



---

# TRIP PLANNER

---

PPP 2019



**FCTUC** FACULDADE DE CIÊNCIAS  
E TECNOLOGIA  
UNIVERSIDADE DE COIMBRA



TRABALHO REALIZADO POR:  
LEANDRO BORGES PAIS Nº 2017251509  
BRUNO CARLOS LUIS FERREIRA Nº2015240050

## *Introdução*

---

No âmbito da cadeira de *Princípios de Programação Procedimental* desenvolvemos, utilizando a linguagem C um planeador de viagens cujo objetivo é fornecendo uma quantidade de variáveis como, por exemplo, locais a visitar, pontos de interesse favoritos entre outros, gerar automaticamente uma viagem ideal e avaliá-la em termos de popularidade.

O objetivo principal deste trabalho é desenvolver competências de programação em linguagens de baixo nível como o C.

## *Estrutura Geral Do Programa*

---

Todo o funcionamento está baseado em listas ligadas e estruturas de dados. As listas ligadas possuem um campo de informação que apontam para as respetivas estruturas que armazenam os vários tipos de dados (sejam, nomes de cidades, horário de funcionamento, etc...). Entendemos ainda que um vetor de dimensão 3 da estrutura “Place” é uma viagem, e cada local possui em si, uma lista ligada de pontos de interesse.

Os menus do programa obedecem todos a uma estrutura semelhante em que a cada opção do menu é atribuída um número e esse número é que é utilizado internamente para tratar da navegação pelo programa.

Os menus em si são apenas switch-cases que utilizam o valor previamente mencionado para navegar de função em função, funções essas que correspondem a outros menus.

O funcionamento em si, é explicado com mais detalhe mais à frente.

## Implementação em C

---

### 1. Estrutura do utilizador

```
//User
typedef struct user{
    char name[BUFSIZE];
    char pw[BUFSIZE];
    char address[BUFSIZE];
    char birth[BUFSIZE];
    char phoneNum[BUFSIZE];
    char hot[BUFSIZE];
    char favPoi[60][BUFSIZE];
    char favLoc[3][BUFSIZE];
    int globalFavLoc;
    int globalFavPoi;
    int globalHot;
    Place trip[3];
}User;
```

Estrutura utilizada para manter todos dados relativos ao utilizador, desde os dados do registo, aos dados necessários para a criação da viagem.

A variável “favPoi” é um vetor que mantém todos os pontos favoritos do utilizador já a “favLoc” faz o mesmo para os locais.

O vetor de dimensão 3 é a viagem associada ao utilizador.

Estes dados, são atualizados no ficheiro quando o utilizador faz “log off” da aplicação.

### 2. Estrutura dos Pontos de Interesse

```
typedef struct poi{
    char name[BUFSIZE];
    char desc[BUFSIZE];
    char wperiod[BUFSIZE];
    int fav;
    int popularity;
    int hot;
}PoI;
```

Estrutura utilizada para manter todos os dados relativos aos pontos de interesse.

### 3. Estrutura dos Locais

```
typedef struct place{
    char cityName[BUFSIZE];
    int popularity;
    int fav;
    poiList listPoi;
}Place;
```

Estrutura utilizada para manter todos os dados relativos aos locais a visitar. É de notar que cada local tem uma lista ligada de pontos de interesse.

#### 4. Lista Ligada dos Pontos de Interesse

```
//PoI linked list
typedef struct lnode *poiList;
typedef struct lnode{
    PoI info;
    poiList next;
}nodeList;
```

Lista ligada de locais, cada nó da lista tem um campo de informação que aponta para a estrutura de dados dos pontos de interesse, previamente vista, e um ponteiro para o nó seguinte.

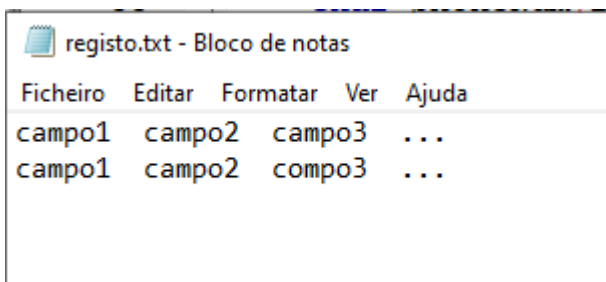
#### 5. Lista Ligada de Locais

```
//Place Linked List
typedef struct llnode *pList;
typedef struct llnode{
    Place info;
    pList next;
}pnodeList;
```

Lista ligada de locais, cada nó da lista tem um campo de informação que aponta para a estrutura dos locais previamente vista, e um ponteiro para o nó seguinte.

Como referido anteriormente estas estruturas são guardadas no final do programa em ficheiros, que têm as seguintes estruturas:

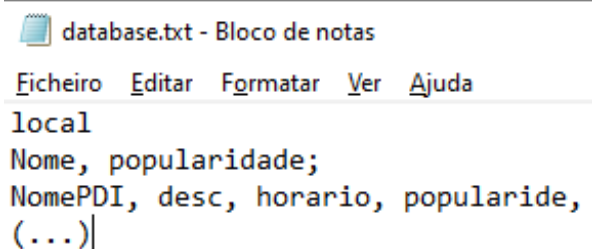
##### 1. Ficheiro de Registos



Ficheiro	Editar	Formatar	Ver	Ajuda
campo1	campo2	campo3	...	
campo1	campo2	compo3	...	

Para o ficheiro de registos optámos por uma estrutura simples em que todos os campos associados ao utilizador são separados por tab's e cada utilizador ocupa pelo menos uma linha, facilitando as funções de procura e escrita.

##### 2. Ficheiros de Base de Dados

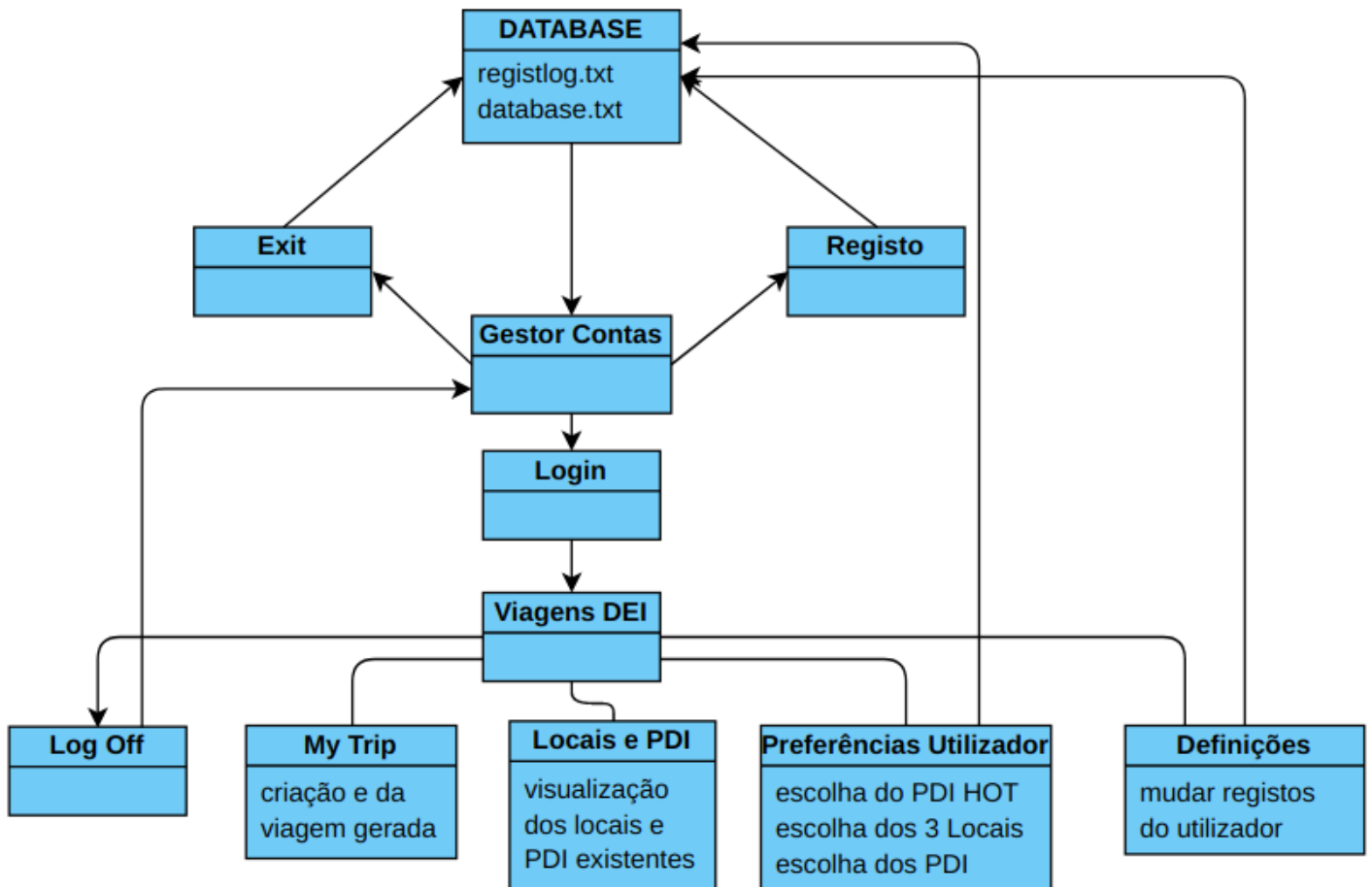


Ficheiro	Editar	Formatar	Ver	Ajuda
local	Nome, popularidade;	NomePDI, desc, horario, popularide,	(...)	

Neste caso, utilizamos uma estrutura mais complexa com delimitadores para que possamos mais facilmente ir buscar toda a informação relativa ao local, inclusive a lista de pontos de interesse.

## Diagrama de Estruturas

---



## *Funcionamento*

---

Em primeiro lugar, é de salientar que toda a interação com o utilizador é feita a partir de consola.

O programa começa por instanciar a estrutura a estrutura User com o nome “usr”, após o começo da sessão, existe um primeiro menu, em que o utilizador se pode registar, se não possuir conta, ou fazer o login com os seus dados caso contrário, seguidamente estamos de caras com o menu da viagem em si que permite, adicionar ou remover (caso tenham sido previamente adicionados) pontos de interesse, locais, e o ponto “hot” após estarem adicionados 3 locais favoritos, passa a ser possível gerar a viagem em si. Existem ainda outras opções como alterar os dados previamente fornecidos pelo utilizador e listar todos os locais e respetivos pontos de interesse por ordem alfabética ou por ordem decrescente de popularidade. No final da sessão o utilizador pode fazer “log off” para dar lugar a outro utilizador ou terminar a sessão completamente caso assim o deseje. Qualquer uma destas ações ativa uma função que descarrega toda a informação que o utilizador forneceu e/ou alterou sobre si e também sobre as listas, nomeadamente a popularidade dos locais e pontos de interesse que são atualizados no final de cada sessão.

## *Conclusão*

---

Com este trabalho conseguimos cimentar conhecimentos teóricos sobre listas e estruturas adquiridos ao longo do semestre, e mais importante ainda conseguimos adquirir conhecimentos práticos sobre a implementação prática de sistemas do género, e por fim, adquirimos conhecimentos básicos do funcionamento de sistemas de “log in”, “bases de dados”, entre outros, contribuindo assim para a nossa formação.