



CENTRO UNIVERSITARIO DE JOÃO PESSOA - UNIPÊ

CENTRO DE TECNOLOGIA

CIÊNCIA DA COMPUTAÇÃO – P2 NOITE

TÉCNICAS E LABORATÓRIO DE DESENVOLVIMENTO DE ALGORITMOS

JOGO DA FORÇA NA LINGUAGEM C

João Pessoa

2022

# SUMÁRIO

1. Lista de Participantes.....	1
2. Introdução.....	2
2.1 Descrição do Jogo.....	2
2.2 Regras.....	2
3. Resultados.....	3
3.3 Problemas e Soluções.....	3-4
3.4 Demonstrativo de Funcionalidades.....	5-6
4. Apêndice.....	7-21

## **LISTA DE PARTICIPANTES**

1. Júlio Pedro Santos Monteiro (RGM: 30199115)
2. Audvânio José Da silva Marinho Filho (RGM: 30196957)

# INTRODUÇÃO

## 1. Descrição do Jogo

O **jogo da forca** é um jogo em que o jogador tem que acertar qual é a palavra proposta, tendo como dica o número de letras da palavra. A cada letra errada, é desenhada uma parte do corpo do enforcado. O jogo termina com o acerto da palavra ou com o término do preenchimento das partes do corpo do enforcado.

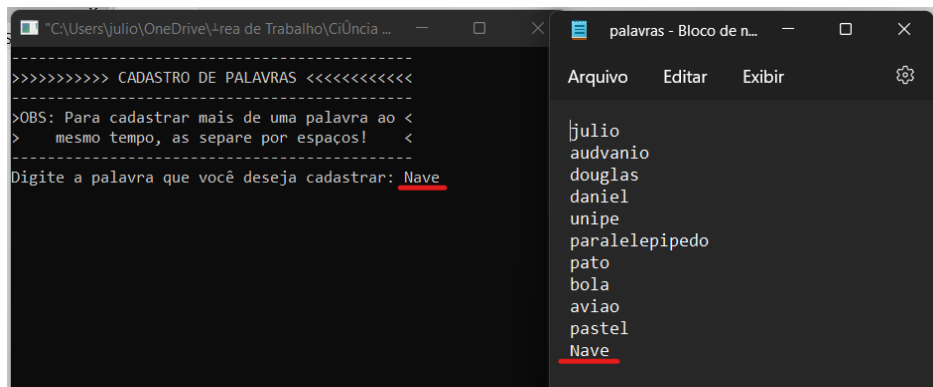
## 2. Regras do Jogo

- O jogador que tenta adivinhar a palavra deve ir dizendo as letras que podem existir na palavra. Cada letra que ele acerta é escrita no espaço correspondente.
- Caso a letra não exista nessa palavra, desenha-se uma parte do corpo a cada erro (iniciando pela cabeça, e depois seguindo para tronco, braços e pernas).
- O jogo é ganho se a palavra é adivinhada. Caso o jogador não descubra qual palavra é ele que perde.
- O limite de erros é de 6 tentativas (uma para cada parte do corpo a ser desenhada), ou seja, chutando seis letras que não estão na palavra secreta, o jogador perde automaticamente, enforcando o bonequinho.

# RESULTADOS

## 1. Problemas e Soluções

- Letras Maiúsculas poderiam causar problemas



O programa recebia e guardava normalmente as palavras, da forma que digitada. Porém, ao rodar o jogo, caso a palavra sorteada tivesse letras maiúsculas, poderia haver um pequeno erro na comparação da letra digitada com a palavra secreta. Então, utilizamos o comando “tolower” em duas situações: ao receber a palavra no cadastro (todas as letras serão guardadas minúsculas), e ao receber a letra digitada pelo jogador (a letra será conferida minúscula, mesmo se digitada maiúscula), assim, com essa padronização, não corre o risco da letra ser dada com errada, mesmo estando presente na palavra. Segue abaixo os códigos das soluções:

```
//Solicita e coleta a palavra que o usuário deseja cadastrar
printf("Digite a palavra que você deseja cadastrar: ");
scanf("%s", &palavra);

//Garantindo que todas as letras da palavra sejam cadastradas minusculas
for(int i = 0; i < strlen(palavra); i++){
    palavra[i] = tolower(palavra[i]);
}

//Salvar palavra no arquivo
abrir_arqPalavras_escrever(); //Abre o arquivo
fprintf(file, palavra); //Pega a variavel palavra e adiciona no arquivo
fprintf(file, "\n"); //Pula a linha no arquivo para quando for cadastrada uma nova palavra
fflush(file); // Limpando o buffer
fclose(file); // Fecha o arquivo.
```

---

*Garantindo que todas letras sejam minúsculas antes de gravar a palavra*

```
//Faz FOR para comparar a letra digitada em todos os indices da palavra secreta.
for (int i = 0; i < strlen(palavra_tela); i++){

    // Entra no if se o jogador acertou a letra.
    if (letra == tolower(palavra_secreta[i])){ //tolower para garantir que a letra digitada estará minúscula para comparação
        palavra_tela[i] = letra;
        errou = 0; //Atribuindo 0 à variavel errou, que como dito, será 0 em caso de acerto
        printf("Pontuação: %d", r.pontuacao);
    }
}
```

---

*Garantindo que a letra do chute do jogador seja minúscula*



Uma funcionalidade que implementamos nessa página, é a possibilidade de cadastro de mais de uma palavra ao mesmo tempo, sem necessidade de ficar retornando à página, basta digitar todas as palavras que deseja cadastrar, as separando por um espaço.





## APÊNDICE

### Código do Code Blocks:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <locale.h>
#include <time.h>
```

```
//Variável para sempre armazenar o arquivo aberto, para não precisar ficar criando toda vez
FILE *file;
```

```
//Corrigir acentuação (metodo encontrado no liguagemc.com.br)
void ortografia() {
    setlocale(LC_ALL, "Portuguese");
}
```

```
// Armazenando informações do ranking em Struct, como solicitado na documentação
struct ranking{
```

```
char nome[30];
```

```
int pontuacao;
```

```
} r;
```

```
//Menu com opções do game (cabeçalho)
```

```
void apresentacao(){
```

```
//Sem operações lógicas, apenas print do cabeçalho
```

```
printf("-----\n");  
    printf(">>>>>>>>>>> JOGO DA FORCA <<<<<<<<<<<< \n");  
printf("-----\n");  
  
printf("                OPÇÕES:                \n\n");
```



Consequência : Alternativa (para exibir a parte do "corpo" apenas com o numero de erros especifico para tal parte)

```
printf("\n | %c%c%c", (qtdeErros >= 2 ? '-' : ' '), (qtdeErros >= 3 ? '|' : ' '), (qtdeErros >= 4 ? '-' : ' '));
```

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

```
if (qtdeErros >= 6){ //Mensagem de derrota, exibida apenas ao atingir o limite de erros!
```

$$\left. \begin{array}{l} \{ \\ \} \end{array} \right\}$$

```
char *sorteiaPalavra(char nomeArquivo[], char escolha[]) {
```

```
int contador = 0, i;
```

```
abrir_arqPalavras_ler();
```

```
//Lê linha do arquivo
```

```
fgets(escolha, 30, file);
```

```
//Retira as quebras extras de linhas
```

```
for (i = 0; i < strlen(escolha); i++)
```

```
    if (escolha[i] == '\n')
```

```
        escolha[i] = '\0';
```

```
//Testa se a string não está vazia e armazena no vetor
```

```
    if (strcmp(escolha, "")) {
```

```
        strcpy(linha[contador], escolha);
```

```
        contador++;
```

```
    }
```

```
} while (!feof(file));
```

```
// Sorteia uma palavra pelo indice
```

```
srand(time(NULL));
```

```
i = rand() % contador + 1;
```

```
strcpy(escolha, linha[i]);
```

```
//Retorna uma palavra sorteada
```

```
return escolha;
```

```
}
```

```
//Cadastro de Palavras
```

```
void cadastrar_palavra(){
```

```
    char palavra[100];
```

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

```
        printf(">>>>>>>>>> CADASTRO DE PALAVRAS <<<<<<<<<< \n");
```

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

```
printf(">OBS: Para cadastrar mais de uma palavra ao <\n");  
printf(">  mesmo tempo, as separe por espaços!  < \n");  
printf("-----\n");
```

```
//Solicita e coleta a palavra que o usuário deseja cadastrar
```

```
printf("Digite a palavra que você deseja cadastrar: ");  
scanf("%s", &palavra);
```

```
//Garantindo que todas as letras da palavra sejam cadastradas minusculas
```

```
for(int i = 0; i < strlen(palavra); i++){  
palavra[i] = tolower(palavra[i]);  
}
```

```
//Salvar palavra no arquivo
```

```
abrir_arqPalavras_escrever(); //Abre o arquivo  
fprintf(file, palavra); //Pega a variavel palavra e adiciona no arquivo  
fprintf(file, "\n"); //Pula a linha no arquivo para quando for cadastrada uma nova palavra  
fflush(file); // Limpando o buffer  
fclose(file); // Fecha o arquivo.
```

```
//Limpa a tela
```

```
system("cls");
```

```
printf("Dados gravados, retornando ao menu principal...");
```

```
//Aguarda 2 segundos até abrir o menu novamente
```

```
sleep(2);
```

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

```
}
```

```
//Jogar
```

```
void jogar(){
```

```
    char palavra_secreta[30], palavra_tela[30], escolha[30];
```

```
    int erros = 0, correta = 0, pontuacao_final = 0, pontuacao_inicial = 100;
```

```
    char letra;
```

```
    //Buscando a palavra aleatoria no banco utilizando a função "palavra_secreta"
```

```
    strcpy(palavra_secreta, sorteiaPalavra("palavras.txt", escolha));
```

```
    //Solicitando nome do jogador
```

```
    printf("JOGADOR:\n");
```

```
    printf("Qual o seu nome? \n");
```

```
    // Limpando o buffer com fflush
```

```
    fflush(stdin);
```

```
    //Coletando o nome do jogador com fgets
```

```
    fgets(r.nome, 30, stdin);
```

```
    //Limpando a tela
```

```
    system("cls");
```

```
    //Dica de quantas letr
```

```
    printf("A palavra tem %lu letras/caracteres", strlen(palavra_secreta));
```

```
    //Copia a palavra secreta para a variável que vai aparecer na tela
```

```
    strcpy(palavra_tela, palavra_secreta);
```

```
    //Substitui a as letras da palavra por underline
```

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

```
        palavra_tela[i] = '_';
```

```
    }
```

```
while (1){ // imprime a forca sem erros (0 atribuido à variavel erros na declaração de variaveis)
e ja inicia os chutes das letras

forca(erros);

printf("\n A palavra secreta é: ");

//Printando os underlines, ou seja, os espaços das letras
for (int i = 0; i < strlen(palavra_tela); i++){
    printf("%c ", palavra_tela[i]);
}

//Solicitando e recebendo a letra digitada
printf("\n \n Digite a letra desejada: ");
scanf(" %c", &letra);

//Limpando a tela
system("cls");

int errou = 1; // Esta variável "errou" será 1 quando o jogador ERRAR e 0 quando ACERTAR.

//Laço FOR para comparar a letra digitada em todos os indices da palavra secreta.
for (int i = 0; i < strlen(palavra_tela); i++){

    // Entra no if se o jogador acertou a letra.
    if (letra == tolower(palavra_secreta[i])){ //tolower para garantir que a letrar digitada estará
minúscula para comparação
        palavra_tela[i] = letra;
        errou = 0; //Atribuindo 0 à váriavel errou, que como dito, será 0 em caso de acerto
        printf("Pontuação: %d", r.pontuacao);
    }
}

// Caso não entre no IF, a letra será errada, então o errou seguirá como 1, então, letra errada
```

```
// Caso o jogador use todas 6 chances e não complete a palavra, não pontua, então,
pontuação = 0
```

// Caso o jogador não cometa nenhum erro, sua pontuação permanecerá igual a inicial (100 pontos)

```
// Compara as palavras (palavra secreta x palavra tela),se forem iguais printa o acerto.
```

```
printf("\n\n-----\n");  
printf(">>>>>>>>>>>> VOCÊ GANHOU! <<<<<<<<<<<<<<< \n");  
printf(" Parabéns %s , você acertou a palavra secreta!|\n", r.nome);  
printf("-----\n");
```

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

}

```
break;
```



```
}
```

```
//Caso atinja o limite de erros, usa a função "forca" para printar que perdeu o jogo
```

```
if (erros == 6){
```

```
    forca(erros);
```

```
    break;
```

```
}
```

```
} // Registra os dados do participante no arquivo de ranking
```

```
    abrir_arqRanking_escrever(); //Abre o arquivo com a função criada
```

```
    fprintf(file, "Nome: %s", r.nome); //Registra nome
```

```
    fprintf(file, "Pontuação: %d\n", r.pontuacao); //Registra Pontuação
```

```
    fclose(file); // Fecha o arquivo.
```

```
}
```

```
void abrir_arqPalavras_escrever(){ // Essa função abre o arquivo "palavras.txt" para a gravação de  
novas palavras
```

```
    file = fopen("palavras.txt", "a"); // O "a" significa append, usado para adicionar dados a um  
arquivo já existente ou, se ele não existir, criar e incluir o texto.
```

```
if (file == NULL)
```

```
{
```

```
    printf("Erro na abertura do Arquivo\n");
```

```
    system("pause");
```

```
    exit(0);
```

```
}
```

```
}
```

```
void abrir_arqPalavras_ler(){ //Essa função abre o arquivo "palavras.txt" para leitura
```

```
    file = fopen("palavras.txt", "rt"); // O "rt" significa abertura para leitura do arquivo de texto
```

```
if (file == NULL)
{

    printf("erro na abertura do Arquivo\n");

    system("pause");

    exit(0);
}
}
```

void abrir\_arqRanking\_escrever(){ // Essa função abre o arquivo "ranking.txt" para a gravação dos usuarios e suas pontuações

file = fopen("ranking.txt", "a"); // O "a" significa append, usado para adicionar dados a um arquivo já existente ou, se ele não existir, criar e incluir o texto.

```
if (file == NULL)
{

    printf("erro na abertura do Arquivo\n");

    system("pause");

    exit(0);
}
}
```

void abrir\_arqRanking\_ler(){ //Essa função abre o arquivo "ranking.txt" para leitura

file = fopen("ranking.txt", "rt");

if (file == NULL)

{

printf("erro na abertura do Arquivo\n");

```
    exit(0);
}
```

}

}

```
int i;
```

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

```
i = 1; //Variável criada para ser utilizada com índice
```

```
pontuacao = fgets(linha_pontos, 100, file); //Segunda linha nome
```

nome -> pontos

}

}

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

```
int main(){
```

```
    ortografia(); //Função para corrigir acentuação
```

```
    int numeroMenuEscolhido = -1; //Iniciando a variável de escolha de opção (não foi iniciada com  
    0 pois essa opção está no menu
```

```
    do{
```

```
        int escolha; //variável que aparece nas páginas de ranking e créditos, dando opção de  
        retornar ao menu
```

```
        apresentacao(); //Função com o cabeçalho do jogo
```

```
        printf("Escolha uma das opções acima para seguir: \n");
```

```
        scanf("%d", &numeroMenuEscolhido);
```

```
        switch (numeroMenuEscolhido){
```

```
            case 1: //jogar
```

```
                system("cls");
```

```
                jogar();
```

```
                break;
```

```
            case 2: //cadastrar palavra
```

```
system("cls");
```

```
cadastrar_palavra();
```

```
break;
```

```
case 3: //ranking
```

```
system("cls");
```

```
exibirRanking();
```

```
//Opção de voltar ao menu ou encerrar o programa
```

```
printf("\n Deseja voltar ao menu principal?\n Digite '1' para sim e '2' para não\n");
```

```
fflush(stdin);
```

```
scanf("%d", &escolha);
```

```
if (escolha == 1){
```

```
    apresentacao();
```

```
    system("cls");
```

```
}else{
```

```
    exit(0);
```

```
}
```

```
break;;
```

```
case 4: //creditos
```

```
system("cls");
```

```
creditos();
```

```
//Opção de voltar ao menu ou encerrar o programa
printf("\n Deseja voltar ao menu principal?\n Digite '1' para sim e '2' para não\n");
fflush(stdin);
scanf("%d", &escolha);

if (escolha == 1){
    apresentacao();
    system("cls");
}else{
    exit(0);
}

break;

case 0: //encerrar

    system("cls");

    printf("Fim do Programa. \n");

    break;

default:

    system("cls");

    printf("Opção Inválida. \n");

    printf("Por favor escolha algum número informado no Menu. \n\n\n");

    break;
}
```

```
}while (numeroMenuEscolhido != 0);
```

```
return 0;
```

```
}
```