Documentação do Gerenciador de Pedidos de Restaurantes

Introdução

- Descrição geral do sistema:

1 - Cadastro de usuários: Os usuários podem se cadastrar no aplicativo fornecendo informações básicas, como nome, endereço, login e senha.

2 - Login de usuários: Os usuários cadastrados podem fazer login no aplicativo usando seu login e sua senha.

3 - Acesso para administrador: login: "adm", senha: "adm"

4 - Menu de administrador: Os administradores têm acesso a um menu especial com funcionalidades adicionais, como adicionar e remover restaurantes da lista de restaurantes cadastrados.

5 - Interagir como cliente: Os usuários podem interagir com o aplicativo como clientes, visualizando a lista de restaurantes cadastrados, escolhendo restaurantes, escolhendo pratos e fazendo pedidos.

6 - Pagamentos: Os usuários podem fazer pagamentos por meio do aplicativo para confirmar seus pedidos.

7 - Avaliação de restaurantes: Após receberem seus pedidos, os usuários têm a opção de avaliar os restaurantes.

8 - Estimativa de entrega: O aplicativo fornece uma estimativa de tempo de entrega para os pedidos feitos pelos usuários, como uma forma de simular o tempo de entrega.

9 - Interface de linha de comando (CLI): O aplicativo é executado em uma interface de linha de comando, com opções e informações exibidas em formato de texto.

- Objetivos do sistema

Utilizar, de maneira adequada, estruturas de dados aprendidas no curso de algoritmos e estruturas de dados 1 (listas, filas e pilhas).

Instalação e Configuração

1 - Compilador C: Para executar o aplicativo, você precisará de um compilador C instalado em sua máquina.

2 - Bibliotecas C padrão: O aplicativo utiliza bibliotecas padrão da linguagem C, portanto, não há dependências adicionais.

3 - Como executar o aplicativo:

Clone este repositório em sua máquina local.

Abra o terminal e navegue até o diretório raiz do projeto.

Compile o código-fonte do aplicativo usando o compilador C. Por exemplo, se estiver usando o GCC, execute o seguinte comando:

make run

4 - Siga as instruções exibidas na interface de linha de comando para interagir com o aplicativo.

Visão Geral do Código Fonte

- Estrutura de diretórios

- Descrição dos arquivos principais

#Módulo: 'structs.h'

Este módulo teve que ser desenvolvido ao longo do projeto para que todas as structs fosse corretamente contempladas pelos outros módulos, respeitando as diretrizes da linguagem C.

O arquivo `structs.h` contém as definições das estruturas de dados utilizadas no sistema de gerenciamento de pedidos de restaurantes. Aqui está uma descrição das principais estruturas presentes no arquivo:

- `Fila`: É uma estrutura que representa uma fila de pedidos de um restaurante. A implementação detalhada da fila não está presente neste arquivo, mas é referenciada como `Fila *fila_pedidos` na estrutura `Restaurante`.

- `Lista_restaurante`: É uma estrutura que representa uma lista de restaurantes. Ela é utilizada para armazenar os restaurantes cadastrados no sistema.

- `Prato`: É uma estrutura que representa um prato oferecido por um restaurante. Ela possui os seguintes campos:

  - `nome`: Uma string que armazena o nome do prato.

  - `bebida`: Uma string que armazena o nome da bebida que acompanha o prato.

  - `preco`: Um valor de ponto flutuante que indica o preço do prato.

  - `qtd_pratos`: Um inteiro que representa a quantidade disponível desse prato.

- `Restaurante`: É uma estrutura que armazena os dados de um restaurante. Ela possui os seguintes campos:

  - `fila_pedidos`: Uma referência a uma fila de pedidos relacionada ao restaurante.

  - `nome`: Uma string que armazena o nome do restaurante.

  - `categoria`: Um caractere que representa a categoria do restaurante.

  - `tipo_culinaria`: Um inteiro que indica o tipo de culinária do restaurante.

  - `qtd_pratos`: Um inteiro que representa a quantidade total de pratos disponíveis no restaurante.

  - `prato`: Um ponteiro para um array de estruturas `Prato`, contendo informações sobre os pratos oferecidos pelo restaurante.

- `Pedido`: É uma estrutura que representa um pedido realizado por um cliente. Ela possui os seguintes campos:

  - `prato`: Uma estrutura `Prato` que representa o prato escolhido no pedido.

  - `quantidade`: Um inteiro que indica a quantidade desejada do prato.

  - `valorTotal`: Um valor de ponto flutuante que representa o valor total do pedido.

Essas estruturas de dados são utilizadas para armazenar informações sobre os restaurantes, pratos e pedidos no sistema de gerenciamento de pedidos de restaurantes.

Módulo: `restaurante.h`

O módulo `restaurante.h` contém as declarações das funções relacionadas à manipulação de restaurantes no sistema de gerenciamento de pedidos de restaurantes. Ele inclui a definição das estruturas de dados utilizadas, como `Restaurante` e `Prato`, e declarações das funções para criar, manipular e buscar restaurantes na lista de restaurantes.

Aqui está uma descrição das principais funcionalidades e estruturas presentes no módulo `restaurante.h`:

- `Lista_restaurante`: É uma estrutura que representa uma lista de restaurantes. Ela é utilizada para armazenar os restaurantes cadastrados no sistema.

- `criar_restaurante()`: Função responsável por criar e retornar uma instância de `Lista_restaurante` vazia.

- `limpar_lista()`: Função que recebe uma instância de `Lista_restaurante` e realiza a remoção de todos os elementos da lista, liberando a memória alocada.

- `qtd_restaurantes()`: Função que retorna a quantidade de restaurantes presentes na lista.

- `insere_restaurante_no_final()`: Função que insere um restaurante no final da lista de restaurantes.

- `insere_restaurantes_cadastrados()`: Função utilizada para inserir uma "simulação de base de dados" dos restaurantes previamente cadastrados no sistema. Essa função é geralmente utilizada durante a inicialização do programa.

- `cadastrar_restaurante()`: Função responsável por cadastrar um novo restaurante na lista de restaurantes.

- `buscar_restaurante()`: Função que busca um restaurante na lista com base em seu nome.

- `buscar_prato_principal()`: Função que busca um prato principal em um restaurante específico, dado o nome do prato.

- `remove_restaurante_inicio()`: Função que remove o primeiro restaurante da lista.

- `remove_restaurante_final()`: Função que remove o último restaurante da lista.

- `remove_restaurante_pos()`: Função que remove um restaurante em uma posição específica da lista.

- `menu_restaurantes_adm()`: Função que exibe o menu de opções para o administrador do sistema realizar operações relacionadas aos restaurantes.

- `menu_restaurante_usuario()`: Função que exibe o menu de opções para o usuário realizar operações relacionadas aos restaurantes.

- `printLetterByLetter()`: Função auxiliar que imprime uma mensagem caractere por caractere com um intervalo de tempo determinado.

O módulo `restaurante.h` fornece as funcionalidades necessárias para criar, manipular, buscar e remover restaurantes na lista de restaurantes do sistema de gerenciamento de pedidos de restaurantes.

Módulo: `usuario.h`

O arquivo `usuario.h` contém as declarações das funções e a definição da estrutura de dados `Usuario` relacionadas aos usuários do sistema de gerenciamento de pedidos de restaurantes. Aqui está uma descrição das principais funcionalidades e estruturas presentes no arquivo:

- `Usuario`: É uma estrutura que armazena os dados de um usuário do sistema. Ela possui os seguintes campos:
  - `nome`: Uma string que armazena o nome do usuário.
  - `cpf`: Uma string que armazena o CPF do usuário.
  - `num_telefone`: Uma string que armazena o número de telefone do usuário.
  - `login`: Uma string que armazena o login do usuário.
  - `senha`: Uma string que armazena a senha do usuário.

- `Lista`: É uma estrutura que representa uma lista de usuários. Ela é utilizada para armazenar os usuários cadastrados no sistema.

- `criar_usuario()`: Função responsável por criar e retornar uma instância de `Lista` vazia.

- `lista_existe()`: Função que verifica se a lista de usuários existe.

- `lista_vazia()`: Função que verifica se a lista de usuários está vazia.

- `tam_lista()`: Função que retorna o tamanho atual da lista de usuários.

- `inserir_no_inicio()`: Função que insere um usuário no início da lista de usuários.

- `inserir_no_fim()`: Função que insere um usuário no final da lista de usuários.

- `remover_usuario()`: Função que remove um usuário da lista de usuários.

- `cadastrar_usuario()`: Função responsável por cadastrar um novo usuário na lista de usuários.

- `verifica_login()`: Função que verifica se o login e a senha fornecidos correspondem a um usuário válido na lista.

- `verifica_administrador()`: Função que verifica se o login e a senha fornecidos correspondem ao administrador do sistema. Retorna 1 se existir um administrador com essas credenciais e 0 caso contrário.

- `insere_adm()`: Função que insere um administrador na lista de usuários. Retorna 0 se o administrador foi cadastrado com sucesso.

- `sleep()`: Função auxiliar que pausa a execução por um determinado período de tempo.

Essas funções e estruturas são utilizadas para gerenciar a lista de usuários e realizar operações como cadastro, autenticação e remoção de usuários no sistema de gerenciamento de pedidos de restaurantes.

Módulo: `security.h`

O arquivo `security.h` contém as declarações das funções relacionadas à segurança da senha no sistema de gerenciamento de pedidos de restaurantes. Essas funções são responsáveis por verificar diferentes critérios de segurança da senha. Aqui está uma descrição das funções presentes no arquivo:

- `temNumerosConsecutivos()`: Função que verifica se a senha contém números consecutivos.

- `temCaracteresEspeciais()`: Função que verifica se a senha contém caracteres especiais. Pelo menos um caractere especial é necessário para satisfazer esse critério de segurança.

- `temLetraMaiuscula()`: Função que verifica se a senha contém pelo menos uma letra maiúscula.

- `temLetraMinuscula()`: Função que verifica se a senha contém pelo menos uma letra minúscula.

- `temComprimentoMinimo()`: Função que verifica se a senha possui o comprimento mínimo especificado. Essa função recebe como parâmetro o comprimento mínimo desejado para a senha.

- `temNumero()`: Função que verifica se a senha contém pelo menos um número.

Essas funções são utilizadas para garantir a segurança das senhas utilizadas pelos usuários no sistema de gerenciamento de pedidos de restaurantes. Cada função verifica um critério específico, como presença de números consecutivos, caracteres especiais, letras maiúsculas e minúsculas, comprimento mínimo e números na senha.

Módulo: `pedido.h`

O arquivo `pedido.h` contém as declarações das funções e a definição da estrutura de dados relacionadas aos pedidos no sistema de gerenciamento de pedidos de restaurantes. Aqui está uma descrição das principais funcionalidades e estruturas presentes no arquivo:

- `Fila`: É uma estrutura que representa uma fila de pedidos de um restaurante. A implementação detalhada da fila não está presente neste arquivo, mas é referenciada como `Fila *fila_pedidos` na estrutura `Restaurante` do arquivo `restaurante.h`.

- `criar_fila()`: Função responsável por criar e retornar uma instância de `Fila` vazia.

- `fila_vazia()`: Função que verifica se a fila de pedidos está vazia.

- `fila_cheia()`: Função que verifica se a fila de pedidos está cheia.

- `qtd_pedidos()`: Função que retorna a quantidade de pedidos presentes na fila.

- `inserir_pedido()`: Função que insere um pedido na fila de pedidos de um restaurante.

- `mostrar_pedido()`: Função que exibe as informações de um pedido na fila, considerando o seu status (em andamento, concluído, etc).

- `remover_pedido()`: Função que remove um pedido da fila de pedidos.

- `pagar_com_cartao()`: Função que lida com o pagamento do pedido utilizando cartão de crédito.

- `pagar_com_dinheiro()`: Função que lida com o pagamento do pedido utilizando dinheiro. Retorna 1 se o pagamento foi realizado com sucesso.

- `pagar_com_pix()`: Função que lida com o pagamento do pedido utilizando a forma de pagamento Pix.

- `mostrar_pagamento()`: Função que exibe o menu de opções de pagamento e retorna a opção escolhida pelo usuário.

- `mostrar_estimativa_entrega()`: Função que exibe a estimativa de entrega do pedido.

- `sleep()`: Função auxiliar que pausa a execução por um determinado período de tempo.

- `mostrar_avaliacao()`: Função que exibe o menu de avaliação do pedido e permite que o usuário faça uma avaliação.

Essas funções e estruturas são utilizadas para realizar operações relacionadas aos pedidos, como inserção, remoção, pagamento, exibição de informações e avaliação dos pedidos no sistema de gerenciamento de pedidos de restaurantes.

Arquivo: `app.c`

- App.c é apenas uma implementação de todos os arquivos acima, no qual é possível mostrar todas as estruturas de dados escolhidas trabalhando em conjunto.

Referências

- Conhecimentos adquiridos em sala de aula com o professor José Gustavo, além de diversos conteúdos disponíveis na internet sobre o tema estrutura de dados em linguagem C