Uma imagem com texto, Tipo de letra, Gráficos, logótipo

Descrição gerada automaticamente

Licenciatura Engenharia Informática

Ano letivo 2022/2023

31-05-2023

Projeto

Princípios de Programação Procedimental

João Pedro Gonçalves Ventura Barata Nº2022245160

Alexandre José Martins Rodrigues Nº2022249408

**Índice**

1. Introdução
2. Ideias fundamentais
3. Estrutura geral do programa
   1. Estruturas de dados
   2. Funções
   3. Ficheiros

**Introdução**

Com este projeto desenvolvemos um programa realizado em C no Visual Studio Code para auxiliar um funcionário de uma oficina na gestão de reservas e pré-reservas. A aplicação mantém informação sobre os clientes e as horas disponíveis para reservas, e também as reservas que estão em espera.

**Ideias fundamentais**

**A aplicação permite ao utilizador, de forma interativa, as seguintes operações:**

1. Reservar lavagem ou manutenção - um cliente faz uma reserva para lavar o carro ou para fazer a sua manutenção.
2. Pré-reserva de lavagem ou manutenção – um cliente pretende reservar uma lavagem ou uma manutenção, mas não tem hora disponível e inscreve-se numa lista de espera para ter prioridade caso uma reserva seja cancelada.
3. Cancelar uma reserva - eliminar uma reserva. Se houver uma pré-reserva compatível com o horário e dia desocupado, esta será imediatamente enquadrada.
4. Cancelar uma pré-reserva - eliminar um pedido de pré-reserva.
5. Listar as reservas e as pré-reservas de lavagens e de manutenções ordenadas por data (as mais antigas primeiro).
6. Listar as reservas e as pré-reservas de um cliente ordenadas por data (as mais recentes primeiro).
7. Realizar uma lavagem ou manutenção. Quando uma operação termina é eliminada da lista de reservas.

**Estrutura geral do programa**

**Estruturas de dados**

* **Localizado no ficheiro trabalho.h:**

Usámos 2 estruturas de dados, sendo elas fila2 e fila\_h onde a fila2 serviu para definir a data do pedido e para recuar e avançar e a fila\_h para determinar o fim e o início da fila.

**Funções**

* **Localizado no ficheiro trabalho.c:**

No que toca a funções utilizámos como base o ficheiro de filas disponibilizados das aulas no UCStudent, e utilizámos e/ou alterámos estas funções: **initialize**, **is\_empty**, **insert**, **remove\_node**, **delete**, **save\_queue** e **print\_queue**.

Na função **initialize** utilizámos apenas para iniciar a fila e na função **is\_empty** para verificar se está vazia, nestas não fizemos grandes alterações.

Na função **insert** tivemos de alterar para conseguirmos introduzir os dados das reservas/pré-reservas.

A função **remove\_node** remove um nó específico de uma fila encadeada, ela percorre a fila até encontrar o nó com o valor desejado, se o nó for encontrado, ele é removido da fila, atualizando os ponteiros de início e fim, caso contrário, uma mensagem informando que a reserva não foi encontrada é exibida, a função lida com diferentes cenários, como remover o primeiro nó, o último nó e um nó no meio da fila.

A função **delete** é responsável por excluir todos os nós de uma fila encadeada, libertando a memória alocada para eles.

A função **save\_queue** salva o conteúdo de uma fila encadeada num arquivo de texto. Ela percorre a fila e escreve as informações de cada nó no arquivo, formatando-as de acordo com um padrão. O arquivo é aberto para escrita, as informações são escritas linha por linha e o arquivo é fechado no final. Isso permite que os dados da fila sejam armazenados e posteriormente recuperados a partir do arquivo.

A função **print\_queue** exibe o conteúdo de uma fila encadeada. Ela percorre a fila e imprime as informações de cada nó, como o nome e a data associados a ele. Os dados são formatados e exibidos de forma legível. Essa função é útil para visualizar o conteúdo da fila durante a execução do programa.

* **Localizado no ficheiro tudo.c:**

O código lê um arquivo chamado "ficheiro.txt" e insere informações em uma fila, e verifica se o arquivo foi aberto corretamente e exibe uma mensagem de erro, caso contrário cada linha do arquivo é dividida em campos usando o delimitador "$", os campos são armazenados em um array bidimensional chamado "fields", se existirem 6 campos, eles são extraídos e inseridos na fila usando a função "insert()", se o número de campos for diferente de 6, é exibida uma mensagem de erro, o arquivo é fechado no final do processo.

Começámos por fazer a interface do menu onde através de printf’s apresentámos as opções pedidas onde, de seguida, utilizámos o scanf juntamente de outras condições como o while e o if. Primeiramente utilizámos um while para definir o comando de saída, de seguida utilizámos um if onde a nossa variável tinha de estar entre 0 e 8, caso contrário aparecia a mensagem “Opção inválida”, depois através de if’s e else’s configurámos o menu consoante os requisitos, utilizando as funções referidas anteriormente presentes no ficheiro “trabalho.c”, no final demos return.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

**Figura 1 –** Interface do menu

**Ficheiros**

No ficheiro ZIP do projeto encontra-se:

“Manual do programador.docx” - manual de programador realizado neste Word;

“.vscode” - para auxiliar na utilização do programa e a sua execução;

“trabalho.h” – ether files;

“trabalho.c” – funções;

“tudo.c” – executar as funções e menu;

“ficheiro.txt” -guardar as listas das marcações das reservas e pré-reservas;

“trabalho.exe” – executável/aplicação dos ficheiros de código combinados.