MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

DESENVOLVIMENTO EM PYTHON DE ROTINAS PARA AUXÍLIO NO PROCESSAMENTO DE DADOS MAGNETOTELÚRICOS

Cecile Lopes Damázio Rodrigues

Relatório de Iniciação Científica do programa PIBIC, orientado pela Dra. Andrea Cristina Lima Santos Matos

INPE
São José dos Campos
2023

RESUMO

Este projeto teve como objetivo desenvolver uma interface de programação de aplicações (API - Application Programming Interface), para auxiliar no processamento Rho+[1] do método Magnetotelúrico (MT), em linguagem Python. O processamento Rho+ é usado para validar as curvas de resistividade e fase magnetotelúricas que serão usadas para determinar a distribuição de condutividade elétrica do interior da Terra e diminuir a relação sinal-ruído nos dados obtidos pelo método MT. O projeto cogita facilitar esta etapa, que consiste em procedimentos manuais e escrita das linhas de comando, por meio do desenvolvimento de um software em linguagem Python que formam um software interativo e simples para a utilização do Rho+. A interface gráfica foi feita usando a biblioteca Tkinter do Python, com diversos outros módulos e bibliotecas para o funcionamento adequado dos scripts. O programa desenvolvido permitirá ao usuário recorrer à interface para seleção dos dados coerentes reais e aqueles que desejam ser modelados por meio do pacote Rho+.

Palavras-Chave: Rho+. Interface Gráfica. Método Magnetotelúrico.

LISTA FE FIGURAS

SUMÁRIO

1. INTRODUÇÃO

O método magnetotelúrico (MT) é um método geofísico que fornece modelos sobre a distribuição da condutividade elétrica na subsuperfície da Terra e caracteriza-se por ser um método eletromagnético que usa as variações naturais do campo geomagnético como fonte do sinal. Esse método deriva funções de transferência entre os campos elétrico e magnético da Terra; as curvas de resistividade e fase MT, que são validadas pela técnica de processamento de dados Rho+ (Parker e Booker, 1996). Esta técnica auxilia na avaliação da qualidade dos dados através da diminuição da sua relação sinal-ruído. A execução do Rho+ é feita por meio de um software cujo uso é inteiramente manual, onde os comandos, seleções e mudanças de etapas da técnica são todos digitados no terminal do computador. Esse processo é propenso a mais erros do usuário devido a existência de longas rotinas manuais, além de não ser prático, e por isso foi feita a proposta da implementação de uma aplicação para a execução do Rho+ que viesse a automatizar essas rotinas.

O objetivo deste trabalho é fazer uma API (Interface de Programação de Aplicações) com uma GUI (Interface Gráfica do Usuário). A API, de acordo com a definição descrita pela International Business Machines Corporation (IBM, 2016), é "um conjunto de funções que fornecem alguma capacidade de negócio ou técnica e pode ser chamada por aplicativos usando um protocolo definido"; Já a GUI, é uma área de interação gráfica do usuário com a máquina. Assim, seria possível unir o Rho+ à uma interface gráfica e à uma aplicação por meio de uma API, de forma a deixar a execução do Rho+ amigável, facilitando o processo.

2. DESENVOLVIMENTO

2.1 Revisão de Literatura

Na programação, há desafios quando se trata de pesquisa, uma vez que, além de profundamente colaborativa, é uma área ampla em que há infinitos modos de resolver um mesmo problema. Por isso, nessa revisão de literatura, cursos, livros e fóruns serão considerados e devidamente referenciados. Para o início do trabalho, foi feito o estudo do curso "Desenvolvendo uma aplicação GUI com Python e Tkinter"[3] na plataforma Udemy[4], lecionado e preparado por Marcos Castro de Souza (UFPI); consultas nos livros Python Crash Course[5], Automate The Boring Stuff With Python[6], e nas documentações oficiais de Python[7] e do Tkinter[8] (TK) (Souza, 2016).

2.1.1 Desenvolvimento por meio do curso "Desenvolvendo uma aplicação GUI com Python e Tkinter"[2]

Neste curso, o professor Marcos Castro de Souza ensina o desenvolvimento de uma aplicação com interface gráfica com a versão 2.7.3 do Python e 2.4 do Tkinter,

onde Python é uma linguagem de programação de alto-nível orientada a objetos e Tkinter, é uma biblioteca de Python usada para fazer interfaces gráficas. Como as versões atuais são 3.10 do Python e 8.6 do TK, foi necessária a consulta constante dos sites de documentação das ferramentas usadas, já que a discrepância da versão do TK usada no curso para a usada no projeto impede que os códigos escritos com a mesma forma de parâmetros, comandos e funções funcionem. Marcos Castro apresenta o uso da biblioteca de forma prática e ensina o básico para a construção da GUI (Udemy, 2023).

A primeira parte começa com a instalação do Python, a recomendação do uso de um IDE (Ambiente de Desenvolvimento Integrado) e a introdução da ferramenta de interface gráfica com a apresentação de suas vantagens, que incluem simplicidade e portabilidade entre sistemas operacionais. Esses foram parâmetros importantes na escolha do Tkinter para a realização da interface do Rho+, pois esses elementos foram explicitados como fundamentais para os usuários. Em seguida, recomenda-se que haja a criação de dois códigos separados, um para a execução do programa usado na API e outro apenas para a interface; conselho em que foi aplicado neste projeto a fim de tornar a organização do projeto mais simples e acessível. A segunda parte do curso se trata do desenvolvimento da aplicação com as ferramentas gráficas, na qual são apresentados os termos Widget, Frame, Listbox e Scrollbar; é feita a integração entre a interface e o código de execução do software e a demonstração do código-exemplo criado ao longo das aulas (Sweigart, 2020; Matthes, 1972).

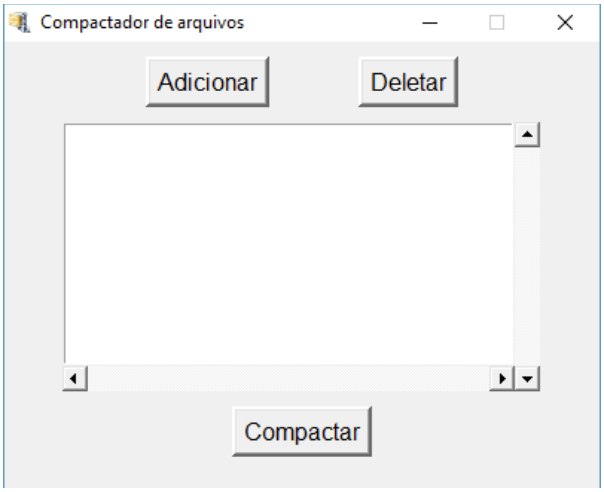
- Frame

Na linguagem de interface gráfica, frame é a superfície em que outros elementos serão colocados.

- Widget

É um elemento interativo de qualquer natureza, cuja ativação pode ser através de um clique, por exemplo. Como há vários tipos de widgets, seu formato varia para o que será destinado. No código descrito por Marcos para fazer um app que compacta arquivos, é possível ver 3 botões e duas barras de rolagem. Ambos elementos são widgets (Figuras 1 e 2).

Figura 1: Aplicativo para compactar arquivos.



Fonte: https://github.com/marcoscastro/compactador_arquivos (2016)

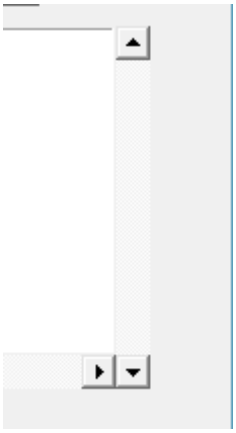
- Listbox

Um tipo de widget que é uma caixa em que será colocada uma lista de arquivos. Está exemplificado na Figura 1, no meio da figura. O espaço vazio em branco é onde ficam os arquivos escolhidos.

- Scrollbar

Widget de barra de rolagem, para “rolar” a página.

Figura 2: Recorte da scrollbar, barra de rolagem, na Figura 1.



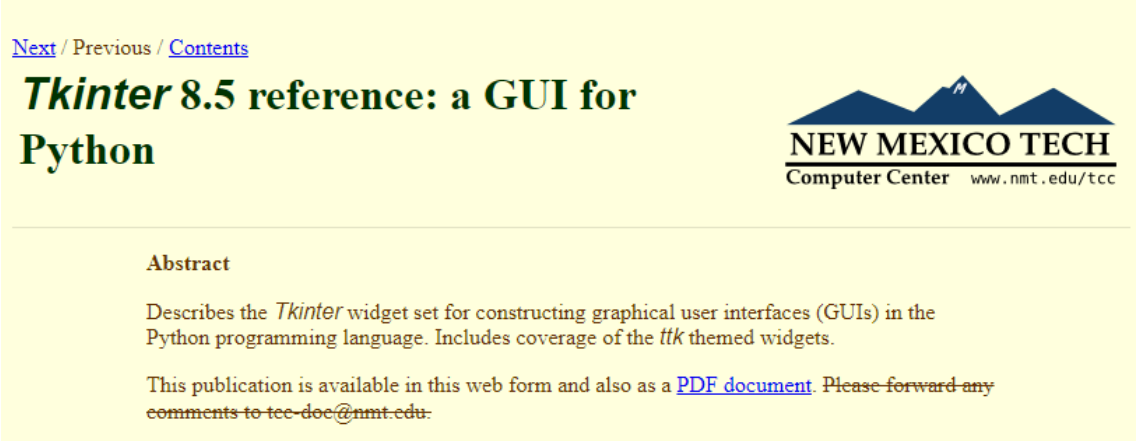
Fonte: Github, 2023. https://github.com/marcoscastro/compactador_arquivos.

Durante as aulas, os códigos em Python eram desenvolvidos durante a gravação e executados, o que permitiu o acompanhamento do curso e entendimento dos conceitos ensinados.

2.1.2 Documentação do Tkinter

Devido à diferença das versões do Python e do Tkinter usadas de base no curso, foi feita a pesquisa em fóruns do GitHub[9], na documentação oficial do Tkinter e também o método da tentativa e erro (Tkinter, 2020). Na programação é muito comum que a atualização de linguagens e bibliotecas inutilizem códigos por incompatibilidade, assim como bugs que dependem de outros fatores; por isso, há o processo chamado de *debugging*, sendo desse conserto de *bugs* (problemas, erros, falhas de design) por meio de tentativas, mudanças no código dependente, ou seja, da habilidade do programador. Para a identificação dos erros, foi checada a área da IDE em que é exibido o motivo de falha e a sua posição, e caso tivesse relação com erro nos comandos, a documentação era verificada e comparações eram feitas entre as versões para o conserto do problema (Figuras 3).

Figura 3: Página de Conteúdos do Tkdocs, página de documentação do Tkinter.



Fonte: https://tkdocs.com/shipman/index.html. Capturada em 5 de agosto de 2023.

2.2 Materiais e Métodos

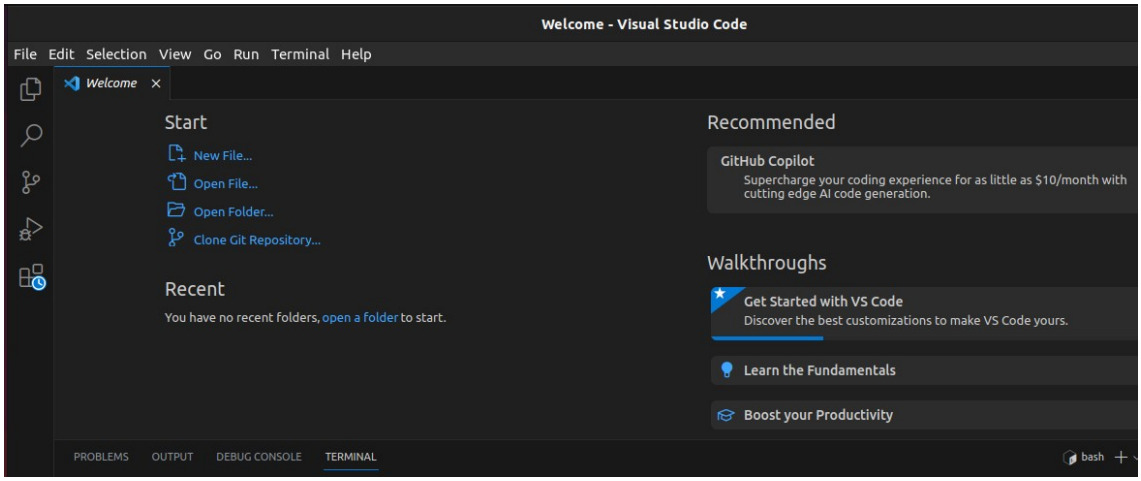
2.2.1 Materiais

2.2.1.1 A IDE VSCode[10]

A IDE, Integrated Development Environment, Ambiente de Desenvolvimento Integrado em português, é uma aplicação para o desenvolvimento e edição de códigos que possui suporte para a identificação de erros, paleta de cores para a identificação de determinadas palavras-chave, ferramentas de debugging e compilação de códigos, dentre outras funcionalidades que variam de uma IDE para a outra. Neste projeto foi

usado o VSCode (Vscode, 2023), IDE desenvolvida para os sistemas operacionais Windows, Linux e MacOS (Figura 4).

Figura 4: Captura de Tela do VSCode



Fonte: Autor.

2.2.1.2 Terminal do Linux (*shell)*

O terminal do Linux, também chamado de *shell*, é uma ferramenta de baixo-nível que envia instruções para o sistema operacional (OS) por meio da digitação de linhas de comando. É pelo terminal que todas as instruções para o Rho+ são feitas. Tem uma linguagem específica e pode ter o uso automatizado por meio de scripts.

2.2.2 Métodos

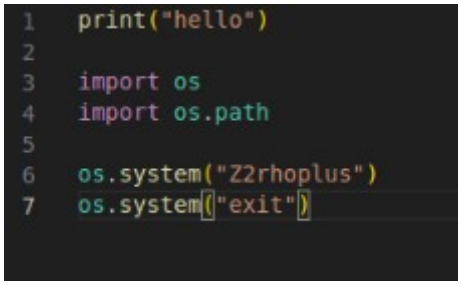
Neste projeto, foram criados dois programas de extensão .py (em Python): aplicacao.py e interface.py, onde o primeiro é para as funções e execuções do Rho+ e, o segundo, inteiramente para a parte gráfica. A API é uma interface (não necessariamente gráfica) composta por funções e comandos que relacionam aplicações e tornam possíveis a comunicação entre elas. No caso do Rho+, este software será conectado à interface gráfica por meio de um programa e a API é o que os une. A realização do trabalho foi feita a partir do seguinte fluxograma (Figura 5).

Figura 5: Fluxograma do Trabalho, feito a partir de um design no Canva (Canva, 2023).

Fonte: Autor.

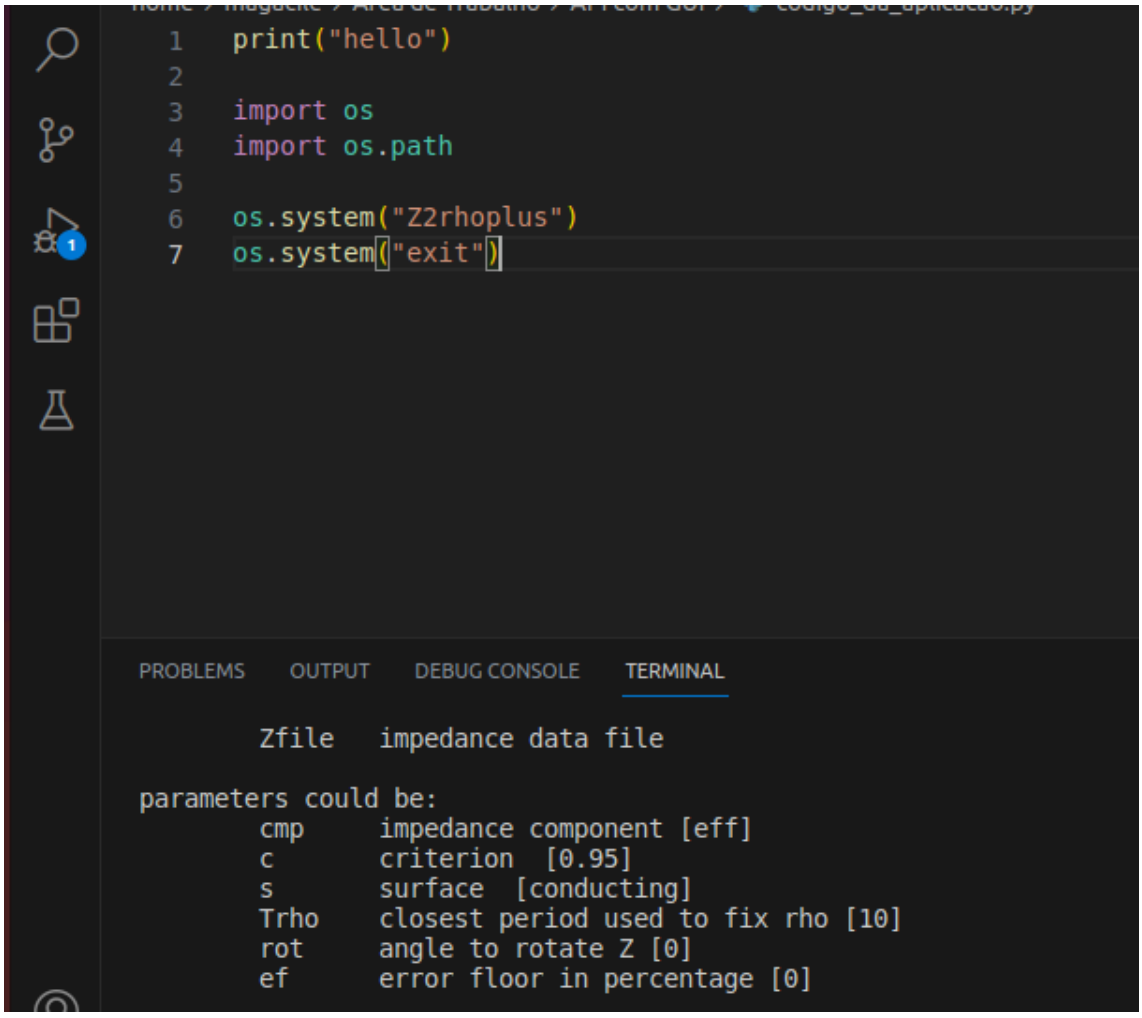
As primeiras etapas consistiram na instalação de módulos e programas, desenvolvidos pelo grupo de Geomagnetismo do Instituto Nacional de Pesquisas Espaciais – GEOMA/INPE. Esses módulos e programas são necessários para executar o Rho+. Após as instalações e testes, houve um estudo para descobrir como seria possível fazer com que os comandos fossem automaticamente postos no terminal. E assim segue-se para a parte 2. O primeiro tipo de comando a ser resolvido foi a partir da biblioteca `os` do python, que serve para fazer uma integração entre a linguagem e o sistema operacional. Como a execução do Rho+ depende muito da manipulação de arquivos com nomes variáveis e comandos específicos do software, seriam necessárias palavras-chave da linguagem que tornassem possível o envio em baixo-nível de informações para o OS. Isso é possível por meio da importação do módulo os.path da biblioteca OS (Figuras 6 e 7).

Figura 6: Teste do módulo os.path



Fonte: Autor.

Figura 7: Retorno do código na figura 6.

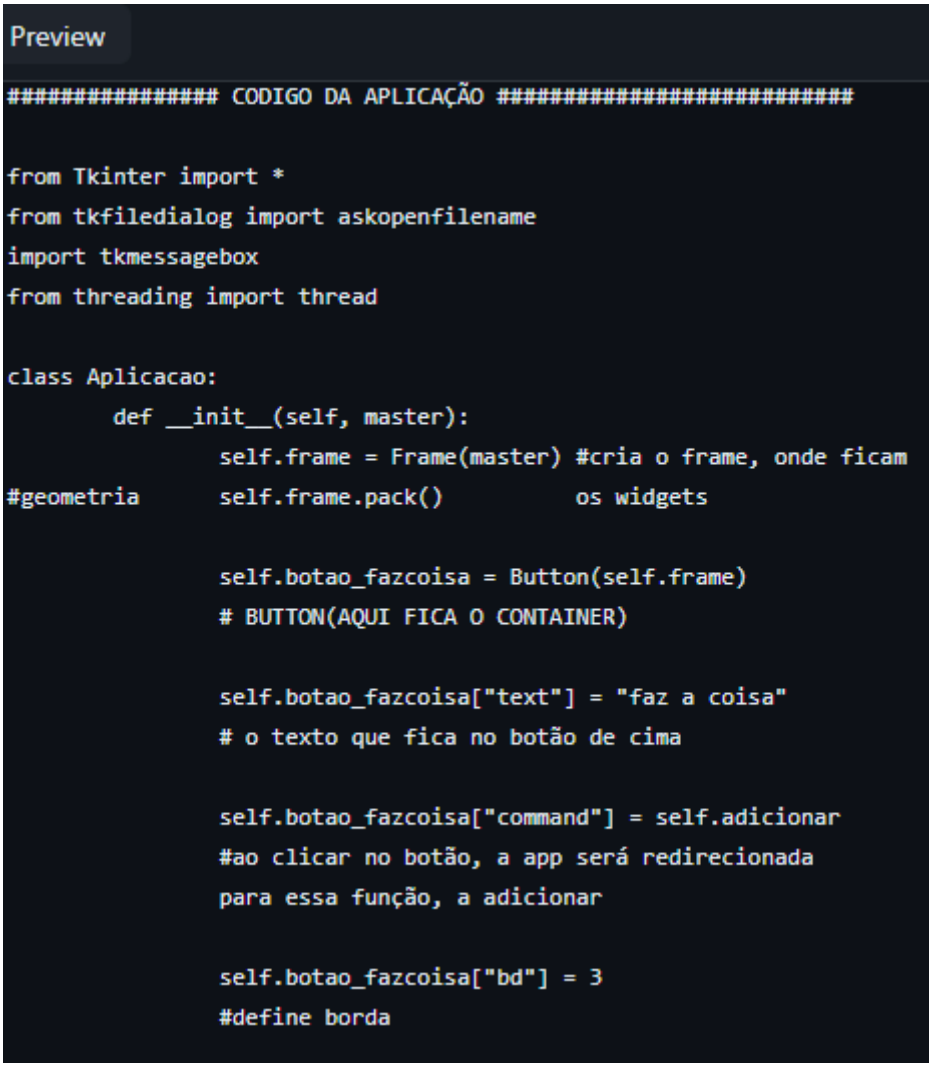


Fonte: Autor.

Por meio deste teste, foi comprovado que era possível fazer a chamada de palavras diretamente para o terminal, independentemente do que fosse escrito. Visto isso, foi iniciada a etapa de desenvolvimento da interface gráfica, para ter uma ideia de

como os widgets se organizavam e o que poderia ativar os comandos para o Rho+. Para o início desta etapa, foi feito um documento com um resumo de como eram formatados os widgets na biblioteca do Tkinter (Figura 8).

Figura 8: Captura de tela do Github, onde está o notas.txt



Fonte: Autor.

Isso foi feito com o objetivo de testar o Tkinter, visualizá-lo e provar se realmente seria adequado para fazer a interface do Rho+. O código atual da interface é mostrado a seguir:

```
#importando

from tkinter import *
from tkinter import filedialog
from tkinter import messagebox
```

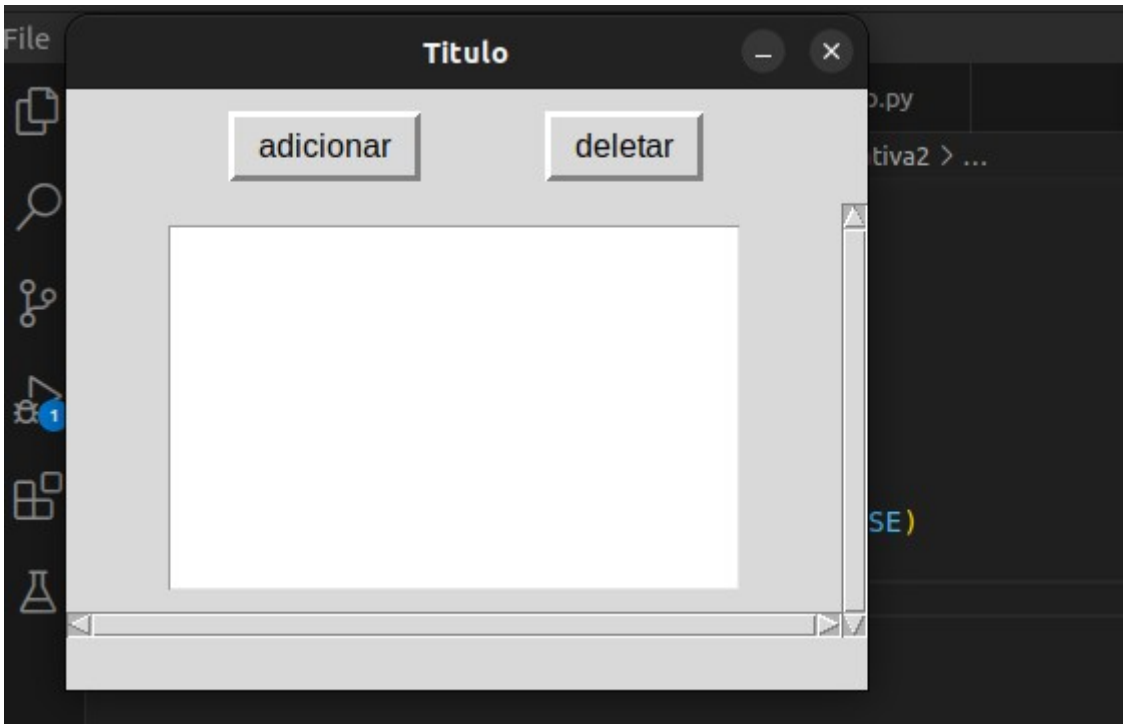
```
      from threading import Thread

      class aplicacao:

          def __init__(self, master):
              self.frame = Frame(master)
              self.frame.pack()



              self.botao_adicionar = Button(self.frame)
              self.botao_adicionar["text"] = "adicionar"
                          self.botao_adicionar["command"]   =
self.adicionar
              self.botao_adicionar["bd"] = 3
              self.botao_adicionar["font"] = ("Arial", 12)

              #agora sera organizada a geometria
                self.botao_adicionar.pack(pady=10, padx= 30,
side="left")

              #agora se faz o botao deletar

              self.botao_deletar = Button(self.frame)
              self.botao_deletar["text"] = "deletar"
              self.botao_deletar["command"] = self.deletar
              self.botao_deletar["bd"] = 3
              self.botao_deletar["font"] = ("Arial", 12)
                          self.botao_deletar.pack(padx=  30,
side="right")

                  #agora será feita uma listbox, na mesma
identação dos self.
              self.frame2= Frame(master)
              #tem que ter dois packs
              self.frame2.pack()

              #agora, para fazer a scrollbar

              # **** Scrollbar em y ******
              self.sby = Scrollbar(self.frame2)
              self.sby.pack(side=RIGHT, fill=Y)
              # **** Scrollbar em x ******
```

```
                        self.sbx   =   Scrollbar(self.frame2,
orient=HORIZONTAL)
            self.sbx.pack(side=BOTTOM, fill=X)



             self.listbox= Listbox(self.frame2, width=50,
height= 10, selectmode=EXTENDED)

            #tem q colocar esse pack aqui em baixo com os
pads, e tem que ser nessa ordem
            self.listbox.pack(pady= 10, padx= 50)

             #agora vamos anexar a listbox para as barras
de rolagem

            self.listbox.config(yscrollcommand=self.sby.se
t)
            self.sby.config(command=self.listbox.yview)

            self.listbox.config(xscrollcommand=self.sbx.se
t)
            self.sbx.config(command=self.listbox.xview)



        def adicionar(self):
            pass
        def deletar(self):
            pass




    root= Tk()
    root.title("Titulo")
    root.geometry("400x300")
    root.resizable(width=FALSE, height=FALSE)
    aplicacao(root)
    root.mainloop()
```

3. RESULTADOS E DISCUSSÃO

O resultado deste trabalho, realizados com os códigos mostrados no capítulo anterior, está apresentado na Figura 9. O projeto ainda não foi finalizado, uma vez que o objetivo principal é fazer toda interface do Rho+. Neste projeto foi possível desenvolver a parte inicial da interface e da API, adquirindo conhecimento da construção de APIs e GUIs desde o nível mais básico, fazendo uma interface gráfica que funciona e recebe interação do usuário e comprovando que o Tkinter é uma ferramenta adequada e simples com potencial para a implementação do projeto final.

Figura 9: protótipo de GUI finalizado.



Fonte: o boslsita.

4. CONCLUSÃO

Neste projeto foi proposta uma interface de programação gráfica para validação de curvas de resistividade e fase magnetotelúricas. Foi também mostrada a capacidade de se chamar o programa rho+, usado como base para as validações, a partir da interface gráfica, parte essencial para a garantia da qualidade dos gráficos que serão exibidos pela interface gráfica quando sua implementação for finalizada em trabalhos futuros.

5. REFERÊNCIAS

Canva, 2023. Disponível em <https://www.canva.com/pt_br/>. Acesso em 6, Ago, 2023.

Egbert, G. D.; D. W. Livelybrooks. Single station magnetotelluric impedance estimation: Coherence weighting and the regression *M*-estimate. Geophysics, v. 61, n. 4, p. 964–970, 1 jul. 1996.

Github, 2023. Disponível em <https://github.com/>. Acesso em 6, Ago, 2023.

IBM, 2021. Criando uma definição de API. Disponível em <https://www.ibm.com/docs/pt-br/api-connect/10.0.1.x?topic=definitions-creating-api-definition>. Acesso em: 6, Ago, 2023.

Matthes, E. 1972. Python Crash Course: A Hands-On, Project-Based Introduction to Programming. 2a Edição. San Francisco: No Starch Press, Inc. 1972.

Parker, R. L.; Booker, J. R. Optimal one-dimensional inversion and bounding of magnetotelluric apparent resistivity and phase measurements. Physics of the Earth and Planetary Interiors, v. 98, n. 3-4, p. 269–282, 1 dez. 1996.

Python, python.org, 2023. Disponível em <python.org>. Acesso em: 6, Ago, 2023.

Souza, M. C. 2016. Desenvolvendo uma aplicação GUI com Python e Tkinter. Jun, 2016. Disponível em <https://www.udemy.com/course/python-tkinter/>. Acesso em 6, Ago, 2023.

Sweigart, A. 2020. Automate the Boring Stuff With Python, 2a Edição. San Francisco: No Starch Press, Inc. 2020.

Tkinter. 8.6. New Mexico, 2020. https://tkdocs.com/pyref/index.html. Acesso em: 6, Ago, 2023.

Udemy, 2023. Disponível em <https://www.udemy.com/>. Acesso em: 6, Ago, 2023.

VSCode, 2023. Disponível em <https://code.visualstudio.com/>. Acesso em: 6, Ago, 2023.