****

**MINISTÉRIO DA EDUCAÇÃO**

**UNIVERSIDADE FEDERAL DO PIAUÍ**

**CENTRO DE CIÊNCIAS DA NATUREZA- CCE**

Campus Universitário Ministro Petrônio Portella, Bloco 06 - Bairro Ininga,

CEP 64049-550 – Teresina-Piauí-Brasil – Fone: (86) 3215-5565 – Fax: (86) 3215-5560

**Curso:** Ciência da Computação

**Disciplina:** Estrutura de Dados

**Professor:** Raimundo Santos

**Aluno:** Otávio C. França

**Relatório: Sistema de Fila com Prioridades**

**Introdução**

Este relatório apresenta um sistema de fila com prioridades desenvolvido em Python usando a biblioteca Tkinter para a interface gráfica. O sistema organiza o atendimento das pessoas na fila, com a opção de prioridade, atendimento das mesmas, listagem dos indivíduos na fila e apresentação de estatísticas relacionadas ao atendimento prioritário e normal.

**Objetivos**

1. Implementar um Tipo Abstrato de Dados (ADT) de uma Fila com Prioridades utilizando uma lista encadeada para o gerenciamento do atendimento de pessoas em um sistema.

2. Desenvolver uma aplicação cliente com um menu contendo as seguintes opções:

i) Registro da chegada de uma pessoa para atendimento normal;

ii) Registro da chegada de uma pessoa para atendimento prioritário;

iii) Atendimento de uma pessoa;

iv) Listagem de todas as pessoas nas filas de atendimento;

v) Geração de estatísticas sobre os atendimentos realizados, incluindo a percentagem de atendimentos prioritários, a percentagem de atendimentos normais e o tamanho das filas Normal e Prioritária.

**Reviw Code**

**1. Classe AplicacaoCliente**

Esta classe é responsável por criar a interface gráfica do sistema de fila com prioridades usando a biblioteca Tkinter.

Métodos:

* \_\_init\_\_(self, master): Método de inicialização da classe que cria a janela principal e configura os widgets.
* criar\_widgets(self): Método para criar e posicionar os widgets na interface.
* chegada\_normal(self): Método chamado quando o botão "Atendimento Normal" é clicado. Adiciona uma pessoa à fila normal.
* chegada\_prioritaria(self): Método chamado quando o botão "Atendimento Prioritário" é clicado. Adiciona uma pessoa à fila prioritária.
* atendimento(self): Método chamado quando o botão "Atendimento" é clicado. Realiza o atendimento da próxima pessoa na fila.
* listar(self): Método chamado quando o botão "Listar" é clicado. Lista todas as pessoas na fila.
* estatisticas(self): Método chamado quando o botão "Estatísticas" é clicado. Mostra estatísticas sobre a fila.
* sair(self): Método chamado quando o botão "Sair" é clicado. Fecha a aplicação se não houver mais pessoas na fila.
* limpar\_campos\_entrada(self): Método para limpar os campos de entrada após adicionar uma pessoa à fila.

**2. Classe Pessoa**

Representa uma pessoa na fila com suas características como nome, idade e se pertence a um grupo prioritário.

Métodos:

* \_\_init\_\_(self, nome, idade, grupo\_prioritario): Método de inicialização da classe que define as propriedades da pessoa.
* definir\_prioridade(self): Método que define a prioridade da pessoa com base em sua idade.

**3. Classe FilaPrioridades**

Implementa a estrutura de dados de uma fila com prioridades.

Métodos:

* \_\_init\_\_(self): Método de inicialização da classe que define o início e o fim da fila como nulos.
* enfileirar(self, pessoa): Adiciona uma pessoa à fila, considerando sua prioridade.
* desenfileirar(self): Remove e retorna a próxima pessoa da fila.
* listar(self): Lista todas as pessoas na fila.
* tamanho(self): Retorna o tamanho atual da fila.
* estatisticas(self): Calcula e retorna estatísticas sobre as pessoas na fila.

**4. Classe No**

Implementa os nós que compõem a fila encadeada.

Métodos:

* \_\_init\_\_(self, pessoa): Método de inicialização da classe que define a pessoa associada ao nó e o próximo nó na fila.

**Resultados**

A interface gráfica foi projetada para proporcionar uma experiência amigável e intuitiva para o usuário. A lógica da fila de prioridades foi implementada de forma eficiente, garantindo que as pessoas com prioridade sejam atendidas antes das pessoas normais. As estatísticas fornecidas pelo sistema à administração, fazem com que eles possam entender melhor o fluxo de clientes. O presente Projeto encontrasse disponível em: https://github.com/otaviofranca/data-structs.