

• U • C •

FCTUC

FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Relatório do Projeto de PPP

Projeto 2018/19
Planeamento de Viagens no DEI



Autores:

Manuel Tolda Fideles & Rui Alexandre Tapadinhas

Introdução:

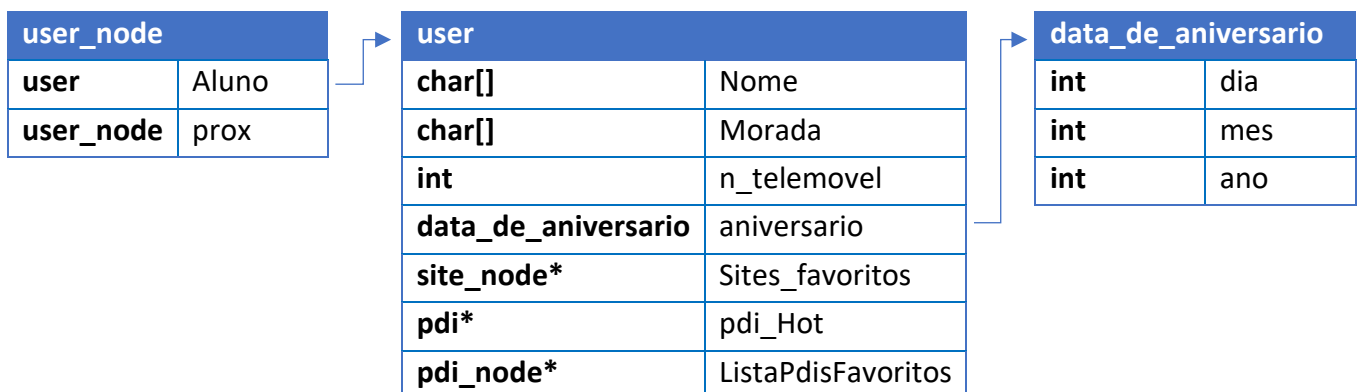
No âmbito da cadeira de Princípios de Programação Procedimental, foi-nos pedido que desenvolvêssemos um programa, escrito na linguagem C, que organizasse uma viagem baseada nas escolhas dos utilizadores. O programa deve ainda receber um ficheiro de texto com os locais e guardar as preferências do utilizador em outros ficheiros de texto.

Este projeto tem como objetivo ensinar os programadores a utilizar listas ligadas da linguagem C. Sendo assim, na nossa aplicação, os dados dos locais e dos Pdl's (Pontos de Interesse) armazenados em ficheiros de texto são carregados para listas ligadas e os dados do registo e as preferências dos utilizadores guardados em listas ligadas são escritos em ficheiros de texto para que quando o programa seja encerrado, os dados não sejam perdidos e para que possam ser recuperados em sessões seguintes.

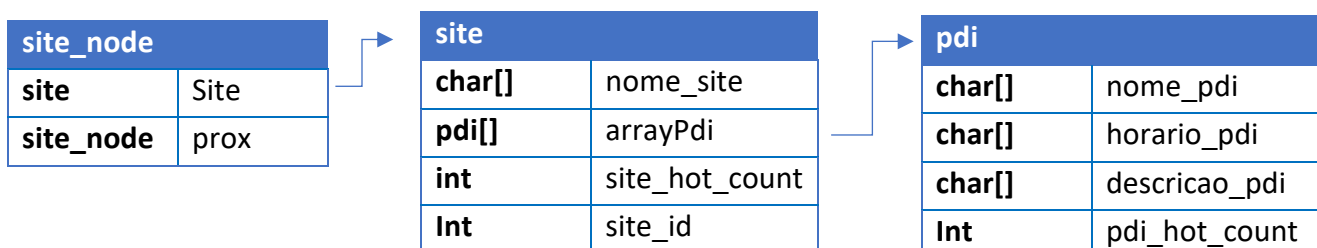
Estruturas:

Todas as estruturas são declaradas num ficheiro auxiliar com o nome "structs_projeto.h".

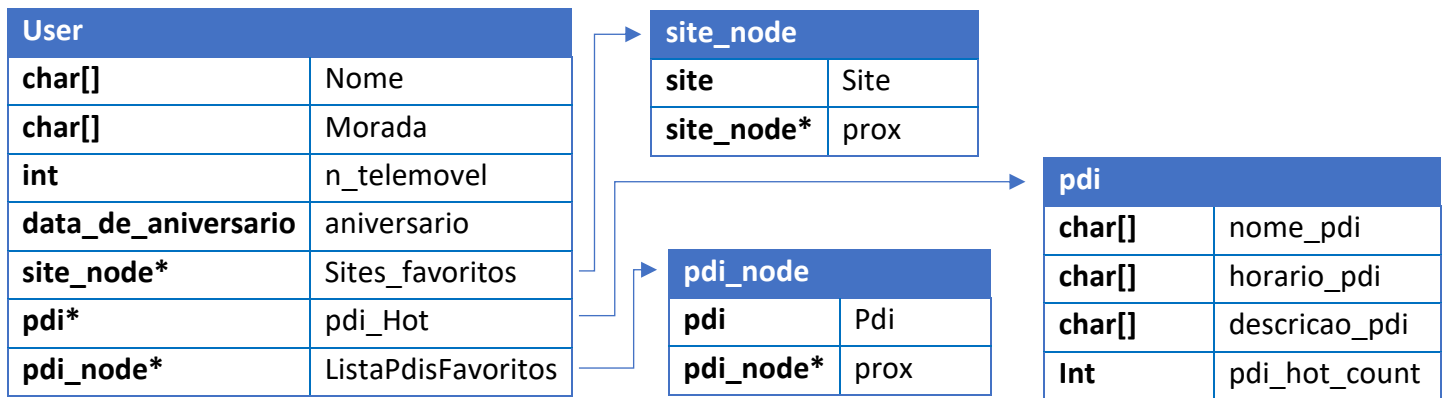
Para organizar os utilizadores usámos uma lista simplesmente ligada na qual usamos as estruturas (**user** e **user_node** e **data_de_aniversario**). Como só precisamos de percorrer a lista uma vez escolhemos utilizar uma lista simplesmente ligada.



Para organizar os locais e Pdl's utilizamos outra lista simplesmente ligada, uma vez que também não precisamos de percorrer a lista nos dois sentidos. Usamos as estruturas **site**, **site_node** e **pdi**.



Para organizar as preferências do utilizador utilizamos uma estrutura (**pdi**) e duas listas simplesmente ligadas (**site_node** e **pdi_node**) dentro da estrutura **user**.



Estrutura dos ficheiros de texto:

- Ficheiro dos locais ("locais2.txt"):

Neste ficheiro encontram-se listados os nomes dos locais seguidos do seu índice de popularidade, e nas 3 linhas seguintes encontram-se os Pdi's de cada local e toda a sua informação, separada por ",".

locais2.txt - Bloco de notas

Ficheiro Editar Formatar Ver Ajuda

```
Viana do Castelo,0
Centro Historico,10h-18h,Bar,0
IPVC,7h-20h,Instituto Politecnico,0
Museu do Traje,10h-18h,Patrimonio cultural,0
Vila Real,0
Quilate 24,15h-00h,Bar,0
Museu de Arqueologia,10h-18h,Arqueologia,0
UTAD,7h-20h,Universidade,0
Braganca,0
Lagoa Azul,11h-16h,Bar,0
Museu do Abade,11h-17h,Museu religioso,0
IPB,7h-20h,Universidade,0
```

Nome do local, Índice de popularidade

Nome do Pdi, Horário, Descrição, Índice de popularidade

Nome do Pdi, Horário, Descrição, Índice de popularidade

Nome do Pdi, Horário, Descrição, Índice de popularidade

Nome do local, Índice de popularidade

[...]

- Ficheiro dos locais organizados ("locais_sorted.txt"):

Este ficheiro é criado por uma função que lê a informação contida no ficheiro dos locais e insere um ID para cada local seguido de toda a informação referente aos locais e aos respetivos Pdi's organizada alfabeticamente pelo nome de cada local.

locais_sorted.txt - Bloco de notas

Ficheiro Editar Formatar Ver Ajuda

```
1
Aveiro,0
Bodegas,20h-05h,Bar,0
UA,7h-20h,Universidade,0
Parque da Macaca,Sempre aberto,Parque Cultural,0
2
Beja,0
Karga Club,17h-02h,Bar,0
IPB,7h-20h,Universidade,0
Jardim das Alcaçarias,sempre aberto,Parque cultur
3
Braga,0
Rossio,22h-03h,Bar,0
UM,7h-20h,Universidade,0
Museu Pio XII,11h-17h,Museu da era monarquica,0
```

ID

Nome do local, Índice de popularidade

Nome do Pdi, Horário, Descrição, Índice de popularidade

Nome do Pdi, Horário, Descrição, Índice de popularidade

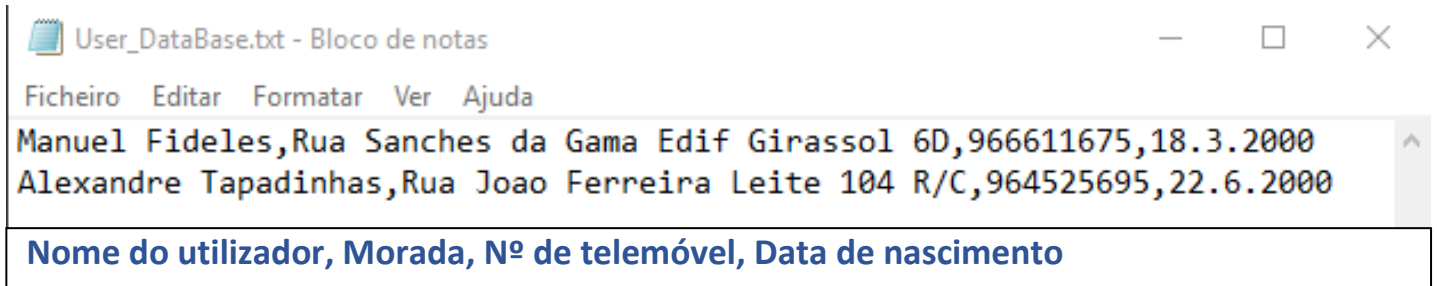
Nome do Pdi, Horário, Descrição, Índice de popularidade

Nome do local, Índice de popularidade

ID

[...]

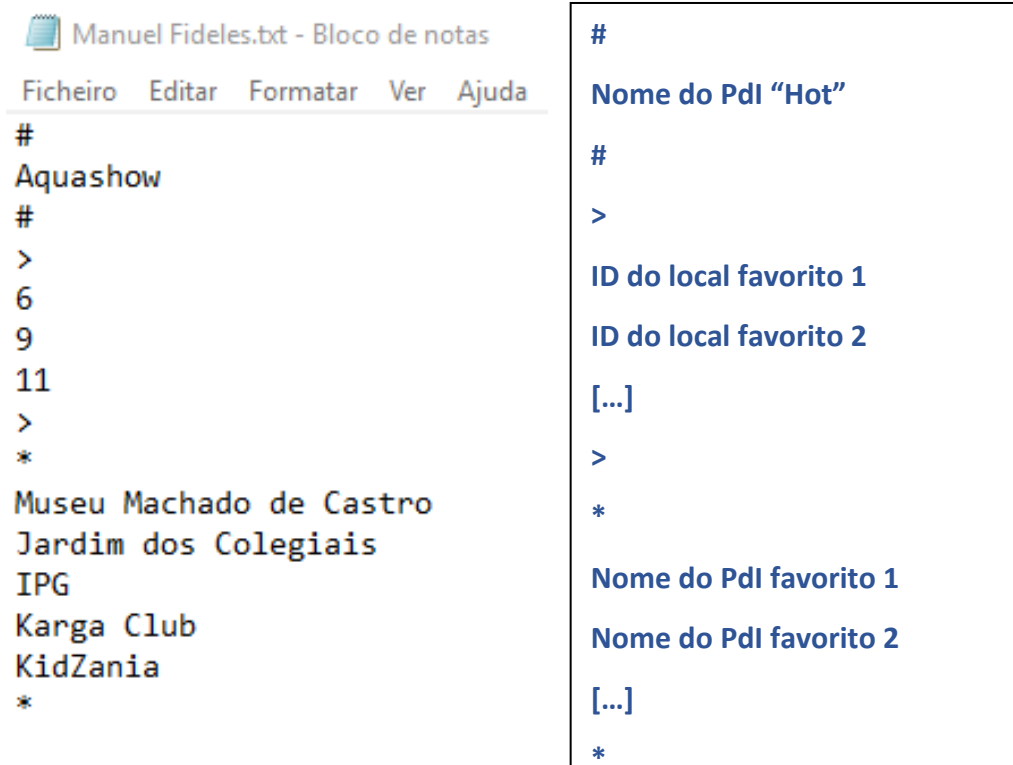
Em cada linha deste ficheiro são guardados os dados referentes a cada utilizador contidos na **struct user**, separados por “,”.



```
User_DataBase.txt - Bloco de notas
Ficheiro  Editar  Formatar  Ver  Ajuda
Manuel Fideles,Rua Sanches da Gama Edif Girassol 6D,966611675,18.3.2000
Alexandre Tapadinhas,Rua Joao Ferreira Leite 104 R/C,964525695,22.6.2000
Nome do utilizador, Morada, Nº de telemóvel, Data de nascimento
```

- Ficheiro das preferências do utilizador (“[nome de utilizador].txt”):

Este ficheiro é criado por uma função assim que o registo do utilizador é concluído. Posteriormente, são escritos neste ficheiro, de forma codificada, os locais e os Pdl’s favoritos e o Pdl “Hot” escolhidos pelo utilizador.



```
Manuel Fideles.txt - Bloco de notas
Ficheiro  Editar  Formatar  Ver  Ajuda
#
Aquashow
#
>
6
9
11
>
*
Museu Machado de Castro
Jardim dos Colegiais
IPG
Karga Club
KidZania
*
#
Nome do Pdl “Hot”
#
>
ID do local favorito 1
ID do local favorito 2
[...]
>
*
Nome do Pdl favorito 1
Nome do Pdl favorito 2
[...]
*
```

Organização das funções segundo os ficheiros “.h” e “.c”:

“**menu_inicial.c**”: Ficheiro com a função **menu** (função que chama a maior parte das funções definidas nos restantes ficheiros).

“**structs_projeto.h**”: Ficheiro com todas as estruturas usadas no trabalho, anteriormente explicadas.

“operações_listas_ligadas.c”: Ficheiro com as funções relacionadas com listas ligadas dos locais e dos Pdl’s e funções de leitura e escrita de ficheiros que usam informação dessas listas ligadas.

“getPreferencias.c”: Ficheiro com funções relacionadas com listas ligadas e listas de Pdl’s das preferências do utilizador.

“getAluno.c”: Ficheiro com funções relacionadas com o registo de utilizadores e com a criação e escrita dos ficheiros de texto da base de dados dos utilizadores e das preferências de cada utilizador.

“user_inputs.c”: Ficheiro com funções de proteção de recolha de dados introduzidos pelo utilizador.

Breve explicação do funcionamento do programa:

Ao iniciar o programa, o ficheiro “locais2.txt”, que contém os dados referentes a cada local e a cada Pdl, é lido e todos os dados são guardados numa lista ligada e uma lista de struct pdi de estruturas. De seguida, através do método *insertion sort*, é criada uma nova lista ligada organizada alfabeticamente pelos nomes dos locais, com os dados anteriores acrescidos de um ID para cada local.

Após termos um ficheiro com os dados organizados alfabeticamente, lê-mos este novo ficheiro (“locais_sorted.txt”), e criamos uma nova lista ligada com todos os dados organizados e pronta a utilizar.

Antes de começar a interação do utilizador com o programa, só falta carregar o perfil de todos os utilizadores, presentes no ficheiro “User_DataBase.txt” para o caso de ser necessário procurar os dados de algum utilizador que já tenha feito o registo noutra sessão. Todos os dados dos utilizadores são carregados para uma nova lista ligada que usa as estruturas **user** e **user_node**. Esta lista é posteriormente recebida pela função **lookup** que a vai percorrer até encontrar um **user** com o número de telemóvel que corresponda ao introduzido pelo operador do programa.

A primeira interação do programa com o utilizador é perguntar ao utilizador se já efetuou o registo ou não. No caso de ser a primeira vez a usar o programa, é pedido ao utilizador que efetue o registo na base de dados a partir da interface do programa. Os dados introduzidos no registo são guardados numa **struct user** e de seguida escritos no ficheiro “User_DataBase.txt”.

De seguida é criado um ficheiro de texto com o nome deste novo utilizador, que tem como intuito guardar as preferências do utilizador após ele as escolher.

No caso de o registo já ter sido efetuado noutra sessão, é apenas pedido ao utilizador que introduza o número de telemóvel associado ao registo previamente feito para que a função **lookup** encontre o utilizador e carregue os seus dados retomando a sessão e as preferências já escolhidas e registadas nos ficheiros.

Visto que a função **lookup** retorna um ponteiro para a estrutura **user**, já temos todos os dados necessários para continuar a execução do programa.

Após os dados do utilizador serem carregados e a sessão ser iniciada ou reiniciada, começa a etapa de planeamento da viagem. Nesta etapa é impressa a lista simplesmente ligada organizada dos locais em que cada local é precedido de um ID e seguido do índice de popularidade atual. De seguida, é pedido ao utilizador que introduza o ID correspondente ao local do qual pretende ver os seus Pdl's.

Neste momento o operador tem de decidir se quer adicionar o local ou algum Pdl à lista de locais ou Pdl's preferidos ou se pretende definir algum dos Pdl como o seu Pdl "hot". (Observação: O Pdl "Hot" não precisa de ser de nenhum dos 3 locais favoritos.).

Após adicionar algum Pdl aos preferidos ou definir o seu Pdl "hot", é apresentado um menu ao utilizador, no qual pode escolher que ação vai fazer a seguir.

```
Adicionou 'Bodegas' a lista de PDIs favoritos!

Insira:
> 1 para remover um local da sua lista de favoritos
> 2 para remover um PDI da sua lista de PDIs favoritos
> 3 para criar automaticamente a sua viagem
> 4 para continuar a escolher PDIs/locais
> 5 para sair
A sua opcao:
```

A lista dos locais favoritos e a lista de Pdl's favoritos está guardada na **struct user**.

As opções 1 e 2 operam apenas entre as listas de locais favoritos e lista de Pdl's favoritos.

A opção 3 só será executada quando houver, exatamente, 3 locais na lista de locais favoritos. (Se houver menos de 3 locais favoritos, o utilizador tem de continuar a adicionar locais aos seus favoritos e se houver mais de 3 locais favoritos o user tem de remover locais até ter exatamente 3 locais na lista dos favoritos). Até a lista ter exatamente 3 locais, se a opção 3 tentar ser executada, o programa vai pedir que adicione ou remova locais da lista dos seus locais favoritos.

A opção 4 pode ser sempre utilizada mesmo que o número de locais favoritos seja maior que 3 visto que o utilizador tem a opção de remover locais dos favoritos.

A opção 5 serve apenas para sair do programa sem gerar a viagem.

Para gerar a viagem, primeiro é verificado se o Pdl "Hot" é de algum local presente na lista de locais favoritos. Se for então o Pdl "Hot" é impresso. Se não for, vai se verificar se algum dos Pdl's favoritos pertence a algum dos locais favoritos, se sim, esse Pdl é impresso. Mas, se nenhum dos Pdl's favoritos pertencer a nenhum dos locais favoritos, então, é impresso um Pdl pertencente a um dos locais favoritos.

Notas:

- Para fazer login decidimos utilizar o número de telemóvel uma vez que é único.
- Escolhemos que a interface do programa nos menus recebesse as opções escolhidas pelo utilizador através de números, de forma a ser eficiente e causar o mínimo de erros possíveis para o utilizador e para o programador.
- Ao longo do programa, aquando dos inputs do utilizador chamamos 3 funções que, no caso de algum erro de introdução do utilizador ou de algum erro de leitura do programa, pedem de novo o mesmo input até que seja aceite pelo programa.