

1 2



9 0

FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE  
COIMBRA

Licenciatura em Engenharia Informática  
Princípios de Programação Procedimental  
2022/2023 - 2º Semestre

Aplicação de gestão de reservas

André Miguel Martinho de Melo

2022213285

Diogo Baptista Costa

2022213506

# Estruturas de dados utilizadas

1. **Estrutura Data:** guarda o dia, o mês e o ano de uma data.
2. **Estrutura Reserva:** guarda as informações sobre uma reserva ou pré-reserva. Se o estado for igual a 1, foi guardada uma pré-reserva, enquanto que se for igual a 2, foi guardada uma reserva.  
Para além disso, é guardado o id do utilizador, o numero da reserva/pré-reserva, o tipo (lavagem ou manutenção), a data da reserva/pré-reserva.  
Por fim é guardado um ponteiro que aponta para a uma Estrutura Hora com a hora de inicio e um outro, também para uma Estrutura Hora, que aponta para a hora de fim.
3. **Estrutura no:** guarda um ponteiro que aponta para uma Estrutura Reserva e que guarda informações de uma pré-reserva e tem um ponteiro para a próxima estrutura no.
4. **Estrutura Lista:** dois ponteiros que apontam para uma Estrutura no, um com o inicio e outro com o fim.

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

struct Reserva{
    int id;
    int n_reserva;
    int tipo; //Lavagem=1, Manutenção=2
    int estado; //Pré reserva=1, Reserva=2
    Data data;
    Hora *inicio;
    Hora *fim;
};

typedef struct no{
    Reserva *pre_reserva;
    struct no *proximo;
}no;

typedef struct{
    no *inicio;
    no *fim;
}lista;
```

```
struct Hora{
    char hora[20];
    Reserva *reserva;
    lista *pre_reservas;
    struct Hora *proximo;
    struct Hora *anterior;
};

typedef struct Lista_Blocos{
    Hora *inicio;
    Hora *fim;
    Data data;
    struct Lista_Blocos *proximo;
    struct Lista_Blocos *anterior;
}Lista_Blocos;

typedef struct{
    Lista_Blocos *inicio;
    Lista_Blocos *fim;
}Lista_Dias;

typedef struct {
    int id;
    Reserva reservas;
} Cliente;
```

5. **Estrutura Hora:** guarda uma string com o bloco de hora correspondente, por exemplo: 08h00, um ponteiro para uma Estrutura Reserva para guardar informação sobre uma reserva, um ponteiro para uma Estrutura Lista para guardar as informações de uma pré-reserva e dois ponteiros para Estrutura Hora, um que aponta para a hora anterior e outro que aponta para a hora seguinte.
6. **Estrutura Lista\_Blocos:** dois ponteiros para Estrutura Hora, um aponta para a hora de inicio e outro aponta para a hora de fim. Guarda-se informação sobre uma data e dois ponteiros para Estrutura Lista\_Blocos que apontam para a hora anterior e para a hora seguinte.
7. **Estrutura Lista\_Dias:** dois ponteiro para Estrutura Lista\_Blocos um para o inicio e outro para o fim.
8. **Estrutura Cliente:** guarda o id do cliente e ainda as reservas deste.

# Estrutura geral do programa

O programa está dividido em um ficheiro .c e um ficheiro .h (header file) de forma a facilitar a organização do programa, em vez de estar tudo contido num único ficheiro main.c.

1. **projeto.c**: ficheiro onde se encontram todas as funções:

1.1. **main()**: Função que define os blocos de horas, define a data inicial do programa, cria a lista de datas, chama a função menu() e que lê as informações contidas nos ficheiros.

1.2. **menu()**: Função que apresenta o menu e que recebe os inputs do utilizador. Consoante o input, são chamadas diversas funções.

1.3. **cria\_fila()**: A função cria uma lista ligada de blocos de tempo representados pela Estrutura Lista\_Blocos. Para cada bloco de tempo, representado pela Estrutura Hora, a função aloca memória, inicializa os membros e adiciona-os no início da lista. Além disso, são alocadas memórias para as Estruturas Reserva e Lista, para armazenar informações sobre reservas e pré-reservas. No final, a função retorna a lista ligada de blocos de tempo.

1.4. **compara\_datas()**: Se a data for menor retorna -1, se for maior retorna 1, se for igual retorna 0.

1.5. **avanca\_data()**: Verifica a que mes pertence, se estiver no último dia muda o mês, caso contrário aumenta o dia. A 31/12/ano mudamos de ano.

1.6. **cria\_lista\_datas()**: Cria os blocos de hora para cada dia.

1.7. **adiciona\_data()**: Adiciona um novo dia e cria blocos de hora para esse dia.

1.8. **imprime\_lista()**: Imprime a lista de blocos de horas para o utilizador ver se estão ocupadas ou livres. Faz a distinção entre lavagem e manutenção. Se for lavagem apresenta blocos de meia hora e se for manutenção apresenta blocos de hora.

1.9. **listarReservas()**: Imprime as reservas e as pré-reservas separadamente. Indica sempre o numero de reserva, o id, o tipo de reserva e a hora de início.

1.10. **listarReservasCliente()**: Através do id do cliente é feita uma filtragem para apenas apresentar as reservas e pré-reservas do cliente com o id fornecido.

1.11. **compara\_strings()**: Verifica se a hora introduzida pelo utilizador é igual a alguma dos blocos criados.

1.12. **procura\_data()**: Procura uma data e dá-lhe print.

1.13. **marcar\_reserva()**: Cria uma reserva se o horario estiver livre ou então cria uma pré-reserva para esse horário.

1.14. **libertar\_memoria()**: Liberta a memoria alocada.

1.15. **le\_ficheiro\_variaveis()**: Lê o ficheiro das variáveis (último id gerado, último número de reserva gerado, data do dia atual).

1.16. **le\_ficheiro\_reservas()** e **le\_ficheiro\_pre\_reservas()**: Leem as informações contidas no ficheiro correspondente.

1.17. **guarda\_no\_ficheiro\_variaveis()**: Escreve as informações de variáveis de controlo num ficheiro, escreve cada variável numa linha. Reescrevemos o ficheiro todo quando se fecha o programa ou se muda de dia.

1.18. **guarda\_no\_ficheiro\_reservas()** e **guarda\_no\_ficheiro\_pre\_reservas()**: Escreve as informações das reservas ou pré-reservas no ficheiro correspondente. Escreve cada informação de cada reserva/pré-reserva numa linha formando um bloco.

1.19. **cancelarReserva()**: Depois de escolhida a opção para cancelar e depois de fornecido o id é chamada a função `listarReservasCliente()` para o cliente poder ver todas as suas reservas. O cliente tem de introduzir o numero da reserva que pretende cancelar.

1. Se quiser cancelar Manutenção:

1.1. Se o que estiver em lista de espera for lavagem passa para reserva. Se o bloco seguinte estiver livre com:

1.1.1. Lavagem em pré-reserva passa para reserva.

1.1.2. Manutenção em pré-reserva vê se o bloco seguinte está livre e passa a reserva.

1.2. Se o que estiver em lista de espera for uma manutenção:

1.2.1. Se a hora de início (da que esta em espera) for a mesma e o bloco seguinte estiver livre passa para reserva.

1.2.2. Se a hora de início (da que esta em espera) for igual à hora anterior e a hora anterior estiver livre passa para reserva.

2. Se quiser cancelar Lavagem:

2.1. Se o que estiver em lista de espera for lavagem passa para reserva.

2.2. Se o que estiver em lista de espera for manutenção:

2.2.1. Se a hora de início for igual à hora da que queremos cancelar e o bloco seguinte estiver livre, passa para reserva.

2.2.2. Se a hora de início for igual à hora anterior e o bloco anterior estiver livre, passa para reserva.

3. Cancelar pré-reservas:

3.1. Cancelar pré-lavagem: remover a pré-reserva e a pré-reserva seguinte (na lista de espera) passa para a frente, fica em primeiro da lista.

3.2. Cancelar pré-manutenção:

3.2.1. Hora de início igual à hora anterior: a pré-reserva seguinte da hora de início e a pré-reserva seguinte da hora anterior passam para primeiro.

3.2.2. Hora de início igual à hora seguinte: a pré-reserva seguinte da hora de início e a pré-reserva seguinte da hora seguinte passam para primeiro.

## Estrutura dos ficheiros de texto

Foram utilizados três ficheiros:

1. reservas.txt: ficheiro que guarda todas as reservas da seguinte forma:

**Primeira linha:** indica o número da reserva  
**Segunda linha:** indica o id da reserva  
**Terceira linha:** indica o tipo de reserva  
**Quarta linha:** indica o estado da reserva  
**Quinta linha:** indica o dia da reserva  
**Sexta linha:** indica o mes da reserva  
**Sétima linha:** indica o ano da reserva  
**Oitava linha:** indica a hora de início da reserva

```
0021
0021
01
02
30
10
2023
08:00
```

2. pre\_reservas.txt: ficheiro que guarda todas as reservas da seguinte forma:

**Primeira linha:** indica o número da pré-reserva  
**Segunda linha:** indica o id da pré-reserva  
**Terceira linha:** indica o tipo de pré-reserva  
**Quarta linha:** indica o estado da pré-reserva  
**Quinta linha:** indica o dia da pré-reserva  
**Sexta linha:** indica o mes da pré-reserva  
**Sétima linha:** indica o ano da pré-reserva  
**Oitava linha:** indica a hora de início da pré-reserva

```
|
0024
0024
01
01
28
09
2023
08:00
```

3. Variáveis.txt: ficheiro que guarda as variáveis de controlo da seguinte forma:

**Primeira linha:** indica o último id gerado  
**Segunda linha:** indica o último número de reserva gerado

Próximas linhas guardam a informação da data atual do programa:

**Terceira linha:** indica o dia  
**Quarta linha:** indica o mes  
**Quinta linha:** indica o ano

```
20
20
16
06
2023
```

# Manual

Agora será explicado, passo a passo, como executar o programa.

1. Inicie o programa e ser-lhe-á apresentado o menu com as diferentes opções:

```
Bem vindo à oficina do Carlos.  
  
Menu:  
1. Listar reservas  
2. Listar reservas de um cliente  
3. Cancelar uma reserva  
4. Fazer uma reserva  
5. Avançar um dia  
6. Sair  
Escolha uma opção: █
```

2. Agora analisaremos cada uma das opções do menu:

2.1. (1) Listar reservas: ao seleccionar esta opção serão listadas todas as reservas e pré-reservas guardadas.

2.2. (2) Listar reservas de um cliente: ao seleccionar esta opção será pedido o id do utilizador e serão apresentadas todas as reservas e pré-reservas do utilizador com o respetivo id.

2.3. (3) Cancelar uma reserva: ao seleccionar esta opção será pedido o id do utilizador e serão apresentadas todas as reservas e pré-reservas do utilizador com o respetivo id. Depois será pedido o número de reserva (numero antes dos dois pontos) associado à reserva que se pretende cancelar e a reserva fica cancelada.

2.4. (4) Fazer uma reserva: ao seleccionar esta opção será pedido o id do utilizador (caso ainda não tenha será atribuído um), será pedido o tipo de serviço pretendido (lavagem ou manutenção), será pedida a data pretendida, será mostrado os horários disponíveis e ocupados e será pedido ao utilizador que escolha um no mesmo formato. Se o horário já estiver ocupado ficará em pré-reserva.

2.5. (5) Sair: Encerra o programa.

2.6. (6) Avançar um dia: Acrescenta à data atual um dia.

