



Pontifícia Universidade Católica de Minas Gerais
Curso de Engenharia de Software
Disciplina: Algoritmos e Estruturas de Dados I
Prof. João Pedro O. Batisteli e Luiz Henrique da Costa Silva

Tema: Jogo de Adivinhação de Letras

Observações:

- O trabalho é individual.
- Cópias de trabalho receberão nota **ZERO**.
- O programa deve ser desenvolvido na linguagem de programação C.
- O trabalho será finalizado em sala na data combinada.

Objetivo: Desenvolver, em etapas, um jogo no qual o jogador tenta adivinhar uma letra escolhida (inicialmente pelo usuário e depois pelo computador). Ao longo das etapas, serão introduzidos conceitos de condicionais, geração aleatória, laços de repetição e pontuação.

Conteúdos Abordados: Estruturas condicionais (`if`, `if-else`, `switch`), estruturas de repetição (`while`, `for`), entrada e saída de dados, geração de valores aleatórios.

Etapas do Trabalho

Etapa 1 – Estrutura Básica e Adivinhação Simples

Objetivo: Criar a primeira versão do jogo permitindo que o jogador tente adivinhar uma letra digitada previamente.

Tarefas:

1. O programa deve permitir que um usuário inicial insira uma letra (maiúscula ou minúscula).
2. Após a inserção, a tela deve ser limpa usando `system("cls")`.
3. O programa deve informar ao jogador se a letra escolhida para ser adivinhada é maiúscula ou minúscula.
4. O jogador terá apenas **uma tentativa** para acertar a letra (sem laços de repetição).
5. Se acertar, deve exibir uma mensagem de vitória; caso contrário, uma mensagem de derrota.

Etapa 2 – Sorteio Automático e Tentativas com Dicas

Objetivo: Melhorar a lógica do jogo com sorteio automático de letras e número limitado de tentativas.

Tarefas:

1. O jogador deve informar se deseja jogar com letras maiúsculas ou minúsculas.
2. O programa deve sortear automaticamente uma letra aleatória no intervalo correspondente:

- Caso o jogador escolha maiúsculas: sorteio entre 'A' (65 na tabela ASCII) e 'Z' (90).
 - Caso escolha minúsculas: sorteio entre 'a' (97 na tabela ASCII) e 'z' (122).
3. O jogador terá, por padrão, **3 tentativas** para acertar a letra sorteada.
 4. A cada erro, o programa deve exibir uma dica informando se a letra digitada está **antes** ou **depois** da letra sorteada no alfabeto.
 5. Se o jogador acertar, o jogo deve terminar imediatamente exibindo uma mensagem de vitória.
 6. Se o jogador errar todas as tentativas, o programa deve revelar qual era a letra sorteada.

Exemplo de código para o sorteio da letra:

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

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

    char letraSorteada;
    int escolha;

    printf("Escolha o tipo de letra:\n");
    printf("1 - Maiúscula\n");
    printf("2 - Minúscula\n");
    scanf("%d", &escolha);

    if (escolha == 1) {
        // Sorteio entre 'A' (65) e 'Z' (90)
        letraSorteada = (char)((rand() % 26)+65);
    } else if (escolha == 2) {
        // Sorteio entre 'a' (97) e 'z' (122)
        letraSorteada = (char)((rand() % 26)+97);
    } else {
        printf("Opção inválida.\n");
        return 0;
    }

    printf("Letra sorteada (oculta no jogo): %c\n", letraSorteada);
    return 0;
}
```

Etapa 3 – Sistema de Pontuação e Menu

Objetivo: Adicionar pontuação e permitir novas partidas.

Tarefas:

1. Implementar um sistema de pontuação:
 - Acerto na 1ª tentativa: 100 pontos
 - Acerto na 2ª tentativa: 80 pontos
 - Acerto na 3ª tentativa: 50 pontos

- Se não acertar, pontuação 0
2. Exibir a pontuação final após cada partida.
 3. Criar um menu inicial com as opções:
 - Iniciar novo jogo
 - Sair
 4. Após terminar uma partida, o jogador deve poder escolher entre começar novamente ou encerrar o programa.