
SENAI

FUNDAMENTOS DE JAVA
TÓPICOS ABORDADOS

- Vetores e Matrizes
  - O que são?
    - ➢ Criação e Atribuição.
  - Leitura e Iteração.
  - Observações

FUNDAMENTOS DE JAVA:

VETORES E MATRIZES: CRIAÇÃO E ATRIBUIÇÃO

SENAI

FUNDAMENTOS DE JAVA
VETORES: O QUE SÃO?

- Um **vetor** ou **array unidirecional** em Java é uma estrutura de dados que armazena uma sequência de elementos do mesmo tipo, em uma única dimensão. Cada elemento tem um **índice**, começando do **zero**.

```java
int[] numeros = new int[5]; // Cria um vetor de inteiros com 5 elementos
numeros[0] = 10;
numeros[1] = 20;
numeros[2] = 30;
numeros[3] = 40;
numeros[4] = 50;

// Acessando os valores do vetor
System.out.println("Primeiro elemento: " + numeros[0]); // Saída: 10
```

FUNDAMENTOS DE JAVA
VETORES: EXEMPLO DE TIPOS DE DADOS

SENAI

- **Vetor de inteiros (int):**

```java
int[] numeros = new int[5]; // Cria um vetor de inteiros com 5 elementos
numeros[0] = 10;
```

- **Vetor de números reais (double):**

```java
double[] valores = new double[3]; // Cria um vetor de double com 3 elementos
valores[0] = 1.5;
```

- **Vetor de caracteres (char):**

```java
char[] letras = new char[3]; // Cria um vetor de char com 3 elementos
letras[0] = 'A';
```

- **Vetor de textos (String):**

```java
String[] nomes = new String[3]; // Cria um vetor de String com 3 elementos
nomes[0] = "Ana";
```

FUNDAMENTOS DE JAVA
MATRIZES: O QUE SÃO?

- Uma **matriz** ou **array multidirecional** é uma estrutura de dados que pode armazenar elementos em mais de uma dimensão. Uma **matriz bidirecional** pode ser vista como uma **tabela com linhas e colunas**. Cada posição é acessada por dois **índices**: um para a **linha** e outro para a **coluna**.

```java
int[][] matriz = new int[2][3]; // Cria uma matriz com 2 linhas e 3 colunas
matriz[0][0] = 1;
matriz[0][1] = 2;
matriz[0][2] = 3;
matriz[1][0] = 4;
matriz[1][1] = 5;
matriz[1][2] = 6;

// Acessando os valores da matriz
System.out.println("Elemento na posição [0][2]: " + matriz[0][2]); // Saída: 3
```

FUNDAMENTOS DE JAVA
MATRIZES: EXEMPLO DE TIPOS DE DADOS

- **Matizes de inteiros (int):**

```
int[][] matriz = new int[3][3]; // Cria uma matriz 3x3 de inteiros
matriz[0][0] = 1;
```

- **Matizes de números reais (double):**

```
double[][] tabela = new double[2][2]; // Matriz 2x2 de double
tabela[0][0] = 1.5;
```

- **Matizes de caracteres (char):**

```
char[][] letras = new char[2][3]; // Matriz 2x3 de caracteres
letras[0][0] = 'A';
```

- **Matizes de textos (String):**

```
String[][] matriz = new String[3][3];

matriz[0][0] = "João";    // Linha 0, Coluna 0
```

FUNDAMENTOS DE JAVA
VETORES E MATRIZES: ATRIBUIÇÃO

Em Java, você pode criar e atribuir valores a uma matriz (ou array multidimensional) diretamente durante a sua declaração, sem a necessidade de inicializar cada elemento separadamente.

Criação e atribuição: Você pode criar e inicializar vetores diretamente durante a declaração usando chaves {} para atribuir os valores.

```
int[] numeros = {1, 2, 3, 4, 5}; // Criação e inicialização do vetor de inteiros
String[] nomes = {"Ana", "Bruno", "Clara"}; // Criação e inicialização do vetor de Strings
```

FUNDAMENTOS DE JAVA
VETORES E MATRIZES: ATRIBUIÇÃO

SENAI

Criação de uma matriz 3x3 de inteiros com valores predefinidos:

```
// Criação e inicialização da matriz 3x3 com valores
int[][] matriz = {
    {1, 2, 3},   // Primeira linha da matriz
    {4, 5, 6},   // Segunda linha da matriz
    {7, 8, 9}    // Terceira linha da matriz
};
```

Criação e atribuição de valores a um vetor de Strings

```
// Criação e inicialização de uma matriz de Strings
String[][] nomes = {
    {"Alice", "Bob", "Carlos"},   // Primeira linha
    {"Diana", "Eva", "Fernando"}, // Segunda linha
    {"Gabriel", "Hugo", "Irene"}  // Terceira linha
};
```

FUNDAMENTOS DE JAVA:

VETORES E MATRIZES: LEITURA E ITERAÇÃO

SENAI

FUNDAMENTOS DE JAVA
VETORES E MATRIZES: LEITURA

Como fazer a leitura do valor de apenas um item do vetor ou matriz?

- **Um índice** é uma posição numérica que indica onde um determinado valor está armazenado em uma estrutura de dados, como um vetor (array unidimensional) ou uma matriz (array bidimensional).
  - **Em um vetor** (array unidimensional), você acessa os elementos por meio de um único índice, que varia de 0 até n-1, onde n é o número de elementos no vetor.
  - **Em uma matriz** (array bidimensional), o índice é composto por dois números: um para a linha e outro para a coluna, também começando em 0.

FUNDAMENTOS DE JAVA
VETORES E MATRIZES: LEITURA

SENAI

Exemplo de Leitura de um Vetor:

```java
public class LeituraVetor {
    public static void main(String[] args) {
        // Criação e inicialização de um vetor de inteiros
        int[] numeros = {10, 20, 30, 40, 50};

        // Leitura do valor na posição de índice 2 (terceiro elemento)
        int valor = numeros[2]; // O valor será 30

        // Exibe o valor lido
        System.out.println("O valor na posição 2 é: " + valor);
    }
}
```

FUNDAMENTOS DE JAVA
VETORES E MATRIZES: LEITURA

SENAI

Exemplo de Leitura de uma matriz:

```java
public class LeituraMatriz {
    public static void main(String[] args) {
        // Criação e inicialização de uma matriz 2x3 de inteiros
        int[][] matriz = {
            {1, 2, 3},
            {4, 5, 6}
        };

        // Leitura do valor na posição [1][2] (segunda linha, terceira coluna)
        int valor = matriz[1][2]; // O valor será 6

        // Exibe o valor lido
        System.out.println("O valor na posição [1][2] é: " + valor);
    }
}
```

FUNDAMENTOS DE JAVA
VETORES E MATRIZES: ITERAÇÃO

Como acessar todos os valores do vetor ou matriz de forma automática?

- **Iteração** é o processo de percorrer os elementos de uma coleção, como vetores e matrizes, de forma sistemática para acessar ou modificar seus valores. Em Java, a iteração geralmente é feita utilizando estruturas de controle como os laços **for**, **while** ou **for-each**.

FUNDAMENTOS DE JAVA
VETORES E MATRIZES: ITERAÇÃO

SENAI

Exemplo de Iteração de um Vetor com <u>for</u>:

```java
public class IteracaoVetor {
    public static void main(String[] args) {
        // Criação e inicialização de um vetor de inteiros
        int[] numeros = {10, 20, 30, 40, 50};

        // Iteração do vetor utilizando um laço for
        for (int i = 0; i < numeros.length; i++) {
            // Exibe o elemento atual do vetor
            System.out.println("Elemento na posição " + i + ": " + numeros[i]);
        }
    }
}
```

FUNDAMENTOS DE JAVA
VETORES E MATRIZES: ITERAÇÃO

SENAI

Exemplo de Iteração de um Vetor com <u>for-each</u>:

```java
public class IteracaoVetorForEach {
    public static void main(String[] args) {
        // Criação e inicialização de um vetor de inteiros
        int[] numeros = {10, 20, 30, 40, 50};

        // Iteração utilizando for-each
        for (int numero : numeros) {
            // Exibe o elemento atual do vetor
            System.out.println("Elemento: " + numero);
        }
    }
}
```

FUNDAMENTOS DE JAVA
VETORES E MATRIZES: ITERAÇÃO

Iterações em uma matriz:

- Uma **matriz** (ou array bidimensional) é uma coleção de vetores, onde cada vetor é uma linha da matriz. Portanto, a matriz tem **linhas** e **colunas**. Para acessar os elementos de uma matriz, precisamos de **dois índices**: um para a linha e outro para a coluna.

```
// Criação e inicialização da matriz 3x3 com valores
int[][] matriz = {
    {1, 2, 3},   // Primeira linha da matriz
    {4, 5, 6},   // Segunda linha da matriz
    {7, 8, 9}    // Terceira linha da matriz
};
```

FUNDAMENTOS DE JAVA
VETORES E MATRIZES: ITERAÇÃO

SENAI

Exemplo de Iteração de um matriz com for:

```java
public class IteracaoMatriz {  new *
    public static void main(String[] args) {  new *
        // Criação e inicialização de uma matriz 2x3 de inteiros
        int[][] matriz = {
                {1, 2, 3},
                {4, 5, 6}
        };

        // Iteração pelas linhas da matriz
        for (int i = 0; i < matriz.length; i++) {
            // Iteração pelas colunas de cada linha
            for (int j = 0; j < matriz[i].length; j++) {
                // Exibe o elemento atual da matriz
                System.out.println("Elemento na posição [" + i + "][" + j + "]: " + matriz[i][j]);
            }
        }
    }
}
```

FUNDAMENTOS DE JAVA
VETORES E MATRIZES: ITERAÇÃO

SENAI

Exemplo de Iteração de um matriz com <u>for-each</u>:

```java
public class IteracaoMatriz {  new *
    public static void main(String[] args) {  new *
        // Criação e inicialização de uma matriz 2x3 de inteiros
        int[][] matriz = {
                {1, 2, 3},
                {4, 5, 6}
        };

        // Iteração pela matriz utilizando for-each
        for (int[] linha : matriz) {
            for (int elemento : linha) {
                // Exibe o elemento atual da matriz
                System.out.println("Elemento: " + elemento);
            }
        }
    }
}
```

FUNDAMENTOS DE JAVA:

VETORES E MATRIZES: OBSERVAÇÕES

SENAI

FUNDAMENTOS DE JAVA
VETORES E MATRIZES: OBSERVAÇÕES

Homogeneidade:

- Todos os elementos de um vetor ou matriz devem ser do mesmo tipo de dado. Por exemplo, se um vetor é declarado como int[], ele só pode armazenar números inteiros.

FUNDAMENTOS DE JAVA
VETORES E MATRIZES: OBSERVAÇÕES

SENAI

Memória Alocada:

- O tipo de dado também determina quanto espaço em memória cada elemento do vetor ou matriz vai ocupar:
  - **int:** 4 bytes
  - **double:** 8 bytes
  - **boolean:** 1 bit (normalmente agrupado em 1 byte)
  - **char:** 2 bytes

FUNDAMENTOS DE JAVA
VETORES E MATRIZES: OBSERVAÇÕES

Acesso por Índice:

- Em vetores, o acesso a cada elemento é feito por um índice que começa do zero. Em matrizes, você acessa os elementos utilizando dois índices: um para a linha e outro para a coluna.
  - **Exemplo de vetor:** vetor[2] acessa o terceiro elemento.
  - **Exemplo de matriz:** matriz[1][2] acessa o elemento na segunda linha e terceira coluna.

FUNDAMENTOS DE JAVA
VETORES E MATRIZES: OBSERVAÇÕES

Estruturas Multidimensionais:

- Em matrizes, você pode ter mais de duas dimensões, mas no ensino básico, normalmente, usamos matrizes bidimensionais. O conceito pode ser expandido para mais dimensões se necessário.

FUNDAMENTOS DE JAVA
VETORES E MATRIZES: OBSERVAÇÕES

Palavra chave *new*:

- Em Java, o uso da palavra-chave **new** ao criar vetores e matrizes é necessário porque arrays são considerados objetos.
- A função do **new** é alocar espaço na memória para o array e inicializar o objeto do array.

FUNDAMENTOS DE JAVA
VETORES E MATRIZES: OBSERVAÇÕES

Tamanho fixo:

- Em **Java**, vetores e matrizes têm **tamanho fixo**, ou seja, uma vez que você define o tamanho de um array ou matriz, esse tamanho **não pode ser alterado**. Isso significa que você não pode aumentar ou diminuir o número de elementos em um array ou matriz após a sua criação.

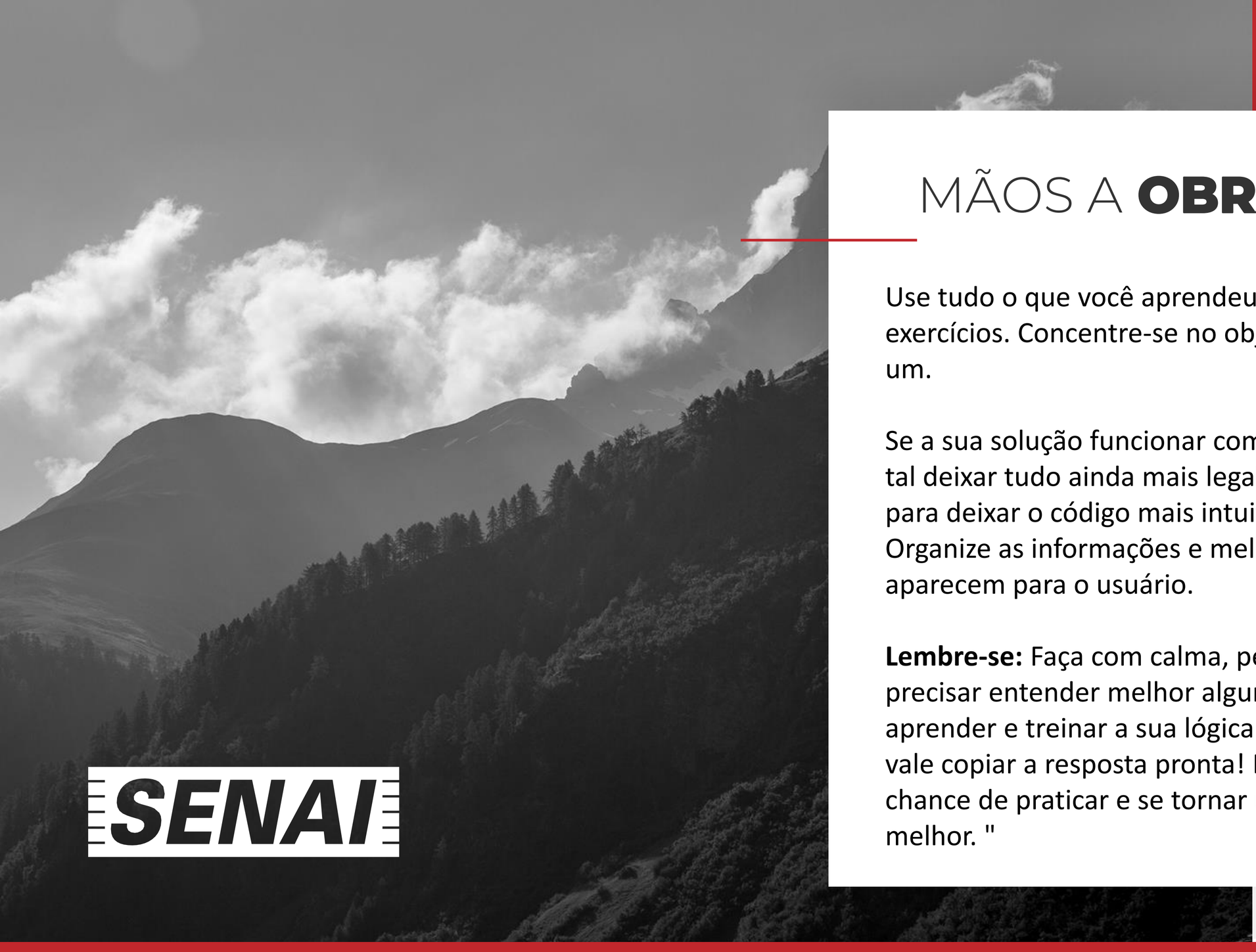
FUNDAMENTOS DE JAVA

EXERCÍCIOS:

VETORES E **MATRIZES:**

- CRIAÇÃO
- ATRIBUIÇÃO
- LEITURA
- ITERAÇÃO

SENAI



SENAI

MÃOS A **OBRA**

Use tudo o que você aprendeu para resolver os exercícios. Concentre-se no objetivo principal de cada um.

Se a sua solução funcionar como o exercício pede, que tal deixar tudo ainda mais legal? Use a sua criatividade para deixar o código mais intuitivo e fácil de entender. Organize as informações e melhore os textos que aparecem para o usuário.

Lembre-se: Faça com calma, peça ajuda quando precisar entender melhor alguma coisa. O importante é aprender e treinar a sua lógica de programação. Não vale copiar a resposta pronta! Esses exercícios são a sua chance de praticar e se tornar um programador ainda melhor. "

FUNDAMENTOS DE JAVA:
EXERCÍCIOS: VETORES

SENAI

Exercício 1: Soma de Elementos Inteiros em um Vetor

Crie um programa que armazene 5 números inteiros em um vetor. Em seguida, some os elementos e exiba o resultado.

Exercício 2: Busca de Caracteres em um Vetor

Armazene 6 caracteres em um vetor de Strings. Pergunte ao usuário uma letra e informe se ela está presente no vetor e em qual posição do vetor.

FUNDAMENTOS DE JAVA:
EXERCÍCIOS: VETORES

Exercício 3: Contagem de Valores Booleanos

Crie um vetor de 8 posições booleanas, onde o usuário pode inserir true ou false. Conte quantos valores true foram inseridos.

Exercício 4: Média de Notas (double)

Leia 4 notas (valores decimais) e armazene-as em um vetor. Calcule e exiba a média dessas notas.

FUNDAMENTOS DE JAVA:
EXERCÍCIOS: VETORES

SENAI

Exercício 5: Vetor de Strings.

Armazene 3 nomes em um vetor de Strings. Pergunte ao usuário qual nome ele deseja verificar, e exiba se o nome está ou não presente no vetor.

FUNDAMENTOS DE JAVA:
EXERCÍCIOS: MATRIZES

SENAI

Exercício 6: Preenchendo uma Matriz

Crie uma matriz 2x2 de inteiros. Preencha a matriz com números fornecidos pelo usuário e exiba os valores inseridos.

Exercício 7: Soma de Valores em uma Matriz

Crie uma matriz 3x3 e peça ao usuário para preencher com números inteiros. Em seguida, some todos os valores da matriz e exiba o total.

FUNDAMENTOS DE JAVA:
EXERCÍCIOS: MATRIZES

Exercício 8: Matriz de Números Reais (double)

Crie uma matriz 2x2 de números reais (double). Preencha a matriz com valores fornecidos pelo usuário, e depois exiba a soma dos valores de cada linha.

Exercício 9: Contando Números Pares em uma Matriz

Crie uma matriz 4x4 de números inteiros. Conte e exiba quantos números pares estão presentes na matriz.

FUNDAMENTOS DE JAVA:
EXERCÍCIOS: MATRIZES

SENAI

Exercício 10: Maior Elemento de uma Matriz

Crie uma matriz 3x3 de números inteiros. Encontre e exiba o maior valor armazenado nessa matriz.

FUNDAMENTOS DE JAVA:
DESAFIO: MATRIZES

SENAI

Sistema de cadastro utilizando uma matriz.

- Você deverá desenvolver um programa em Java que simule um sistema de cadastro de pessoas utilizando uma matriz para armazenar os dados. A primeira linha da matriz será o cabeçalho da tabela, contendo as colunas: "ID", "Nome", "E-mail" e "Telefone". As linhas subsequentes serão utilizadas para armazenar os dados das pessoas cadastradas.

FUNDAMENTOS DE JAVA:
DESAFIO: MATRIZES

SENAI

O programa deverá realizar as seguintes funções:

1. **Entrada de dados:**
    - O usuário deverá informar o número de pessoas que deseja cadastrar.
    - Para cada pessoa, solicite as seguintes informações: ID (número), Nome, E-mail e Telefone.

```
Quantas pessoas você deseja cadastrar? 3
Cadastro da pessoa 1:
ID: 1
Nome: Rafael
E-mail: rafael@email.com
Telefone: 0000-0000
```

FUNDAMENTOS DE JAVA :
DESAFIO: MATRIZES

SENAI

2. **Armazenamento de dados:**
    - Armazene os dados informados pelo usuário em uma matriz de Strings.
    - A matriz deve ter tamanho dinâmico baseado na quantidade de pessoas a serem cadastradas.

FUNDAMENTOS DE JAVA :
DESAFIO: MATRIZES

SENAI

3. **Exibição de dados:**
   - Após o cadastro, o programa deverá exibir os dados cadastrados em formato de tabela, onde a primeira linha corresponde ao cabeçalho.

Exemplo:

```
ID      Nome      E-mail             Telefone
1       João      joao@email.com     1234-5678
2       Maria     maria@email.com    9876-5432
```


SENAI
DEPARTAMENTO REGIONAL
DE SÃO PAULO
www.sp.senai.br