



## Ramo Estudantil IEEE - UEL



---

Aline Arissa Nakao Horiye (aline.arissahoriye@uel.br)  
Bruna de Lima Dantas (bruna.dantas.13@uel.br)  
Kaue Ernany Tavares Cruz da Silva(kaueernany.cruz@uel.br)  
Luiz Henrique Botega Beraldi (luizhenrique.beraldi@uel.br)

### **RELATÓRIO FINAL:** Projeto Campo Minado

Londrina  
2023



# Ramo Estudantil IEEE - UEL



---

Aline Arissa Nakao Horiye  
Bruna de Lima Dantas  
Kaue Ernany Tavares Cruz da Silva  
Luiz Henrique Botega Beraldi

## RELATÓRIO FINAL: Projeto Campo Minado

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

---

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



## Ramo Estudantil IEEE - UEL



---

HORIYE, Aline Arissa Nakao. DANTAS, Bruna de Lima. SILVA, Kaue Ernany Tavares Cruz. BERALDI, Luiz Henrique Botega. **Relatório Final:** Projeto Campo Minado. 2023. 14 folhas. Relatório apresentado ao Ramo Estudantil IEEE da Universidade Estadual de Londrina, Londrina, 2023.

### RESUMO

O projeto Campo Minado consiste em um código feito em C que simula o jogo Campo Minado, podendo ser jogado no terminal do compilador desejado.

**Palavras-chave:** Campo minado, bomba, C, programação.



## SUMÁRIO

### Sumário

1. INTRODUÇÃO.....	6
2. FUNDAMENTAÇÃO TEÓRICA .....	7
2.1 FUNÇÕES DO CÓDIGO .....	7
2.1.1 jogo .....	7
2.1.2 inicia .....	7
2.1.3 sortear .....	7
2.1.4 coordenadaValida .....	7
2.1.5 quantBombasVizinhas .....	8
2.1.6 contarBombas .....	8
2.1.7 imprimirN .....	8
2.1.8 imprimirL .....	8
2.1.9 imprimirLN.....	8
2.1.10 abrirNo.....	9
2.1.11 ganhou .....	9
2.1.12 traduz .....	9
2.2 PARTICULARIDADES DO CÓDIGO .....	9
2.2.1 Dificuldade .....	9
2.2.1 Apresentação do Tabuleiro .....	10
3. METODOLOGIA.....	11
4. RESULTADOS E DISCUSSÕES.....	12



## 1. INTRODUÇÃO

Jogos do gênero Campo Minado (do inglês, *Minesweeper*) datam desde o início dos anos de 1980, juntamente com os primeiros modelos de computadores domésticos. Do *Mined-Out* lançado em 1983 até a versão do *Minesweeper* lançada pela *Microsoft* em 1990 (e depois continuamente a cada nova versão do sistema operacional *Windows*), o *game* se mantém relevante. De campeonatos competitivos a aparições como *minigame* em outros títulos grandes na indústria, tal qual a atualização de Primeiro de Abril de *Minecraft* em 2015, Campo Minado é atemporal.

O jogo (e o gênero todo, em geral) se baseia num campo quadriculado de espaços selecionáveis, no qual 'minas' se escondem. O objetivo é 'limpar' o campo sem detonar as minas, com a ajuda de dicas numéricas dadas por espaços 'limpos'. Cada número representa a quantidade de minas escondidas nos quadrados adjacentes. Portanto, o *game* se enquadra na categoria *puzzle* (quebra-cabeça ou lógica, em português).

O projeto tutorial em questão, portanto, tem como finalidade um código feito com linguagem C que simula o jogo Campo Minado e podendo ser jogado no terminal do compilador desejado.



## 2. FUNDAMENTAÇÃO TEÓRICA

### 2.1 FUNÇÕES DO CÓDIGO

#### 2.1.1 jogo

Função principal do programa, é a única função da main, sendo dela que o programa começa. Ela contém a seleção de dificuldade e já designa os valores pré-definidos para as variáveis.

#### 2.1.2 inicia

Mandamos as pré-definições escolhidas na função jogo para ela e aqui criamos o tabuleiro de acordo com isso. São criados 3 tabuleiros: um com as bombas, um com as posições abertas e outro com os números de vizinhos.

#### 2.1.3 sortear

Serve para sortear as bombas em lugares aleatórios usando a função “rand”.

#### 2.1.4 coordenadaValida

Confere se a coordenada que foi mandada para ela é válida.



## 2.1.5 quantBombasVizinhas

Verifica cada posição em volta da coordenada escolhida para saber quantas bombas existem na vizinhança dela.

## 2.1.6 contarBombas

Manda as linhas e colunas (ou seja, a localização) para a função QuantBombasVizinhas mostrada acima, o retorno da função é adicionado em cada posição no tabuleiro de vizinhos.

## 2.1.7 imprimirN

Imprime o tabuleiro com as pré-definições pedidas no começo do código, nesse caso, o tabuleiro é impresso com números para jogar.

## 2.1.8 imprimirL

Imprime o tabuleiro com as pré-definições pedidas no começo do código, nesse caso, o tabuleiro é impresso com letras para jogar.

## 2.1.9 imprimirLN

Imprime o tabuleiro com as pré-definições pedidas no começo do código, nesse caso, o tabuleiro é impresso com números nas linhas e letras nas colunas para jogar, semelhante ao campo do jogo Batalha Naval.



## 2.1.10 abrirNo

Abre a coordenada escolhida.

## 2.1.11 ganhou

Verifica se a quantidade de espaços que faltam são todas bombas (requisito para vitória) ou se existe um que não é bomba.

## 2.1.12 traduz

Caso o usuário escolha jogar com alguma opção que contém letras, o código vem para essa função, aqui ele “traduz” as letras para seus equivalentes em números. Essa função foi criada para facilitar o código, uma vez que ter que recriar diversas funções apenas para aceitar a entrada alfanumérica iria dificultar seu entendimento, além de aumentar seu tamanho desnecessariamente.

## 2.2 PARTICULARIDADES DO CÓDIGO

### 2.2.1 Dificuldade

São disponibilizados três níveis de dificuldade ao jogador: **Fácil** (com um tabuleiro 10x10 e 10 minas, ou seja, 10% de probabilidade de um quadrado qualquer ser uma bomba), **Médio** (16x16 e 40 minas, ou seja, 15,625% de chance) e **Difícil** (21x21 e 90 minas, e probabilidade um pouco mais de 20%). A escolha por pré-definir o tamanho do campo e a quantidade de bombas aconteceu para tentar manter o jogo o menos “quebrado” possível. Desta maneira, o jogador é garantido poder de escolha, desde um modo mais casual até um mais exigente, sem perder uma gameplay equilibrada.



## Ramo Estudantil IEEE - UEL



---

### 2.2.1 Apresentação do Tabuleiro

A versão produzida no tutorial possui três opções de tabuleiro: a completamente numérica (linhas e colunas representadas por números), a versão completamente alfabética (linhas e colunas representadas por letras) e a alfanumérica (linhas como números e colunas como letras), semelhante ao jogo Batalha Naval. Essa possibilidade de escolha de formato de tabuleiro surgiu numa tentativa de se adaptar melhor ao gosto do possível jogador e melhorar sua experiência ou *gameplay*.



## Ramo Estudantil IEEE - UEL



---

### 3. METODOLOGIA

O código do projeto se baseou em outros disponíveis na internet, em especial ao do Wagner Gaspar, disponível em <<https://wagnergaspar.com/testando-nosso-jogo-campo-minado/>>. Apesar das semelhanças, o tutorial possui diversas adaptações e melhorias, a destacar: correção de verificação de bombas para todas os espaços vizinhos, e não apenas os de lado; adição de opção de escolha do tipo de tabuleiro (numérico, alfabético e alfanumérico); adição de escolha de nível de dificuldade atrelado ao tamanho do tabuleiro e ocorrência de minas. O código se encontra disponível e comentado no GitHub <<https://github.com/LurdesBB/Campo-Minado>>.



## 4. RESULTADOS E DISCUSSÕES

O código criado em C cumpre seu objetivo, sendo um Campo Minado capaz de rodar em qualquer compilador, além de possuir possibilidades de customização em aparência de tabuleiro e tamanho/dificuldade.

```
3. Dificil
Opcao: 1

Deseja jogar usando os numeros ou as letras do teclado?
Digite (1) para numeros, (2) para letras ou (3) para numeros e letras: 3

CAMPO MINADO

  A  B  C  D  E  F  G  H  I  J
-----
0 |  |  |  |  |  |  |  |  |  |
-----
1 |  |  |  |  |  |  |  |  |  |
-----
2 |  |  |  |  |  |  |  |  |  |
-----
3 |  |  |  |  |  |  |  |  |  |
-----
4 |  |  |  |  |  |  |  |  |  |
-----
5 |  |  |  |  |  |  |  |  |  |
-----
6 |  |  |  |  |  |  |  |  |  |
-----
7 |  |  |  |  |  |  |  |  |  |
-----
8 |  |  |  |  |  |  |  |  |  |
-----
9 |  |  |  |  |  |  |  |  |  |
-----

Digite as coordenadas de linha (numero) e coluna (letra): 0a

  A  B  C  D  E  F  G  H  I  J
-----
0 | 1 |  |  |  |  |  |  |  |
-----
1 |  |  |  |  |  |  |  |  |  |
```



## Ramo Estudantil IEEE - UEL



---

### 5. CONCLUSÕES

Campo Minado é um jogo de alto valor histórico, cuja programação, apesar de resultar em um jogo fácil de se aprender e se divertir, possui diversos obstáculos: variáveis, estruturas condicionais, loops, manipulações de matrizes entre outras.



## REFERÊNCIAS BIBLIOGRÁFICAS

**The History of Minesweeper.** Disponível

em:<<https://freeminesweeper.org/minesweeper-history.php>>. Acesso em: 20 set. 2023.

**Minesweeper (video game).** Disponível

em:<[https://en.wikipedia.org/wiki/Minesweeper\\_\(video\\_game\)#:~:text=to%20solved%20state-,History,the%20ZX%20Spectrum%20in%201983.>](https://en.wikipedia.org/wiki/Minesweeper_(video_game)#:~:text=to%20solved%20state-,History,the%20ZX%20Spectrum%20in%201983.>)>. Acesso em: 20 set. 2023