDireito.pack(side=RIGHT)

```

```

102 # --- imagem logo ---
103 logoLabel.place(x=18, y=30)
104
105 # --- peso ---
106 pesoLabel.place(x=18, y=20)
107 pesoEntry.place(x=100, y=26, height=30)
108 # --- pas ---
109 pasLabel.place(x=18, y=89)
110 pasEntry.place(x=350, y=95, height=30)
111 # --- pad ---
112 padLabel.place(x=18, y=158)
113 padEntry.place(x=355, y=164, height=30)
114 # --- pulsacao ---
115 pulsacaoLabel.place(x=18, y=227)
116 pulsacaoEntry.place(x=150, y=233, height=30)
117
118 # --- botões ---
119 sairButton.place(x=300, y=320, height=30)
120 cadastrarButton.place(x=100, y=320, height=30)
121
122 # ----- Tamanho da janela -----
123 # --- width x height + left + top ---
124 janela.geometry("800x400+250+180")
125 janela.protocol("WM_DELETE_WINDOW", click_sair)
126 janela.mainloop()
  
```

Após a etapa de criação da janela e do banco de dados, é válido fazer um teste de como está o projeto, e para tal, após a execução do código tem-se a janela e integração conforme demonstração das imagens a seguir.

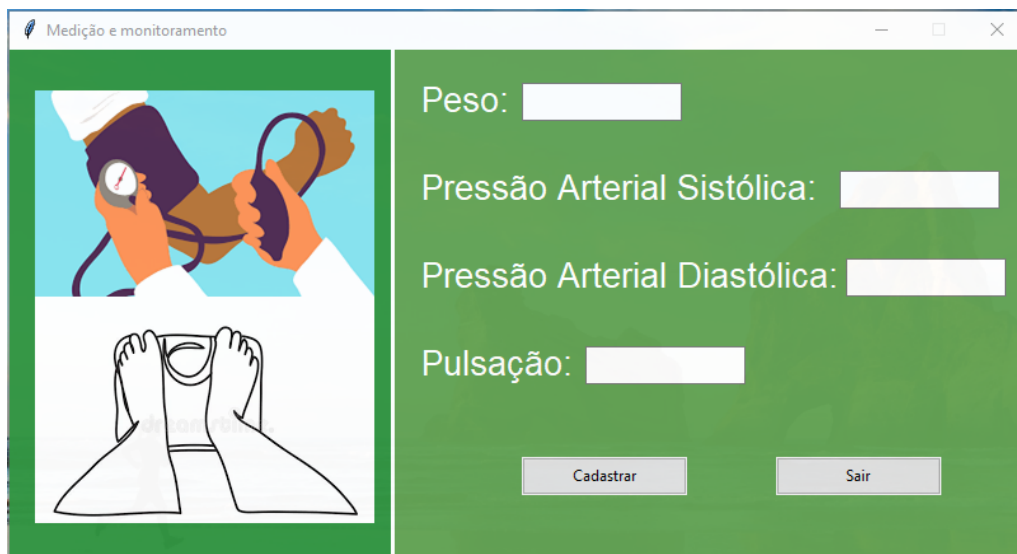


Figura 9. Janela inicial do medidor

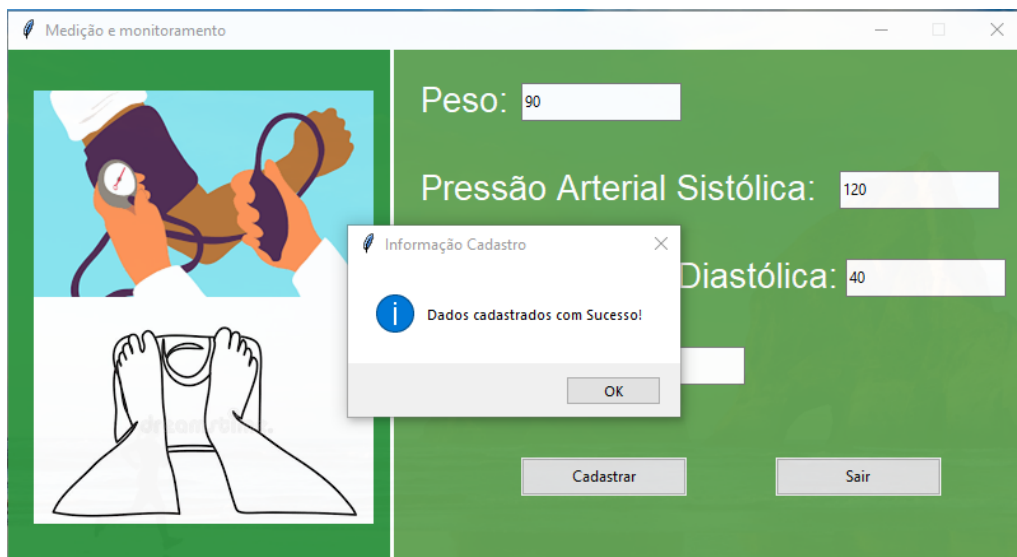


Figura 10. Janela após clicar no botão de cadastro

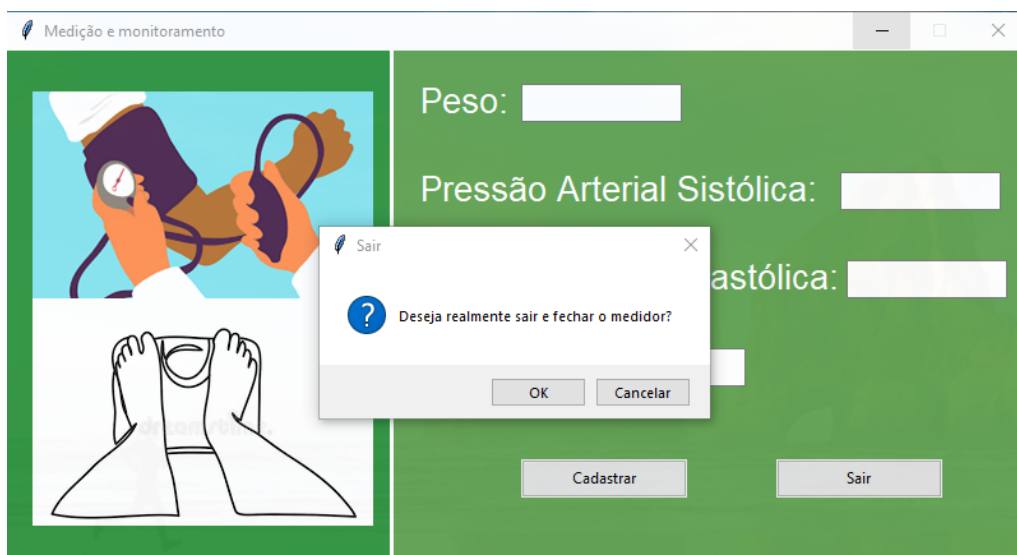


Figura 11. Janela após clicar no botão de fechar ou sair do medidor

3.2.3. Criação do Aplicativo Desktop

Após a execução do teste da criação da janela e do registro no banco de dados, é hora de criar o aplicativo desktop para ser possível o voluntário abrir este por meio da tela inicial ou qualquer localidade do computador, sem a necessidade de abrir um editor de códigos. Abaixo será demonstrado por meio de imagens os comandos necessários para a criação do aplicativo como também o passo a passo para ele rodar, como também após estes passos todos os códigos para serem executados no terminal serão disponibilizados.

No terminal do editor, ou no cmd do windows, já no diretório da pasta do projeto, seguir os passos abaixo:

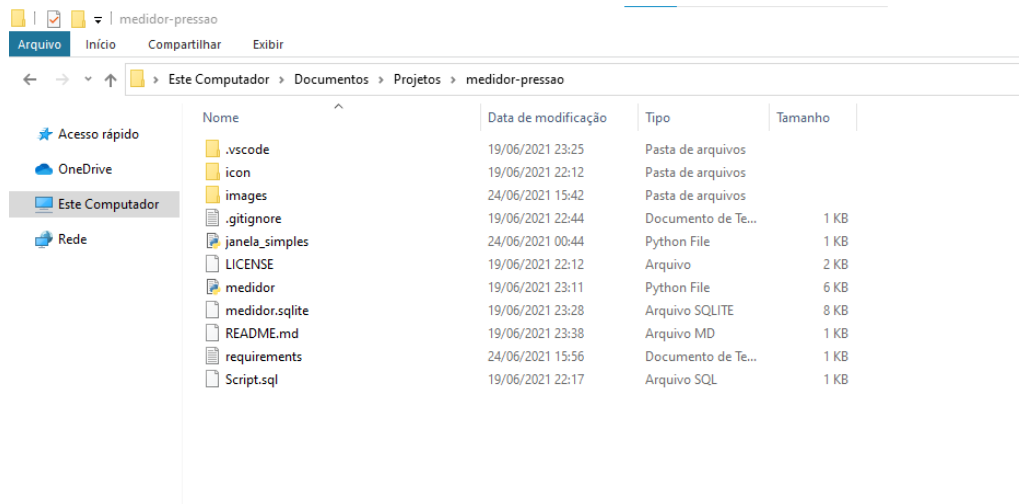


Figura 12. Passo 01 - Pasta do diretório do projeto



Figura 13. Passo 02 - Verificando instalação da biblioteca PyInstaller



Figura 14. Passo 03 - Criando .exec no diretório do projeto

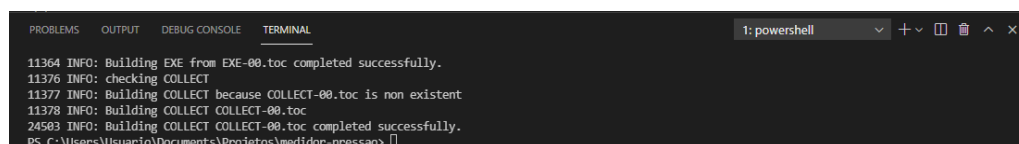


Figura 15. Passo 04 - Após ter sido criando .exec aparecerá a seguinte mensagem

A partir deste ponto já foi criado o executável no windows. Para confirmar, verifica-se os arquivos criados no diretório e logo após, verificar o local do executável.

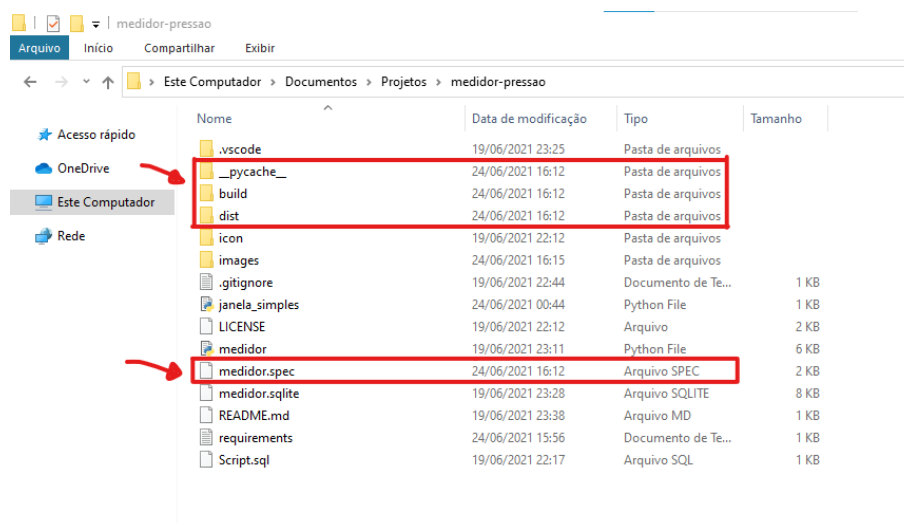


Figura 16. Passo 05 - Verificando arquivos criados no diretório do projeto

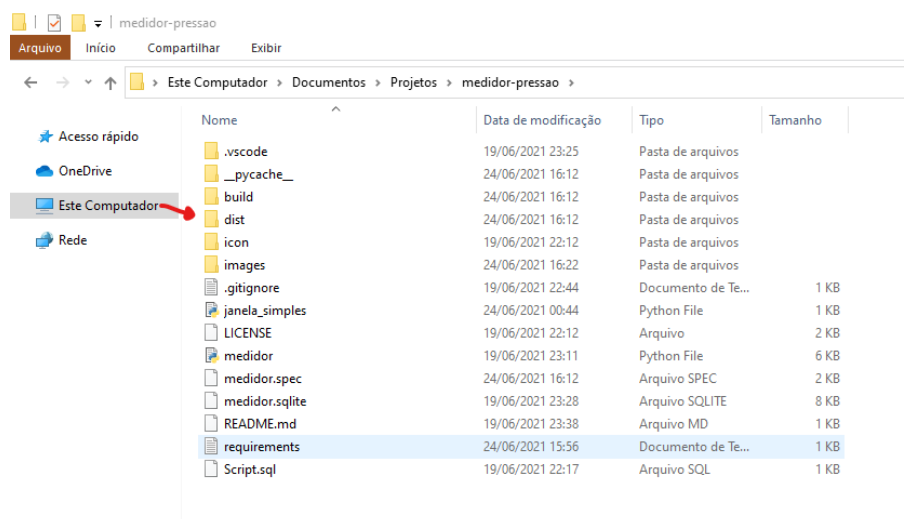


Figura 17. Passo 06 - Abrindo pasta do aplicativo criado e seus arquivos

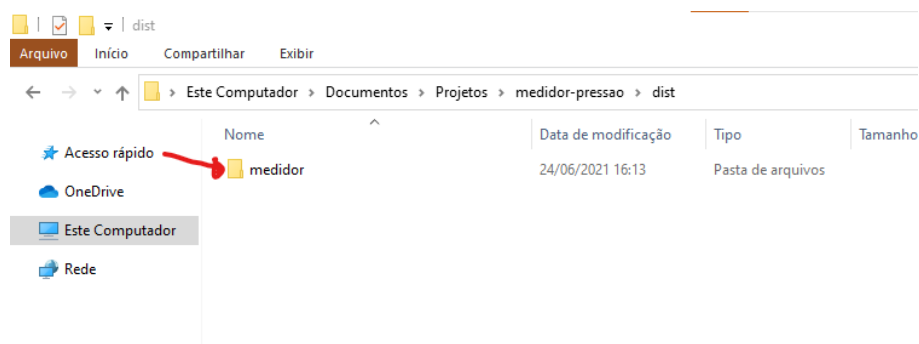


Figura 18. Passo 07 - Abrindo pasta do executável

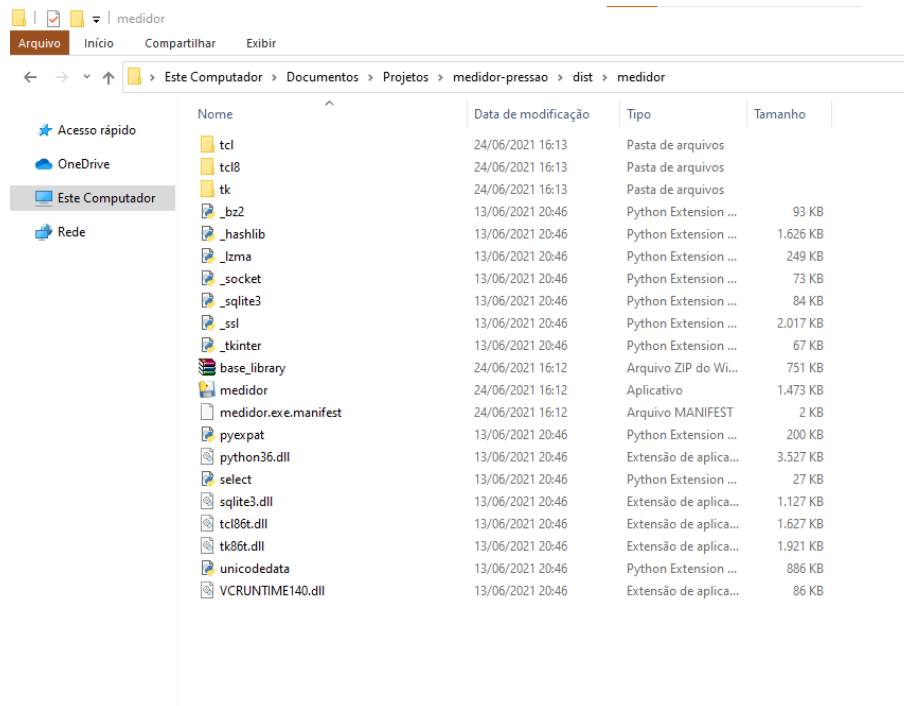


Figura 19. Passo 08 - Verificando arquivos na pasta do executável criado

Código do passo 02 e 03 estão a seguir:

```

1 # Passo 02 - Verificando instalação da biblioteca PyInstaller.
2
3 pyinstaller --version
4
5 # Passo 03 - Criando .exec no diretório do projeto.
6
7 pyinstaller --windowed medidor.py

```

3.2.4. Primeiro teste

A partir deste ponto, será realizado o teste do aplicativo, para verificar se tudo ocorreu corretamente, e se não faltou nenhum arquivo para a execução correta do aplicativo.

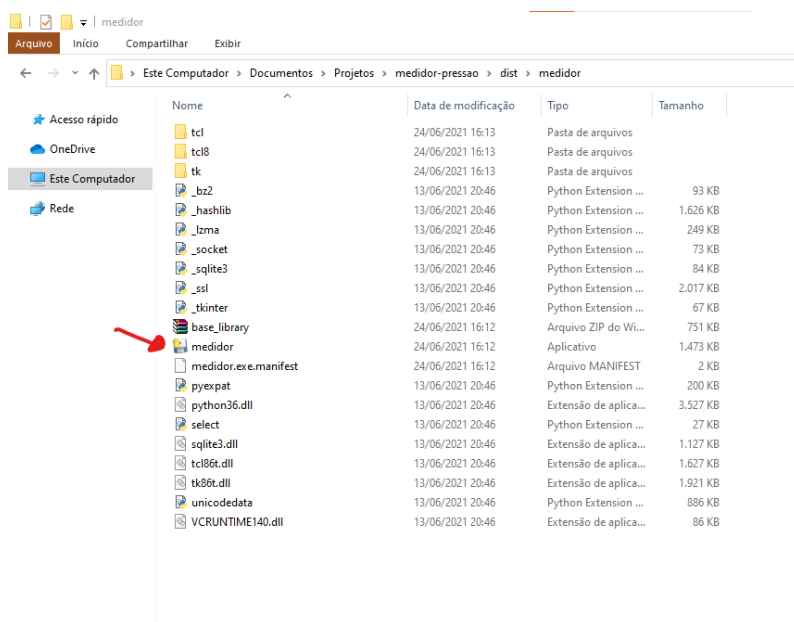


Figura 20. Passo 01 - Abrindo o executável para primeiro teste

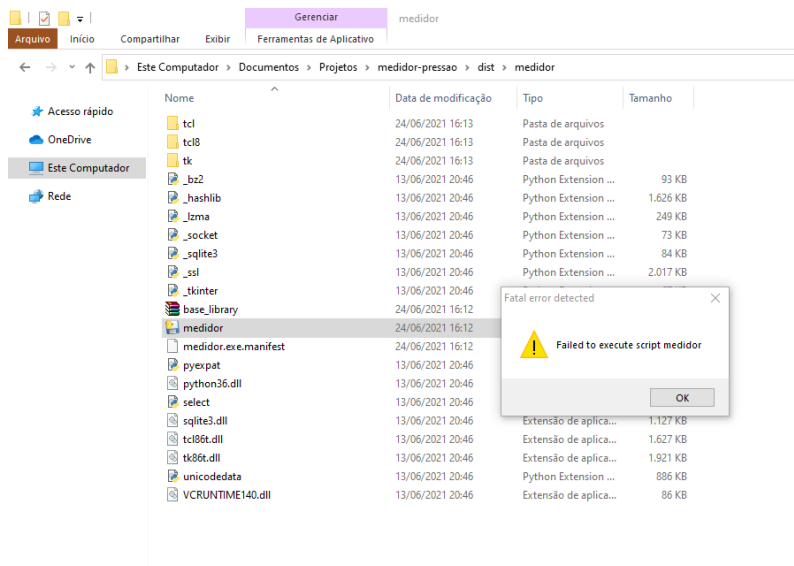


Figura 21. Passo 02 - Realizando primeiro teste no executável e verificação de mensagem

Retornado uma mensagem de erro, opa, será que deu errado? A resposta é sim. Mas vamos verificar o porque de ter ocorrido tal feito e assim mesmo a criação ter ocorrido toda por completo sem erro.

3.2.5. Erro e a solução

Após algumas pesquisas, encontrei a solução do problema, e para isso, verificaremos o diretório do projeto:

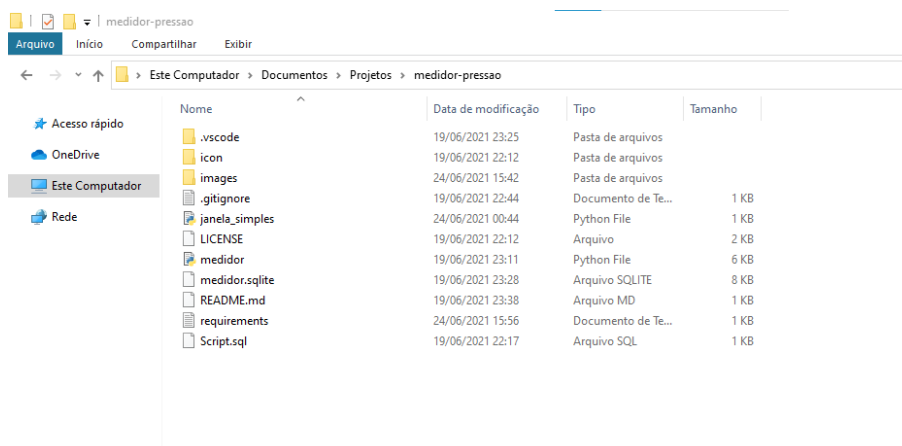


Figura 22. Diretório do projeto

Verificaremos o diretório do aplicativo criado e realizar a comparação com o diretório do projeto, e será observado todos os arquivos:

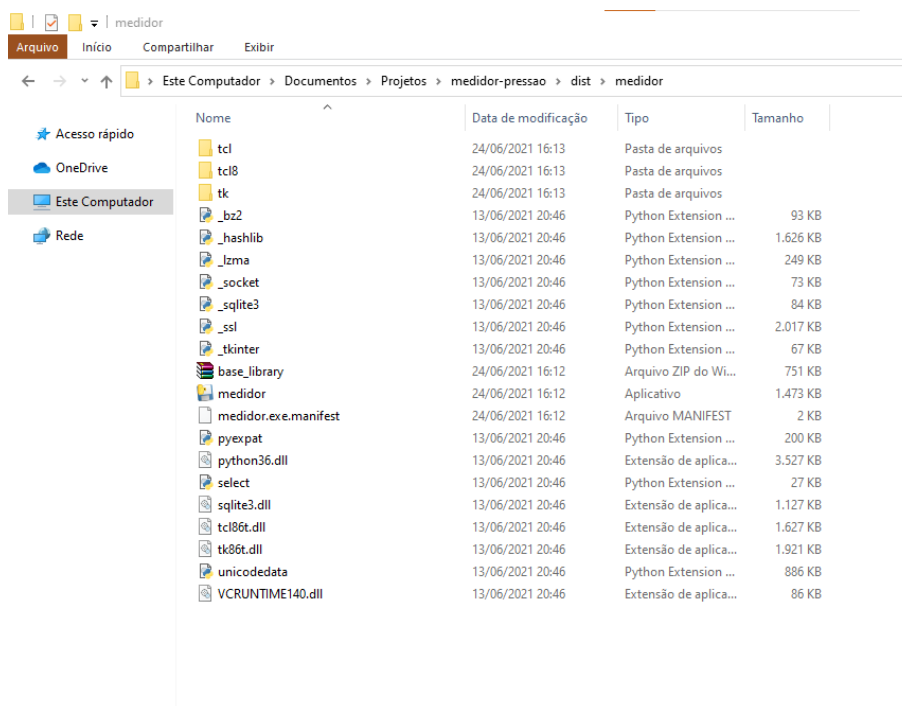


Figura 23. Diretório do aplicativo

Após uma comparação foi possível perceber que na criação do aplicativo, faltou dois arquivos principais para o aplicativo rodar da forma correto. Quais arquivos faltaram:

- Pasta icon com a logo.
- Arquivo do banco de dados.

Vamos verificar estes arquivos no diretório do projeto e iremos copiar estes arquivos com as teclas Ctrl + C:

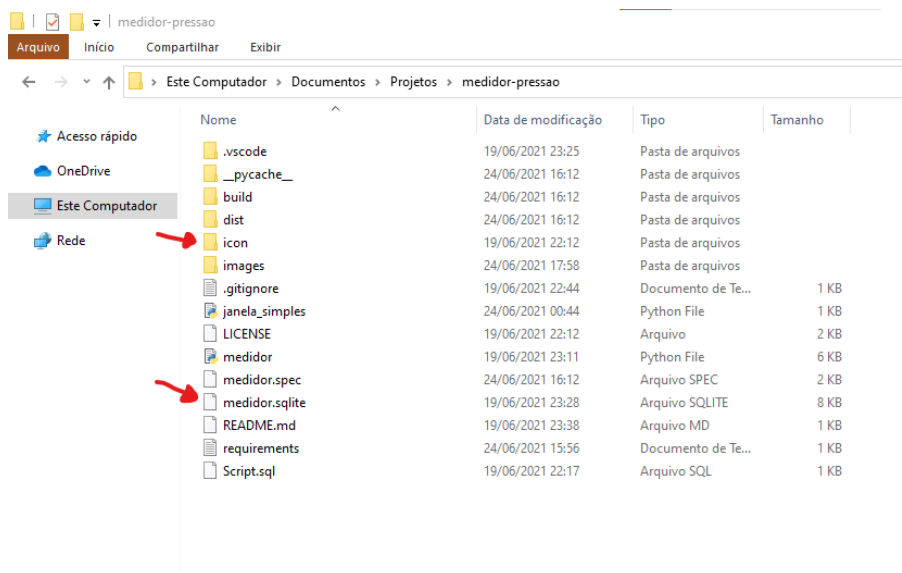


Figura 24. Cópia dos arquivos no diretório do projeto

Após a cópia, iremos colar estes arquivos no diretório do aplicativo com as teclas Ctrl + V:

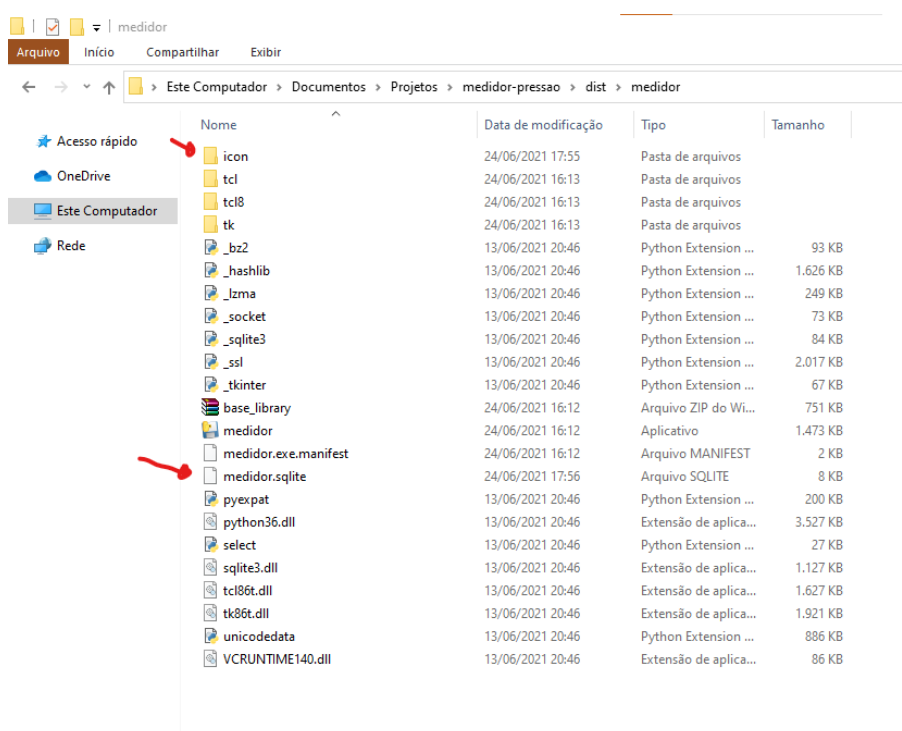


Figura 25. Colando os arquivos no diretório do aplicativo

3.2.6. Funcionamento do aplicativo

Pronto, após estas etapas, iremos realizar o teste do aplicativo novamente, e verificar se ocorreu tudo corretamente. Para isso clicaremos duas vezes no ícone do aplicativo, conforme figura abaixo:

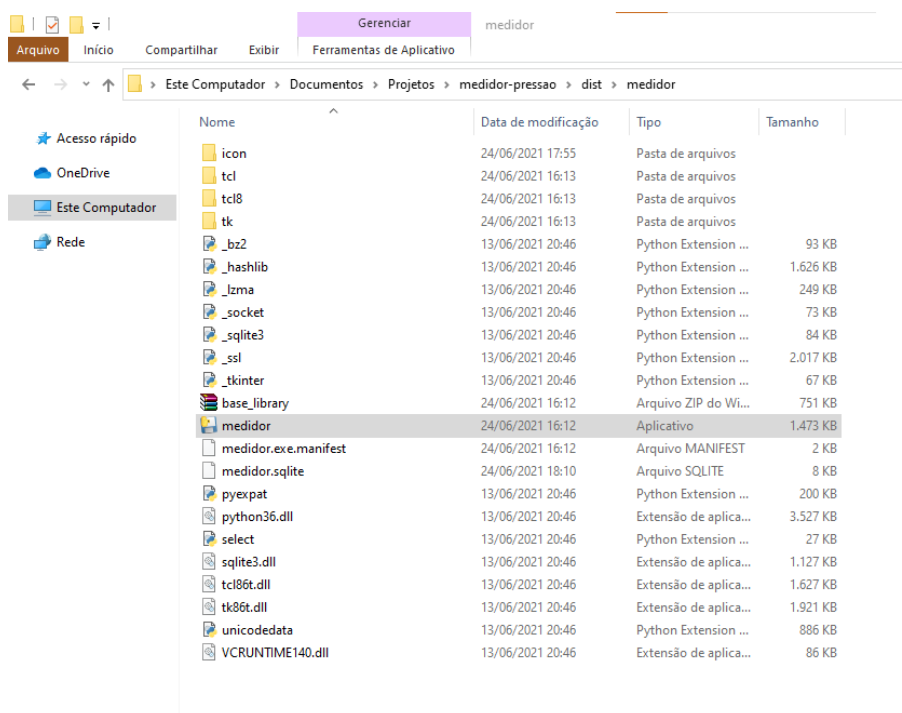


Figura 26. Executando o segundo teste

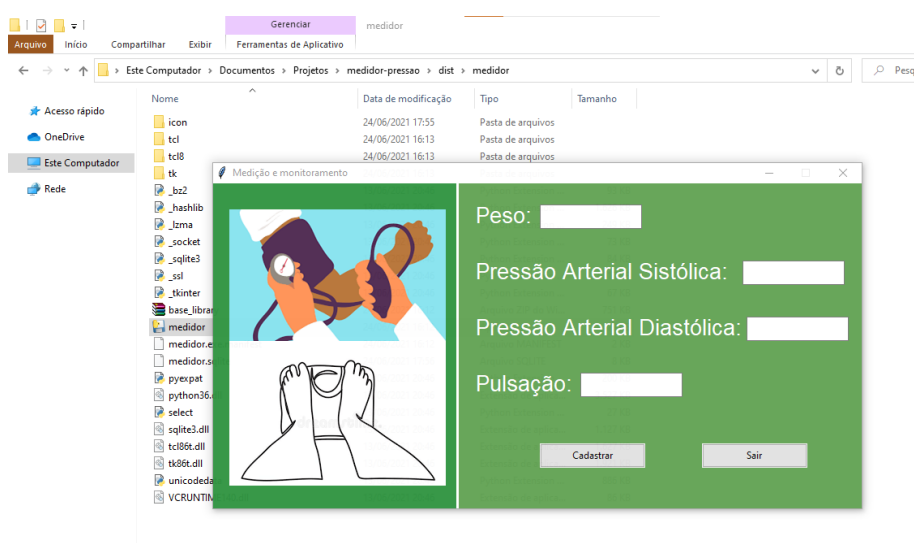


Figura 27. Funcionamento do aplicativo

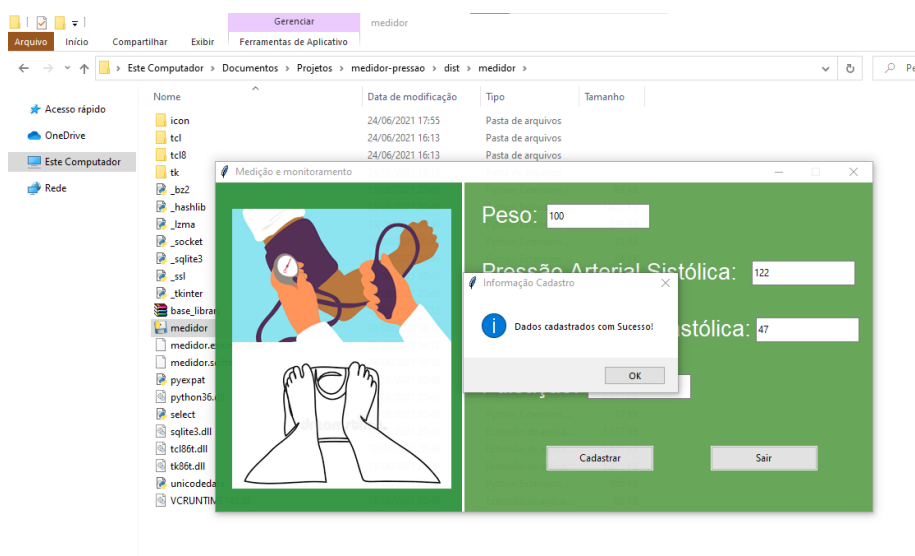


Figura 28. Teste do cadastro

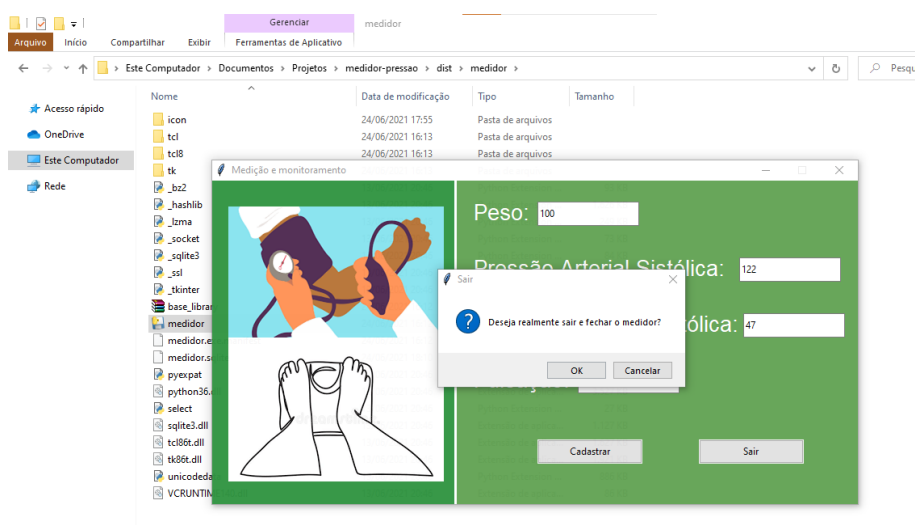


Figura 29. Teste para sair ou fechar o aplicativo

3.2.7. Criação do atalho

Para uma melhor utilização do voluntário, será criado um atalho do aplicativo, para que o voluntário escolha o melhor local para sempre lembrar e coletar suas medidas mais rapidamente, mas sendo esta etapa opcional. E para tal, seguir conforme imagem a seguir e colocar no local desejado da preferência do voluntário.

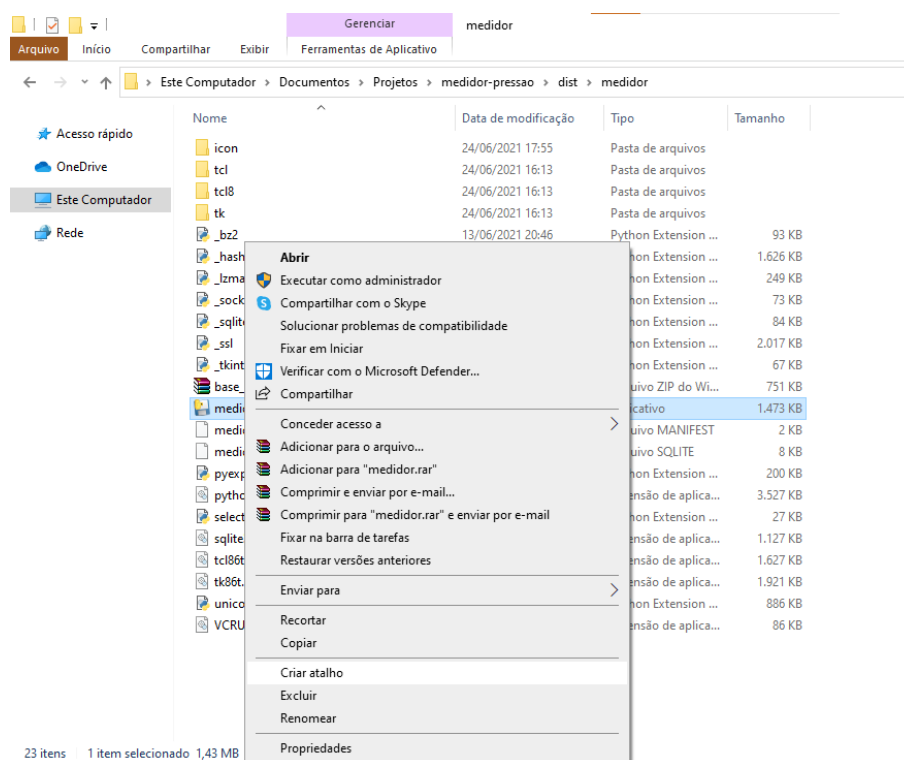


Figura 30. Criação do atalho do aplicativo

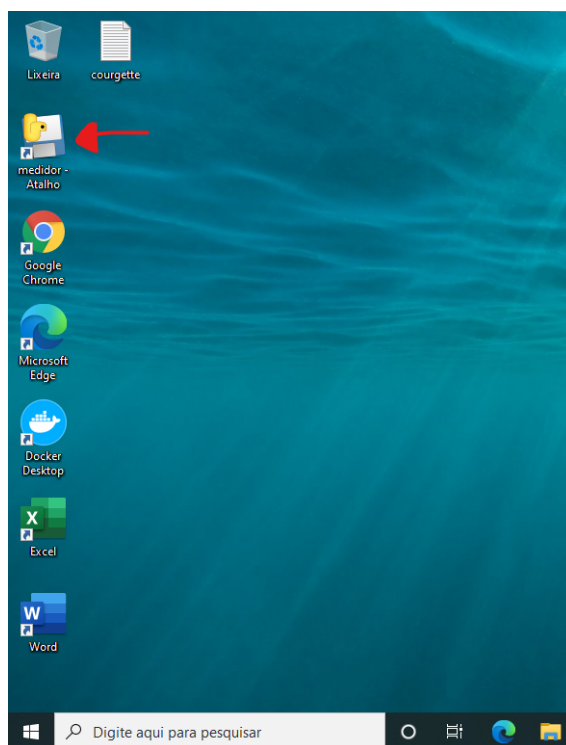


Figura 31. Atalho do aplicativo criado e na área de trabalho do Windows

3.2.8. Análise Exploratória dos dados

Nesta etapa foi realizada a análise exploratória dos dados, por meio do código abaixo, sendo este criado através da plataforma do Google Colab.

```

1 # Abrindo conexão com o Google Drive.
2 from google.colab import drive
3 drive.mount('/content/drive')
4
5 # Importando Bibliotecas
6 import sqlite3
7 import pandas as pd
8 from datetime import datetime
9 import numpy as np
10 import matplotlib.pyplot as plt
11 import seaborn as sns
12
13 #Apresentas os dados no notebook.
14 %matplotlib inline
15
16 # Definindo estilo para os gráficos.
17 sns.set_style("darkgrid")
18
19 # Tamanho da imagem.
20 plt.rcParams['figure.figsize'] = (12,8)
21
22 # Parametros do Pandas, limitando a quantidade máxima e a largura das
    colunas.
23 pd.set_option('max_columns', 120)
24 pd.set_option('max_colwidth', 5000)
25
26 # Caminho do arquivo.
27 caminho = "/content/drive/MyDrive/pi_4/medidor.sqlite"
28
29 # Criando a conexão com o banco de dados
30 def conectaBanco(caminhoarquivo: caminho, query: str):
31     try:
32         conn = sqlite3.connect(caminhoarquivo)
33         return pd.read_sql_query(query, conn)
34     except sqlite3.Error as error:
35         print("Erro ao conectar com o banco de dados SQLite", error)
36
37 # Consulta a tabela medida.
38 query = """SELECT * FROM medida"""
39 dados = conectaBanco(caminho, query)
40
41 # Proporção dos dados.
42 dados.shape
43
44 # Verificando informações dos dados.
45 dados.info()
46
47 # Primeiros dados.
48 dados.head()
49

```

```

50 # Últimos dados.
51 dados.tail()
52
53 # Estatística Básica dos dados numéricos.
54 dados.describe()
55
56 # Transformando dados em formato datetime.
57 dados['data_hora'] = pd.to_datetime(dados['data_hora'],
    format='%Y-%m-%d %H:%M:%S')
58 dados['data_hora'].head()
59
60 # Formatando a data e hora.
61 dados['data'] = dados['data_hora'].apply(lambda x:
    x.strftime('%Y-%m-%d'))
62 dados['hora'] = dados['data_hora'].apply(lambda x: x.strftime('%H:%M'))
63
64 # Obtendo dia da semana e periodo da hora se é AM ou PM.
65 dados['dia_semana'] = dados['data_hora'].apply(lambda x:
    x.strftime('%A'))
66 dados['periodo_hora'] = dados['data_hora'].apply(lambda x:
    x.strftime('%p'))
67
68 # Verificando informações dos dados após transformação.
69 dados.info()
70
71 # Verificando a proporção dos dados.
72 dados.shape
73
74 # Primeiros dados.
75 dados.head()
76
77 # Últimos dados.
78 dados.tail()
79
80 # Plotando gráfico das medidas do peso.
81 dados['peso'].hist()
82 plt.title('Frequência das medidas de peso')
83 plt.xlabel('Peso em Kg')
84 plt.ylabel('Frequência')
85 plt.show()
86
87 # Plotando gráfico das medidas do sistólica.
88 dados['sistolica'].hist()
89 plt.title('Frequência das medidas de PAS')
90 plt.xlabel('Pressão Arterial Sistólica')
91 plt.ylabel('Frequência')
92 plt.show()
93
94 # Plotando gráfico das medidas do diastólica.
95 dados['diastolica'].hist()
96 plt.title('Frequência das medidas de PAD')
97 plt.xlabel('Pressão Arterial Diastólica')
98 plt.ylabel('Frequência')
99 plt.show()
100
101 # Plotando gráfico das medidas da pulsação.

```

```

102 dados['pulsacao'].hist()
103 plt.title('Frequência das medidas de pulsação')
104 plt.xlabel('Pulsação')
105 plt.ylabel('Frequência')
106 plt.show()
107
108 # Gráfico de comparação entre a peso e a data da medição.
109 sns.lineplot(data=dados, x='data', y='peso', style_order='whitegrid')
110 plt.title('Peso por data da medição')
111 plt.ylabel('Peso em Kg')
112 plt.xlabel('Data')
113 plt.show()
114
115 # Gráfico de comparação entre a PAS e a data da medição.
116 sns.lineplot(data=dados, x='data', y='sistolica',
117               style_order='whitegrid')
118 plt.title('PAS por data da medição')
119 plt.ylabel('PAS')
120 plt.xlabel('Data')
121 plt.show()
122
123 # Gráfico de comparação entre a PAD e a data da medição.
124 sns.lineplot(data=dados, x='data', y='diastolica',
125               style_order='whitegrid')
126 plt.title('PAD por data da medição')
127 plt.ylabel('PAD')
128 plt.xlabel('Data')
129 plt.show()
130
131 # Gráfico de comparação entre a pulsação e a data da medição.
132 sns.lineplot(data=dados, x='data', y='pulsacao',
133               style_order='whitegrid')
134 plt.title('Pulsação por data da medição')
135 plt.ylabel('Pulsação')
136 plt.xlabel('Data')
137 plt.show()
138
139 # Agrupando por periodo_hora e dia_semana.
140 dados.groupby(by=['periodo_hora', 'dia_semana']).mean()

```

4. Avaliação dos Resultados

Com este projeto aprendi muito sobre o aspecto da criação de GUIs (Interface Gráfica do Usuário) em português. Onde por meio da biblioteca Tkinter na linguagem de programação Python, foi possível criar uma janela para coleta dos dados de medição de peso, pressão arterial sistólica, pressão arterial diastólica, pulsação, e gravar em um banco de dados em arquivo SQLite de forma estrutura.

Na criação do aplicativo desktop, me esbarrei com um problema em que a janela após a criação do aplicativo não aparecia, conforme descrição no ponto 3.2.4. Primeiro teste, e 3.2.5. Erro e a solução. Neste após pesquisas encontrei uma direção para a solução do problema encontrado, onde me deparei com um relato parecido, de acordo com [e Programação 2020].

5. Conclusões

Por meio da análise exploratória dos dados, que pode ser encontrada neste link do repositório do GitHub:

```
https://github.com/rafhaelom/DataScience/blob/master/ProjetosFaculdade/Medidor\_Peso\_e\_Pressao\_Arterial/Analise\_medidor.ipynb
```

Foi possível perceber, que pelo fato de não ter tido mais voluntários para realizar as medidas, não pude criar perguntas a se responder pois os dados se refletem há um único indivíduo/voluntário. Portanto não poderia com os resultados criar uma análise aprofundada e dizer que para todos seria o mesmo resultado, ou seja, estaria criando viés com os dados disponíveis.

5.1. Relato da Experiência

Por meio deste projeto integrador, pude colocar em prática os conhecimentos adquiridos em programação, análise de dados, banco de dados, pesquisa sobre o tema de obesidade, hipotensão arterial, hipertensão arterial, como também desenvolvimento de aplicativos, terminal Linux e CMD do Windows.

Pretendo continuar pesquisando o tema deste projeto integrador, como também como próximos passos, pretendendo procurar um meio de encontrar novos voluntários, criação de janela de login e tabela com dados de usuários, para uma melhor separação dos dados de medição de cada voluntário, e criação de um painel interativo, também conhecido como Dashboard como atualização em tempo real para cada registro de medição adicionado no banco de dados, como devidos filtros para usuário e demais filtros a serem criados e suas visualizações, para ser possível a ideia principal do projeto de criar um medidor para o monitoramento de medidas de saúde.

Referências

- Association, A. H. (2021). Understanding blood pressure readings. <https://www.heart.org/en/health-topics/high-blood-pressure/understanding-blood-pressure-readings>.
- Biblioteca Virtual em Saúde, M. d. S. 04/3 - dia mundial da obesidade. <https://bvsmis.saude.gov.br/04-3-dia-mundial-da-obesidade/>.
- BR, D. (2020). Integrando banco de dados com tkinter (python) temp 01 sistema de login. <https://www.youtube.com/playlist?list=PLusI35eERodl2U3CD5ajrB3p9McKiuduL>.
- Claw20017, S. (2021). Tkinter. <https://wiki.python.org/moin/TkInter>.
- e Programação, E. (2020). Gerar executável python com interface gráfica. <https://www.youtube.com/watch?v=m49ftKzge0k>.
- eXcript (2018). Curso de python - módulo tkinter. <https://www.youtube.com/playlist?list=PLesCEcYj003ShHnUT83gQEH6KtG8uysUE>.
- Ferg, S. Pensando em Tkinter. Grupo Python, Departamento de Engenharia Mecânica, UNESP - Ilha Solteira.
- Foundation, P. S. (2001-2021a). Interfaces gráficas de usuário com tk. <https://docs.python.org/pt-br/3/library/tk.html>.
- Foundation, P. S. (2001-2021b). tkinter - interface python para tcl/tk. <https://docs.python.org/pt-br/3/library/tkinter.html#tkinter-modules>.
- Levi D. Procter, MD, V. C. U. (2020). Hipotensão arterial. <https://www.msmanuals.com/pt-br/casa/dist%C3%BArbios-dor%C3%A7%C3%A3o-e-dos-vasos-sangu%C3%ADneos/hipotens%C3%A3o-arterial-e-choque/hipotens%C3%A3o-arterial>.
- Luiz, R. D. S. O que é hipotensão arterial? <https://www.rededorsaoluiz.com.br/sintomas/hipotensao-arterial>.
- Profª. Drª. Cibele Isaac Saad Rodrigues, D. d. d. d. H. d. S. Entenda o que é pressão arterial. <https://www.sbn.org.br/orientacoes-e-tratamentos/doencas-comuns/hipertensao-arterial/>.
- Ribeiro, J. (2019). Python tkinter. https://www.youtube.com/playlist?list=PLXik_5BrzO_m8NaaEix1pyQOsCZM7t1h.