

# RELATÓRIO TÉCNICO - PROJETO DE CAMPO MINADO EM C

**Aluno:** João Davi dos Santos Araújo

**Professor:** Fernando Marques Figueira Filho

**Disciplina:** Introdução às Técnicas de Programação

Universidade Federal do Rio Grande do Norte

Outubro de 2025

## INTRODUÇÃO E CONTEXTO

O projeto “Campo Minado em C” tem como objetivo implementar uma versão textual do clássico jogo Campo Minado, utilizando os principais conceitos de lógica de programação aprendidos até o momento. O programa desafia o jogador a descobrir as posições seguras em um tabuleiro 5x5, evitando as minas ocultas. O objetivo é aplicar de forma prática estruturas condicionais, laços de repetição, vetores e funções em um contexto lúdico e interativo.

O Campo Minado oferece uma aplicação dinâmica, permitindo compreender a lógica de controle de fluxo e manipulação de dados em múltiplas dimensões. O projeto demonstra como o uso combinado de estruturas básicas pode resultar em um sistema funcional completo.

A escolha do Campo Minado se deu por ser um desafio de programação que exige o domínio dos fundamentos e por ser algo de caráter lúdico, fazendo com que o processo de desenvolvimento seja não só um desafio de aprendizado, mas seja até mesmo recreativo, o que torna todo o processo mais agradável. Além disso, é um jogo amplamente conhecido, o que facilita compreender o funcionamento e identificar erros de lógica.

## ANÁLISE TÉCNICA

## Metodologia:

Foram usadas algumas ferramentas importantes para o desenvolvimento do projeto. São elas: compilador GCC (GNU Compiler Collection), editor de texto Visual Studio Code, sistema operacional Windows e as bibliotecas `stdio.h`, `stdlib.h`, `time.h` e `locale.h`, todas padrão da linguagem C.

## Aplicação dos conceitos da unidade 1:

**Estruturas Condicionais:** Utilizadas para verificar se uma posição contém mina, se o jogador venceu, ou se uma coordenada informada é válida. A estrutura `if/else` garante o controle do fluxo do jogo.

**Estruturas de Repetição:** Os laços `for` e `while` controlam a criação do campo, o posicionamento das minas e o loop principal do jogo. O uso do `while` permite que o jogo continue até que o jogador perca ou vença.

**Vetores:** Foram aplicados vetores para representar o campo e as posições das minas. Cada célula do vetor guarda um estado (mina, campo seguro, jogada aberta).

**Funções:** Foram criadas funções separadas para inicializar o campo, imprimir o tabuleiro e processar as jogadas. Essa organização facilita a manutenção e a leitura do código.

## Estruturas de dados:

A estrutura de dados principal é uma matriz de inteiros 5x5 que representa o tabuleiro. Outra matriz guarda as posições reveladas. Variáveis simples armazenam coordenadas, estado de jogo e número de minas. Essas estruturas permitem manipular facilmente as informações do jogo sem recorrer a alocação dinâmica.

## IMPLEMENTAÇÃO E REFLEXÃO

### Dificuldades encontradas:

As principais dificuldades encontradas durante a implementação foram:

controlar corretamente os índices da matriz para evitar acessos inválidos; evitar que as minas fossem posicionadas duas vezes na mesma coordenada; exibir o campo de forma compreensível para o jogador; manter o código limpo e modularizado. Esses desafios exigiram revisões constantes do código e testes repetidos após cada alteração.

## Soluções Implementadas

As soluções aplicadas incluíram verificações adicionais nas funções de geração do campo, o uso de loops de repetição controlados, e uma função dedicada para mostrar o campo ao jogador de maneira clara. A organização modular também ajudou a isolar e corrigir erros com maior facilidade.

## Organização do Código

O código foi estruturado em funções específicas: uma para gerar o campo, outra para exibir e outra para processar jogadas. A função principal (main) coordena o fluxo geral do jogo. Essa divisão permitiu clareza e reutilização de código, que é exatamente o papel que as funções têm de repetir instruções.

## Fluxo Lógico do Programa

O programa possui a seguinte organização lógica:

1. O programa inicia e gera o campo 5x5 com 5 minas aleatórias.
2. O campo é mostrado ao jogador com todas as posições ocultas.
3. O jogador informa uma linha e uma coluna (de 1 a 5).
4. O programa verifica se há mina naquela posição: - Se houver, o jogo termina (derrota); - Se não houver, a posição é revelada e o jogo continua.
5. O processo se repete até o jogador vencer (revelar todas as posições sem minas) ou perder.

## Conclusão

O desenvolvimento do Campo Minado permitiu consolidar os conceitos iniciais da linguagem C. Foi possível compreender melhor o uso de matrizes, o funcionamento de loops e condicionais e a importância da organização modular. Como melhorias futuras, podem ser adicionados níveis de dificuldade, um contador de tentativas e um sistema de pontuação.