



Ramo Estudantil IEEE - UEL



Guilherme Ramolla (guilherme.ramolla@uel.br)
Gustavo Marcucci (gustavo.marcucci@uel.br)
Jordão Asato (jordão.asato@uel.br)
Rodrigo Macedo Silva Junior (rodrigo.macedo@uel.br)

RELATÓRIO FINAL: Gerador de Caça Palavras

Londrina
2023

Contato do Ramo: sb.uel@ieee.org
Institute of Electrical and Electronics Engineers – IEEE
Universidade Estadual de Londrina - UEL • Paraná - Brasil



Ramo Estudantil IEEE - UEL



Guilherme Ramolla
Gustavo Marcucci
Jordão Asato
Rodrigo Macedo Silva Junior

RELATÓRIO FINAL: Gerador de Caça Palavras

Relatório apresentado ao Ramo Estudantil
IEEE da Universidade Estadual de Londrina.

Diretor de Projetos: Nathan Andreani Netzel
Gestores de Projetos: Daniel Tresse Dourado, Levi Monteiro dos Santos

Londrina
2023



Ramo Estudantil IEEE - UEL



RAMOLLA, Guilherme. MARCUCCI, Gustavo. ASATO, Jordão. SILVA, Rodrigo Macedo. **Relatório Final:** Gerador de Caça Palavras. 2023. 10 folhas. Relatório apresentado ao Ramo Estudantil IEEE da Universidade Estadual de Londrina, Londrina, 2023.

RESUMO

O projeto do gerador de caça palavras consiste na utilização de uma linguagem de programação, no caso desde projeto a linguagem escolhida foi C, para criação de um programa de computador capaz de gerar um arquivo contendo um jogo de caça palavras de acordo com as especificações fornecidas pelo usuário. Esse programa recebe um arquivo de texto como entrada e entrega ao usuário outro arquivo de texto com o caça palavras já pronto para ser jogado.



Ramo Estudantil IEEE - UEL



Palavras-chave: Programação, C, gerador, caça-palavras.



Ramo Estudantil IEEE - UEL



SUMÁRIO

Sumário

1. INTRODUÇÃO.....	6
2. METODOLOGIA.....	7
3. RESULTADOS E DISCUSSÕES.....	10
4. CONCLUSÕES	11



1. INTRODUÇÃO

Jogos de palavras, como caça-palavras, têm desafiado mentes e proporcionado entretenimento por gerações. No entanto, a crescente influência da tecnologia moldou a maneira como esses quebra-cabeças são criados e distribuídos. Este trabalho é referente a um projeto de um gerador de caça-palavras, um software desenvolvido na linguagem de programação C. Este software oferece aos entusiastas de quebra-cabeças a capacidade de criar quebra-cabeças de palavras personalizados, permitindo que usuários definam suas preferências e especificações para um jogo sob medida. O programa recebe um arquivo de texto como entrada, processa as informações fornecidas pelo usuário e gera um novo arquivo de texto contendo um caça-palavras pronto para ser desfrutado. Neste artigo, vamos mergulhar na análise de como esse gerador de caça-palavras em C funciona e atende às demandas de jogadores de quebra-cabeças, destacando os aspectos de programação que tornam essa aplicação possível.



2. METODOLOGIA

Foi empregada uma metodologia de divisão, tentativa e erro. Inicialmente foram designadas partes separadas para cada integrante. Depois de concluído o que seria o esqueleto do código final, através de feedback, tentativa e erro, o código evoluiu, sendo removidos todos os bugs percebidos até então, e sendo implementada uma nova funcionalidade, a de permitir que as palavras se cruzassem umas sobre as outras.

Definição da Estrutura: A estrutura "palavra" é usada para armazenar informações sobre cada palavra, incluindo sua posição na grade e a letra correspondente.

Tabela 1 – Funções do código e respectivas explicações

geraEsp	Essa função é responsável por gerar as posições possíveis para uma palavra na grade de caça-palavras com base em várias direções (esquerda, direita, cima, baixo e suas diagonais). Ela também verifica se as letras correspondentes já estão presentes nas posições geradas. Se não houver espaço suficiente ou se uma letra não corresponder, ela redefine as variáveis de direção.
coloca	Esta função coloca a palavra gerada pela função geraEsp na grade de letras. Ela escolhe aleatoriamente uma das oito direções (esquerda, direita, cima, baixo e suas diagonais) para colocar a palavra na grade. Ela também armazena informações sobre a palavra, como sua posição e letra correspondente na estrutura palavra.
main	A função principal do programa começa lendo informações sobre a grade (quantidade de linhas e colunas) do arquivo "dados.txt". Em seguida, gera uma grade de letras aleatórias, lê as palavras desse arquivo e as coloca na grade de caça-palavras usando as funções geraEsp e coloca. O programa também verifica se as palavras se cruzam em alguma posição e exibe essas posições. No final, ele escreve a grade de caça-palavras resultante em um arquivo chamado "caca_palavra.txt".

Fonte: os autores

Figura 1 – Primeira versão do código

```
1 #include <stdio.h>
2
3 int main() {
4     int linhas, colunas;
5
6     printf("Digite o número de linhas da matriz: ");
7     scanf("%d", &linhas);
8     printf("Digite o número de colunas da matriz: ");
9     scanf("%d", &colunas);
10
11     if (linhas <= 0 || colunas <= 0) {
12         printf("Número inválido de linhas ou colunas.\n");
13     }
14     else {
15         char matriz[linhas][colunas];
16
17         char letra = 'A';
18
19         for (int i = 0; i < linhas; i++) {
20             for (int j = 0; j < colunas; j++) {
21                 matriz[i][j] = letra;
22             }
23         }
24
25         printf("Matriz gerada:\n");
26
27         for (int i = 0; i < linhas; i++) {
28             for (int j = 0; j < colunas; j++) {
29                 printf("%c ", matriz[i][j]);
30             }
31             printf("\n");
32         }
33     }
34     return 0;
35 }
```

Fonte: os autores



Figura 2 – Versão final do código

The image shows a screenshot of a code editor window. The code is written in a dark-themed editor with a light background. It appears to be a C++ program, likely for a microcontroller, given the use of `volatile` and `uint8_t` types. The code includes several function definitions and a `main` function. The code is wrapped in a light gray box with a thin border.

Fonte: os autores



Ramo Estudantil IEEE - UEL



3. RESULTADOS E DISCUSSÕES

Após finalizado o código, foram realizados diversos testes por todos os integrantes do grupo, garantindo que funcionasse como esperado. Realizamos com indivíduos de fora do grupo, tanto na parte funcional como na parte prática.

O projeto final diferiu da idealização feita no início. Com mais tempo e um planejamento diferente seria possível ter completado a ideia inicial de fazer uma interface mais bonita para o usuário e possibilitar que o caça palavras gerado fosse jogável.



Ramo Estudantil IEEE - UEL



4. CONCLUSÕES

A partir da testagem do funcionamento do código foi possível aferir que ele estava funcionando de acordo com o desejado, e logo, que não havia erros de sintaxe ou lógica no código, assim, finalizando o projeto.