

UNIPÊ  
CENTRO UNIVERSITÁRIO DE JOÃO PESSOA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO - BACHARELADO

FERRAMENTA DE APOIO AO APRENDIZADO DO PROJETO  
DE XADREZ PROGRAMADO EM C

JOÃO PESSOA

2021

# FERRAMENTA DE APOIO AO APRENDIZADO DO PROJETO DE XADREZ PROGRAMADO EM C

Por

**Bianca Schiochet Tiepolo** – RGM: 26140489

**Célio Santos da Silva** – RGM: 27219551

**Kleber Herivelto de Lima Cavalcanti Filho** - RGM: 25365037

**Matheus Fialho Barbosa** - RGM: 27296377

**Matheus Henrique Meireles de Almeida Silva** – RGM: 26565901

Novembro de 2021

## **Resumo**

O presente relatório apresenta o desenvolvimento de um algoritmo de xadrez com funcionamento automatizado com um programa para o tabuleiro. O objetivo é permitir que um jogador possa jogar no tabuleiro contra outro jogador, através do software. Deste modo, as peças do tabuleiro devem se mover sem necessidade de intervenção do primeiro jogador, para que o outro jogador possa executar o segundo movimento. Foi utilizado funções da programação C como estação base, a qual recebe e envia informações para o algoritmo.

## Sumário

1.1 Objetivo .....	1
--------------------	---

## Xadrez

2.1 Estrutura e disposição das peças .....	2
2.2 Movimentação e ataque das peças .....	3
2.3 Promoção .....	4
2.4 Captura de peças .....	4
2.5 Final do jogo .....	4

## Elaboração do jogo de xadrez

3.1 Representação do tabuleiro .....	5
3.2 Controle de jogadas .....	7
3.3 Algoritmo em C .....	8
3.4 Gerador de movimentos .....	9
3.5 Conclusão .....	10

## Extras

4.1 Ranking .....	11
4.2 Menu .....	12

# 1 Introdução

Gerar um jogo de xadrez entre dois usuários que por opção possa resoluto se a pessoa vai jogar com outro usuário pelo mesmo software, com isso simular um jogo de xadrez e assim tomar resoluções a partir das funções em C.

O xadrez é um dos jogos de tabuleiro mais conhecidos no mundo. Existem versões eletrônicas do jogo, as quais não necessitam de dois jogadores para que o jogo aconteça. As versões eletrônicas também permitem o jogo remoto entre dois jogadores pela internet e o jogo entre usuário e máquina que utiliza de inteligência artificial para promover as jogadas.

## 1.1 Objetivo

Este trabalho teve por objetivo dar ênfase na aplicação de um algoritmo dentro de um jogo de xadrez. Também será apresentado um jogo de xadrez, o qual foi desenvolvido, onde dois jogadores tem o movimento de suas peças controladas pelos mesmos (Sendo gerados e avaliados por linguagem de programação C), simulando assim a disputa entre um jogo de xadrez.

## 2 Xadrez

O jogo de xadrez é jogado entre dois adversários que movimentam peças em um tabuleiro. O objetivo do jogo é colocar o rei do oponente sob ataque de um modo que não seja possível escapar deste ataque.

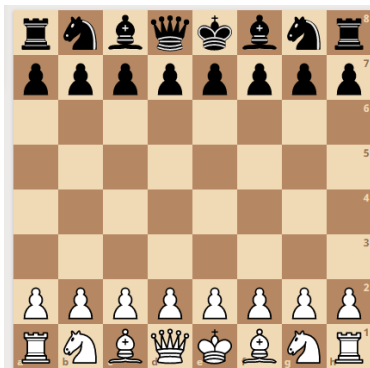
A origem do jogo ainda não foi confirmada, mas estima-se que o jogo se originou na Ásia, mais especificamente na Índia por volta do século VI d.C. O que se sabe é o jogo surgiu com outras formas e regras. A forma atual do jogo se iniciou no século XV no sudoeste da Europa, sofrendo algumas alterações até chegar ao modo que conhecemos hoje.

Segundo Carolus (2006), o xadrez é um jogo baseado em turnos, ou seja, cada jogador pode realizar suas ações separadamente, enquanto o outro jogador aguarda. O xadrez é um jogo onde toda a informação está sendo vista por ambos os jogadores. Essa característica torna-o um bom jogo para aplicação de técnicas de inteligência artificial. Realça-se que ele é um jogo determinístico, não dependendo de nenhum fator aleatório como a sorte.

### 2.1 Estrutura e disposição das peças

O jogo ocorre em um tabuleiro quadriculado, uma matriz 8 X 8 com 64 casas de mesmo tamanho, possuindo cores alternadas em cada casa. Quanto às peças, cada jogador tem a sua disposição 16 peças de uma determinada cor (geralmente as peças são formadas por um conjunto de peças pretas e outro composto por peças brancas).

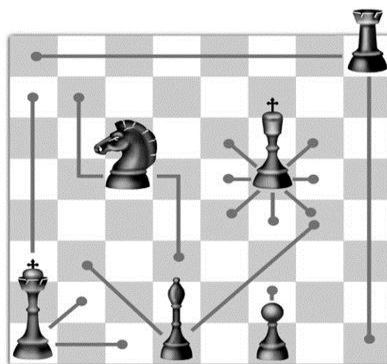
Cada jogador possui a sua disposição as seguintes peças: 8 peões, 1 rei, 2 bispos, 2 cavalos e 1 rainha. A Figura 2.1 apresenta a disposição inicial das peças no tabuleiro.



Fonte: <https://dialecticalnelson.medium.com/guia-b%C3%A1sico-de-xadrez-m%C3%B3dulo-1-4a7c5a2fd88d>

## 2.2 Movimentação e ataque das peças

- Peão (pawn): se desloca somente para frente, uma casa por jogada (frisando que no primeiro movimento de cada peão, o mesmo pode se deslocar duas casas a frente), onde seu ataque só pode ser realizado caso a peça adversário esteja uma casa a frente e uma casa a direita ou uma casa a frente e a esquerda. Diferente das demais peças, o peão é a única peça, a qual ataca de modo diferente ao qual se desloca;
- Rei (king): ele pode se deslocar e atacar em qualquer sentido no tabuleiro, deslocando-se somente uma única casa em cada movimento, desde que este não o coloque sob ataque. Realça-se que sua captura resulta no fim do jogo;
- Rainha (queen): também chamada de dama, esta peça é considerada a peça mais poderosa do jogo, pois pode se mover/atacar em posição horizontal, vertical ou diagonal. A quantidade de casas ficará a critério do jogador;
- Torre (rook): esta peça se move na horizontal ou vertical, onde a quantidade fica a critério do jogador;
- Bispo (bishop): cada bispo move-se em diagonal, nas casas da mesma cor em que a peça se encontrava no início da partida. Do mesmo modo que a torre e rainha, ela pode se mover quantas casas o jogador desejar;
- Cavalo (knight): esta peça move-se em "L", ou seja, duas peças horizontalmente e uma verticalmente ou vice-versa.



Fonte: <https://www.palpitedigital.com.br/2013/05/08/como-jogar-xadrez-quais-as-regras/>

É importante mencionar que nenhuma peça, exceto o cavalo, pode pular outra peça em seu caminho. Realça-se também que as peças brancas sempre devem realizar o primeiro movimento da partida.

## 2.3 Promoção

Apesar de ser a única peça que não pode recuar de posição, o peão é a única peça que pode ser promovida. Se um peão conseguir avançar até a última fileira do outro lado do tabuleiro, ele é promovido e pode escolher qual peça deseja se tornar: rainha, torre, bispo ou cavalo. Normalmente, a peça escolhida é a rainha (por ser a peça mais poderosa). Assim, é possível que um time tenha até 9 rainhas disponíveis dentro de uma mesma partida.

## 2.4 Captura de peças

Dentro do jogo, é possível capturar (tomar) peças do adversário. Se uma peça inimiga estiver no raio de ataque de uma determinada peça, aquela pode ser capturada. Quando uma peça é capturada, esta é retirada do jogo e a peça que a atacou assume sua posição da casa.

## 2.5 Final do jogo

Conforme dito anteriormente, o jogo de xadrez pode acabar de três maneiras: derrota, empate ou vitória.

Uma vitória ou derrota pode ocorrer de em duas formas: quando um jogador abandonar a partida ou quando ocorrer um xeque mate.

Para entendermos o que é o xeque mate, precisamos primeiro entender o que é o xeque. O xeque é um estado do jogo, o qual indica que há uma ameaça de captura ao Rei. Já o xeque mate, se refere ao estado em que não há nenhum movimento capaz de fazer com que o rei saia da posição de xeque, resultando no fim do jogo. É importante frisar que se um rei está em xeque, o movimento das peças do mesmo time que esse rei só é válido se este tirá-lo do estado de xeque.

O empate pode ocorrer em diversas situações:

- Quando ambos os jogadores decidem encerrar a partida;
- Por insuficiência de peças: quando não há peças suficientes para garantir o xeque mate;
- Por ‘afogamento’ do rei: é uma situação em que o rei está encurralado em uma posição, a qual não consegue sair, pois as opções são inválidas ou resultam em xeque mate para o adversário. É importante frisar que essa situação é diferente do xeque mate, pois o rei não está em xeque;
- Por regra dos 50 movimentos: estado do jogo em que o empate pode ser solicitado, devido ao fato dos 50 últimos movimentos terem sido realizados sem que nenhuma peça fosse capturada e que houvesse movimento de qualquer peão;

- Por regra das três posições: situação em que um jogador pode solicitar empate quando uma posição for repetida três vezes durante a partida com o mesmo jogador (importante frisar que possibilidade de execução de movimentos especiais é considerada para validar esse tipo de empate);

## 3 Elaboração do jogo de xadrez

Conforme proposto, um programa foi criado, cuja sua finalidade é representar um jogo de xadrez entre usuários (peças brancas), (peças pretas). O mesmo foi desenvolvido em linguagem C, voltado para a plataforma Windows, tendo sua interface apresentada no prompt de comando do Windows.

Esse jogo procura atender as propriedades básicas de um jogo de xadrez, como tabuleiro, peças, movimentação das mesmas, validação dos movimentos, propriedades de ataque e outras regras básicas.

Sales (2008) define o jogo de xadrez como um problema de busca com os componentes a seguir:

- **Estado inicial:** posições atuais das peças no tabuleiro e o jogador que fará o próximo movimento.
- **Operadores:** jogadas legais para em um determinado estado da partida.
- **Teste de término:** verifica se o estado corresponde ao fim do jogo.

### 3.1 Representação do tabuleiro

Na aplicação proposta, o posicionamento das peças dentro do tabuleiro foi feito através de uma matriz bidimensional com dimensões 8x8. Cada elemento dessa matriz possui um determinado valor e esse valor definirá se há uma peça naquela determinada posição. As vantagens dessa metodologia são a fácil implementação e facilidade em avaliar o valor das peças, sua desvantagem é a necessidade de um loop para percorrer todas as posições dessa matriz.



Veja na Figura 3.1 como foi elaborado a versão do xadrez em C.

```
char xadrez [8][8]= {
    { 'T', 'C', 'B', 'Q', 'R', 'B', 'C', 'T'},
    { 'P', 'P', 'P', 'P', 'P', 'P', 'P', 'P'},
    { ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '},
    { ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '},
    { ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '},
    { ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '},
    { 'P', 'P', 'P', 'P', 'P', 'P', 'P', 'P'},
    { 't', 'c', 'b', 'q', 'r', 'b', 'c', 't'}
};

//tabuleiro

int pintartabuleiro() {
    system("cls");
    int linha, coluna, r;
    for (linha=0; linha < 8 ; linha++){
        for (r = 0; r < 3; r++){
            for(coluna=0; coluna < 8 ; coluna++){
                char peca = xadrez[linha][coluna];
                char letra = ((linha + coluna) % 2 == 0) ? '\xB2':' '; //marca o centro do local
                letra = (r == 1 && peca != ' ') ? peca: letra ;
                if( (linha + coluna) % 2 == 0){ //pinta o local preto e branco
                    printf("\xB2\xB2%c\xB2\xB2", letra); //local onde fica a peça
                }else{
                    printf("  %c  ",letra);
                }
            }
            printf("\n");
        }
    }
}
```

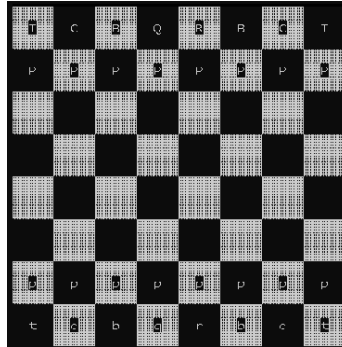
As principais funções utilizadas para compor o tabuleiro de xadrez no algoritmo foi o comando 'For' para definir a linha e coluna do tabuleiro e assim encaixar cada peça com o 'If', foi declarado a variável 'char peca' para elaborar cada peça dentro de algoritmo com os movimentos correspondentes.

Abaixo está disposta, a maneira como as peças foram definidas;

- Peças cujo dígito é letra maiuscula, representam as peças pretas.
- Peças cujo dígito é letra minúscula, representam as peças brancas
- Peças cujo valor é zero, representam posições não ocupadas no tabuleiro.
- No que se refere ao tipo de peça, a identificação ocorre do seguinte modo:
- p e P representam os peões brancos e os peões pretos, respectivamente.
- b e B representam os bispos brancos e bispos pretos, respectivamente.
- c e C representam os cavalos brancos e os cavalos pretos, respectivamente.
- t e T representam as torres brancas e as torres pretas, respectivamente.
- q e Q representam a rainha branca e a rainha preta, respectivamente.

- r e R representam o rei branco e o rei preto, respectivamente.

Logo, para acessar uma determinada peça, basta informar a posição x e a posição y em que ela se encontra dentro da matriz. A Figura 3.1 apresenta a formação inicial do tabuleiro no algoritmo.



### 3.3 Controle de jogadas

Após a construção do tabuleiro, o programa fica guardando que alguma jogada seja executada. Ele possui o controle de jogadas, o qual identifica o tipo de movimento que será realizado. Quando é turno do jogador, o programa aguarda qual peça ele deseja movimentar e qual posição ele deseja que assuma.

Para realizar uma jogada, o jogador precisa digitar no prompt a linha e coluna de origem da peça e avançar digitando a linha e coluna de destino da mesma.

```
int main() {
    int linhaorigem, linhadestino, colunaorigem, colunadestino; // Variáveis da linha e origem e da peça
    while(1) {
        pintartabuleiro();
        printf("informe a linha e coluna de origem: "); // Verifica a posição da peça
        scanf("%d %d", &linhaorigem, &colunaorigem);

        printf("\ninforme a linha e coluna de destino: "); // Verifica o destino da peça que irá ser movimentada
        scanf("%d %d", &linhadestino, &colunadestino);

        if (moverpeca(linhaorigem, colunaorigem, linhadestino, colunadestino) != 1) {
            printf("\ncoordenadas invalidas"); // Alerta do programa caso a linha e coluna seja inválida
            getch();
        }
        getch();
        return 0;
    }
}
```

Na aplicação desenvolvida, foi utilizado comando 'While(1)' para testar com base na condição, se for falsa ele pulará pra até depois dos comandos de while e se for verdadeira ele irá executar cada um dos comandos. Com isso, se o comando que o jogador digitar for incorreto para inicializar o programa, o mesmo vai alertar para o jogador que o comando não existe. Entre outras funções básicas do algoritmo em C que se especifica na imagem acima.

## 3.3 Algoritmo em C

Criada pelo cientista da computação Dennis Ritchie, em 1972, a linguagem C foi derivada de outras duas: a BCPL e a Algol 68. Embora tenha sido pensada com o propósito exclusivo de ser usada no desenvolvimento de uma nova versão do sistema operacional Unix, hoje é aplicada nos mais variados tipos de projeto.

Considerada uma linguagem de alto nível genérica, a C pode ser usada em diversos tipos de projeto, como a criação de aplicativos, sistemas operacionais, drivers, entre outros. Trata-se de uma linguagem estruturada que se tornou muito popular nos anos 80 — tanto que é difícil encontrar arquiteturas para as quais não existam compiladores para a C, o que garante o seu elevado nível de portabilidade.

Na linguagem C, existem diversas funções básicas que já estão definidas nas bibliotecas e que foram utilizadas para compor o algoritmo, como;

- `Printf(“”)`: Permite apresentar na tela qualquer tipo de dados, nesse caso, pedir para o jogador a posição das peças.
- `Scanf(“”)`: Grava e ler os dados que o usuário utilizou para a aplicação.
- `If`: Estrutura de decisão gerada para identificar qual peça foi selecionada e a forma que ela deve se comportar no tabuleiro, ao mesmo tempo somando linha e coluna.
- `For`: Estrutura de repetição utilizada para coluna e linhas para limitar o comando ate determinada casas.
- `System(cls)`: Faz parte da biblioteca “`stdlib. h`”, ela basicamente irá aplicar comandos no console/terminal.
- `Char, int`: Declarar os tipos de variáveis que serão informadas no algoritmo do tabuleiro.
- `While(1)`: Estrutura de repetição que deve se repetir apenas uma vez, caso o `if` seja incorreto, ele pulará os comandos para depois do `while`.

## 3.4 Gerador de movimentos

No algoritmo, é necessário avaliar se a movimentação da peça que a gente selecionou, corresponde com a movimentação das regras do xadrez para os dois jogadores. É possível realizar essa operação limitando o avanço das peças em linhas e colunas para que elas não fujam do padrão de jogabilidade.

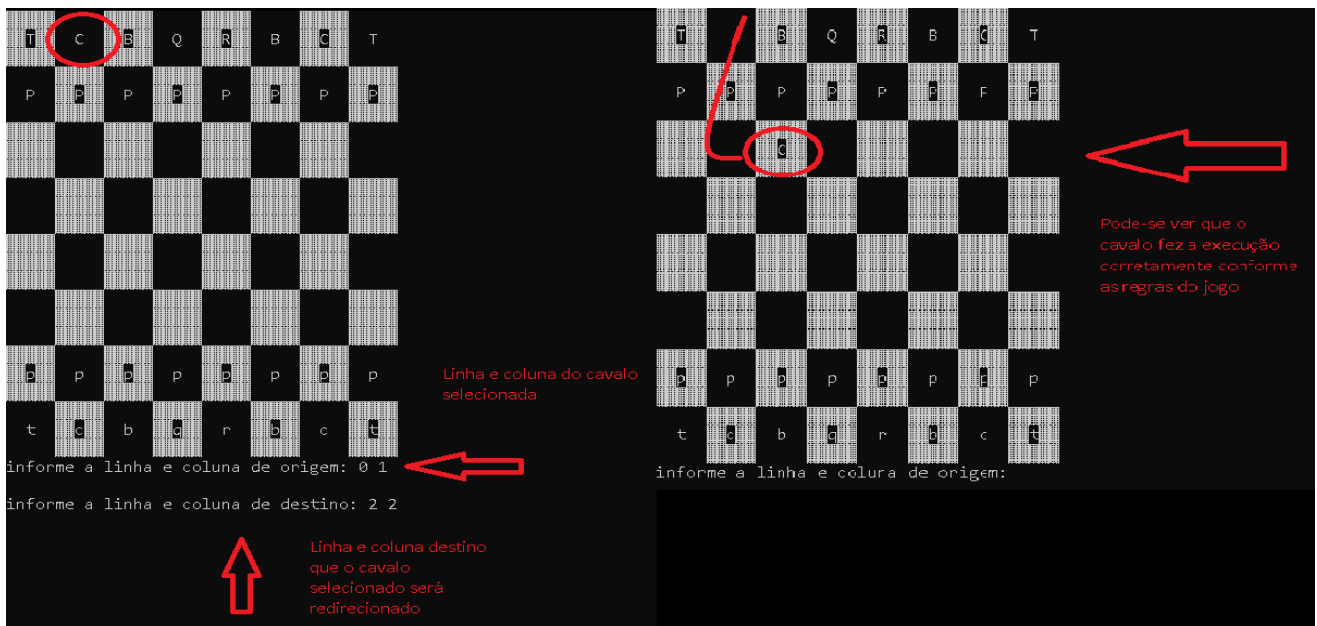
O código 3.4 abaixo apresenta um trecho do programa desenvolvido, qual retrata a parte da movimentação de cada peça.

```
int moverpeca(int linhaorigem, int colunaorigem, int linhadestino, int colunadestino) { // Movimento de peças
    int mover = 0;
    int peca = xadrez[linhaorigem][colunaorigem];
    int deslvertical = abs(linhadestino - linhaorigem);
    int deslhorizontal = abs(colunadestino - colunaorigem);
    if ( (linhaorigem >= 0 && linhaorigem < 8 && colunaorigem >= 0 && colunaorigem < 8) &&
        (linhadestino >= 0 && linhadestino < 8 && colunadestino >= 0 && colunadestino < 8) ) {

        if ( (peca == 'T' || peca == 't') && (deslvertical == 0 || deslhorizontal == 0) ) { //movimentação da torre
            mover = 1;
        }
        if ( (peca == 'B' || peca == 'b') && (deslvertical == deslhorizontal) ) { //movimentação do bispo
            mover = 1;
        }
        if ( (peca == 'C' || peca == 'c') && (deslvertical == 1 && deslhorizontal == 2)
            || (deslvertical == 2 && deslhorizontal == 1) ) { //movimentação do cavalo
            mover = 1;
        }
        if ( (peca == 'Q' || peca == 'q') && (deslvertical == deslhorizontal)
            || (deslvertical == 0 || deslhorizontal == 0) ) { //movimentação da rainha
            mover = 1;
        }
        if ( (peca == 'R' || peca == 'r') && (deslvertical >= 0 && deslvertical <= 1
            || (deslhorizontal >= 0 && deslhorizontal <= 1) ) { //movimentação do rei
            mover = 1;
        }
        if ( (peca == 'P') && (linhadestino - linhaorigem) == 1 && (deslhorizontal == 0) ) { // movimentação do peão
            mover = 1;
        }
        if ( (peca == 'p') && (linhadestino - linhaorigem) == -1 && (deslhorizontal == 0) ) {
            mover = 1;
        }
        if (mover) {
            xadrez[linhadestino][colunadestino] = xadrez[linhaorigem][colunaorigem];
            xadrez[linhaorigem][colunaorigem] = ' ';
        }
        return 1;
    }
}
```

Com isso foi utilizada a função 'If', se a mesma for verdadeira o algoritmo realizará a movimentação da determinada peça que ele selecionou naquela casa vista pela variável, dentro do algoritmo. Ou seja se o jogador selecionar a linha e coluna que abriga o cavalo, ela realizará a movimentação conforme o algoritmo diz para o cavalo e o mesmo para as outras peças do tabuleiro, sendo assim, seguindo as regras do jogo.

Veja a Figura 3.4 da aplicação sendo executada conforme o algoritmo em C;



Veja como exemplo o cavalo que se movimenta em 'L' da mesma conforme as regras do xadrez. Dessa maneira o jogo sendo rodado de forma otimizada e correta para a realização de jogadas contra outro usuário ou até mesmo máquina.

## 3.5 Conclusão

Utilizando o software, o usuário consegue jogar contra um adversário, sendo suas jogadas movimentadas pelo algoritmo com os devidos comandos realizados.

Devido a esse projeto inicial, pode-se ver que é possível aprimorar mais e tornar desse simples algoritmo entre dois usuários, algo mais realista até mesmo com uma inteligência artificial comandando as outras peças e jogadas. A realização desse projeto fez com que a equipe ganhasse mais experiência em comandos da programação C, que se abrange em diversas funções diferentes e formas de utilização.

Um trabalho futuro do projeto também pode permitir que campeonatos sejam realizados sem a presença dos dois competidores no mesmo local, assim como a integração de um temporizador para as jogadas no software e no tabuleiro do jogo.

A equipe também adquiriu mais conhecimento sobre rankings aderindo arquivos txt para ordenagem, forma de aplicar menus e diversas variedades de funções compostas.

## 4 Ranking

Os rankings servem para mostrar seu mérito e conquista através dos seus feitos, seja numa faculdade, escola ou jogo, funciona de uma maneira competitiva aonde mostra que pessoa fez maior pontuação naquela determinada coisa ou fez mais feitos que a outra.

No xadrez não é diferente, para separar as pessoas mais experientes daquelas menos experientes, para mostrar quem tem mais vitórias, empates ou ate mesmo derrotas.

Com isso foi implementado também no algoritmo um sistema de ranking que tem a funcionalidade de após uma partida entre usuário 1 e usuário 2, ele gravar e mostrar no menu e no final do jogo a pontuação entre eles, fazendo com quem ganhou fique com 1 ponto e quem perdeu fique com 0.

```
case 3: //Ranking
    rank = fopen("ranking.txt", "rb");
    fseek(rank, 0, SEEK_END);
    long fsize = ftell(rank);
    fseek(rank, 0, SEEK_SET);

    char *string = malloc(fsize + 1);
    fread(string, fsize, 1, rank);
    fclose(rank);

    printf("%s\n", string);
    break;
case 4: //Créditos
    printf("\n\n\n\nEsse jogo foi feito por:");
    break;
case 5: //Sair
    break;
default: {}
}
getch();
return 0 ;
}
```

O ranking foi composto por um arquivo .txt que é lido pelo o algoritmo no final do jogo e transmite na tela os jogador que venceu e o que faz essa parte é a função 'fopen' no algoritmo.

A função 'long' foi utilizada para calcular os valores e 'fread' para ler a pontuação correta que será aplicado para o vencedor.

## 4.1 Menu

Os menus querendo ou não, eles são sempre parte dos jogos e podem chamar a atenção ou não, podem ser mais funcionais ou pouco intuitivos, belos e chamativos ou apenas uma lista com duas ou três escolhas.

Nesse caso não é muito diferente, foi elaborado um pequeno menu com o algoritmo C para o nosso jogo de xadrez, com a função de ficar mais intuitivo para os dois usuários após abrirem o algoritmo.

O menu foi implementado com a função 'Switch' para a estrutura do menu e 'Case' para as opções de realização dos comandos no algoritmo.

```
int main() {
    int jogador = -1;
    int xequemate = 0;
    int escolha;      // escolher a opção no menu

    char P1[15];
    char P2[15];

    int linhaorigem, linhadestino, colunaorigem, colunadestino;

    printf("Escolha uma opção:\n1- 1 Pessoa\n2- 2 Pessoas\n3- Ranking\n");
    scanf("%d", &escolha);
}
```

Veja que primeiro o algoritmo declara as variáveis e coloca as opções e logo abaixo coloca as funções necessárias:

```
case 2: {
    printf("P1, Digite seu nome: ");
    scanf("%s", P1);
    printf("P2, Digite seu nome: ");
    scanf("%s", P2);

    while(jogador != 0) {
        pintartabuleiro();

        if(xeque == 1) {
            printf("Seu rei está sobre ataque, mova-o\n");
        }

        if(jogador == -1) {
            printf("P1, informe a linha e coluna de origem: ");
        }
        else if(jogador == 1) {
            printf("P2, informe a linha e coluna de origem: ");
        }
        scanf("%d %d", &linhaorigem, &colunaorigem);

        printf("\nPeça: %c, informe a linha e coluna de destino: ", xadrez[linhaorigem][colunaorigem]);
        scanf("%d %d", &linhadestino, &colunadestino);

        if (testexeque(jogador, linhaorigem, colunaorigem, linhadestino, colunadestino)) {
            printf("\ncoordenadas invalidas");
            getch();
        }
        else if(moverpeca(jogador, linhaorigem, colunaorigem, linhadestino, colunadestino) == 1) {
            pecasatacadas();
            jogador *= -1;
        }
        else {
            printf("\ncoordenadas invalidas\n");
            getch();
        }

        if(xeque == 1) {
            if(testexequemate(jogador)) {
                jogador *= -1;
                if(jogador == -1) {
                    printf("\nP1 venceu\n");
                }
            }
        }
    }
}
```

'Case' serve para levar o jogador a um determinado resultado de acordo com a resposta dele dentro do algoritmo, com isso informando as respostas através do algoritmo.

FONTES:

<https://blog.betrybe.com/linguagem-de-programacao/linguagem-c/>

<https://www.comp.uems.br/~PFC/PFC%20179.pdf>

<https://www.palpitedigital.com.br/tag/aprenda-a-jogar-xadrez/>

<https://dialecticalnelson.medium.com/guia-b%C3%A1sico-de-xadrez-m%C3%B3dulo-1-4a7c5a2fd88d>

-----fim-----