



**CENTRO UNIVERSITÁRIO DE JOÃO PESSOA - UNIPÊ**

Gabriel Victor Santos de Lima - RGM 30729696

João Wictor Ferreira - RGM 30304768

Samuel Primo Arcoverde - RGM 30137918

**Projeto Conjunto TDA e LDA**  
Jogo da Força

JOÃO PESSOA - PB  
2022

## **INTRODUÇÃO**

Acredita-se que o jogo da forca tenha sido criado em 1894 na Inglaterra, tendo seu primeiro registro oficial no livro “Traditional Games”.

O jogo contém a proposta que o jogador deverá acertar a palavra oculta corretamente, que pode ser definida pelo seu oponente ou gerada aleatoriamente pelo computador, tendo como dica o número de letras ligado ao tema escolhido.

A cada letra citada, o jogador pode ter uma das lacunas preenchidas, ou mais um traço desenhado em seu boneco, chegando assim ao total Máximo de 6 erros, que ao ser completo, perde instantaneamente o jogo.

## RESULTADO

O planejamento do algoritmo foi desenvolvido com base em pesquisas e leituras de documentação, a mecânica do jogo, se baseia numa entrada de dados por parte do usuário e num menu já programado.

Assim que executar o programa, é exibido um menu interativo, onde o usuário pode escolher entre as opções existentes.

- Iniciar o Jogo;
- Cadastrar novas palavras;
- Ver o Ranking;

Ao selecionar a opção jogar, o usuário é redirecionado a um input, onde o mesmo deverá cadastrar o seu nome de jogar, e do segundo usuário, que caso não tenha, pode ser o mesmo nome escolhido para si.

```
C:\Users\WINDOWS 10\Desktop\hangman-main\ave.exe
```

```
| |-----| | | | | | | | |
| |                                     | |  
| |                                     | |  
| |      _ _ _      _ _ _      _ _ _      _ | |  
| |      |   |      |   |      |   |      |   | |  
| |      | _ _ _ |   |   | _ _ _ /   |   |   | |  
| |      |   |   |   |   |   |       | _ _ _ | |  
| |      |   |   |   |   |   |       |       | |  
| |      |   |   |   |   |   |       |       | |  
| |      |   | _ _ _ |   |   | _ _ _     |   | |  
| |                                     | |  
| |-----| |
```

```
1 - Jogar  
2 - Ver Ranking  
3 - Cadastrar Plavras  
4 - Creditos  
5 - Sair  
Escolha uma das opcoes? (1-5)
```

## Menu da forza

[illegible]

imput do nome dos jogadores

Logo em seguida será exibido a força e linhas para representar cada letra da palavra secreta, deve ser deixado em adendo que a palavra é gerada aleatoriamente do banco de dados “palavras.txt” existente na pasta local do projeto

[illegible]

Assim é feito o início das partidas, sendo um jogador por vez. A cada letra errada, o boneco ganha uma nova parte do corpo e a vez é passada ao próximo jogador. O primeiro a concluir todas as letras da palavra, vence. Já o primeiro que completar a força ou errar 6 vezes perde o jogo.

```

Voce errou: a palavra NAO tem a letra S

Puxa, samuel foi enforcado!

O vencedor e leonardo
A palavra era **CALCINHA**

  .==.==.=.
  \::  7/
 | (|:: |) |
  -|:: |
  .|:: |
    \:: |
      \::
        ) (
      .-.-.
      -----

```

Mas, caso o jogador opte por alguma das outras opções do menu, ele irá se deparar com sua respectiva tela.

A opção de cadastrar novas palavras, conta com um input que recebe as palavras digitadas e armazena no arquivo “palavras.txt” que exerce o papel de banco de dados no projeto.

```

||=====||
1 - Jogar
2 - Ver Ranking
3 - Cadastrar Plavras
4 - Creditos
5 - Sair
Escolha uma das opcoes? (1-5)3

Voce deseja adicionar uma nova palavra no jogo (S/N)?S
Digite a nova palavra, em letras maiusculas: LAGOSTA

```

(print tela de cadastro das palavras)

Caso opte por ver o ranking, será exibido a tela com os nomes dos jogadores cadastrados anteriormente, se o usuário ainda não tiver cadastrado os novos nomes, às variáveis irão conter seu nome padrão de "jogador 1" e "jogador 2" com seus respectivos pontos.

```

1 - Jogar
2 - Ver Ranking
3 - Cadastrar Plavras
4 - Creditos
5 - Sair
Escolha uma das opcoes? (1-5)2

=====RANKING=====

LEONARDO = 1

=====

Deseja voltar ao menu? (S/N)

```

(print da tela de raking)

## APÊNDICE

Arquivos extras, que devem constar na pasta local da aplicação, os mesmos já vem como padrão na hora do download ou git clone do projeto, mas, conferir antes de rodar localmente.

- creditos.txt;
- forca.c;
- forca.h;
- palavras.txt;
- ranking.txt;
- README.md;

Código do projeto. (main)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include "forca.h"
#include <locale.h>
#define TAMANHO_PALAVRA 20
#define MAX_LEN 200
#define BUFFER_SIZE 1000

char palavrasecreta[TAMANHO_PALAVRA];
char chutes[26];

int chutesdados = 0;
int stop = 0;
int vez = 1;

typedef struct Player {
    int points;
    char name[50];
} player;

player p1;
player p2;
player vencedor;
player perdedor;

int acabou = 0;
```

```

int letraexiste(char letra) {
    for(int j = 0; j < strlen(palavrasecreta); j++) {
        if(letra == palavrasecreta[j]) {
            return 1;
        }
    }
    return 0;
}

```

```

int chuteserrados() {
    int erros = 0;

    for(int i = 0; i < chutesdados; i++) {
        if(!letraexiste(chutes[i])) {
            erros++;
        }
    }
    return erros;
}

```

```

int enforcou() {
    return chuteserrados() >= 5;
}

```

```

int ganhou() {
    for(int i = 0; i < strlen(palavrasecreta); i++) {
        if(!jachutou(palavrasecreta[i])) {
            return 0;
        }
    }
    return 1;
}

```

```

int validaNomes(){
    for(int i = 0; i<strlen(p1.name);i++){
        if(isalpha(p1.name[i]) == 0){
            printf("\nOs nomes nao podem conter numeros nem caracteres especiais!\n");
            return 0;
        }
    }
    for(int i = 0; i<strlen(p2.name);i++){
        if(isalpha(p2.name[i]) == 0){

```

```

        printf("\nOs nome nao podem conter numeros nem caracteres especiais!\n");
        return 0;
    }
}
return 1;
}

void abertura() {
    int valid = 0;
    while(valid == 0){
        printf("\n/*****\n");
        printf("\n/ Jogo da Forca *\n");
        printf("\n/*****\n");

        printf("\nDigite o nome do primeiro jogador:");
        scanf("%s", &p1.name);
        printf("\nDigite o nome do segundo jogador:");
        scanf("%s", &p2.name);
        p1.points = 0;
        p2.points = 0;
        valid = validaNomes();
    }
}

void trocarVez(){
    if (vez==1){
        vez=2;
    }
    else{
        vez=1;
    }
}

void chuta() {
    char chute;
    printf("Qual letra? ");
    scanf("%s", &chute);

    if(letraexiste(chute)) {
        printf("Voce acertou: a palavra tem a letra %c\n\n", chute);
        if (vez == 1){
            p1.points++;
        }
        else{
            p2.points++;
        }
    }
}

```



```

    }
} else {
    printf("\nVoce errou: a palavra NAO tem a letra %c\n\n", chute);
}

chutes[chutesdados] = chute;
chutesdados++;
trocarVez();
}

int jachutou(char letra) {
    int achou = 0;
    for(int j = 0; j < chutesdados; j++) {
        if(chutes[j] == letra) {
            achou = 1;
            break;
        }
    }
    return achou;
}

void desenhaforca() {

    int erros = chuteserrados();

    printf("Veza do jogador %s\n", (vez==1? p1.name : p2.name));
    printf(" _____ \n");
    printf(" |/      | \n");
    printf(" |      %c%c%c \n", (erros>=1?' ':'_'), (erros>=1?' ':'_'), (erros>=1?' ':' '));
    printf(" |      %c%c%c \n", (erros>=2?'\\':' '), (erros>=2?'|':' '), (erros>=2?'/':' '));
    printf(" |      %c \n", (erros>=3?'|':' '));
    printf(" |      %c %c \n", (erros>=4?'/':' '), (erros>=4?'\\':' '));
    printf(" |      \n");
    printf(" _|_____ \n");
    printf("\n\n");

    for(int i = 0; i < strlen(palavrasecreta); i++) {

        if(jachutou(palavrasecreta[i])) {
            printf("%c ", palavrasecreta[i]);
        } else {
            printf("_ ");
        }
    }
}

```

```
    }  
    printf("\n");  
}
```

```
void escolhepalavra() {  
    char line[1024];  
    int count = 0;  
  
    FILE* f;  
  
    f = fopen("palavras.txt", "r");  
    if(f == 0) {  
        printf("Banco de dados de palavras nao disponivel\n\n");  
        exit(1);  
    }  
  
    while (fgets(line , sizeof(line) , f )!= NULL)  
    {  
        count++;  
  
    }  
  
    srand(time(NULL));  
    int r = rand() % count + 1;  
    count = 0;  
  
    rewind(f);  
    while (fgets(line , sizeof(line) , f )!= NULL)  
    {  
        count++;  
        if(count == r){  
            int size = strlen(line);  
            strcat(palavrasecreta, line);  
            palavrasecreta[size - 1] = NULL;  
            break;  
        }  
  
    }  
  
    fclose(f);  
}
```

```
void adicionapalavra() {
```

```

char quer;

printf("\nVoce deseja adicionar uma nova palavra no jogo (S/N)?");
scanf(" %c", &quer);

if(quer == 'S') {
    char novapalavra[TAMANHO_PALAVRA];

    printf("Digite a nova palavra, em letras maiusculas: ");
    scanf("%s", novapalavra);

    FILE* f;

    f = fopen("palavras.txt", "r+");
    if(f == 0) {
        printf("Banco de dados de palavras nao disponÃvel\n\n");
        exit(1);
    }

    fprintf(f, "%s\n", novapalavra);

    fclose(f);
}

void continuar(){
    char cont;
    printf("\nDeseja voltar ao menu? (S/N)");
    scanf("%s", &cont);
    if (cont == 'S' || cont == 's'){
        stop = 0;
    }
    else{
        stop = 1;
    }
}

void checarGanhadores(){
    if (p1.points > p2.points ){
        vencedor = p1;
        perdedor = p2;
        acabou = 1;
    }
    else if(p1.points == p2.points){

```

```

        printf("O jogo terminou em empate!");
        acabou = 2;
    }
    else{
        vencedor = p2;
        perdedor = p1;
        acabou = 1;
    }
}

int jogo(){
    abertura();
    escolhepalavra();

    do {
        desenhaforca();
        chuta();
    } while (!ganhou() && !enforcou());

    if(ganhou()) {
        checarGanhadores();

        printf("\nO vencedor e %s\n", vencedor.name);

        printf("\nParabens, voce ganhou!\n\n");

        printf("
_____ \n");
        printf(" '._==_==_._.' \n");
        printf(" .-\\:  /-. \n");
        printf(" | (|:.  ) | \n");
        printf(" '-|:.  |-' \n");
        printf("  \\::  / \n");
        printf("   ':. .' \n");
        printf("   ) ( \n");
        printf("  _.' '._ \n");
        printf("  '-----' \n\n");

    } else {
        checarGanhadores();

        printf("\nPuxa, %s foi enforcado!\n", perdedor.name);

        printf("\nO vencedor e %s\n", vencedor.name);

```



```

printf("\n||=====
==||\n");
    printf("\n1 - Jogar");
    printf("\n2 - Ver Ranking");
    printf("\n3 - Cadastrar Plavras");
    printf("\n4 - Creditos");
    printf("\n5 - Sair");
    printf("\nEscolha uma das opcoes? (1-5)");
    scanf("%d", &option);

    return option;
}

void print_image(FILE *fptr)
{
    char read_string[MAX_LEN];

    while(fgets(read_string,sizeof(read_string),fptr) != NULL)
        printf("%s",read_string);
}

int adicionarRanking(char name[50], int points){
    char line[1024];
    char s1[10];
    int count = 0;
    int countTmp = 0;
    int exist = 0;

    FILE *file;
    FILE * fTemp;

    file = fopen("ranking.txt", "r+");
    while (fgets(line , sizeof(line) , file )!= NULL)
    {
        count++;
        if (strstr(line , name)!= NULL){
            exist = 1;

            for(int i = 0; i<strlen(line); i++){
                if(isdigit(line[i]) == 1){
                    for(int a = 0; a<sizeof(s1); a++){
                        if (isdigit(s1[a]) == 0){
                            s1[a] = line[i];

```

```

        break;
    }
}
}
break;
}
}
if (exist == 1){
    int i = atoi(s1);
    char s2[10];
    fTemp = fopen("replace.tmp", "w");
    points += i;
    sprintf(s2, "%d", points);
    strcat(name, " = ");
    strcat(name, s2);
    strcat(name, "\n");

    rewind(file);
    while ((fgets(line, sizeof(line), file)) != NULL)
    {
        countTmp++;
        if (countTmp == count){
            fputs(name, fTemp);
        }
        else{
            fputs(line, fTemp);
        }
    }

    fclose(file);
    fclose(fTemp);

    remove("ranking.txt");
    rename("replace.tmp", "ranking.txt");
}
else{
    fprintf(file, "%s = %d \n", name, points);
    fclose(file);
}

}

void verRanking(){

```

```

int c;
FILE *file;
file = fopen("ranking.txt", "r");
printf("\n");
printf("====RANKING====\n");

if (file) {
    while ((c = getc(file)) != EOF){
        putchar(c);
    }

    fclose(file);
}
printf("\n=====\\n");
continuar();
}

void creditos(){
    char *filename = "creditos.txt";
    FILE *fptr = NULL;

    if((fptr = fopen(filename,"rw+")) == NULL)
    {
        fprintf(stderr,"Nao foi possivel encontrar o arquivo %s\\n",filename);

    }

    print_image(fptr);
    fclose(fptr);
}

int main() {
    setlocale(LC_ALL, "");

    while (stop == 0){
        int option = menu();
        switch (option)
        {
            case 1:
                if(acabou == 0){
                    jogo();
                    continue;
                }

```



```
        else{
            printf("\nO jogo ja acabou!, renincie o programa!");
            continue;
        }
    case 2:
        verRanking();

        continue;
    case 3:
        adicionapalavra();
        continue;
    case 4:
        creditos();
        continuar();
        continue;
    case 5:
        stop = 1;
    }
}

}
```