

# **Grupo**

## **Integrantes:**

**Carlos Eduardo da Silva Papa – 232013390.**

**Joao Arthur Barbosa De Farias – 211067282.**

**Vinicius Magno Da Silva Nascimento – 222005528.**

**Younes Boumediene – 212009040.**

# **Apresentação do Código C++ para um Jogo de Bingo**

**Análise e Explicação do Código Fonte**

# sumário

- **Introdução.**

- **Composição do Código:**

- Análise detalhada das classes e funções utilizadas.
- Explicação dos tipos de dados e estruturas usados.

- **Funcionamento do Jogo:**

- Descrição passo a passo do fluxo do jogo.
- Demonstração prática da execução do código.

- **Conclusão e Discussão:**

- Resumo das principais funcionalidades.
- Discussão sobre possíveis melhorias e extensões futuras.

# Introdução

## O que é o Bingo?

- O Bingo é um jogo de azar onde os jogadores marcam números em uma cartela, conforme eles são sorteados aleatoriamente.
- O objetivo é completar uma linha, coluna ou diagonal na cartela para ganhar o jogo.

## Objetivo do Projeto:

- Criar uma simulação do jogo de Bingo utilizando a linguagem de programação C++.
- Prover funcionalidades como geração de cartelas, sorteio de números, marcação de números, verificação de vitória e exibição de resultados.

# Visão Geral do Código

- **Explicação das classes principais:**
  - BingoCard: Representa uma cartela de Bingo.
  - Player: Representa um jogador do Bingo.
  - BingoGame: Gerencia o jogo de Bingo.

# Classe Player

## - Construtor e Variáveis Membro:

- name: Nome do jogador.
- card: Cartela do jogador.

## - Descrição:

- A classe Player associa um nome a uma cartela de Bingo.

# Classe BingoCard

## - Construtor e Variáveis Membro:

- card: Matriz 2D que armazena os números da cartela.
- marked: Matriz 2D que armazena os estados marcados dos números.

## - Funções Membro:

- generateCard(): Gera a cartela com números aleatórios.
- displayCard(): Exibe a cartela no console.
- markNumber(int number): Marca um número na cartela.
- checkWin(): Verifica se há uma linha, coluna ou diagonal completa.
- operator==: Compara duas cartelas para garantir unicidade.

# Classe BingoGame

## - Construtor e Variáveis Membro:

- players: Vetor de objetos Player.
- drawsCount: Contador de números sorteados.
- numbersDrawn: Vetor de números já sorteados.
- podium: Mapa que rastreia as vitórias dos jogadores.



# Classe BingoGame

## - Funções Membro:

- `play()`: Método principal do jogo.
- `drawNumber()`: Sorteia um número não repetido.
- `generateUniqueCards()`: Gera cartelas únicas para os jogadores.
- `updatePodium(const string& winner)`: Atualiza o pódio com o vencedor.
- `displayPodium()`: Exibe o pódio no console.

# Tipos de Dados Utilizados

- **Strings:** Utilizadas para armazenar nomes dos jogadores.
- **Vetores:**
  - `vector<int>`: Para armazenar números.
  - `vector<vector<int>>`: Para representar a cartela de Bingo.
  - `vector<bool>`: Para marcar números na cartela.
- **Mapas:** `map<string, int>` para rastrear vitórias dos jogadores.
- **Funções Lambda:** Utilizadas em `all_of` para verificar condições nas matrizes.

# Funções e Lógica do Jogo

## - Sorteio de Números:

- `drawNumber()`: Gera um número aleatório que ainda não foi sorteado.

## - Marcação de Números:

- `markNumber(int number)`: Marca o número sorteado na cartela do jogador.

## - Verificação de Vitória:

- `checkWin()`: Verifica se a cartela tem uma linha, coluna ou diagonal completa.

## - Atualização e Exibição do Pódio:

- `updatePodium(const string& winner)`: Atualiza o número de vitórias do jogador.
- `displayPodium()`: Exibe o pódio no console.

# Demonstração

- **Passo a passo do funcionamento do jogo:**
  - Inicialização do jogo com jogadores.
  - Geração e exibição das cartelas.
  - Sorteio de números e marcação nas cartelas.
  - Verificação de vitória e atualização do pódio.
- **Exemplo visual da execução do jogo no console.**

# Conclusão

## - Resumo das Funcionalidades:

- Classes Principais: BingoCard, Player, BingoGame

## - Principais Funções:

- generateCard(), markNumber(int number), checkWin(), drawNumber(), play()

## -Estruturas de Dados:

- Strings: Nomes dos jogadores
- Vetores: Números, cartelas, estados de marcação
- Mapas: Rastreamento de vitórias

# Perguntas

**Ficamos abertos para demais explicações que sejam necessárias.**

**Obrigado pela atenção.**