



**FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE NOVA DE LISBOA**

Departamento de Engenharia Electrotécnica

## **PROGRAMAÇÃO DE MICROPROCESSADORES**

**2021 / 2022**

Mestrado Integrado em Engenharia Electrotécnica  
e Computadores

1º ano

1º semestre

### **Trabalho nº 8 Estruturas de Dados**

# 1 Introdução

No capítulo 11 do livro “Linguagem C” de Luís Damas, recomendado para a disciplina de Programação de Microprocessadores, são apresentadas as estruturas. Esta aula visa consolidar estas matérias através de um conjunto de exercícios.

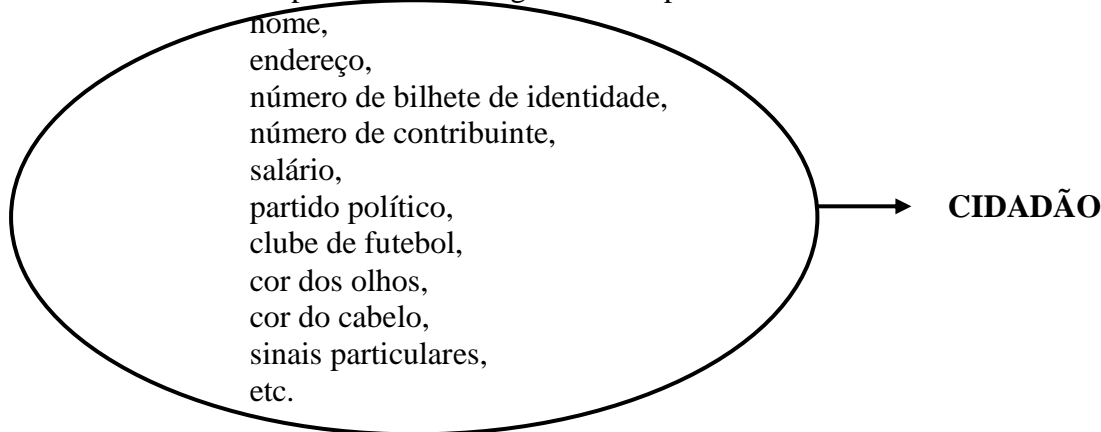
Os trabalhos executados nesta aula vão ter continuação na aula seguinte que trata de ficheiros.

Faça todos os exercícios pedidos em ficheiros separados e **GUARDE O CÓDIGO desenvolvido na memória USB**. Durante a aula o docente pode pedir-lhe para mostrar o código desenvolvido.

## 2 As estruturas: são tipos ou variáveis?

Entre muitos usos possíveis, as estruturas servem para agregar informação de vários tipos sobre uma certa entidade.

Por exemplo, numa base de dados de cidadãos podíamos definir uma estrutura para descrever uma pessoa com os seguintes campos:



Ao fazer isto estamos de facto a definir um novo tipo de dados para o C, um pouco mais complexo do que os tipos básicos (`char`, `int`, etc.).

A definição do tipo tem de estar colocada logo no início do ficheiro de modo a que todas as funções desse ficheiro conheçam o tipo novo.

Como este novo tipo não é um tipo básico de C, não é possível usar as operações que conhecemos com as variáveis deste tipo. Por exemplo, se `comp` e `larg` forem duas variáveis do tipo inteiro, podemos fazer `comp * larg` para calcular a área, ou `comp = larg` se for um quadrado. Se `pessoa1` e `pessoa2` forem duas variáveis de um certo tipo de estrutura, não existe `pessoa1+pessoa2`, ou `pessoa1*pessoa2`, ou `pessoa1 = pessoa2` ou outra qualquer expressão com as operações conhecidas. As operações que conhecemos só estão definidas para os tipos básicos de C que só são cinco (lembra-se quais são?).

Resumindo, as estruturas formam tipos não básicos em C. É claro que pode haver variáveis de todos os tipos em C, incluindo deste novo tipo. As variáveis são definidas onde for preciso. Lembre-se que em Programação de Microprocessadores não podem existir variáveis globais. Assim elas não podem ser definidas no início do ficheiro.

Se as estruturas são um tipo, pode-se definir também apontadores para elas. Aliás os apontadores são um instrumento muito poderoso para se trabalhar com estruturas. Este trabalho contém algumas situações.

### 3 A Base de Dados de um Povo

Imagine uma base de dados com informação sobre os cidadãos. Neste trabalho vai-se definir esta base de dados (exercício 1), construir algumas funções para a poder preencher com dados (exercício 2) e, finalmente, construir um programa pequeno que permita visualizar a informação (exercício 3).

Na aula seguinte vai-se usar ficheiros para termos mais informação sem ter de a escrever no teclado, e também podermos guardar a base de dados.

#### 3.1 EXERCÍCIO 1: INFORMAÇÃO POR CIDADÃO

Vai-se usar estruturas dentro de estruturas pois assim é mais fácil de estruturar o pensamento. O tipo global chama-se `baseDados`. É constituído por cinco sub-agregados, como está descrito em baixo e mostrado na figura.

Número para os serviços secretos:

Número de série do cidadão

Informação pessoal:

Nome, data de nascimento, altura, cor dos olhos, cor do cabelo.

Informação numérica:

Cartão do cidadão, cartão de contribuinte, cartão de eleitor, cartão de segurança social, carta de condução.

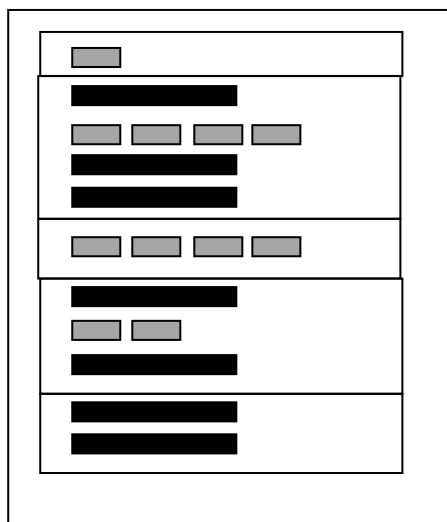
Morada

Rua, código postal

Confidencial:

Partido político, clube de futebol

A figura em baixo mostra as cinco sub-estruturas em que os inteiros são rectângulos pequenos cinzentos e os textos são retângulos maiores e pretos.



O código em baixo mostra a declaração de estruturas a usar.

```
/*
 * base de dados de cidadaos
 * Ficheiro: basedados.c
 */
#include <stdio.h>

struct num_serv_sec
{
    int num_serie;
};

struct info_pessoal
{
    char nome [100];
    int dia_nasc;
    int mes_nasc;
    int ano_nasc;
    int altura;
    char olhos [100];
    char cabelo [100];
};

struct info_numer
{
    int bi;
    int contribuinte;
    int seg_social;
    int carta_cond;
};

struct morada
{
    char rua [100];
    int cod_postal1;
    int cod_postal2;
    char cod_postal3 [100];
};

struct confid
{
    char partido [100];
    char clube [100];
};

struct cidadao
{
    struct num_serv_sec    nss;
    struct info_pessoal    infp;
    struct info_numer      infn;
    struct morada          address;
    struct confid          clubes;
};
```

**EXERCÍCIO 1:** Em vez de o copiar para um ficheiro, faça a declaração usando o typedef e designe o novo tipo de baseDados.

## 3.2 EXERCÍCIO 2: INTRODUÇÃO DE INFORMAÇÃO

Se reparar, o tipo `baseDados` só contém inteiros e vetores de caracteres (em vez de *strings* para ser mais fácil escrever nos ficheiros). Um modo muito simples de efectuar a entrada de dados é construir uma função genérica que leia inteiros (na sua lista de parâmetros até pode ter os intervalos em que os inteiros têm de estar, para haver uma verificação prévia), e outra que leia vetores de caracteres (na sua lista de parâmetros pode ter o comprimento máximo). Até pode enviar na lista de parâmetros de cada uma das funções o texto da pergunta que quer que se mostre ao utilizador.

A função para ler um cidadão deve primeiro chamar as seguintes funções:

```
le_nss  
le_info_pessoal  
le_info_numer  
le_morada  
le_confidencial
```

Cada uma destas escreve o que pretende ler e chama então as funções que lêem os inteiros e os vetores de caracteres.

Assuma que o máximo de cidadãos é 50.

## 3.3 EXERCÍCIO 3: O PROGRAMA

Construa agora um programa que permita criar um cidadão, apagar um cidadão, e escrever alguma, ou toda a informação de um cidadão

O menu está mostrado em baixo.

Tente programar de modo a ter funções e não repetir código entre as várias tarefas.

```
/***** PROGRAMA SIRP – Registo dos cidadaos *****/
```

- a – Mostrar os número de série de todos os elementos da base de dados
- b – Mostrar toda a informação do cidadão com um certo número (a pedir)
- c – Modificar a informação do cidadão com um certo número (a pedir)
- d – Criar um novo registo para um novo cidadão
- e – Apagar o registo com um certo número (a pedir)
  
- s – Sair do programa