

# BINGO EM C++

---

PEDRO LUCAS FERREIRA ALVES

211067380

ERICOVIEIRA ZANARDI

180138880



# DESCRIÇÃO DO PROJETO

---

- Cartela 5X5 com números gerados aleatoriamente;
- Números de 1 até 75 são sorteados e preenchidos em cada cartela;
- O primeiro jogador que tiver todos os números da cartela sorteados vence;
- No fim do jogo, é gerado um arquivo que resume a partida.

# FUNCIONALIDADE

---

## BIBLIOTECAS:

- IOSTREAM: Entrada e saída de dados (cin e cout);
- FSTREAM: Criação e edição de arquivos;
- CSTDLIB: Geração de números aleatórios;
- CTIME: Função que inicia a geração dos números;
- SET: Armazena os números sorteados para evitar repetições;
- IOMANIP: Formatação da saída.

```
1  #include <iostream>
2  #include <fstream>
3  #include <cstdlib>
4  #include <ctime>
5  #include <set>
6  #include <iomanip>
7
```

# FUNCIONALIDADE

---

## DEFINIÇÃO DE CONSTANTES

- Define algumas constantes que serão usadas varias vezes durante o código:
  - Tamanho da matriz;
  - Máximo de jogadores;
  - Intervalo de números que podem ser sorteados.

```
10  const int TAMANHO = 5;  
11  const int MAX_JOGADORES = 5;  
12  const int MAX_NUMERO = 75;  
13
```

# FUNCIONALIDADE

---

## GERAÇÃO DAS CARTELAS

- O `set<int>` garante que os números não repitam;
- `Rand()` gera um numero aleatório entre 1 e 75;
- Insere o numero na cartela.

```
14 void gerarCartela(int cartela[TAMANHO][TAMANHO]) {  
15     set<int> numerosUsados;  
16     for (int i = 0; i < TAMANHO; i++) {  
17         for (int j = 0; j < TAMANHO; j++) {  
18             int numero = rand() % MAX_NUMERO + 1;  
19             while (numerosUsados.find(numero) != numerosUsados.end()) {  
20                 numero = rand() % MAX_NUMERO + 1;  
21             }  
22             numerosUsados.insert(numero);  
23             cartela[i][j] = numero;  
24         }  
25     }  
26 }
```



# FUNCIONALIDADE

## IMPRESSAO DAS CARTELAS

- Imprime na tela as cartas
- Se o número tiver sido sorteado, será substituído por X.

```
28 void imprimirCartelas(int numJogadores, string jogadores[MAX_JOGADORES], int cartelas[MAX_JOGADORES][TAMANHO][TAMANHO], bool marcados[MAX_JOGADORES][TAMANHO][TAMANHO]) {
29     cout << "\nCartelas Atualizadas:\n";
30     for (int p = 0; p < numJogadores; p++) {
31         cout << "Jogador: " << jogadores[p] << "\n";
32         for (int i = 0; i < TAMANHO; i++) {
33             for (int j = 0; j < TAMANHO; j++) {
34                 if (marcados[p][i][j]) {
35                     cout << " X ";
36                 } else {
37                     cout << setw(3) << cartelas[p][i][j] << " ";
38                 }
39             }
40             cout << "\n";
41         }
42         cout << "\n";
43     }
44 }
45 }
```

# FUNCIONALIDADE

---

- SALVAR CARTELAS EM ARQUIVO
- Esta função é responsável para que ao final do bingo é criado um arquivo "bingo\_cartelas.txt".
- No arquivo estará, em ordem, as cartelas iniciais de cada jogador, os números sorteados e as Cartelas Atualizadas

# FUNCIONALIDADE

```
41 void salvarCartelas(int numJogadores, string jogadores[MAX_JOGADORES], int cartelas[MAX_JOGADORES][TAMANHO][TAMANHO], bool marcados[MAX_JOGADORES][TAMANHO][TAMANHO], const set<int>& numerosSorteados) {
42     ofstream arquivo("bingo_cartelas.txt");
43     arquivo << "Cartelas Iniciais:\n";
44     for (int p = 0; p < numJogadores; p++) {
45         arquivo << "Jogador: " << jogadores[p] << "\n";
46         for (int i = 0; i < TAMANHO; i++) {
47             for (int j = 0; j < TAMANHO; j++) {
48                 arquivo << setw(3) << cartelas[p][i][j] << " ";
49             }
50             arquivo << "\n";
51         }
52         arquivo << "\n";
53     }
54     arquivo << "\nNúmeros sorteados: ";
55     for (int numero : numerosSorteados) {
56         arquivo << numero << " ";
57     }
58     arquivo << "\n\n";
59     arquivo << "Cartelas Atualizadas:\n";
60     for (int p = 0; p < numJogadores; p++) {
61         arquivo << "Jogador: " << jogadores[p] << "\n";
62         for (int i = 0; i < TAMANHO; i++) {
63             for (int j = 0; j < TAMANHO; j++) {
64                 if (marcados[p][i][j]) {
65                     arquivo << " X ";
66                 } else {
67                     arquivo << setw(3) << to_string(cartelas[p][i][j]) << " ";
68                 }
69             }
70             arquivo << "\n";
71         }
72         arquivo << "\n";
73     }
74     arquivo.close();
75 }
```



# FUNCIONALIDADE

---

- VERIFICAÇÃO VENCEDOR
- Este arquivo irá sempre verificar se a cartela de todos os jogadores estão preenchidas.

```
76 - bool verificarVitoria(bool marcados[TAMANHO][TAMANHO]) {  
77 -     for (int i = 0; i < TAMANHO; i++) {  
78 -         for (int j = 0; j < TAMANHO; j++) {  
79 -             if (!marcados[i][j]) {  
80 -                 return false;  
81 -             }  
82 -         }  
83 -     }  
84 -     return true;  
85 - }
```

# FUNCIONALIDADE

- Int main()
- Lógica do código por completo

```
86 int main() {
87     srand(time(0));
88     int numJogadores;
89     string jogadores[MAX_JOGADORES];
90     int cartelas[MAX_JOGADORES][TAMANHO][TAMANHO];
91     bool marcados[MAX_JOGADORES][TAMANHO][TAMANHO] = {false};
92     cout << "##### VAI COMEÇAR O BINGO! #####\n";
93     cout << "Quantos jogadores? (Máx: 5) ";
94     cin >> numJogadores;
95     while (numJogadores < 1 || numJogadores > MAX_JOGADORES) {
96         cout << "Número inválido! Digite novamente: ";
97         cin >> numJogadores;
98     }
99     for (int i = 0; i < numJogadores; i++) {
100         cout << "Nome do jogador " << i + 1 << ": ";
101         cin >> jogadores[i];
102         gerarCartela(cartelas[i]);
103     }
104     set<int> numerosSorteados;
105     bool vencedorEncontrado = false;
106     while (!vencedorEncontrado) {
```

# FUNCIONALIDADE

```
105     bool vencedorEncontrado = false;
106     while (!vencedorEncontrado) {
107         cout << "Pressione ENTER para sortear um número...";
108         cin.ignore();
109         cin.get();
110         int numeroSorteado = rand() % MAX_NUMERO + 1;
111         while (numerosSorteados.find(numeroSorteado) != numerosSorteados.end()) {
112             numeroSorteado = rand() % MAX_NUMERO + 1;
113         }
114         numerosSorteados.insert(numeroSorteado);
115         cout << "Numero sorteado: " << numeroSorteado << endl;
116         for (int p = 0; p < numJogadores; p++) {
117             for (int i = 0; i < TAMANHO; i++) {
118                 for (int j = 0; j < TAMANHO; j++) {
119                     if (cartelas[p][i][j] == numeroSorteado) {
120                         marcados[p][i][j] = true;
121                     }
122                 }
123             }
124         }
125         imprimirCartelas(numJogadores, jogadores, cartelas, marcados);
126         for (int p = 0; p < numJogadores; p++) {
127             if (verificarVitoria(marcados[p])) {
128                 cout << "\nBINGO! " << jogadores[p] << " venceu!\n";
129                 vencedorEncontrado = true;
130             }
131         }
132         salvarCartelas(numJogadores, jogadores, cartelas, marcados, numerosSorteados);
133     }
134     return 0;
135 }
```

# CONTEÚDOS ABORDADOS

---

- Condicionais;
- Laços de repetição;
- Vetores;
- Função;
- Arquivos.



# DISTRIBUIÇÃO DO DESENVOLVIMENTO

---

Pedro:

- Gerar cartela;
- Imprimir cartela.

Erico:

- Salvar cartelas no arquivo;
- Declarar o vencedor.



# OBRIGADO!

---

PEDRO LUCAS FERREIRA ALVES

211067380

ERICOVIEIRA ZANARDI

180138880