Uma imagem com logótipo

Descrição gerada automaticamente

**Instituto Tecnológico e Profissional da Figueira da Foz**

Ano letivo

Curso de Programação e informática

**Terminal Beta 2.1.21: Um Terminal Simplificado para Windows**

Autor: Gabriel Wagner

2024

Índice

[Introdução 3](#_30j0zll)

[Capítulo A 4](#_1fob9te)

[Capítulo B 4](#_3znysh7)

[Capítulo C 4](#_2et92p0)

[Capítulo D 4](#_tyjcwt)

[Conclusão 5](#_3dy6vkm)

[Bibliografia 6](#_1t3h5sf)

# Introdução

Este trabalho é sobre o desenvolvimento de um terminal simplificado para o sistema operacional Windows, denominado "Terminal Beta 2.1.21". O objetivo deste projeto é proporcionar uma maneira mais prática de utilizar algumas funções do terminal, facilitando o acesso a informações do sistema e a execução de comandos frequentemente usados.

É objetivo deste trabalho:

* Facilitar a interação do usuário com o terminal do Windows.
* Automatizar comandos comuns do terminal.
* Proporcionar uma interface gráfica simples e intuitiva.

Está organizado em quatro capítulos:

1. No capítulo 1, será abordado o desenvolvimento da interface gráfica utilizando a biblioteca Tkinter do Python.
2. No capítulo 2, exploramos a implementação de comandos do sistema operacional.
3. No capítulo 3, discutimos as funcionalidades adicionais incluídas no terminal.
4. No capítulo 4, apresentamos testes e resultados obtidos com o uso do terminal.

A metodologia utilizada foi a pesquisa bibliográfica, complementada com a implementação prática e testes do software desenvolvido.

# Desenvolvimento da Interface Gráfica

O desenvolvimento da interface gráfica foi realizado utilizando a biblioteca Tkinter, que é uma das bibliotecas padrão do Python para a criação de interfaces gráficas. A interface foi projetada para ser simples e intuitiva, com botões que executam funções específicas ao serem clicados.

from tkinter import \*

def fechar\_root():

root.destroy()

root = Tk()

root.geometry('250x90')

root.title('Terminal Beta')

root.resizable(False,False)

texto = Label(root, text='Seja bem-vindo a nova versão do terminal \nclique entrar pra usufruir!')

texto.grid(column=0, row=0)

butao = Button(root, text='Entrar', command=fechar\_root)

butao.place(relx=0.40,rely=0.35)

root.mainloop()

A primeira parte da interface exibe uma mensagem de boas-vindas e um botão "Entrar" que, ao ser clicado, fecha a janela inicial e abre a janela principal do terminal.

# Implementação de Comandos do Sistema Operacional

Neste capítulo, discutimos a implementação dos comandos do sistema operacional que são executados quando os botões da interface gráfica são pressionados. Utilizamos a biblioteca subprocess do Python para executar comandos do terminal do Windows.

Exemplo de comando para exibir a configuração de rede:

def ipconfig():

COMMAND = 'start cmd /k "color 0c && ipconfig"'

coiso = subprocess.run(COMMAND, shell=True)

print(coiso.stdout)

Outros comandos implementados incluem a exibição de drivers, informações do sistema, processos em execução, entre outros.

# Funcionalidades Adicional

Além dos comandos básicos do sistema operacional, o terminal inclui funcionalidades adicionais como:

* **Star Wars**: Executa o comando para assistir o filme "Star Wars" em ASCII via Telnet.
* **Relatório de Bateria**: Gera um relatório detalhado do estado da bateria.
* **Limpeza de Arquivos Temporários**: Abre a ferramenta de limpeza de disco do Windows.

def pastas():

command = "start cmd /k telnet towel.blinkenlights.nl"

coiso = subprocess.run(command, capture\_output=True, text=True, shell=True)

print(coiso.stdout)

Estas funcionalidades visam proporcionar uma experiência mais completa e divertida para o usuário.

# Testes e Resultados

Durante o desenvolvimento, foram realizados vários testes para garantir que todos os comandos e funcionalidades estavam funcionando corretamente. Os testes incluíram a execução de cada comando em diferentes versões do Windows para verificar a compatibilidade e a robustez do terminal.

Os resultados mostraram que o Terminal Beta 2.1.21 é uma ferramenta eficaz para simplificar a interação com o terminal do Windows, oferecendo uma interface amigável e funcional.

# Conclusão

Este trabalho foi muito importante para o aprofundamento no desenvolvimento de interfaces gráficas e na automação de comandos do terminal do Windows. A criação do Terminal Beta 2.1.21 permitiu compreender melhor a integração entre o Python e o sistema operacional Windows, além de desenvolver competências em programação, organização e comunicação de informações.

Este projeto facilitou o acesso a comandos importantes do sistema, proporcionando uma ferramenta útil tanto para usuários comuns quanto para profissionais de TI.

# Bibliografía

<http://www.clicaqui.com>

<https://docs.python.org/3/library/tkinter.html>

<https://docs.python.org/3/library/subprocess.html>

<https://stackoverflow.com>

<https://pythonacademy.com.br/blog/a-biblioteca-os-do-python>

<https://discuss.python.org/t/tkinter-text-insert/44527>

<https://dtnetwork.com.br/comandos-basicos-do-cmd/>

https://www.treinaweb.com.br/blog/manipulando-arquivos-com-python