



UnB

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA
ELÉTRICA**

**Projeto de Computacional da Disciplina
Computação para Engenharia**

Pedro Guimarães Batista

Rodrigo Corrêa Pereira

Brasília, 7 de Fevereiro de 2025.

SUMÁRIO

1.INTRODUÇÃO	3
2.OBJETIVO	3
3. METODOLOGIA.....	3
3.1. Conceitos Abordados	3
3.2. Funções Principais	4
3.3. Fluxo do Programa.....	4
3.4. Bibliotecas Utilizadas	5
4. CONCLUSÃO	5

1. INTRODUÇÃO

O jogo da velha é um tradicional passatempo popular para dois jogadores, cujo objetivo é formar uma linha horizontal, vertical ou diagonal com três símbolos igual (X ou O) em um tabuleiro 3x3. O presente trabalho consiste na implementação de um código em C++ para uma competição de jogo da velha, utilizando conceitos abordados na disciplina, tais quais estruturas condicionais, funções, vetores, laços de repetição, structs, ponteiros e manipulação de arquivos. O programa desenvolvido oferece duas modalidades de jogo: a *multiplayer*, onde dois jogadores disputam entre si, e a *singleplayer*, em que um jogador disputa contra o computador. A implementação do jogo foi desenvolvida com ênfase na clareza do código, modularidade e usabilidade, assegurando uma experiência interativa e iterativa para o usuário.

2. OBJETIVO

O objetivo deste projeto é desenvolver um código em C++ que simule o jogo da velha, possuindo duas modalidades de jogo:

- a) Modo *Multiplayer*: dois jogadores competem entre si;
- b) Modo *Singleplayer*: um jogador compete contra o computador, que realiza jogadas aleatórias.

3. METODOLOGIA

3.1. Conceitos Abordados

- a) Structs: no código desenvolvido, utiliza-se a Struct Jogador para facilitar o armazenamento de informações referentes a cada jogador, sendo elas o nome e a pontuação;
- b) Funções: foram desenvolvidas várias funções ao longo do código para implementar cada tarefa necessária em um jogo da velha. As funções criadas no código serão explicadas em mais detalhes na seção 3.2;
- c) Ponteiros: várias funções do código utilizam argumentos passados por referência, utilizando a lógica praticada na aprendizagem de ponteiros;

- d) **Laços de Repetição e Estruturas Condicionais:** essas estruturas foram amplamente utilizadas ao longo do programa para controlar o fluxo de execução e garantir o funcionamento correto do jogo;
- e) **Arquivos:** no momento em que se encerra uma seção de jogos, o código gera um arquivo txt da última competição realizada, mostrando os nomes dos jogadores e o placar de vitórias.

3.2. Funções Principais

- a) *ExibirTabuleiro*: Exibe o estado atual do tabuleiro no terminal;
- b) *VerificaVencedor*: Verifica se há um vencedor com base no estado atual do tabuleiro;
- c) *TabuleiroCheio*: Determina se todas as posições do tabuleiro estão ocupadas, indicando um possível empate;
- d) *Jogo*: Gerencia a lógica principal de uma partida, alternando entre os jogadores e verificando o resultado a cada jogada;
- e) *Singleplayer*: Implementa o modo de jogo contra o computador, com jogadas aleatórias realizadas pela máquina;
- f) *Multiplayer*: Implementa o modo de jogo para dois jogadores humanos, permitindo que ambos interajam alternadamente.

3.3. Fluxo do Programa

1. O programa inicia exibindo um menu para o usuário escolher entre os modos *SinglePlayer*, *MultiPlayer* ou sair do jogo.
2. No modo *SinglePlayer*, o computador realiza jogadas aleatórias.
3. No modo *MultiPlayer*, os jogadores alternam entre si para realizar suas jogadas.
4. Após cada partida, o placar é atualizado e o usuário pode escolher jogar novamente ou voltar ao menu principal.

3.4. Bibliotecas Utilizadas

- <iostream>: Para entrada e saída de dados;
- <cstdlib>: Para geração de números aleatórios;
- <string>: Para manipulação de strings;
- <fstream>: Para a manipulação de arquivos;
- <iomanip>: Para a formatação de saída.

4. CONCLUSÃO

Este projeto demonstrou a aplicação de conceitos fundamentais de programação em C++ para a criação de um jogo interativo. A implementação do jogo da velha permitiu explorar técnicas como manipulação de matrizes, controle de fluxo, modularização de código e interação com o usuário. Além disso, o programa foi desenvolvido com foco na usabilidade, garantindo que os jogadores tenham uma experiência clara e intuitiva. O código pode ser expandido com funcionalidades adicionais, como:

- Implementação de uma inteligência artificial mais avançada para o modo Singleplayer.
- Adição de uma interface gráfica para melhorar a experiência do usuário.
- Suporte para salvar e carregar partidas.

Em suma, o projeto atingiu seus objetivos, proporcionando uma base sólida para o desenvolvimento de jogos mais complexos no futuro.

