**Relatório do Trabalho Final de Computação II**

**Nomes:** Douglas Salustrino de Oliveira, Pedro Kauã dos Santos do Prado e Rafael Martins Ferreira

**DREs:** 122116292, 122169067 e 122128744

**Tema:** Sistema de Gerenciamento de Tarefas

**- Como rodar o código?**

Pode-se usar qualquer programa de desenvolvimento de códigos em Python, porém recomendamos o VSCode, ambiente em que o código foi criado, desenvolvido e implementado. Estaremos enviando um arquivo .zip que contém o código principal e o arquivo tasks.csv que tem as tarefas a serem feitas, também aconselhamos deixar a pasta dos arquivos na pasta raíz de seu programa.

**- Introdução**

Escolhemos o tema de criar um sistema que envolve gerenciamento de tarefas que propõe o uso dos principais conteúdos de programação orientada a objetos além de manipulação de arquivos e exceções. Nosso código tem duas classes como a 'Task' e 'TaskManager', além de usar a biblioteca 'tkinter' como interface gráfica, 'uuid' trazendo a função de criar uma ID aleatória sendo muito útil para o cadastramento e 'datetime' para alocação de data e hora do gerenciamento de tarefas usando o calendário real para evitar o registro de datas inválidas, como por exemplo, 31/02. A TaskManager visa em gerenciar uma lista de tarefas representadas pela classe Task.

**-** **A classe Task**

A primeira classe apresentada no trabalho tem os princípios básicos para exibir uma tarefa, utilizando os seguintes atributos dados no método construtor:

**task\_id:** armazena a ID da tarefa, obtida aleatoriamente pela 'uuid';

**task\_name:** armazena o nome da tarefa, digitada pelo usuário;

**deadline:** armazena o prazo da tarefa, digitada pelo usuário;

**status:** marca o estado de uma tarefa, inicialmente definida como pendente;

**data\_criacao:** armazena a data criada, obtida realmente usando a biblioteca datetime.

\*Ela ainda tem dois métodos nomeados como detalhes, que apresenta ao usuário na tela principal da interface os detalhes conforme foram armazenados nos atributos acima, um exemplo disso é nome e status (detalhes2), e ao dar dois cliques numa tarefa qualquer, abre uma segunda tela demonstrando os detalhes completos (detalhes) da mesma, como ID, nome, prazo, status e o dia que ela foi criada.

**- A classe TaskManager**

A maior e principal classe trabalhada neste código apresenta diversas funções que montam o corpo inteiro desse projeto. Como é bastante grande e com muitas coisas para explicar, vamos esclarecer dividindo em alguns tópicos:

• Para começar, criamos o 'init' como método construtor para criar a interface gráfica usando atributos para defini-la e também vimos a primeira exceção, caso o arquivo não leia.

• Em seguida, a função 'data\_valida' verifica se a data digitada para a finalização da tarefa é válida ou não, contando com processos se o ano é bissexto, se o mês tem 30, 31 dias e etc.

• Logo após isso, vemos a parte extremamente fundamental utilizando o tkinter, começando com a função 'criar\_interface' tendo oss seguintes métodos:

**reproduzir\_gif:** Reproduz um GIF especial de natal pra esse trabalho;

**sair:** Adiciona os botões 'sim', 'não' em formato de radiobutton e 'confirmar' para deixar o programa;

**confirmar\_saida:** Vindo da função sair, ao clicar no botão sair, aparecem dois novos botões na tela caso queira deixar o programa, com 'não' valendo 0, assim não fechando e 'sim' valendo 1, fechando a interface;

**ajuda:** Exibe uma mensagem em uma nova tela descrevendo como utilizar o sistema de gerenciamento de tarefas caso uma pessoa nova use-o;

**deletar\_tarefa:** Após selecionar qualquer tarefa e clicando no botão de deletar, abre um novo pop-up desejando se quer excluir aquela tarefa;

**adicionar\_tarefa:** Cria uma nova tarefa totalmente com input do usuário pedindo um nome e uma data válida futura para ser o seu prazo, caso a data for menor que o dia atual ou for digitado uma string ou número qualquer, a tarefa não é criada;

**marcar\_concluido:** Muda o status da tarefa de pendente pra concluído;

**editar\_nome/editar\_data:** Altera nome e/ou data de uma tarefa, digitadas pelo próprio usuário;

**abrir\_janela\_edicao:** Abre uma janela pro usuário selecionar se quer trocar o nome ou data de uma tarefa, pra poder chamar a função 'editar\_nome' ou 'editar\_data';

**editar\_tarefa:** Ao selecionar uma tarefa já existente, clicando no botão de editar tarefa, chama as funções 'abrir\_janela\_edicao' e em sequência 'editar\_nome' e/ou 'editar\_tarefa';

\***exibir\_tarefa:** Ao clicar duas vezes numa tarefa, exibe um pop-up com todos os seus detalhes, já explicado lá em cima na classe Task;

**exibir\_tarefas:** Percorre lista das tarefas e armazena seus detalhes de cada tarefa numa lista;

**atualizar\_tkinter:** Atualiza a interface do tkinter caso crie ou edite uma tarefa;

**atualizar\_arquivo:** Atualiza o arquivo 'tasks.csv' caso haja alguma alteração ou criação de tarefa.