Uma imagem com logótipo

Descrição gerada automaticamente

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

Ano letivo 2023/2024

Curso de AC

**Criação de um bloco de notas pessoal**

Autor:Jeizy do Nascimento luciano

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

O presente trabalho é sobre a criação de um bloco de notas pessoais, mais concretamente por um bloco de anotação.

É objetivo deste trabalho abordar sobre tudo que aprendeste sobre python durante as aulas .

Está organizado em 4 capítulos. No capítulo 1, será abordado como está formado. No capítulo 2 optei por abordar sobre a interface gráfica do código e no capítulo 3 optei por falar sobre o funcionamento do aplicativo .

A metodologia utilizada foi a pesquisa bibliográfica, enriquecida com algumas entrevistas.

Como está formado a estrutura do meu código

A estrutura do código é bem organizada e dividida em duas principais classes: **DailyLog** e **DailyLogApp**. Cada classe tem um papel específico na funcionalidade do aplicativo. Abaixo, detalho a estrutura do código:

**Importações**

1. Bibliotecas padrão do Python: os e json
2. Bibliotecas do Tkinter: tk, messagebox, e simpledialog

**Classe DailyLog**

1. **\_\_init\_\_(self, date)**: Inicializa a classe com uma data e carrega as tarefas desta data.
2. **add\_task(self, description)**: Adiciona uma tarefa à lista e salva no arquivo JSON.
3. **list\_tasks(self)**: Retorna a lista de tarefas atuais.
4. **delete\_task(self, task\_index)**: Remove a tarefa no índice fornecido e salva as mudanças.
5. **save\_tasks(self)**: Salva as tarefas no arquivo JSON.
6. **load\_tasks(self)**: Carrega as tarefas do arquivo JSON para a data específica.

**Classe DailyLogApp**

1. **\_\_init\_\_(self, root)**: Inicializa a interface gráfica com vários componentes (rótulos, entradas, botões e lista de tarefas).
2. **set\_date(self)**: Define a data para o log diário e carrega as tarefas.
3. **add\_task(self)**: Adiciona uma nova tarefa e atualiza a lista de tarefas.
4. **delete\_task(self)**: Remove a tarefa selecionada e atualiza a lista de tarefas.
5. **update\_tasks\_list(self)**: Atualiza a exibição da lista de tarefas na interface gráfica.

**Função Principal**

1. **if \_\_name\_\_ == "\_\_main\_\_":**
   1. Cria a instância da janela principal do Tkinter.
   2. Inicializa a aplicação **DailyLogApp**.
   3. Executa o loop principal do Tkinter (root.mainloop()).

A interface gráfica

Este código é um aplicativo de bloco diário construído usando Python e a biblioteca Tkinter para criar uma interface gráfica de usuário (GUI). Ele permite que os usuários registrem e gerenciem tarefas diárias, salvas em um arquivo JSON. Abaixo, uma explicação detalhada das partes do código:

### Componentes da Interface Gráfica

1. **Label e Entry para Data**: Permitem que o usuário insira e confirme a data para o log diário.
2. **Listbox para Tarefas**: Exibe as tarefas atuais para a data definida.
3. **Entry para Nova Tarefa**: Campo onde o usuário pode inserir a descrição de uma nova tarefa.
4. **Botões**: Incluem botões para confirmar a data, adicionar tarefas, eliminar tarefas e sair do aplicativo.

Funcionamento deste meu aplicativo

1. **Início**: O aplicativo inicia e mostra a janela principal onde o usuário pode inserir uma data.
2. **Definir Data**: O usuário insere uma data e clica no botão "Confirmar Data". Isso cria uma instância de **DailyLog** para essa data e carrega as tarefas associadas.
3. **Adicionar Tarefa**: O usuário insere a descrição de uma nova tarefa e clica no botão "Adicionar Tarefa". A tarefa é adicionada à lista e salva no arquivo JSON.
4. **Eliminar Tarefa**: O usuário seleciona uma tarefa da lista e clica no botão "Eliminar Tarefa". A tarefa é removida da lista e o arquivo JSON é atualizado.
5. **Sair**: O usuário pode fechar o aplicativo clicando no botão "Sair".

Conclusão

Este trabalho foi muito importante para o meu  conhecimento, a compreensão e o aprofundamento deste tema permitiu-me conhecer melhor, compreender melhor sobre python  além de ter-me permitido desenvolver o meu aprendizado que tive durante as aulas , também aperfeiçoar competências de investigação,selecção, organização e comunicação da informação.

Este trabalho foi de grande proveito para mim , e tenho a certeza que aprende muito além da perspetiva,tudo isso graças ao grande professor que tenho.Ele ajudou-me muito a entender bem sobre o python e não só sobre outras coisas que ele já tinha falado na aula. Hoje o conhecimento não é como antes.

Bibliografia

[Bloco de notas desenvolver em Python](https://youtu.be/Fj5yOPOKIJI?si=HnYMD67UX7qnrkQJ)