



**Instituto Superior  
de Engenharia**

Politécnico de Coimbra

# **Relatório do trabalho prático**

Departamento de Engenharia Informática e de Sistemas  
Licenciatura em Engenharia Informática  
Ramo de Sistemas de Informação  
Unidade Curricular de Programação  
Isabel Castro 2018013160  
Coimbra, 11 de junho de 2021

## Índice

Introdução.....	2
Estruturas de dados.....	3
Estrutura jogadores .....	3
Estrutura jogadas.....	3
Estrutura tabuleiro .....	3
Ficheiros usados.....	4
Estruturas dinâmicas.....	4
Tomadas de decisão .....	5
Inicialização e jogo .....	5
Manual de utilização.....	6

## Introdução

Este trabalho foi realizado no âmbito da unidade curricular de Programação, orientada pelo professor Francisco Pereira, sendo o tema proposto a recriação do jogo do semáforo.

O programa implementado deve permitir a realização do jogo do semáforo. Este é um jogo de tabuleiro entre 2 pessoas que efetuam jogadas alternadas, até que uma delas vença ou que se verifique um empate, para além da lógica do jogo foi também pedido outras funcionalidades.

Na primeira fase, o programa confirma se há jogos anteriores para voltar a jogar ou então a criação de um novo, de seguinte, segue-se todo o processo do jogo. Para acabar o jogo têm que haver vitória de algum dos jogadores e termina o programa gravando um ficheiro texto com algumas informações.

## Estruturas de dados

Para este trabalho e sendo que não considero que seja necessário para a realização do mesmo uma grande divisão ou várias estruturas, optei por utilizar apenas 3.

### Estrutura jogadores

Esta estrutura é simples e nem recorre a memória dinâmica pois só há sempre a possibilidade de haver 2 jogadores e também os dados que estava a guardar nela são simples.

```
typedef struct jogador jogador, *pJogador;  
struct jogador {  
    char nome;  
    char jogada;  
    int nPedras, nAdicoes;  
};
```

### Estrutura jogadas

Nesta estrutura, guardei toda a informação necessária para manter um histórico de jogadas, devido à necessidade de mais tarde reaver o jogo e pela forma como criei a estrutura jogadores então tive que também guardar dados do jogador em cada jogada, assim como o tamanho do tabuleiro e a linha/coluna da peça.

Isto está em forma de lista ligada dupla, com ponteiro para o próximo nó assim como ponteiro para o anterior pois aquando da implementação da parte de mostrar no ecrã as sucessivas jogadas apercebi-me que era mais complicado fazer com uma lista ligada simples visto que a manutenção do código estava a ficar demorosa, com uso de muitas funções recursivas e o tabuleiro não estava a mostrar todas as jogadas anteriores.

```
typedef struct jogadas noJogada, *pJogada;  
struct jogadas {  
    jogador x;  
    int linha, coluna;  
    int tamTab[2];  
    pJogada prox;  
    pJogada prev;  
};
```

### Estrutura tabuleiro

O tabuleiro é feito com uma matriz alocada dinamicamente e a sua primeira presença no programa está definido como:

```
char **tabuleiro=NULL
```

Visto que era dado a opção de fazer uma tabela dinâmica não utilizei estrutura para este caso, no entanto, penso que teria sido mais vantajoso o uso de uma lista ligada, pois tive a necessidade de criar matrizes auxiliares.

## Ficheiros usados

Para o desenvolvimento do trabalho prático e para uma melhor estruturação do código e facilidade na sua manipulação, acabei por usar 5 ficheiros \*.c e 4 ficheiros \*.h, sem conta com o código já fornecido inicialmente.

### Main.c

Trata do controlo do fluxo inicial do código e do jogo. Todas as funções pertencem a outros ficheiros.

### Jogo.c

Contém toda a lógica necessária ao fluxo e término do jogo, assim como todas as opções escolhidas por cada jogador e determina as ações correspondentes, invoca, também, algumas funções a outros ficheiros.

### Tabuleiro.c

Trata da criação da matriz dinâmica, alterações no seu tamanho etc

### Histórico.c

Tem todas as funções relativas à parte da manutenção da sucessão de jogadas.

### Ficheiros.c

É o ficheiro com menos funções, mas serve para separar tudo que envolva a criação/leitura de ficheiros de texto ou binários.

## Estruturas dinâmicas

Para o processamento da recuperação de x jogadas e sendo obrigatória a utilização de listas ligadas, usei uma lista ligada dupla pois aquando da necessidade de imprimir num mesmo tabuleiro todas as jogadas anteriores à jogada x ocorria o problema de o tabuleiro poder mudar o seu tamanho então tinha que ter alguma forma de verificar se naquela jogada teria que mostrar um tabuleiro maior ou não. Assim, o uso de um ponteiro para a estrutura anterior permitiu colmatar estas dificuldades com as condições.

Para a recuperação do jogo anterior também usei a mesma estrutura.



## Tomadas de decisão

Como já referi, optei por simplificar a minha forma de armazenar dados do jogador pois à partida não é necessário mais que 2 jogadores e o uso desta tornou mais fácil a sua integração posteriormente na lista ligada. No entanto, tenho noção que era possível de tratar esta estrutura de forma mais eficiente que permitisse adaptar a outra situação.

## Inicialização e jogo

Optei por permitir que aquando do término do jogo e depois de guardar um relatório sobre as jogadas que o jogador tenha a oportunidade de escolher se quer jogar de novo, ler regras ou sair. Sendo que é com a opção sair que termina oficialmente o ciclo e permite a libertação de memória dinâmica.

A nível de escolha de jogadas, optei por usar valores ASCII para fazer validações e também colocar a obrigatoriedade de usar os caracteres tal como exposto no ecrã. Caso não seja respeitado volta a pedir até que esteja correto. Para a primeira jogada de ambos os jogadores estes também só podem colocar uma peça verde pois o tabuleiro está vazio, a única exceção a esta regra é quando se retoma um jogo anterior.

O jogador automático é todo feito com recurso a funções rand que devolve valores que correspondem a cada jogada, o computador pode fazer tudo como um jogador normal exceto a parte de interromper o jogo e pedir para visualizar jogadas.

Por fim, a nível da sucessão de estados optei pela criação de um tabuleiro auxiliar para imprimir no ecrã a informação e este é libertado a sua memória depois da sua utilização. Ao nível do ficheiro de texto, apenas guardei a quantidade de jogadas, se jogou com computador ou não, e uma lista completa de todas as jogadas. Ponderei colocar a informação de quem ganhou o jogo, mas como o jogo só termina e grava o ficheiro se alguém ganhar então a última jogada é o vencedor, também é fácil de saber porque tem a identificação da ordem das jogadas lá.

## Manual de utilização

A minha interface com o utilizador inicia-se com uma subdivisão de tópicos, onde o utilizador selecciona se quer iniciar um jogo ou ler as regras.

```
-----  
-                                     -  
-           -----  
-           JOGO DO SEMAFORO  
-           -----  
-           1. Iniciar jogo  
-           2. Regras  
-           3. Sair  
-  
-----  
>>
```

Caso exista um jogo há a opção de o continuar, senão o utilizador avança e pode escolher se quer jogar contra o computador ou com outra pessoa.

```
-----  
-                                     -  
-           -----  
-           JOGO DO SEMAFORO  
-           -----  
-           1. Iniciar jogo  
-           2. Regras  
-           3. Sair  
-  
-----  
>>1  
Existe um jogo para recuperar!  
-----  
Pretende recuperar o jogo?  
Sim(S) / Nao(N)  
-----  
>>
```

```

-----
Jogo vai começar!
-----

-----
A criar tabuleiro...
-----
      x0      x1      x2
-----
x0    | -  | -  | -  |
-----
x1    | -  | -  | -  |
-----
x2    | -  | -  | -  |
-----

Jogador A e a sua vez!
Pretende visualizar o estado do tabuleiro? (S)/(N):
>>

```

De seguida, o jogo inicia-se e basta seguir as instruções na consola. Caso seja o primeiro turno e o primeiro jogador então vai aparecer um aviso que não há jogadas anteriores pra ver, isto se escolher a opção de visualizar!

```

      x0      x1      x2
-----
x0    | G  | -  | -  |
-----
x1    | -  | -  | -  |
-----
x2    | G  | -  | -  |
-----

Jogador A e a sua vez!
Pretende visualizar o estado do tabuleiro? (S)/(N):
>>N

-----
Ja pode fazer as seguintes jogadas
-----
(A)- Colocar uma peça Verde numa célula vazia
(B)- Trocar uma peça Verde que esteja colocada no tabuleiro por uma peça Amarela
(C)- Trocar uma peça Amarela que esteja colocada no tabuleiro por uma peça Vermelha
(D)- Colocar uma pedra numa célula vazia.(Usado 0 vezes)
(E)- Adicionar uma linha ou uma coluna ao final do tabuleiro.(Usado 0 vezes)
(F)- Interromper o jogo e guardar para mais tarde!

>>

```



```

      x0      x1      x2
-----
x0 | G | - | - |
-----
x1 | G | - | - |
-----
x2 | G | - | - |
-----

-----
Coluna 0 completa!
-----

-----
Jogador A e o vencedor!
Jogada vencedora na posicao: [1][0]
-----

```

Depois de detetado o vencedor, indica quem venceu e mostra uma lista de todas as jogadas. Termina pedindo o nome de um ficheiro para guardar os dados e volta ao menu inicial para que o utilizador decida se quer jogar de novo ou sair.

```

-----
Voltando ao menu inicial....
-----

-----
Indique o nome do ficheiro(xxx.txt) para gravar as jogadas
-----
>> jogo1.txt

-----
Relatorio guardado!
-----

-----
-                                     -
-           -----                 -
-           JOGO DO SEMAFORO         -
-           -----                 -
-           1. Iniciar jogo          -
-           2. Regras                 -
-           3. Sair                   -
-----
>>

```