

## Licenciatura em Engenharia Informática – 19/20

# Programação

## 0. Revisão de Conceitos

Francisco Pereira (xico@isec.pt)

---

- Escrever 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

Guião  
laboratorial 0

- Escrever programas simples em C
  - 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  $3 \times 3$
- Gere as jogadas efetuadas por 2 jogadores
- Deteta final do jogo e anuncia vencedor

- **Simplificações:**

- Interface básico
- Validação de inputs limitada

- 1. Inicializar** tabuleiro  $3 \times 3$
- 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

	<b>O</b>	
	<b>X</b>	<b>O</b>
		<b>X</b>

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

- 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 N 3
```

Constante

```
int main()
```

Início da função main

```
{
```

```
    char tab[N][N];
```

```
    int joga, n_jogadas, ganhou;
```

Declaração de  
variáveis

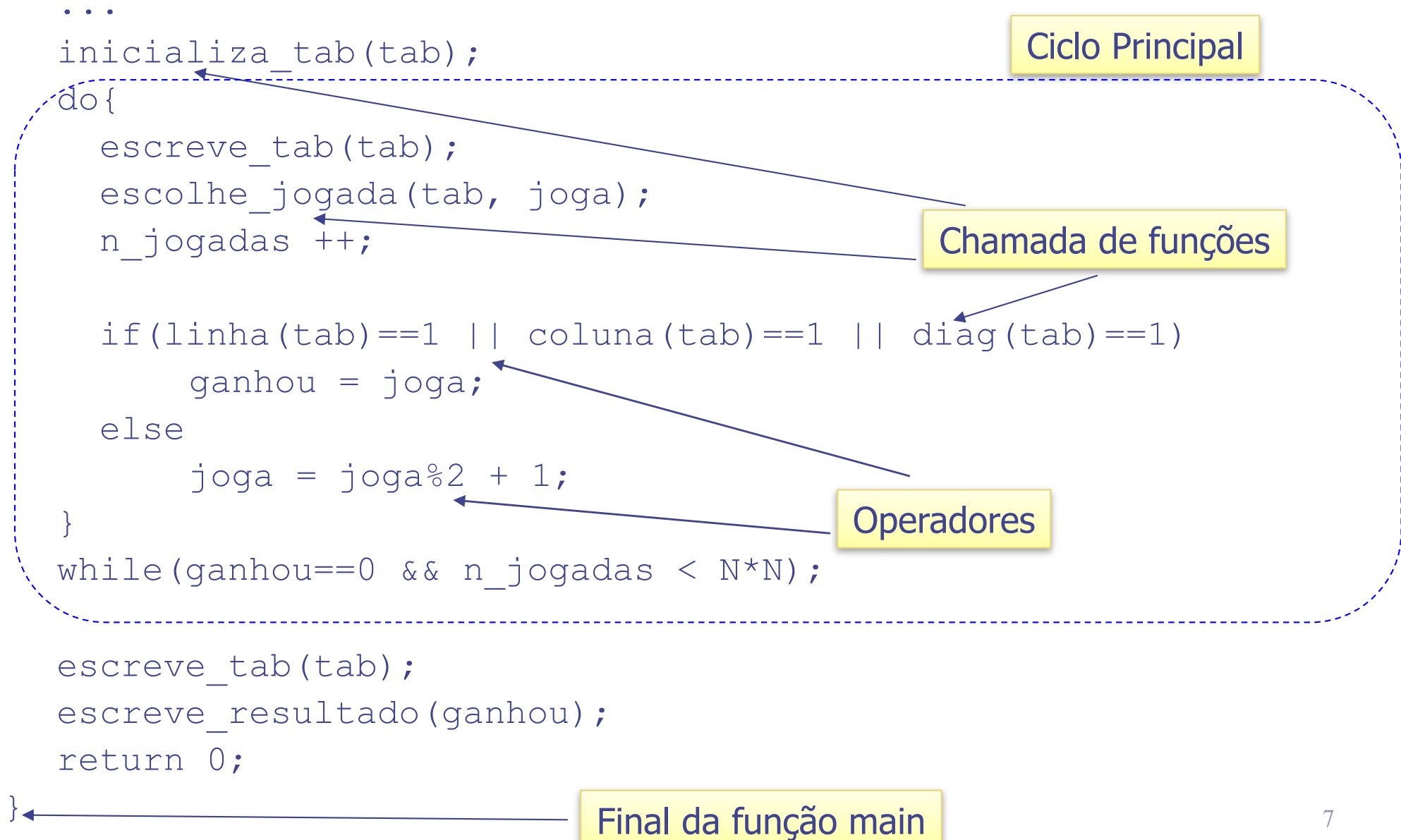
```
    joga=1;
```

```
    n_jogadas=0;
```

```
    ganhou=0;
```

```
    ...
```

# A função main(): parte II





# Função inicializa\_tab()

```
void inicializa_tab(char c[][N]);
```

Declaração / Protótipo

Argumentos

```
void inicializa_tab(char t[][N])  
{  
    int i,j;  
  
    for(i=0; i<N; i++)  
        for(j=0; j<N; j++)  
            t[i][j] = '_';  
}
```

Definição  
(Código da função)

Percorrer  
uma matriz

Manipulação  
de uma tabela

# Função escreve\_tab()

---

```
void escreve_tab(char t[][N])
{
    int i, j;

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

# Função escolhe\_jogada()

Passagem por  
referência

Passagem  
por valor

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

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

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

Operadores

# Função linha()

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

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

Manipulação  
de uma tabela

Devolver um  
resultado

# Funções coluna() e diag()

---

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

Completar

```
int diag(char t[][N])  
{  
    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. Permitir que os jogadores escolham a dimensão do tabuleiro (por exemplo, entre 3 e 5)
4. ...