

Licenciatura em Engenharia Informática – 2023/24

Programação

1.1. Introdução: Revisão de Conceitos

Francisco Pereira (xico@isec.pt)

- Implementação de programas simples em C

1. Definição da estratégia

- **Exemplo:** Encontrar o maior elemento num conjunto não ordenado
- Variantes
 - E se o conjunto estiver ordenado?
 - Encontrar o menor elemento
 - Encontrar a posição do maior elemento

Ficha
Prática 1

- Implementação de programas simples em C

2. Escrever código

- Divisão em funções
- Estrutura do código
 - Sequencial, Seleção, Repetição
- Tipos de dados
 - Simples
 - `int`, `char`, `double`, `float`
 - Variáveis ou constantes
 - Tabelas (Vetores, Matrizes, Strings)

- **Funcionalidades**

- Tabuleiro 3×3
- Gere as jogadas efetuadas por 2 jogadores
- Deteta final do jogo e anuncia vencedor

- **Simplificações:**

- Interface básico
- Validação de inputs limitada

1. **Inicializar** tabuleiro 3×3
2. **Enquanto** o final do jogo não for atingido **Repetir:**
 1. Mostrar tabuleiro
 2. Obter e validar jogada do jogador atual
 3. Atualizar tabuleiro
3. **Escrever** o resultado final

Onde Armazenar a Informação?

- Tabuleiro

- *tab*: Matriz de caracteres com 3 linhas e 3 colunas

	O	
	X	O
		X

```
#define DIM      3  
char tab[DIM][DIM];
```

- Variáveis inteiras auxiliares:

- *n_jogadas*: Número de jogadas efetuadas
- *joga*: Jogador atual (1 ou 2)
- *ganhou*: Assinala o vencedor

A função main(): parte I

```
#include <stdio.h>
```

Biblioteca standard

```
#define DIM 3
```

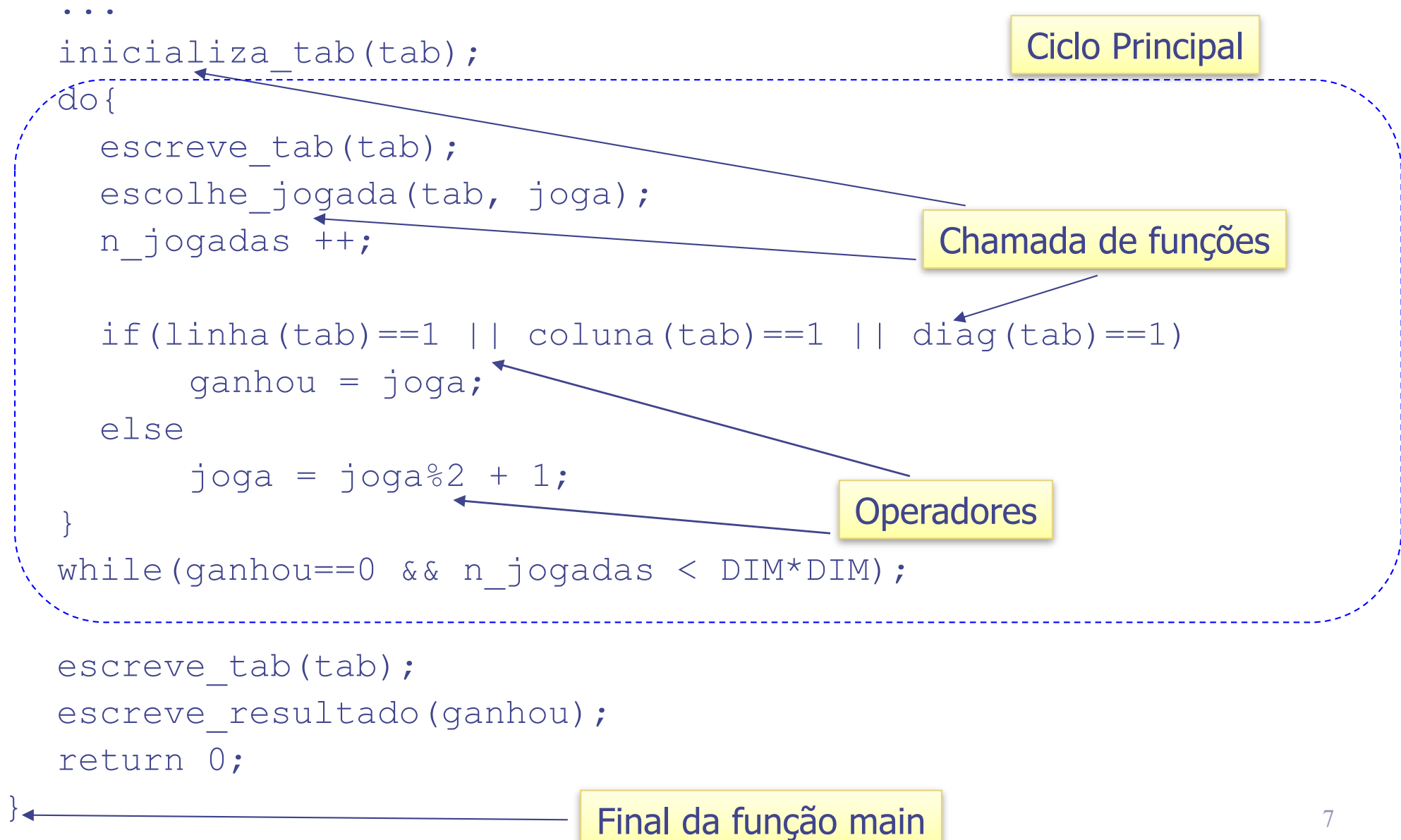
Constante visível em
todo o programa

```
int main() {  
    char tab[DIM][DIM];  
    int joga, n_jogadas, ganhou;  
  
    joga=1;  
    n_jogadas=0;  
    ganhou=0;  
    ...  
}
```

Início da função main

Declaração de
variáveis

A função main(): parte II



Função inicializa_tab()

```
void inicializa_tab(char t[][DIM]);
```

Declaração / Protótipo

Argumentos

```
void inicializa_tab(char t[][DIM]) {  
    int i,j;
```

Definição
(Código da função)

```
    for(i=0; i<DIM; i++)  
        for(j=0; j<DIM; j++)  
            t[i][j] = '_';
```

Percorrer
uma matriz

```
}
```

Manipulação
de uma tabela

Função escreve_tab()

```
void escreve_tab(char t[][DIM]) {  
    int i, j;  
  
    printf("\n\n");  
    for(i=0; i<DIM; i++)  
    {  
        for(j=0; j<DIM; j++)  
            printf(" %c ", t[i][j]);  
        putchar('\n');  
    }  
}
```

Função escolhe_jogada()

Passagem por
referência

Passagem
por valor

```
void escolhe_jogada(char t[][DIM], int jogador){
    int pos;

    printf("\nÉ a vez do jogador %d\n", jogador);
    do{
        printf("Posição: ");
        scanf(" %d", &pos);
    }while(pos<1 || pos>DIM*DIM || t[(pos-1)/DIM][(pos-1)%DIM]!='_');

    if(jogador == 1)
        t[(pos-1)/DIM][(pos-1)%DIM] = 'X';
    else
        t[(pos-1)/DIM][(pos-1)%DIM] = 'O';
}
```

Operadores

Função linha()

```
int linha(char t[][DIM])
{
    int i, j;

    for(i=0; i<DIM; i++)
        if(t[i][0] != '_') {
            for(j=0; j<DIM-1 && t[i][j] == t[i][j+1]; j++)
                ;
            if(j==DIM-1)
                return 1;
        }
    return 0;
}
```

Manipulação
de uma tabela

Devolver um
resultado

Funções coluna() e diag()

```
int coluna(char t[][DIM])  
{  
    return 0;  
}
```

Completar

```
int diag(char t[][DIM])  
{  
    return 0;  
}
```

Completar

Desafio: Completar/Melhorar o código

1. Escrever as funções `coluna()` e `diag()`
2. Adicionar um sistema de recomendação sobre qual a jogada que deve ser feita
3. Adicionar um jogador automático
4. Permitir que os jogadores escolham a dimensão do tabuleiro (por exemplo, entre 3 e 5)
5. Permitir que o jogo seja interrompido e retomado mais tarde
6. ...