



Instituto Superior de
Engenharia de Coimbra

Relatório - Programação

Aluquer de Guitarras

Trabalho Prático - 2017/2018

Programação

- **Ano Lectivo:** 2017/2018
- **Nome:** João Pedro Alves Branco
- **Nº aluno:** 21250086

Índice

Estruturas de Dados	2
Manipulação de Ficheiros	5
Estrutura do Programa	7
Manual de utilização:	10

Estruturas de Dados

1. Cliente

```
typedef struct cliente {
    char nome[100];
    int nif, cont_estado, n_alugueres;
    struct cliente *prox; //ponteiro para o proximo nó da lista
    Aluguer *lista; //ponteiro para a lista dos alugueres
    //cont_estado -> contador para nº de guitarras danificadas
    //n_alugueres -> contador para o nº de alugueres
} Cliente;
```

2. Aluguer

```
typedef struct aluguer {
    int id, estado;
    Data inicio, fim;
    struct aluguer *prox; //ponteiro para o próximo nó da lista de alugueres
} Aluguer;
```

3. Guitarra

```
typedef struct {
    int id, valor, estado;
    char nome[100];
    float preco;
} Guitarra;
```

4. Data

```
typedef struct {
    int dia, mes, ano;
} Data;
```

5. Cliente Banido

```
typedef struct banido {
    char motivo[100];
    int nif;
} Banido;
```

O trabalho prático contém 5 estruturas de dados que estão guardadas no *header* file *structs.h*, sendo 2 delas auxiliares para o trabalho, que é a estrutura da *Data* e a estrutura *Banido*. A estrutura **clientes** guarda os dados relativos aos clientes da loja de aluguer de guitarras, esta estrutura é usada como **lista ligada**, todos os clientes serão inseridos na lista ligada e depois passados para o ficheiro e vice-versa. A estrutura **alugueres** guarda os dados relativos aos alugueres de cada cliente. A estrutura das **guitarras**, guarda informação das guitarras que podem ser alugadas pelos clientes e vai ser usada como **Array Dinâmico**.

Por último temos a estrutura da **data** que vai ser útil para a gestão das datas que é necessário fazer por parte do programa, onde vai ser usada nas datas iniciais e finais dos alugueres.

Na estrutura dos **clientes** tenho as informações de identificação relativas ao cliente, sendo elas o nome, o NIF. Depois de declaradas estas variáveis, optei pela criação de 2 variáveis inteiras auxiliares que estão relacionadas com o aluguer, que são o *conta_estado* (contador do nº de guitarras que foram entregues danificadas) e o *n_alugueres* (contador do nº de alugueres que o cliente tem).

Após a criação destas variáveis, declarei as variáveis que vão estar relacionadas com o propósito do trabalho e da estrutura dos clientes e com os alugueres, ou seja, declarei uma variável do tipo ponteiro que está a apontar para uma estrutura do tipo cliente (*struct cliente *prox*) e uma variável do tipo ponteiro a apontar para uma estrutura do tipo aluguer (*aluguer *lista*).

Com isto tenho a informação necessário para poder guardar a informação do cliente, tanto pessoal como do aluguer, podendo então guardar dinamicamente os dados do cliente.

Na estrutura dos **alugueres**, declarei 2 variáveis inteiras sendo elas o ID da guitarra (*ID*) a que o aluguer se referia e o estado desse mesmo aluguer (*estado*) (se estava entregue, a decorrer, ou se tinha sido entregue mas com danos na guitarra).

Usei a estrutura auxiliar da data (*Data início* e *Data fim*) para poder guardar a informação da data inicial e final de um aluguer, que será introduzida pelo utilizador.

E por fim criei uma variável do tipo ponteiro que estará a apontar para uma estrutura do tipo aluguer (*struct aluguer *prox*), sendo assim possível a criação de uma lista de uma lista (lista de lista).

Na estrutura das **guitarras** tenho variáveis declaradas que irão guardar informações como o nome da guitarra, o seu identificador (ID), o estado em que se encontra a guitarra (se alugada, disponível para ser alugada ou danificada). Tenho também o valor da guitarra e o seu preço que é cobrado por dia de aluguer. As estruturas das guitarras servirão para um Array Dinâmico.

A estrutura para os **clientes banidos**, armazenará informações como o NIF do Cliente em causa e o motivo do seu *ban*.

Manipulação de Ficheiros

Foram utilizados 2 tipos de ficheiro, ficheiro binário e ficheiro de texto, o ficheiro de texto serviu para guardar a informação relativa aos clientes e possíveis alugueres dos mesmos como também as guitarras da loja.

O ficheiro binário serviu para guardar toda a informação relativa aos clientes que são banidos por danos materiais (às guitarras) ou por excesso de dias de atraso (mais de 20 dias).

No início do programa tanto o ficheiro de texto como o ficheiro binário são abertos em modo de leitura para a leitura dos dados existentes para as estruturas de dados. Assim que o utilizador adicione um cliente, crie um aluguer ou adicione uma guitarra o ficheiro é actualizado com a informação acabada de introduzir por parte do utilizador.

Os dados do ficheiro de texto são manipulados através das funções *fgets()*, *fscanf()* e *fprintf()*, enquanto que no ficheiro binário são usadas as funções *fread()* e *fwrite()*.

Exemplo de manipulação de ficheiro para carregamento de clientes:

```
FILE *f = fopen("clientes.txt", "rt");
Cliente *aux = NULL;
char buffer[300], c;

if (f == NULL) //verifica se foi possivel abrir o ficheiro
    printf("erro a abrir ficheiro %s", f);

while ((c = getc(f)) != EOF) { //Ler até ao final do ficheiro
    ...
    fgets(buffer, sizeof (buffer), f); //Passar a linha toda do .txt para o buffer
    sscanf(buffer, " %d %d %99[^\n]s", &aux->nif, &aux->n_alugueres, aux->nome); //separa a linha do buffer para as variaveis
    ...
}
fclose(f);
```

Exemplo de leitura dos dados dos clientes banidos:

```
FILE *f = fopen("clientes.dat", "rb");
Banido aux;

if (f == NULL) //verifica se foi possivel abrir o ficheiro
    printf("erro a abrir ficheiro %s", f);

while (fread(&aux, sizeof (aux), 1, f) == 1) { //le os dados do ficheiro binário para a estrutura dos clientes banidos
    ...
}
fclose(f);
```

O ficheiro que guarda os dados dos clientes banidos tem o nome de clientes.dat, é um ficheiro binário que guarda a estrutura dos clientes banidos.

O ficheiro clientes.txt guarda os dados dos clientes e possíveis alugueres, e é um ficheiro de texto simples.

A forma como carrego os dados dos alugueres é a mesma como faço para a dos clientes.

Estrutura do Programa

Nesta parte vou colocar as funções que criei com a designação da utilidade de cada uma delas.

- Funções principais dos clientes e alugueres:

```
//cria um novo nó da lista e insere os dados pedidos ao utilizador nesse nó e actualiza a lista e o ficheiro .txt, recebendo um //ponteiro para o início da lista ligada dos clientes
Cliente *adiciona_cliente(Cliente *c);

//função chamada quando o utilizador quiser remover um cliente, enviando assim o NIF do cliente a remover, actualizando logo a lista //ligada dos clientes
Cliente *remove_cliente_lista(Cliente *c, int nif);

//remove um cliente do ficheiro clientes.txt
void remove_cliente_ficheiro(Cliente *c);

//função que cria um novo aluguer, onde a função recebe o ponteiro para o início lista dos clientes, um ponteiro para o array de //guitarras, o seu tamanho, o NIF do cliente que quer criar o aluguer e o ID da guitarra que o cliente quer alugar
void adiciona_aluguer(Cliente *a, Guitarra *g, int total, int nif, int id);

//função que mostrar todos os aluguer activos, ou seja, que estão a decorrer ainda (ainda não foram entregues)
void alugueres_activos(Cliente *c);

//função que conclui um aluguer, recebendo por parâmetro o ponteiro para o início da lista ligada dos clientes, um ponteiro para o //array de guitarras, o seu tamanho, e o NIF do cliente que quer concluir o seu aluguer
void conclui_aluguer(Cliente *c, Guitarra *g, int total, int nif);
```


- Funções principais relativas às guitarras:

```
//função que adiciona uma nova guitarra ao Array Dinâmico, recebendo por parâmetro o ponteiro para o início do Array e o seu tamanho
Guitarra *adiciona_guitarra(Guitarra *g, int *total);

//função que mostra todas as guitarras, recebendo por parâmetro o ponteiro para o início do Array e o seu tamanho
void mostra_guitarras(Guitarra *g, int tam);

//função que mostra as guitarras alugadas no momento, recebendo por parâmetro o ponteiro para o início da Lista dos Clientes, o //ponteiro
para o início do Array e o seu tamanho
void mostrar_guitarras_alugadas(Cliente *c, Guitarra *g, int tam);

//função que mostra o histórico de uma determinada guitarra, recebendo o ponteiro para o início da Lista ligada dos Clientes e o ID da
//Guitarra que quer investigar
void historico_guitarras(Cliente *c, int id);

//função que verifica se já existe a guitarra com o ID dado (passado por argumento na função), função útil para quando o utilizador
//adicionar uma nova guitarra
int verifica_guitarras(Guitarra *g, int total, int id);

//função que verifica se a guitarra com o ID dado (passado por argumento na função) está danificada ou não, é útil para quando o //utilizador
quiser criar um aluguer novo, onde não poderá alugar uma guitarra danificada
int verifica_guitarra_danificada(Guitarra *g, int total, int id);

//função que retornará o valor do preço diário que a guitarra com o ID dado (passado por parâmetro)
float multa_guitarra(Guitarra *g, int id, int total);

//função que actualiza o estado da guitarra, ou seja, se passa a estar disponível, danificada ou se está alugada no momento
Guitarra *actualiza_estado_guitarra(Guitarra *g, int total, int estado, int id);

//função que escreve a informação que está contida no Array de Guitarras para o ficheiro guitarras.txt
void escreve_ficheiro_guitarras(Guitarra *g, int total);
```

- Funções diversas:

```
//função para mostrar toda a informação dos clientes, e possíveis alugueres, recebendo o ponteiro para o início da lista dos Clientes
void mostrar_info(Cliente *c);

//função que escreve o cliente banido no ficheiro clientes.dat, recebendo a estrutura do cliente que acabou por ser banido
void escreve_clientes_banidos(Banido ban);

//função que mostra a informação presente no ficheiro clientes.dat, ou seja, os clientes banidos
void mostra_clientes_banidos();

//função que escreve a informação dos clientes e seus alugueres (se existirem) no ficheiro clientes.txt
void escreve_ficheiro(Cliente *a);

//função que verifica se já existe o cliente com o NIF passado por argumento na função, útil quando o utilizador quer adicionar novo
//cliente
int verifica_cliente(int nif);

//função que serve para verificar se o NIF introduzido pelo utilizador é válido (se contém 9 dígitos)
int verifica_nif(int nif);

//função que calcula a data final do aluguer (os 7 dias após a data do início do aluguer)
Data verifica_data(Aluguer *aux);

//função que verifica se o cliente está banido, útil quando o utilizador quer adicionar novo cliente (se introduzir um NIF igual ao do
//cliente banido, apresenta uma mensagem de erro que o cliente com o NIF introduzido está banido)
int verifica_cliente_banido(int nif);

//função que verifica o pagamento do cliente
int verifica_atraso(Guitarra *g, int total, Aluguer *aux, Data temp);

//função que verifica se o ficheiro está vazio ou não
int verifica_ficheiro(FILE *f);
```

Manual de utilização:

A interface do programa é relativamente simples, as diversas opções encontram-se organizadas nos respectivos menus com pequenas informações, pelo que será relativamente simples a sua utilização.

Menu principal:

No menu principal encontramos 4 opções, são elas:

- 1 - Clientes
- 2 - Guitarras
- 3 - Aluguer
- 4 - Terminar

A opção número **1** permite-nos aceder ao menu relativo aos **clientes** que posteriormente nos permite manipular e visualizar a informação relativa aos mesmos.

A opção número **2** permite-nos manipular e visualizar a informação relativa às **guitarras**.

A opção **3** permite-nos manipular e visualizar a informação relativo aos **alugueres**

A opção **4** **termina** o programa.

Menu Clientes:

Neste menu encontramos as seguintes opções :

- 1 - Adicionar novo Cliente
- 2 - Remover Cliente
- 3 - Lista de Clientes Activos
- 4 - Lista de Clientes Banidos
- 5 - Voltar Atrás

Na 1ª opção temos a funcionalidade de adicionar um novo cliente e onde é pedido ao utilizador informações como o Nome do Cliente e o seu NIF (tem de ter obrigatoriamente 9 dígitos).

Na 2ª opção temos a possibilidade de remover um Cliente, onde é pedido ao utilizador que insira o NIF do Cliente a remover.

Na 3ª opção é-nos mostrada a informação de cada utilizador activo. (Nome, NIF, Nº de Aluguers que o cliente tem de momento, e a informação de cada aluguer que fez até ao momento, ID da Guitarra, o seu estado e a Data Inicial e de Entrega (se for o caso) do Aluguer).

Na 4ª opção temos a funcionalidade de listar a informação dos clientes que estão na lista negra (banidos), sendo mostrado ao utilizador o NIF do cliente banido e o seu motivo.

Por último temos a opção 5 que permite ao utilizador retornar ao menu anterior.

Menu Guitarras:

- 1 - Adicionar Guitarra
- 2 - Histórico de Guitarra
- 3 - Listagem de Guitarras
- 4 - Listagem de Guitarras Alugadas
- 5 - Voltar Atrás

Na 1ª opção é pedido ao utilizador os dados da guitarra, como o seu ID, o nome da mesma, o preço diário a cobrar, o seu valor e o estado.

Na 2ª opção temos a funcionalidade de “investigar” uma guitarra introduzindo o seu ID, onde nos é mostrado os dados do utilizador que está a alugá-la (NIF e Nome do Cliente) e os dados da guitarra (ID, Preço/dia, Valor e o Estado da mesma).

Na Listagem de Guitarras (3ª opção) é onde é mostrada toda a informação das guitarras, ou seja, ID, Nome, Preço por Dia, Valor e o Estado de cada uma delas.

Por último na Listagem de Guitarras Alugadas (4ª opção) é somente mostrada a informação de Guitarras que estão a ser alugadas no momento.

A opção 5 serve para voltar para o menu anterior.

Menu Aluguer:

- 1 - Criar Aluguer
- 2 - Concluir Aluguer
- 3 - Lista de Alugueres
- 4 - Voltar Atrás

Na 1ª opção temos a funcionalidade criar um novo aluguer, sendo pedido ao utilizador o NIF do cliente, o ID da guitarra a alugar e a data inicial do aluguer. É mostrado de seguida a data final do aluguer (onde até a essa mesma data não paga multa por excesso de dias).

Na 2ª opção temos a oportunidade de concluir um aluguer, sendo pedido ao utilizador o NIF do cliente que quer concluir o aluguer, a data de entrega da guitarra e o seu estado (se danificada ou se foi entregue em boas condições).

Na 3ª opção é listado ao utilizador as informações dos alugueres activos, sendo apresentada as informações de cada cliente (Nome e NIF) e cada um dos seus alugueres (ID da Guitarra, Data Inicial de Aluguer).

Na última opção o utilizador pode retornar para o menu anterior.