PROJETO CONJUNTO DAS DISCIPLINAS DE TÉCNICAS E LABORATÓRIO DE DESENVOLVIMENTO DE ALGORITMOS

PROFESSORES:

DANIEL DOS SANTOS BRANDÃO

DOUGLAS ANDRADE DE MENESES

ALUNOS:

MARIANA BARROS DA NÓBREGA CHAGAS **RGM:** 30698057

ANTÔNIO LUCAS DANTAS DE ABRANTES **RGM:** 30555922

JOGO DA PALAVRA

UNIPÊ
Centro Universitário
de João Pessoa

Sumário

1. INTRODUÇÃO

O *Jogo da Palavra*, desenvolvido pelos alunos Mariana Barros e Antônio Abrantes, teve como inspiração o jogo *Termo,* que se popularizou no Brasil e pode ser comparado ao Jogo da Forca.

O objetivo principal do *Termo* é fazer com que o usuário tente adivinhar uma palavra em até 6 tentativas. Enquanto o jogador não acertar, ele precisará arriscar palavras ao invés de dar palpites de letras. Além disso, uma nova palavra é sorteada a cada 24h e é a mesma para todos que estejam jogando naquele dia.

A grande diferença entre os jogos citados aqui se dá no seguinte ponto: no Jogo da Palavra, o usuário pode jogar quantas vezes quiser. Enquanto isso, o Termo só pode ser jogado uma única vez ao dia pelo mesmo jogador. Ademais, no Jogo da Palavra o termo a ser descoberto é sorteado cada vez que o usuário inicia o jogo.

Há outras características marcantes no Jogo da Palavra, entre elas cabe destacar a possibilidade do jogador realizar seu cadastro e o cadastro de novas palavras, além da visualização da pontuação em um ranking.

1.1. REGRAS E INFORMAÇÕES ADICIONAIS

O Jogo da Palavra possui algumas regras, são elas:

- O jogador precisa realizar seu cadastro antes de jogar a sua primeira partida;
- O jogador pode jogar quantas vezes quiser;
- O jogador só possui 5 chances para acertar a palavra sorteada;
- O jogador pode cadastrar novas palavras, mas elas devem conter 5 letras;
- O jogador só verá seu nome no Ranking se ele adivinhar o termo ao menos uma vez;
- Não há diferenciação de letras maiúsculas ou minúsculas, porém acentos não são considerados;
- Pode haver palavras com letras repetidas;
- A cada sugestão, o jogador recebe dicas do quanto está próximo da resposta correta.

2. RESULTADOS

2.1. DESCRIÇÃO GERAL

A principal característica do Jogo da Palavra é o sorteio de uma palavra nova sempre que for iniciado o jogo. Sendo assim, foi feita uma função chamada ***sorteaPalavra().*** Nessa função é feita a abertura de um arquivo no modo leitura, e ela retornará o índice do arquivo da palavra sorteada, além de fazer a contagem de quantas palavras há no arquivo para que possa ser usada junto à função ***rand()*** da biblioteca stdlib.

CÓDIGO:

```
int sorteaPalavra() {
	FILE *palavras;
	palavras = fopen("ArqPalavras.txt","r");
	if(palavras == NULL){
		printf("Erro na abertura do arquivo!");
		system("pause");
	}
	int nPalavras = 0;
	int nSorteio; //variável que receberá o índice do arquivo da palavra sorteada
	int i;

	while(fscanf(palavras,"%s",v_palavra[nPalavras]) != EOF ){
		nPalavras++;
   }
   srand(time(NULL));
   nSorteio = rand() % nPalavras;
	fclose(palavras);
	return nSorteio;
}
```

Uma outra função importante no Jogo é o cadastro de usuário, sem ela não há como iniciar uma partida. Ela é identificada como ***cadastro_usuario()***, na qual há a abertura do arquivo com todos os usuários no modo *append* para que seja adicionado um novo sem a exclusão de outros.

Para o entendimento dessa função, é necessário saber que é feita a leitura/carregamento de todos os usuários salvos no arquivo **ArqUsuarios** pela função ***carregar_usuarios()*** e feita a consulta da existência ou não do jogador dentro do arquivo pela função ***existe_usuario()***. Dessa forma, caso o usuário já exista é solicitado o cadastro de outro nome ou dada a opção de iniciar a partida.

CÓDIGO:

```
void cadastro_usuario(){
	int i=1;
```

```
        system("color 80");
   printf("\n_____CADASTRO DE USUÁRIO_____");

        FILE *cad_usuario;
        cad_usuario = fopen("ArqUsuarios.txt","a+");

        char usuario[50];
        if(cad_usuario == NULL){
                printf("Erro na abertura do arquivo!");
                system("pause");
        }else{
                carregar_usuarios(cad_usuario);
                do{
                        printf("\nDigite o nome: ");
                        scanf("%s",usuario);
                        if(existe_usuario(usuario)){
                                printf("Usuário já existe.\n");
                                printf("\n\nDigite 1 para cadastrar outro usuário ou 0 para voltar ao Menu Principal.\n");
                                scanf("%d",&i);
                        }
                        else{
                                fprintf(cad_usuario,"%s\n",usuario);
                                printf("Usuário cadastrado com sucesso! \n\nDigite 1 para continuar cadastrando ou 0 para voltar ao Menu Principal.\n");
                        scanf("%d",&i);
                        }

                }while (i==1);
                if(i!=1){
```

```
                fclose(cad_usuario);
            }
        }

printf("\n_____
_____\n\n");
        system("pause");
        system("cls");
}
```

O cadastro de novas palavras é também um diferencial nesse jogo. Por meio da função **cadastro_palavra()**, é feita a abertura de um arquivo no modo *append,* para que seja permitida a adição de novas palavras sem a exclusão de outras. Nessa função também há a verificação do tamanho da palavra que o usuário deseja cadastrar, se ela for maior ou menor do que 5 letras, a palavra não será válida.

CÓDIGO:

```
void cadastro_palavra(){
        int i=1;
        char palavra[6];
        system("color 50");
        printf("\n_____CADASTRO DE
PALAVRA_____");

        FILE *palavras;
        palavras = fopen("ArqPalavras.txt","a");

        if(palavras == NULL){
            printf("Erro na abertura do arquivo!");
            system("pause");
        }else{
            do{
                printf("\n\nEscreva a palavra (com 5 letras) que deseja cadastrar: ");
```

```
                scanf("%s",palavra);
                // Verifica se o tamanho da palavra é válido
                if(strlen(palavra)!= 5){
                        printf("Palavra inválida, digite novamente!\n");
                }
                else{
                        //Salva palavra no arquivo
                        fprintf(palavras,"%s\n",palavra);
                        printf("Palavra cadastrada com sucesso!\n");
                }
                printf("\nDigite 1 para continuar cadastrando ou 0 para voltar ao
Menu Principal.\n");
                scanf("%d",&i);
        }while (i==1);

        if(i!=1){
                fclose(palavras);
        }
    }

printf("\n_____
_____\n\n");
    system("pause");
    system("cls");
}
```

A opção de visualizar o ranking também torna o jogo mais divertido e competitivo, essa funcionalidade é descrita por meio da aplicação de algumas funções.

A primeira delas é ***pontuacao()***, ela recebe como parâmetro a palavra que retornará quantas letras o jogador acertou, e tem como finalidade verificar se já foi atingido o total de 5 letras certos. Pois, nesse caso, o jogo já pode ser finalizado.

CÓDIGO:

```
int pontuacao(char * palavra){
	int i;
	int pontos = 0;

//Contar quantas letras acertou
	for(i=0;i<5;i++){
		if(palavra[2*i] != '_')
			pontos++;
	}
	return pontos;
}
```

A segunda função utilizada para a criação do ranking é a ***carregar_ranking().*** Nela é aberto um arquivo no modo leitura com todos os usuários e suas respectivas pontuações, e verifica quantos usuários há no arquivo.

CÓDIGO:

```
void carregar_ranking(){
	int i;
	FILE *f_ranking = fopen("Ranking.txt","r");
	i = 0;
	while(i < 1000 && fscanf(f_ranking,"%s %s",ranking[i][0],ranking[i][1]) != EOF ){
		i++;
	}
	tam_ranking = i;
}
```

A terceira função referente a Ranking é ***atualizar_ranking()*** que serve para atualizar a pontuação dos jogadores, ela recebe como parâmetro o nome do usuário e abre o arquivo de ranking no modo de escrita. Sendo assim, caso o usuário já exista no ranking a pontuação dele é atualizada e, caso não exista, ele é adicionado ao final do ranking já que significará que é sua primeira partida.

CÓDIGO:

```
void atualizar_ranking(char *usuario){
	int i;
```

```
    FILE *f_ranking = fopen("Ranking.txt","w");
    bool existe = false;

    for(i = 0; i < tam_ranking;i++){
        //procura se existe o nome do usuário no ranking
        if(strcmp(ranking[i][0],usuario) == 0){
            sprintf(ranking[i][1],"%d",atoi(ranking[i][1]) + 1);
            existe = true;
            break;
        }
    }
    if(!existe){
        strcpy(ranking[tam_ranking][0],usuario);
        sprintf(ranking[tam_ranking][1],"%d",1);
        tam_ranking++;
    }
    for(i = 0;i < tam_ranking;i++){
        fprintf(f_ranking,"%s %s\n",ranking[i][0],ranking[i][1]);
    }

    fclose(f_ranking);
}
```

Por último, a função que determina toda a jogabilidade e une as funções é a ***jogo()***, ela se inicia na abertura do arquivo **ArqUsuarios** no modo leitura para verificar se o usuário já realizou seu cadastro antes de iniciar a primeira partida. Em seguida, recebe a palavra sorteada e salva na variável **palavra_a_descobrir.**

Após esses passos, é iniciado o laço de repetição para contar as tentativas de adivinhação do jogador. Caso o usuário acerte a posição de um ou mais caracteres, é feita a substituição de "_" salvos na variável para **palavra_final** para o caracter correto.

Para saber se o jogador acertou ou não a palavra, a função ***pontuacao(),*** que já foi explicada anteriormente, recebe como parâmetro a **palavra_final** e salva em uma variável o valor retornado pela função. Caso seja igual a 5, o usuário acertou e o jogo é encerrado com uma

mensagem de parabenização. Caso seja menor que 5, o jogador errou e o jogo é encerrado com uma página informando a palavra correta.

CÓDIGO:

```
void jogo(){
	system("color 30");
	int num;
	int n_palavra,i,j,k,cont_certas;
	char palavra_final[] = "_ _ _ _ _";
	char palavra_a_descobrir[6],entrada[6],usuario[50];

	FILE *cad_usuario;
	cad_usuario = fopen("ArqUsuarios.txt","r");
	carregar_usuarios(cad_usuario);
	fclose(cad_usuario);

	carregar_ranking();

	printf("\n_____JOGO DA PALAVRA_____");

	cont_certas = 0;
	n_palavra = sorteaPalavra(); //Recebe a palavra sorteada
	strcpy(palavra_a_descobrir,v_palavra[n_palavra]);
	printf("\n\nDigite o usuário: ");
	scanf("%s",usuario);

	if(!existe_usuario(usuario)){
		printf("\n\nUsuário inexistente.\nCrie o usuário para poder jogar!\n\n");
		system("pause");
		system("cls");
		cadastro_usuario();
```

```
            return;
        }

        for(i=0;i<5;i++){
            int contagem[30] = {0};
            printf("\n\n%s\n\ \n\n",palavra_final);
            scanf("%s",entrada);

            for(j = 0; j<5;j++){
                if(toupper(palavra_a_descobrir[j]) == toupper(entrada[j])){
                    palavra_final[2*j] = toupper(entrada[j]);
                }
            }

            for(j=0;j<5;j++){
                for(k=0;k<5;k++){
                    if(toupper(palavra_a_descobrir[j]) == toupper(entrada[k])  &&
j != k  && abs(j-k) > 1) {
                        contagem[toupper(palavra_a_descobrir[j]) - 'A'] = 1;
                    }
                }
            }

            for (j = 0 ; j< 30; j++){
                if(contagem[j] != 0)
                    printf("\n%c Tente uma palavra com a letra %c em outra
posição.\n",16,j+'A');
            }
            cont_certas = pontuacao(palavra_final);

            if(cont_certas == 5)
```

```
                break;
        }

        if(cont_certas == 5){
                system("cls");
                system("color 20");

printf("_____
_____\n");
                printf("\n\n\n\t\tParabéns, você ganhou!\n\n\n");

printf("_____
_____\n");
                atualizar_ranking(usuario);
                printf("\nDigite 1 para continuar jogando ou 0 para voltar ao Menu
Principal.\n");
                scanf("%d",&num);

                if(num==1){
                        system("cls");
                        jogo();
                }else{
                        system("cls");
                }
        }

        else{
                system("cls");
                system("color 40");

printf("_____
_____\n");
                printf("\n\n\n\t\tQue pena, você errou!\n");
```

```
                printf("\n\n\t\tA palavra era %s.\n\n\n",palavra_a_descobrir);

printf("_____
_____\n");
                printf("\n\n\nDigite 1 para continuar jogando ou 0 para voltar ao Menu
Principal.\n");
                scanf("%d",&num);


                if(num==1){
                        system("cls");
                        jogo();
                }else{
                        system("cls");
                }
        }
}
```

2.2. DIFICULDADES E SOLUÇÕES

Algumas dificuldades foram encontradas ao longo do desenvolvimento desse projeto, entre elas cabe destacar o sorteio da palavra a ser adivinhada pelo jogador. Optamos por utilizar a função **rand()** da biblioteca stdlib e, como ela retorna um número inteiro, no caso desse jogo foi retornado o índice da palavra do arquivo e não a palavra em si.

Para solucionar esse problema, foram criadas duas variáveis, uma do tipo int chamada **n_palavra** para receber o índice e outra do tipo char chamada **palavra_a_descobrir** para, posteriormente, armazenar a palavra que foi sorteada. Em seguida, foi criado um vetor **v_palavra[1000][6]** que armazena todas as palavras presentes no arquivo.

Dessa forma, ao fazer **strcpy(palavra_a_descobrir,v_palavra[n_palavra])**, foi possível armazenar a string.

Outro desafio foi verificar e informar ao usuário se algum caracter da palavra digitada por ele existe na palavra sorteada, mas está na posição errada e, em caso de haver duas letras repetidas, não informar duas vezes que há essa letra na palavra.

Utilizamos, primeiramente, a função **toupper()** da biblioteca ctype, para não haver diferenciação de letras maiúsculas ou minúsculas no momento da comparação. Além disso, foi preciso criar um vetor que armazena a quantidade de letras do alfabeto chamado

contagem[30] com todos os valores nulos**.** Após isso, foram criados laços de repetição percorrendo toda a palavra para comparar dois caracteres entre si.

Por fim, para entender qual era a letra em questão, foi feito o uso da Tabela Asscii. Então, ao chegar ao índice da letra que estava correta, mas na posição errada, foi subtraído do valor de ’A’, que corresponde a 65, o valor da letra correta. Ao obter esse valor, ele foi igualado a 1 e salvo no vetor **contagem[30]**.

Ou seja, o vetor que antes era composto por valores nulos, agora possui o valor 1 no índice correspondente a letra do alfabeto. Dessa forma: **contagem[toupper(palavra_a_descobrir[j]) - 'A'] = 1**

Assim, quando **contagem[j]** for diferente de 0, significa que a letra existe na palavra sorteada.

2.3. FUNCIONALIDADE

Ao abrir o Jogo da Palavra o Menu Principal é apresentado ao usuário, conforme a Figura 1. Nessa tela há 7 escolhas que podem ser feitas pelo jogador.
A primeira opção é para iniciar o jogo, a segunda é para cadastrar uma nova palavra no banco de palavras, a terceira é para o usuário realizar o seu cadastro (necessário apenas seu nome), a quarta é para visualizar o ranking, a quinta é para acessar o menu de ajuda com a listagem de regras, a sexta é para os créditos (informações dos desenvolvedores) e, por último, a sétima opção é para encerrar o jogo.

FIGURA 1: TELA DE MENU PRINCIPAL

Como uma das regras do Jogo da Palavra é o cadastro do usuário, caso ele tente jogar antes de realizar essa etapa, o jogador é redirecionado para a página de Cadastro de Usuário, apresentada na figura abaixo.

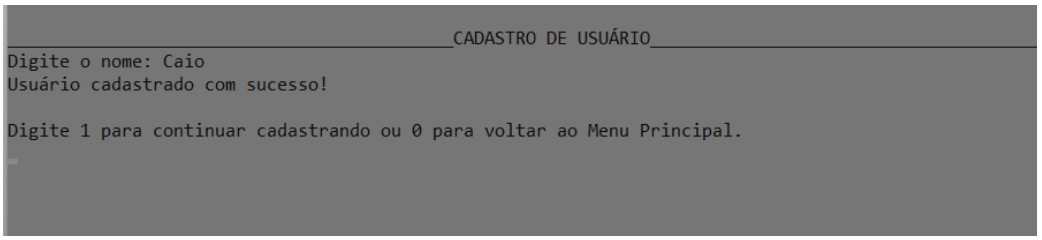

FIGURA 2: TELA DE CADASTRO DE USUÁRIO

Ao iniciar o jogo, é solicitado o nome do jogador para que seja verificado se o usuário já foi cadastrado. Caso a resposta seja positiva, o jogo é iniciado efetivamente.

Como mostrado na figura abaixo, a palavra contém 5 letras e, à medida que o jogador for dando seus palpites e as posições das letras corresponderem, a palavra vai sendo formada e, além disso, são apresentadas dicas no caso de existir a letra na palavra sorteada.

FIGURA 3: TELA DO JOGO

O jogador possui apenas 5 tentativas para tentar acertar a palavra que foi sorteada para ele. No caso de erro, ele não somará pontos e será informado sobre qual palavra era a correta. Em caso de acerto, o jogador somará um ponto.

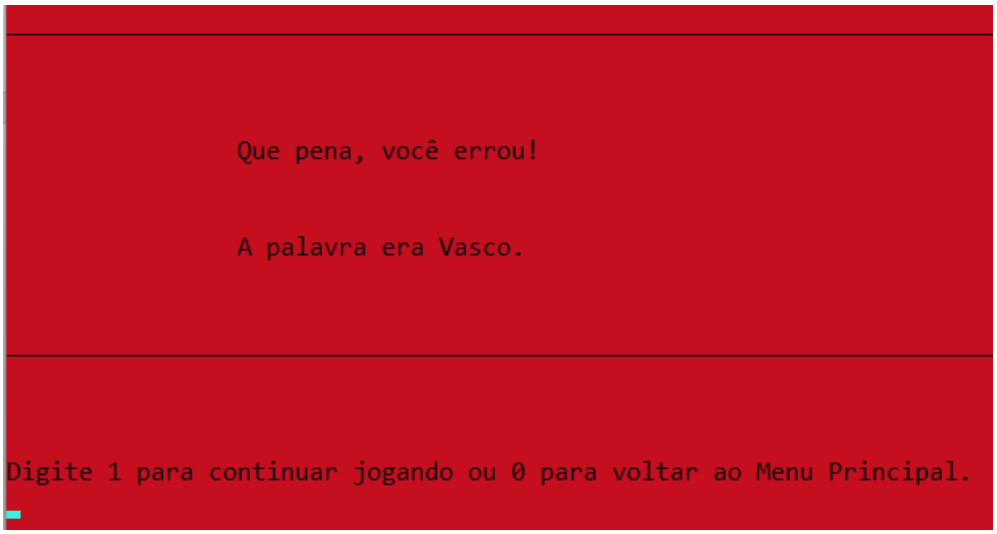

FIGURA 4: TELA DO FINAL DO JOGO EM CASO DE ERRO

Após o acerto de ao menos uma palavra, o jogador poderá visualizar seu nome e sua pontuação na página de Ranking.

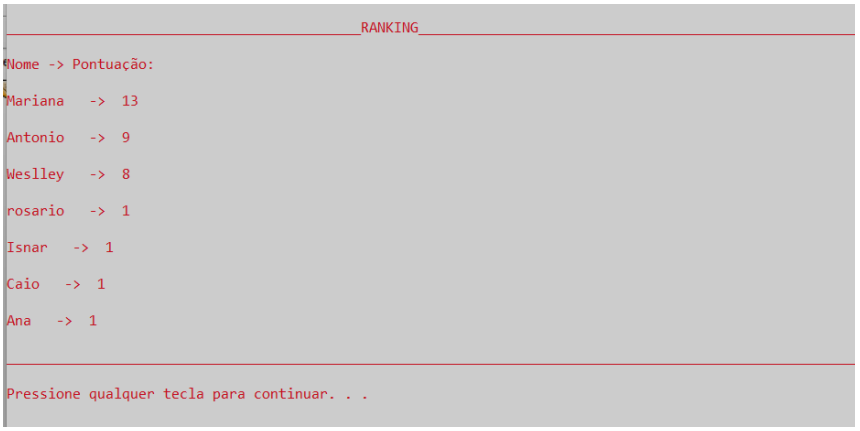

FIGURA 5: TELA DE RANKING

3. APÊNDICE

- Arquivo *jogo.c*

```
#include "jogo.h"

//VARIÁVEIS GLOBAIS (ACEITAS EM TODAS AS FUNÇÕES)
int op; //Opção para menu principal
char usuarios[30][50]; // vetor de usuarios
```

```
char v_palavra[1000][6]; // vetor que receberá todas as palavras do arquivo

char ranking[1000][30][10]; //vetor que receberá os valores que estão no arquivo ranking, no máximo 1000 ususários. [quantidade][nomes][pontuação]

int tam_ranking; // Para saber quantos usários tem no ranking


//Função para sortear palavra

int sorteaPalavra() {


        FILE *palavras;

        palavras = fopen("ArqPalavras.txt","r");


        if(palavras == NULL){

                printf("Erro na abertura do arquivo!");

                system("pause");

        }


        int nPalavras = 0;

        int nSorteio; //variável que receberá o índice do arquivo da palavra sorteada

        int i;


        while(fscanf(palavras,"%s",v_palavra[nPalavras]) != EOF ){ //ler todas as palavras do arquivo

    nPalavras++; //conta quantas palavras há no arquivo

  }


  srand(time(NULL)); //inicializa o gerador de números aleatórios

  nSorteio = rand() % nPalavras;

        fclose(palavras);

        return nSorteio;

}
```

```
//Função para ranking
int pontuacao(char * palavra){
	int i;
	int pontos = 0;

//Contar quantas letras acertou
	for(i=0;i<5;i++){
		if(palavra[2*i] != '_')
			pontos++;
	}

	return pontos;
}

//Função para saber o tamanho do ranking
void carregar_ranking(){
	int i;
	FILE *f_ranking = fopen("Ranking.txt","r");
	i = 0;

	//lê do arquivo duas strings, uma com nome e outra com a pontuação
	while(i < 1000 && fscanf(f_ranking,"%s %s",ranking[i][0],ranking[i][1]) != EOF ){
		i++;
	}

	tam_ranking = i;
}

//atualiza a pontuação do usuário
```

```
void atualizar_ranking(char *usuario){
	int i;
	FILE *f_ranking = fopen("Ranking.txt","w");
	bool existe = false;

	for(i = 0; i < tam_ranking;i++){
		//procura se existe o nome do usuário no ranking
		if(strcmp(ranking[i][0],usuario) == 0){
			 //Atoi é utilizado para convertar a pontuação para inteiro e somar +1.
E sprintf transforma novamente para string, para continuar no vetor de strings (ranking)
			sprintf(ranking[i][1],"%d",atoi(ranking[i][1]) + 1);
			existe = true;
			break;
		}

	}
	if(!existe){
		//Se o usuário ainda não existe no ranking, ele é adicionado no final do ranking
		strcpy(ranking[tam_ranking][0],usuario);
		//Atualiza a pontuação para 1
		sprintf(ranking[tam_ranking][1],"%d",1);
		//Como adicionou mais um usuário, o tamanho do ranking é aumentado em 1
		tam_ranking++;
	}

	for(i = 0;i < tam_ranking;i++){
		//Salvar o nome e a pontuação no arquivo
		fprintf(f_ranking,"%s %s\n",ranking[i][0],ranking[i][1]);
	}

	fclose(f_ranking);
```

```
}

//Função para auxiliar na alteração da ordem do ranking
void troca_str(char *s1, char * s2){
	char aux[100];
	strcpy(aux,s1);
	strcpy(s1,s2);
	strcpy(s2,aux);
}

//Ordena o ranking de forma decrescente
void ordernar_ranking(){
	int i ,j;

	//Compara uma pontuação do início do ranking com uma do final do ranking
	for(i = tam_ranking-1; i>0; i--){
		for(j = 0; j<i;j++){
			if(atoi(ranking[i][1]) > atoi(ranking[j][1])) {
				troca_str(ranking[i][0],ranking[j][0]); //troca posição do nome
				troca_str(ranking[i][1],ranking[j][1]); //troca posição da
pontuação
			}
		}
	}
}

//Função para imprimir o ranking
void mostrar_ranking() {
	int i;
	carregar_ranking(); //carrega novamente o ranking, para o caso de ter havido mudança
ou não ter sido carregado ainda
```

```
	ordernar_ranking(); //ordena de forma decrescente

	system("color 74");

printf("\n_____RANKING_____
_____\n\n");
	printf("Nome -> Pontuação: \n");
	for(i = 0; i < tam_ranking;i++){
		printf("\n%s  ",ranking[i][0]);
		printf(" ->  %s\n",ranking[i][1]);
	}

printf("\n_____
_____\n\n");
	system("pause");
	system("cls");
}


//Função para carregar usuários do arquivo em um vetor de usuários
void carregar_usuarios(FILE *cad_usuario){
	int i = 0;

	while(i < 50 && fscanf(cad_usuario,"%s",usuarios[i]) != EOF){
		i++;
	}
}

//Função para testar se o usuário já existe no arquivo
bool existe_usuario(char *usuario){

	int i = 0;
```

```
	for(i = 0; i < 50; i++){
		if(strcmp(usuario,usuarios[i]) == 0)
			// Usuário já existe no cadastro
			return true;
	}
	//Usuário não existe no cadastro
		return false;
}

//Função para cadastro de usuário
void cadastro_usuario(){
	int i=1;

	system("color 80");
   printf("\n_____________________________________________CADASTRO DE
USUÁRIO_____________________________________________________");

	FILE *cad_usuario;
	cad_usuario = fopen("ArqUsuarios.txt","a+");

	char usuario[50];

	if(cad_usuario == NULL){
		printf("Erro na abertura do arquivo!");
		system("pause");
	}else{
		carregar_usuarios(cad_usuario);
		do{
			printf("\nDigite o nome: ");
			scanf("%s",usuario);
			if(existe_usuario(usuario)){
```

```
                                printf("Usuário já existe.\n");
                                printf("\n\nDigite 1 para cadastrar outro usuário ou 0 para voltar ao Menu Principal.\n");
                                scanf("%d",&i);
                        }
                        else{
                                //Salva no arquivo
                        fprintf(cad_usuario,"%s\n",usuario);
                        printf("Usuário cadastrado com sucesso! \n\nDigite 1 para continuar cadastrando ou 0 para voltar ao Menu Principal.\n");
                        scanf("%d",&i);
                        }

                }while (i==1);

                if(i!=1){
                        fclose(cad_usuario);
                }
        }

printf("\n_____\n\n");
        system("pause");
        system("cls");
}

//Função para cadastrar palavras novas
void cadastro_palavra(){
        int i=1;
        char palavra[6];
```

```
	system("color 50");
	printf("\n_____CADASTRO DE PALAVRA_____");

	FILE *palavras;

	palavras = fopen("ArqPalavras.txt","a");

	if(palavras == NULL){
		printf("Erro na abertura do arquivo!");
		system("pause");
	}else{
		do{
			printf("\n\nEscreva a palavra (com 5 letras) que deseja cadastrar: ");
			scanf("%s",palavra);
			// Verifica se o tamanho da palavra é válido
			if(strlen(palavra)!= 5){
				printf("Palavra inválida, digite novamente!\n");
			}
			else{
				//Salva palavra no arquivo
				fprintf(palavras,"%s\n",palavra);
				printf("Palavra cadastrada com sucesso!\n");
			}
			printf("\nDigite 1 para continuar cadastrando ou 0 para voltar ao Menu Principal.\n");
			scanf("%d",&i);
		}while (i==1);

		if(i!=1){
			fclose(palavras);
```

```
                }
        }

printf("\n_____
_____\n\n");
        system("pause");
        system("cls");
}

void menu_principal(){
                system("color 60");

printf("\n\t_____\
n");

printf("\t_____\n")
;

printf("\t_____\n")
;
        printf("\t_____          _____\n");
        printf("\t_____          _____\n");
        printf("\t_____  JOGO DA PALAVRA
_____\n");
        printf("\t_____          _____\n");

printf("\t_____\n")
;

printf("\t_____\n")
;

printf("\t_____\n\
n\n");
        printf("\tEscolha uma das opções abaixo:\n\n",135,228);
```

```
        printf("\t%c 1 Jogar\n\t%c 2 Cadastrar Palavra\n\t%c 3 Cadastrar Usuário \n\t%c 4 Ver Ranking \n\t%c 5 Ajuda \n\t%c 6 Créditos \n\t%c 7 Sair\n\n",16,16,16,16,16,16,16);

        printf("\tOpção: ");

                scanf("%d",&op);

        system("cls");

}


void ajuda(){

        system("color 90");

   printf("\n_____QUAL O OBJETIVO DO JOGO?_____");

   printf("\n\n\t     O Jogo da Palavra foi feito para você advinhar a palavra que escolhemos para você.\n");

   printf("\n_____REGRAS O JOGO_____");

   printf("\n\n\t1- Você deve realizar seu cadastro antes de jogar a primeira vez.");

   printf("\n\n\t2- Você pode jogar quantas vezes quiser, mas tem apenas 5 tentativas para tentar acertar a palavra.");

   printf("\n\n\t3- A cada novo jogo, uma nova palavra é sorteada. Você pode nos ajudar a aumentar nosso banco de palavras cadastrando uma nova.");

   printf("\n\n\t4- Para cadastrar uma nova palavra, é preciso que ela possua 5 letras.");

   printf("\n\n\t5- Lembre-se que pode haver palavras com letras repetidas. Se a sua possuir, nós vamos te dar uma dica.");

   printf("\n\n\t6- Seu nome só aparecerá em nosso Ranking se você conseguir acertar ao menos uma palavra.");

   printf("\n\n\t7- Cada acerto te concederá 1 ponto.");

   printf("\n\n\n\n\t Está pronto pra jogar? \n");


printf("_____\n");

        system("pause");

   system("cls");

}
```

```
void creditos(){
	system("color 90");

printf("\n_____CRÉDITOS_____
_____");
   printf("\n\n\t    Criação: Mariana Barros e Antônio Abrantes.\n");
	printf("\t    Alunos do segundo período de Ciências da Computação.\n");
	printf("\n\t    Professores: Douglas e Daniel.\n");

printf("_____
_____\n");
	system("pause");
   system("cls");
}



//Função para jogar
void jogo(){
	system("color 30");
	int num;
	int n_palavra,i,j,k,cont_certas;
	char palavra_final[] = "_ _ _ _ _"; //palavra final digitada pelo usuário
	char palavra_a_descobrir[6],entrada[6],usuario[50];

	FILE *cad_usuario;
	cad_usuario = fopen("ArqUsuarios.txt","r");

	//Para conferir se o usuário existe
	carregar_usuarios(cad_usuario);
	fclose(cad_usuario);
```

```
	carregar_ranking();


	printf("\n_____JOGO DA PALAVRA_____");


	cont_certas = 0;

	n_palavra = sorteaPalavra(); //Recebe a palavra sorteada

	strcpy(palavra_a_descobrir,v_palavra[n_palavra]); //palavra_a_descobrir receberá a palavra sorteada


	printf("\n\nDigite o usuário: ");

	scanf("%s",usuario);


	if(!existe_usuario(usuario)){

		printf("\n\nUsuário inexistente.\nCrie o usuário para poder jogar!\n\n");

		system("pause");

		system("cls");

		cadastro_usuario();

		return; //Volta ao Menu Principal.

	}


	for(i=0;i<5;i++){

		int contagem[30] = {0}; //Vetor de letras do alfabeto. Garante que todos são nulo


		printf("\n\n%s\n\ \n\n",palavra_final); //imprime os caracteres digitados corretamente, substituindo o "_"

		scanf("%s",entrada);


		for(j = 0; j<5;j++){

			if(toupper(palavra_a_descobrir[j]) == toupper(entrada[j])){
```

```
                                palavra_final[2*j] = toupper(entrada[j]);  //O 2*j é para pular
espaços em branco da variável palavra_final e substituir o "_" pelo caracter

                                 // que foi digitado corretamente

                                }

                        }


                        for(j=0;j<5;j++){

                                for(k=0;k<5;k++){

                                        //Verifica se cada caracter da entrada (palavra digitada) existe
e e se está na posição errada da palavra sorteada

                                        if(toupper(palavra_a_descobrir[j]) == toupper(entrada[k])  &&
j != k  && abs(j-k) > 1) { //O último parâmetro é para não pegar letras repetidas

                                                contagem[toupper(palavra_a_descobrir[j]) - 'A'] = 1;
//O valor do A na tabela Asscii é 65 e o vetor contagem começa em 0.

                                        }

                                }

                        }


                        for (j = 0 ; j< 30; j++){

                                if(contagem[j] != 0)

                                        printf("\n%c Tente uma palavra com a letra %c em outra
posição.\n",16,j+'A'); //Imprime a letra correta

                        }


                        cont_certas = pontuacao(palavra_final);


                        if(cont_certas == 5) //se o usuário acertar, finaliza o jogo

                                break;

        }


        if(cont_certas == 5){

                system("cls");
```

```
                system("color 20");

printf("_____
_____\n");

                printf("\n\n\n\t\tParabéns, você ganhou!\n\n\n");

printf("_____
_____\n");

                atualizar_ranking(usuario);

                printf("\nDigite 1 para continuar jogando ou 0 para voltar ao Menu
Principal.\n");

                scanf("%d",&num);


                if(num==1){

                        system("cls");

                        jogo();

                }else{

                        system("cls");

                }

        }


        else{

                system("cls");

                system("color 40");

printf("_____
_____\n");

                printf("\n\n\n\t\tQue pena, você errou!\n");

                printf("\n\n\t\tA palavra era %s.\n\n\n",palavra_a_descobrir);

printf("_____
_____\n");

                printf("\n\n\nDigite 1 para continuar jogando ou 0 para voltar ao Menu
Principal.\n");
```

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

                if(num==1){
                        system("cls");
                        jogo();
                }else{
                        system("cls");
                }
        }
}
```

- Arquivo *jogo.h*

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <locale.h> // para utilizar acentos
#include <time.h> //para usar time()
#include <ctype.h> //para utilizar toupper()

extern int sorteaPalavra();
extern int pontuacao();
extern void carregar_ranking();
extern void atualizar_ranking(char *usuario);
extern void troca_str(char *s1, char * s2);
extern void ordernar_ranking();
extern void mostrar_ranking();
extern void carregar_usuarios(FILE *cad_usuario);
extern bool existe_usuario(char *usuario);
extern void cadastro_usuario();
```

```
extern void cadastro_palavra();
extern void menu_principal();
extern void ajuda();
extern void creditos();
extern void jogo();
```

- Arquivo main.cpp

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <locale.h> // para utilizar acentos
#include <time.h> //para usar time()
#include <ctype.h> //para utilizar toupper()
#include "jogo.h" //biblioteca com implementação das funções

extern int op; //Opção para menu principal
extern int tam_ranking; // Para saber quantos usários tem no ranking

//Função para aceitar acentos
void acento(){
	setlocale(LC_ALL,"Portuguese");
}

int main(){

	acento();
  tam_ranking = 0;

	do{
		menu_principal();
```

```
switch(op){
   case 1: //usuário joga
      jogo();
            system("pause");
      system("cls");
          break;
          case 2: //cadastra palavra
                cadastro_palavra();
   break;
       case 3: //cadastra usuário
                cadastro_usuario();
       break;
       case 4:
                mostrar_ranking();
       break;
   case 5:  //usuário acessa Ajuda
                ajuda();
         break;
   case 6: //Créditos
                        creditos();
                break;
   case 7: //encerra
       return 1;
   break;
                default:
                        printf("Opção inválida.\n\n");
                        system("pause");
                        system("cls");
       }
```

```
  }while(op!=7);


   return 0;
}
```