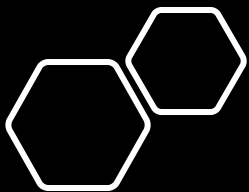


Vetores []

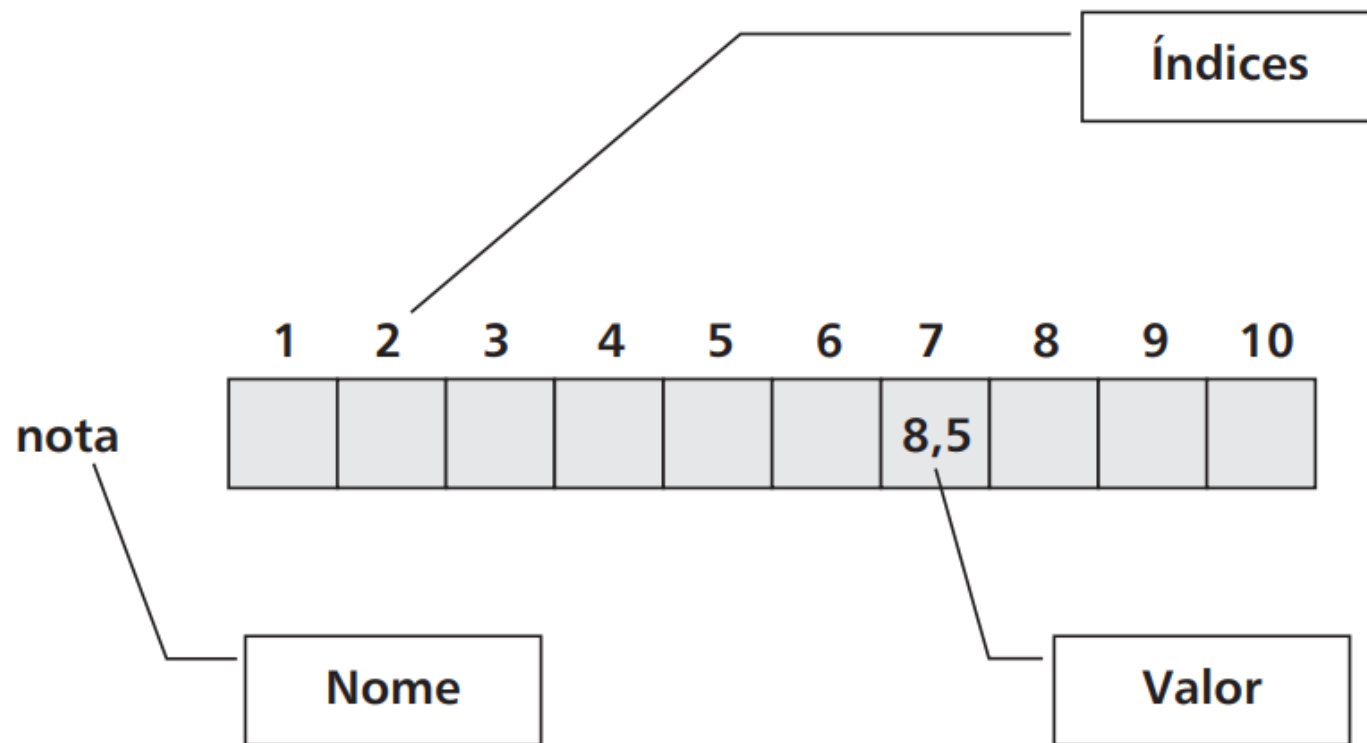
Professora: Fernanda dos Santos Cunha

Estagiário: Vinicius de Aquino Piai



Vetores

- Vetor é uma coleção de variáveis;
- Acessado a partir de um índice;
- Tem o mesmo tipo de dado;



Declaração de Vetores - Visualg

<variável> : vetor [<n>..<>m>] de <tipo de dado>

- variável - Nome da variável ;
- n - Índice do primeiro elemento;
- m - Índice do segundo elemento;
- tipo de dado - tipo de dado do vetor;

Declaração de Vetores - C++

`<tipo de dado> <variável>[<n>;`

- tipo de dado - Tipo de dado do vetor;
- variável - Nome da variável ;
- n - Índice do primeiro elemento;

Declaração de Vetores - Exemplo

Visualg

Var

```
numeros : vetor [1..5] de inteiro  
nomes : vetor [1..4] de caractere  
| valores : vetor [1..10] de real
```

Inicio

Fimalgoritmo

C++

```
#include <iostream>  
  
using namespace std;  
  
int main()  
{  
  
    int numeros[5];  
    string nomes[4];  
    char opcoes[3];  
    float valores[10];  
    bool pago[3];  
  
}
```

Inicialização de Vetores - C++

```
int numeros[5] = {1, 2, 4, 5, 6};  
string nomes[4] = {"Lucas", "Davi", "Maria", "Jessica"};  
char opcoes[3] = {'A', 'B', 'C'};  
float valores[3] = {3.5, 9.0, 5.1};  
bool pago[3] = {true, false, true};
```

```
int numeros[] = {1, 2, 4, 5, 6};  
string nomes[] = {"Lucas", "Davi", "Maria", "Jessica"};  
char opcoes[] = {'A', 'B', 'C'};  
float valores[] = {3.5, 9.0, 5.1};  
bool pago[] = {true, false, true};
```

```
int numeros = {1, 2, 4, 5, 6};  
string nomes = {"Lucas", "Davi", "Maria", "Jessica"};  
char opcoes = {'A', 'B', 'C'};  
float valores = {3.5, 9.0, 5.1};  
bool pago = {true, false, true};
```

```
int numeros[5] = {0};  
string nomes[5] = {""};  
char genero[3] = {'M'};  
float valores[7] = {0.0};  
bool pago[100] = {true};
```

Implemente um algoritmo em Visualg e C++ que leia o nome de 4 alunos e os escreva na tela de forma inversa utilizando um vetor de caracteres e string

Var

```
i : inteiro  
nomes : vetor[1..4] de caractere
```

Inicio

```
para i de 1 ate 4 faca  
    escreva("Digite o nome ", i ,": ")  
    leia(nomes[i])  
fimpara  
  
escreval("=====")  
para i de 4 ate 1 passo - 1 faca  
    escreval(nomes[i])  
fimpara
```

Fimalgoritmo

```
#include <iostream>  
using namespace std;  
#define TMAX 4  
  
int main()  
{  
    string nomes[TMAX];  
  
    for(int i = 0; i < TMAX; i++){  
        cout << "Digite o nome do aluno: ";  
        getline(cin, nomes[i]);  
    }  
  
    cout << "=====" << endl;  
  
    for(int i = TMAX - 1; i >= 0; i--){  
        cout << nomes[i] << endl;  
    }  
}
```

Pesquisa de Elemento



Percorrer vetor;



Encontrar o elemento;

Visualg

Var

```
numeros : vetor [1..10] de inteiro
i, elemento : inteiro
encontrou : logico
```

Inicio

```
numeros[1] <- 5
numeros[2] <- 10
numeros[3] <- 4
numeros[4] <- 4
numeros[5] <- 10
numeros[6] <- 123
numeros[7] <- 99
numeros[8] <- 15
numeros[9] <- 11
numeros[10] <- 0
```

```
escreva("Digite o elemento que deseja procurar: ")
leia(elemento)
```

```
i <- 1
encontrou <- falso
enquanto (i <= 10) e (nao(encontrou)) faca
    se numeros[i] = elemento entao
        encontrou <- verdadeiro
    senao
        i <- i + 1
    fimse
fimenquanto
se encontrou entao
    escreva("Encontrado")
senao
    escreva("Elemento não esta presente")
fimse
```

Fimalgoritmo

The C++ logo is displayed in white text on a dark gray background. The background is a rectangle with a right-pointing arrow shape on its right side. The text "C++" is centered within the rectangle.

C++

```
#include <iostream>
using namespace std;
#define TMAX 10

int main()
{
    int numeros[TMAX] = {1, 2, 3, 4, 1, 6, 7, 8, 1, 10}, i = 0, elemento;
    bool encontrou = false;

    cout << "Digite o elemento que deseja procurar: ";
    cin >> elemento;

    while(i < TMAX and not(encontrou)){
        if(numeros[i] == elemento){
            encontrou = true;
        }else{
            i++;
        }
    }
    if(encontrou){
        cout << "Encontrado na posicao " << i;
    }else{
        cout << "Elemento não esta presente";
    }
}
```



Contar elementos

- Uma extensão da pesquisa de elementos;
- Percorre todo o vetor;

Visualg

Var

```
numeros : vetor [1..10] de inteiro  
i, contador, elemento : inteiro
```

Inicio

```
numeros[1] <- 1  
numeros[2] <- 2  
numeros[3] <- 3  
numeros[4] <- 4  
numeros[5] <- 1  
numeros[6] <- 34  
numeros[7] <- 3  
numeros[8] <- 1  
numeros[9] <- 4  
numeros[10] <- 5  
contador <- 0
```

```
escreva("Digite o elemento que deseja procurar:")  
leia(elemento)
```

```
para i de 1 ate 10 faca  
    se numeros[i] = elemento entao  
        contador <- contador + 1  
    fimse  
fimpara
```

```
escreva("Foram encontradas ", contador)  
escreva(" presencas do elemento ", elemento)
```

Fimalgoritmo

C++

```
#include <iostream>
using namespace std;
#define TMAX 10

int main()
{
    int numeros[TMAX] = {1, 2, 3, 4, 1, 6, 7, 8, 1, 10}, elemento, contador = 0;

    cout << "Digite o elemento que deseja procurar: ";
    cin >> elemento;

    for(int i = 0; i < TMAX; i++){
        if(numeros[i] == elemento){
            contador++;
        }
    }

    cout << "Foram encontradas " << contador << " presencas do elemento " << elemento;
}
```

The background of the image is a dense, overlapping collage of numerous small, rectangular sticky notes. These notes are in various colors, including shades of blue, green, yellow, and pink. Each sticky note features a large, bold, black question mark. The notes are scattered across the entire frame, creating a textured and busy visual effect. In the center of this collage, the word "Dúvidas" is written in a large, white, serif font.

Dúvidas