

PROJETO FINAL DE ALGORITMOS E ESTRUTURA DE DADOS

- **Projeto:** aplicativo de gerenciamento de pedidos de Delivery
- **Justificativa:** utiliza de forma eficiente os conceitos apresentados durante a disciplina (Lista encadeada do tipo “fila”, TAD (tipo abstrato de dados) e alocação dinâmica)
- **Metodologia:** utilização da linguagem C no ambiente de desenvolvimento VS Code

TAD – Sistema de gerenciamento de pedidos de um aplicativo de “Delivery”

- Separado em três files:
- `main.c` – Código principal que contém a função “main” e as chamadas das funções correspondentes
- `delivery.c` – Implementação das funções
- `Delivery.h` – Declaração das estruturas e protótipos das funções

Definição das estruturas

- **Produto:** Armazena informações sobre produtos (ID, nome e preço).
- **Cliente:** Armazena informações sobre clientes (ID, nome e endereço).
- **Pedido:** Representa um pedido, contendo um cliente, um produto e um ponteiro para o próximo pedido na fila.
- **Fila:** Estrutura para gerenciar a fila de pedidos, com ponteiros para o início (frente) e o fim (trás) da fila, e um contador de pedidos.

```
typedef struct Produto {  
    int id;  
    char nome[50];  
    float preco;  
} Produto;
```

```
typedef struct Cliente {  
    int id;  
    char nome[50];  
    char endereco[100];  
} Cliente;
```

```
typedef struct Pedido {  
    int id;  
    Cliente cliente;  
    Produto produto;  
    struct Pedido* proximo;  
} Pedido;
```

```
typedef struct {  
    Pedido* frente;  
    Pedido* tras;  
    int quantidade;  
} Fila;
```

Funções da fila:

```
// Função para criar e inicializar a fila
Fila* criaFila();

// Verifica se a fila está vazia
int filaVazia(Fila* fila);

// Enfileira um pedido
void enfileira(Fila* fila, Pedido* pedido);

// Desenfileira um pedido
Pedido* desenfileira(Fila* fila);

// Função para criar um novo pedido
Pedido* criaPedido(int id, Cliente cliente, Produto produto);
```

Manipulação de arquivos:

```
// Salva os produtos em um arquivo
void salvaProdutos(Produto* produtos, int numProdutos, const char* nomeArquivo);

// Salva os clientes em um arquivo
void salvaClientes(Cliente* clientes, int numClientes, const char* nomeArquivo);

// Carrega os produtos de um arquivo
void carregaProdutos(Produto* produtos, int* numProdutos, const char* nomeArquivo);

// Carrega os clientes de um arquivo
void carregaClientes(Cliente* clientes, int* numClientes, const char* nomeArquivo);
```

Função main:

- Cria uma fila e popula arquivos com produtos e clientes.
- Carrega os produtos e clientes dos arquivos.
- Cria e enfileira dois pedidos.
- Exibe o conteúdo da fila de pedidos.
- Desenfileira e exibe um pedido, liberando a memória alocada.
- Exibe novamente a fila após o desenfileiramento e libera a memória restante dos pedidos e da fila.

Notas importantes:

- O código inclui verificações de erros para operações de alocação de memória e manipulação de arquivos.
- O programa faz uso de estruturas dinâmicas (alocação com malloc) e manipulação básica de arquivos binários para persistência de dados.
- Essencialmente, o código demonstra a criação e gerenciamento de uma fila de pedidos, com a capacidade de salvar e carregar dados de arquivos.