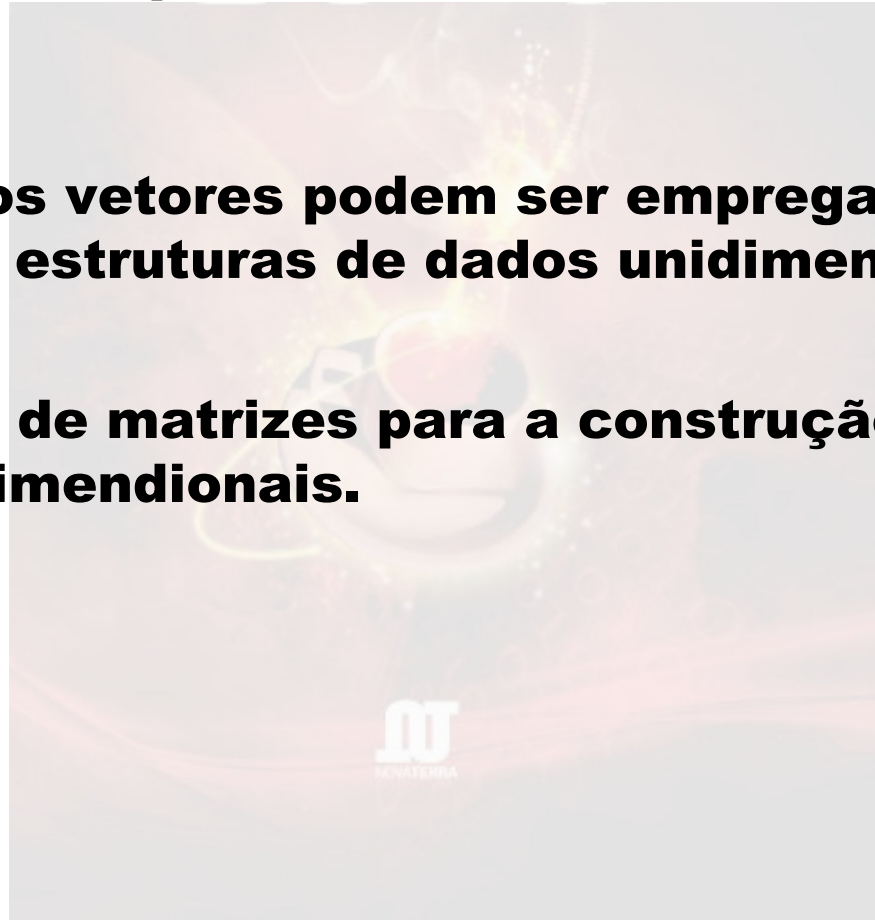


Capítulo 9

Vetores e Matrizes

Objetivos do Capítulo

- Apresentar os dois tipos fundamentais de estruturas de dados estáticas disponíveis no Java: os vetores e as matrizes.**
- Indicar como os vetores podem ser empregados para a construção de estruturas de dados unidimensionais.**
- Explorar o uso de matrizes para a construção de estruturas de dados multidimensionais.**



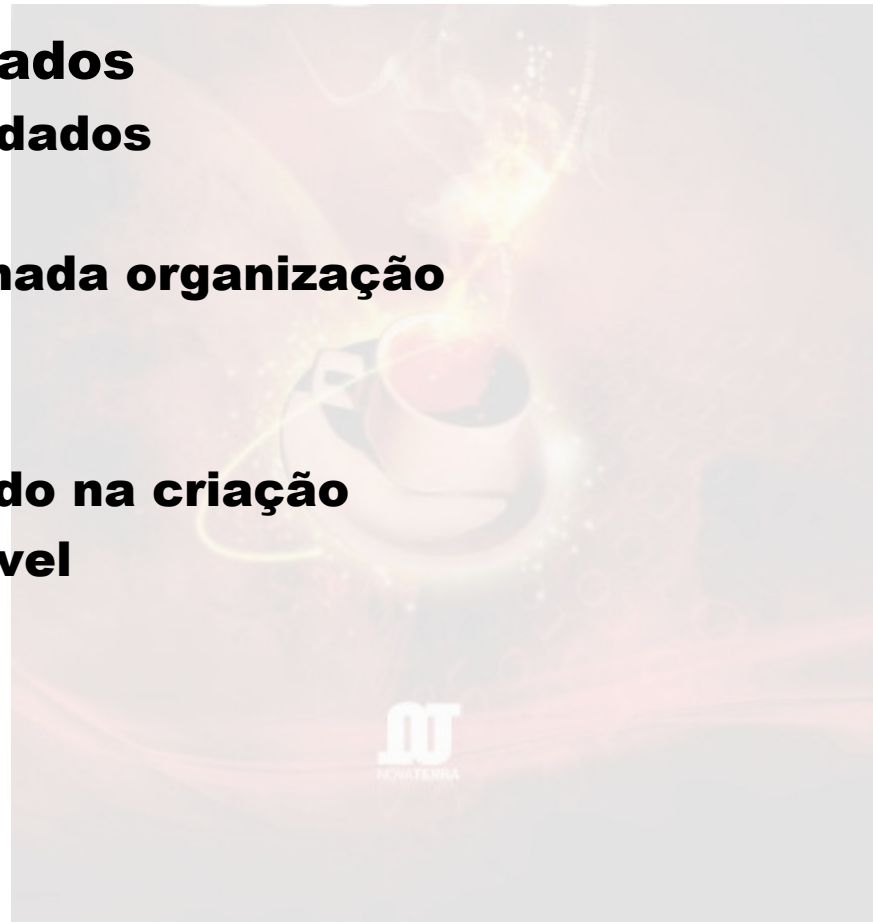
Introdução

- ❑ **Variável: armazena um dado**

- ❑ **Estrutura de dados**
 - **Conjunto de dados**
 - **Agrupados**
 - **Sob determinada organização**

- ❑ **Estática**
 - **Limite definido na criação**
 - **Limite imutável**

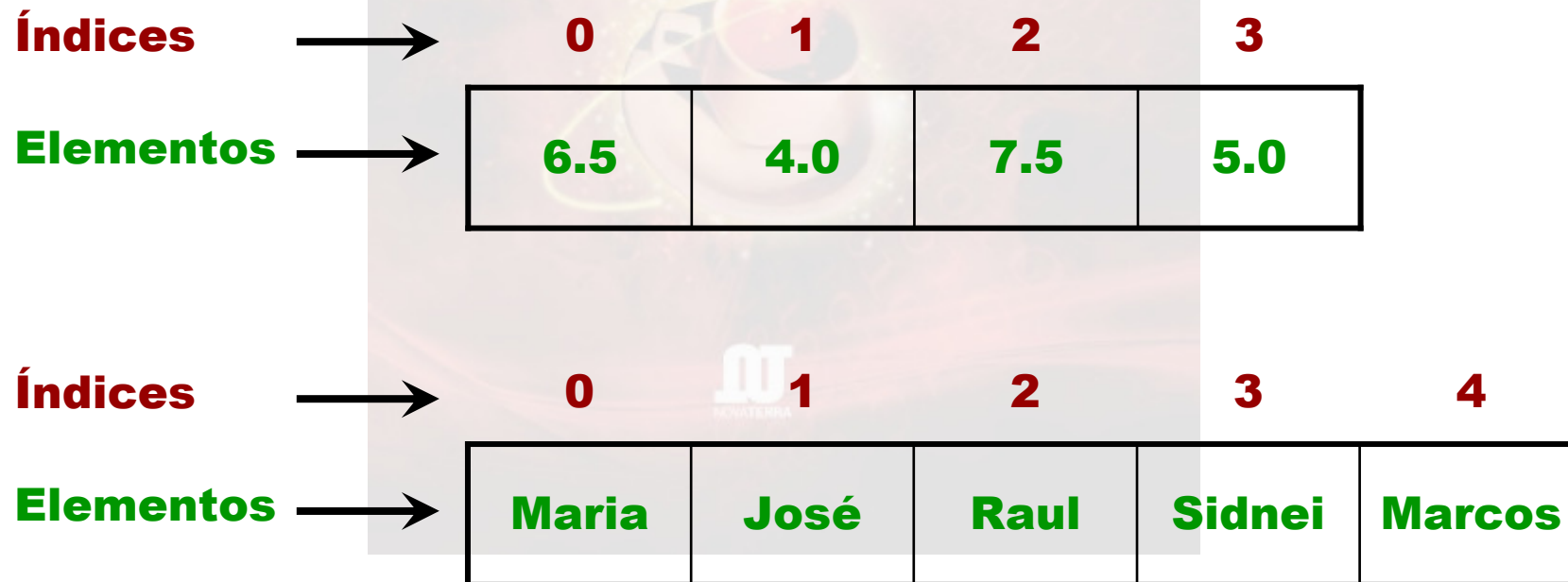
- ❑ **Tipos**
 - **Vetores**
 - **Matrizes**



Vetores

□ Estruturas unidimensionais

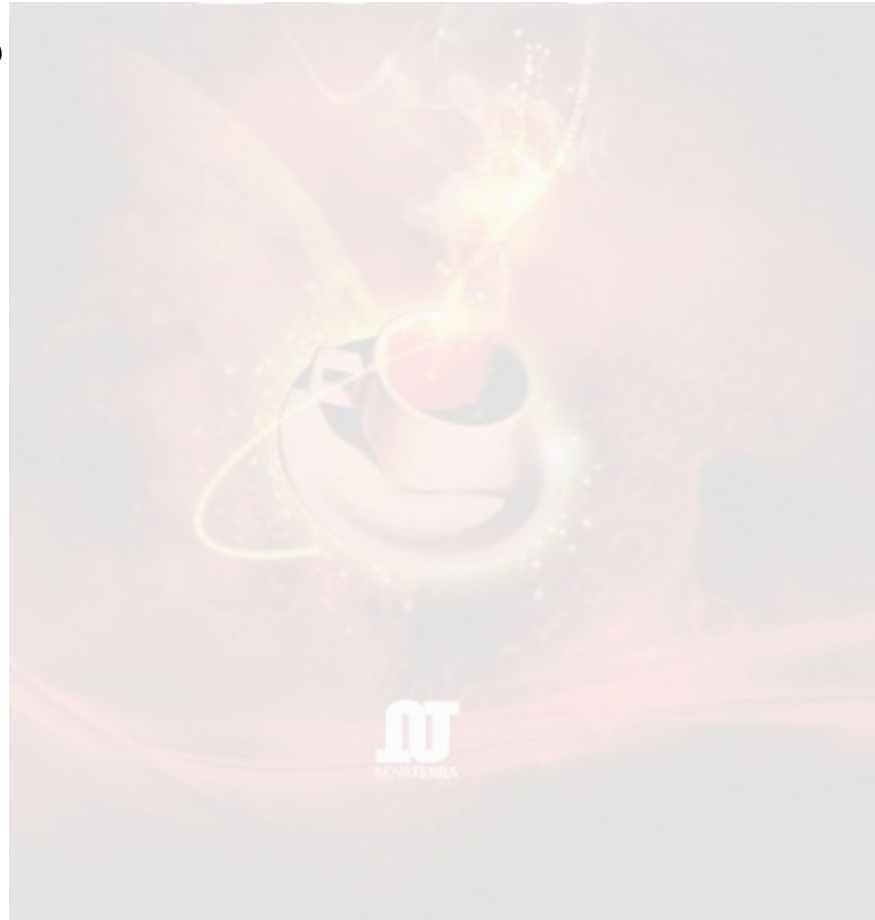
- **Tamanho:** limite de valores
- **Tipo:** o que pode armazenar
- **Elemento:** cada dado armazenado
- **Índice:** posição de cada elemento



Vetores

❑ Operações:

- **Declaração**
- **Instanciação**
- **Inicialização**
- **Consulta**



Vetores

❑ Declaração:

- Similar às variáveis
- Duas formas

❑ Sintaxe:

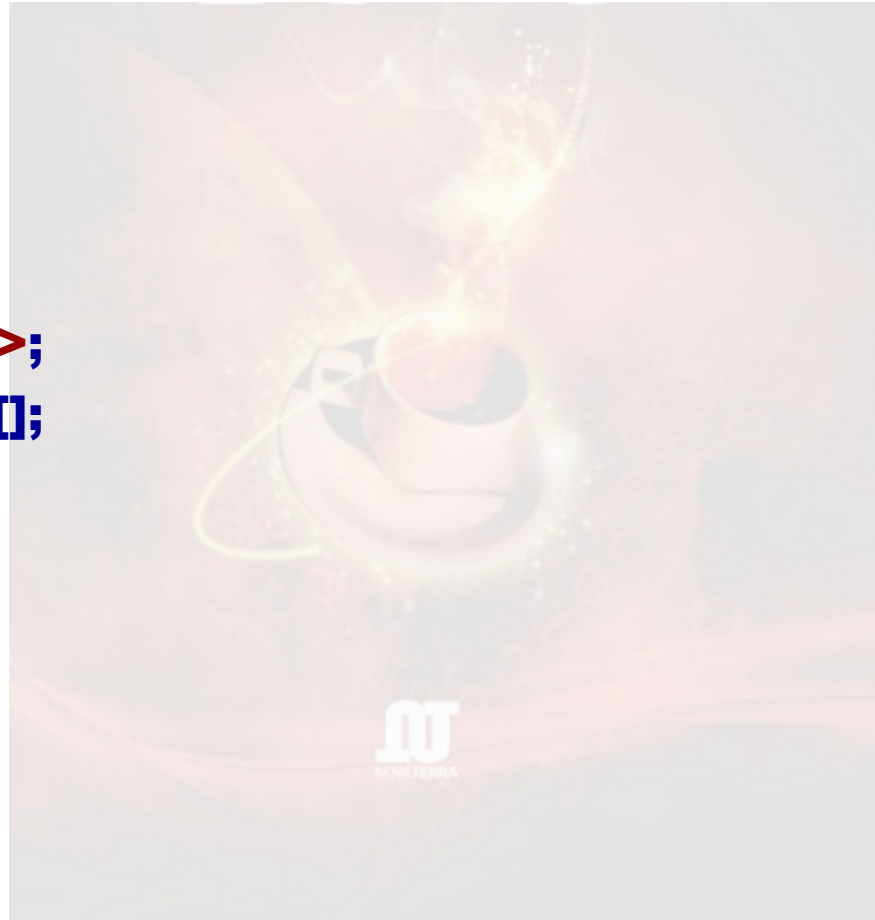
```
<tipo>[] <nome>;
```

```
<tipo> <nome>[];
```

❑ Exemplos:

```
int[] it;
```

```
char ch[];
```



Vetores

❑ Instanciação:

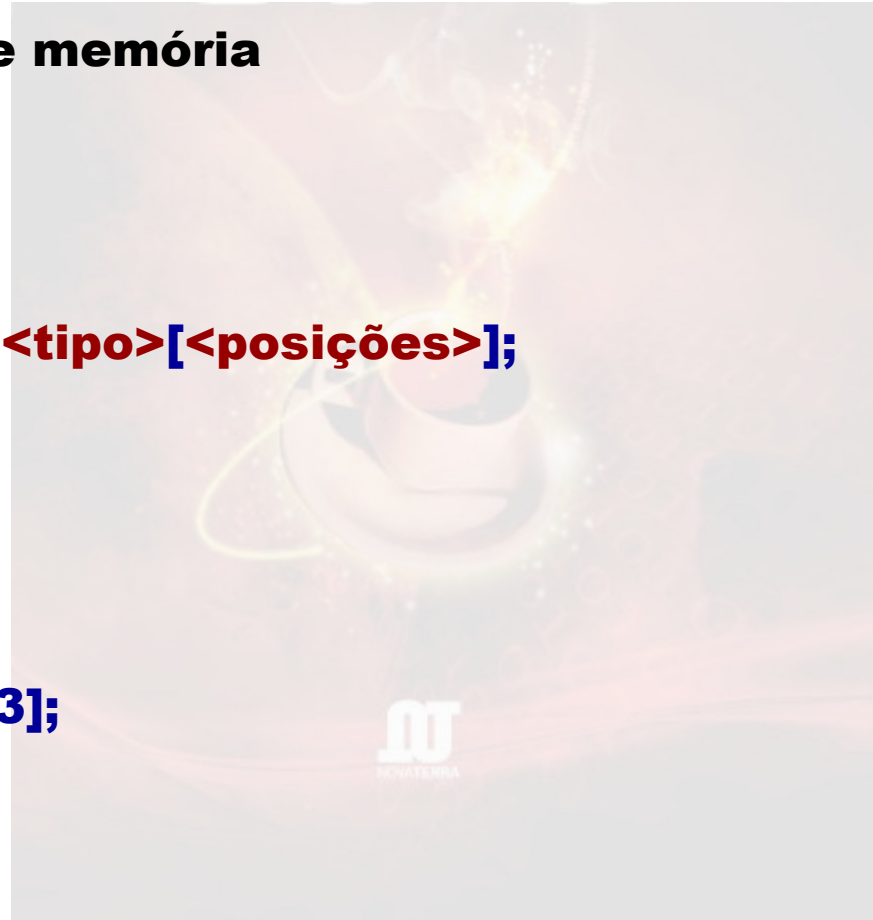
- Criação do vetor
- Atribuição de memória

❑ Sintaxe:

<nome> = new <tipo>[<posições>;

❑ Exemplos:

```
it = new int[4];  
ch = new char[3];
```



Vetores

❑ Declaração e instanciação:

❑ Sintaxe:

```
<tipo>[] <nome> = new <tipo>[<posições>];  
<tipo> <nome>[] = new <tipo>[<posições>];
```

❑ Exemplos:

```
int[] it = new int[4];  
char ch[] = new char[3];
```


Vetores

❑ Inicialização:

- Armazenar valores

❑ Sintaxe:

<nome>[<posição>] = <valor>;

❑ Exemplos:

it[0] = 5;

it[1] = 8;

it[2] = 3;

it[3] = 9;

ch[0] = 'A';

ch[1] = 'B';

ch[2] = 'C';



Vetores

❑ Consulta:

➤ Recuperação de valores

❑ Sintaxe:

<nome>[<posição>]

❑ Exemplos:

```
int it2 = it[3];
```

```
int it3 = (it[0] + it[1] + it[2]) / it[3];
```

```
System.out.println("A posição 1 do vetor ch contém: " + ch[1]);
```

Vetores

❑ Declaração, instanciação e inicialização:

❑ Sintaxe:

```
<tipo>[] <nome> = {<valor1>, <valor2>, ... , <valorN>;
```

```
<tipo> <nome>[] = {<valor1>, <valor2>, ... , <valorN>;
```

❑ Exemplos:

```
int[] it = {5,8,3,9};
```

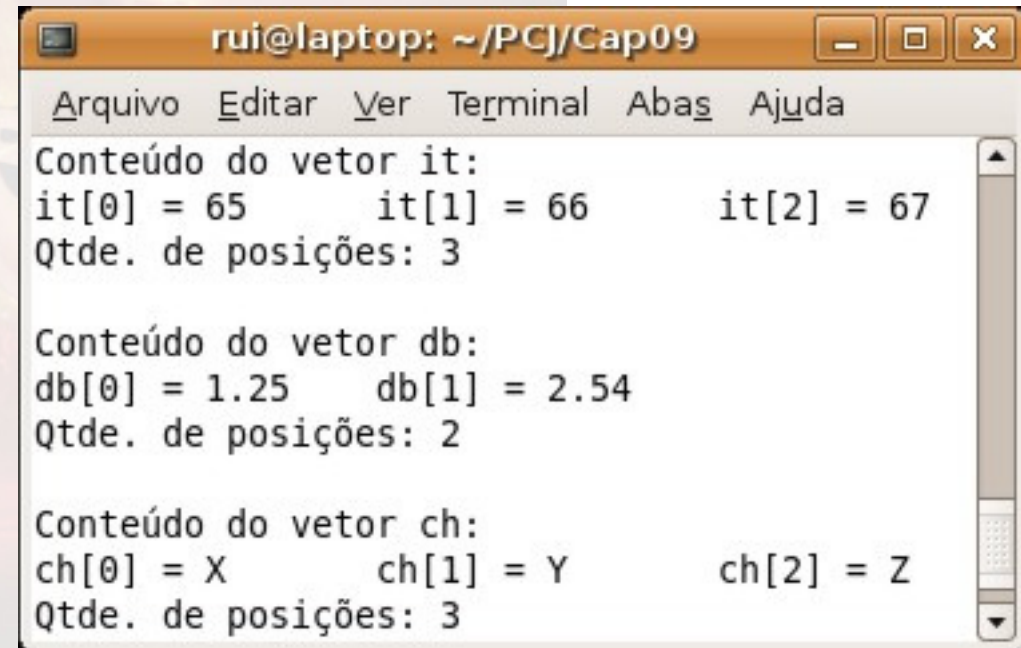
```
char ch[] = {'A','B','C'};
```

Vetores

❑ Código 9.1 – Vetor.java

- **Declaração, instanciação, inicialização e consulta a vetor de números inteiros, números decimais e de caracteres.**
- **Uso do laço for para percorrer o vetor de caracteres.**

65	66	67
1.25	2.54	
X	Y	Z



```
rui@laptop: ~/PCJ/Cap09
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
Conteúdo do vetor it:
it[0] = 65      it[1] = 66      it[2] = 67
Qtde. de posições: 3

Conteúdo do vetor db:
db[0] = 1.25    db[1] = 2.54
Qtde. de posições: 2

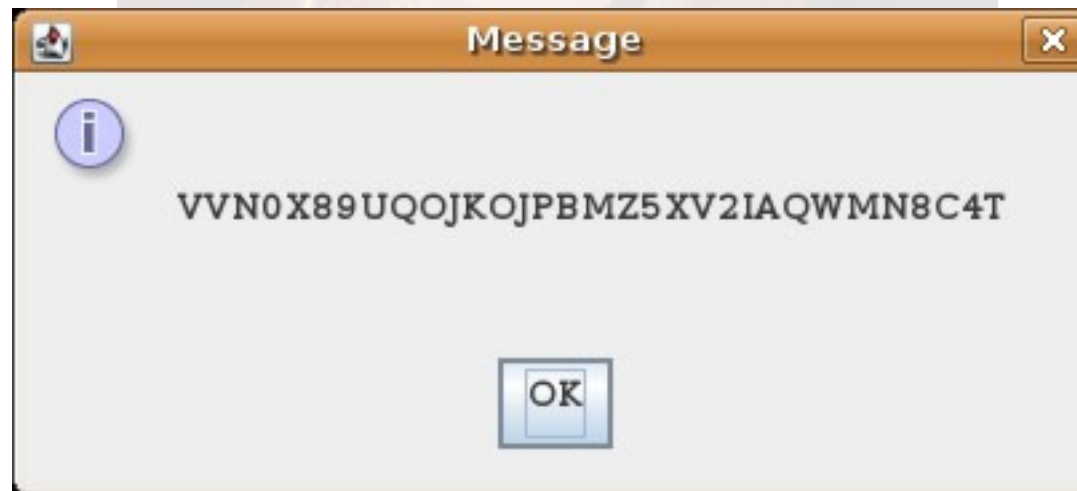
Conteúdo do vetor ch:
ch[0] = X      ch[1] = Y      ch[2] = Z
Qtde. de posições: 3
```

Vetores

❑ Código 9.2 – Codigo.java

- **Crie um programa que gere um código de autenticação para o usuário.**
- **Composição do código: 32 caracteres alfanuméricos.**
- **Passos:**
 - ❖ **Crie um vetor de caracteres com os números e letras do alfabeto.**
 - ❖ **Utilize um laço para sortear 32 destes caracteres.**

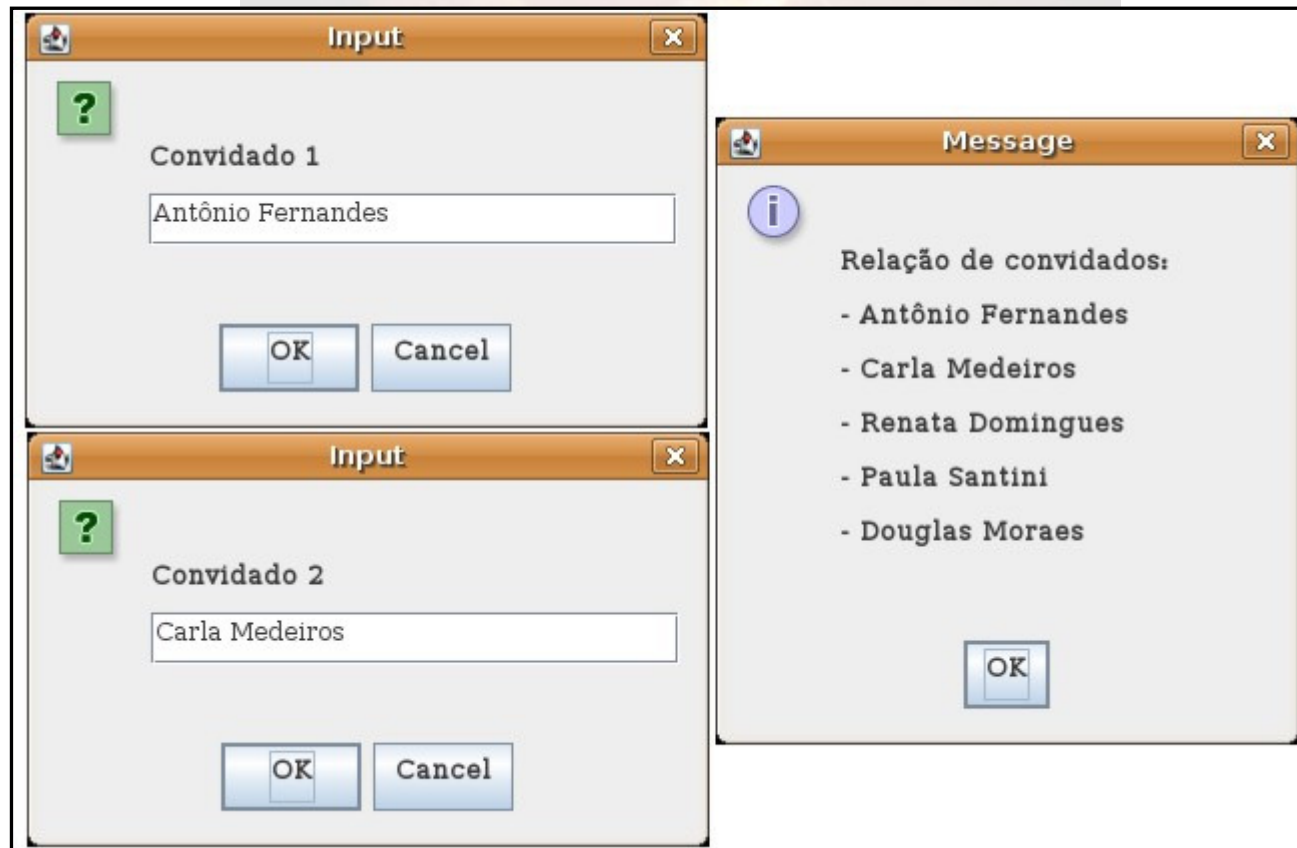
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



Vetores

❑ Código 9.3 – Convidados.java

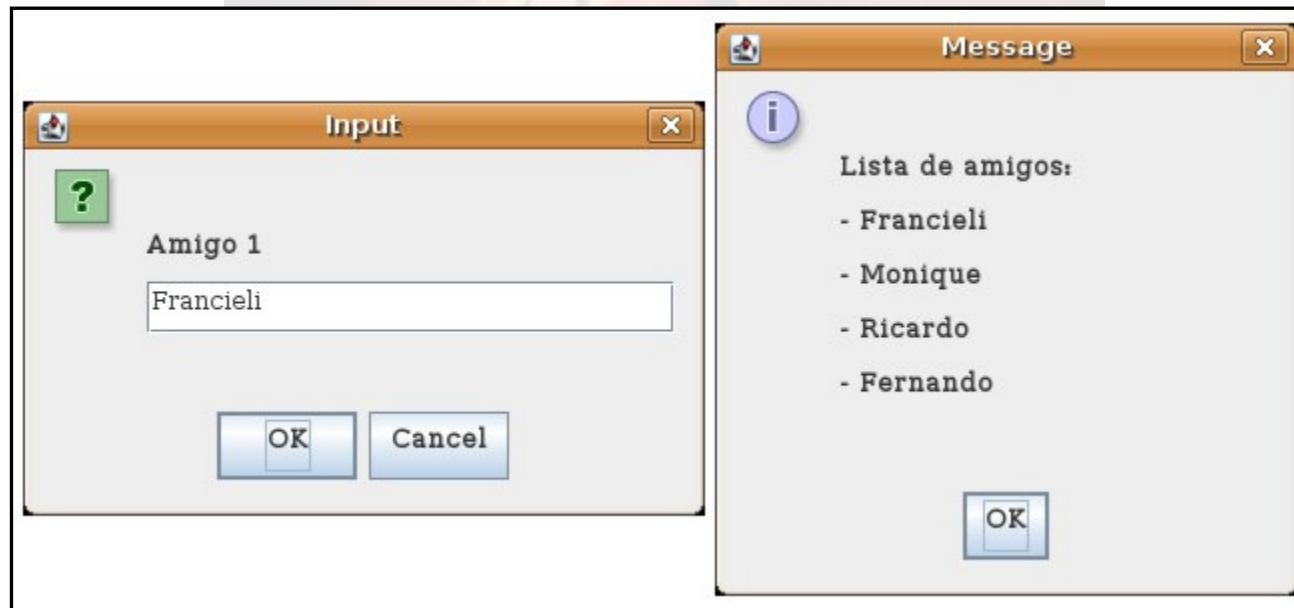
- Crie um vetor capaz de armazenar nomes de **até 20 convidados**.
- Utilize um laço for para captar e armazenar os nomes.
- Utilize um laço for para recuperar e exibir a lista de convidados.



Vetores

❑ Código 9.4 – Amigos.java

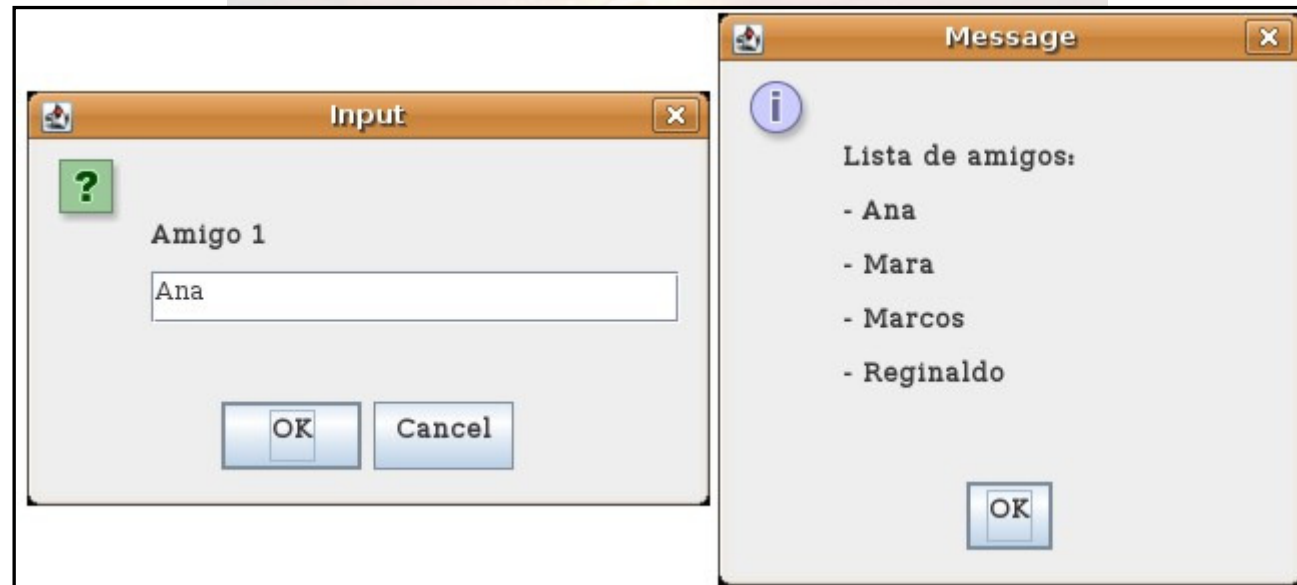
- **Crie um vetor capaz de armazenar nomes de amigos.**
- **O usuário poderá indicar quantos amigos deseja registrar.**
 - ❖ **Este número deve ser maior que zero.**
 - ❖ **O usuário poderá registrar um número menor de amigos.**
- **Utilize um laço for para captar e armazenar os nomes.**
- **Utilize um laço for para recuperar e exibir a lista de amigos.**



Vetores

❑ Código 9.5 – ForEach.java

- Crie um vetor capaz de armazenar nomes de **até 10 amigos**.
- Utilize um laço for para captar e armazenar os nomes.
- Utilize um laço for sem contador para recuperar e exibir a lista de amigos (**laço foreach**).



Matrizes

□ Estruturas multidimensionais

- **Duas ou mais dimensões**
- **Tamanho: limite de valores (linhas x colunas)**
- **Tipo: o que pode armazenar**
- **Elemento: cada dado armazenado**
- **Índice: posição de cada elemento (linha + coluna)**

The diagram illustrates a 2D array with three rows and two columns. The rows are indexed from 0 to 2, and the columns are indexed from 0 to 1. The data is as follows:

0	Maria	9988-2244
1	José	9972-3322
2	Raul	9925-4497

Labels: **Linhas** (rows) and **Colunas** (columns).

Matrizes

❑ Operações:

- Declaração
- Instanciação
- Inicialização
- Consulta



Matrizes

❑ Declaração:

➤ Duas formas

❑ Sintaxe:

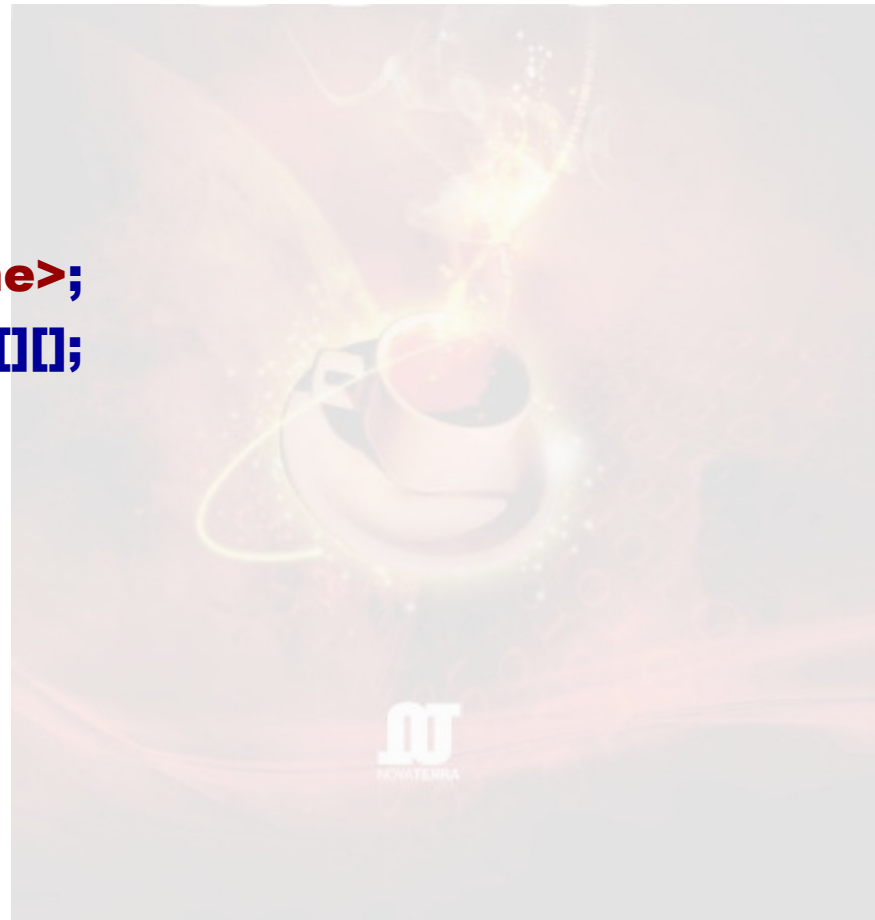
<tipo> [][] <nome>;

<tipo> <nome> [][];

❑ Exemplos:

int [][] it;

char ch [][];



Matrizes

❑ Instanciação:

- Criação da matriz
- Atribuição de memória

❑ Sintaxe:

<nome> = new <tipo>[<linhas>] [<colunas>];

❑ Exemplos:

it = new int[3][4];

ch = new char[2][3];

Matrizes

❑ Declaração e instanciação:

❑ Sintaxe:

```
<tipo> [][] <nome> = new <tipo> [<linhas>] [<colunas>];
```

```
<tipo> <nome> [][] = new <tipo> [<linhas>] [<colunas>];
```

❑ Exemplos:

```
int [][] it = new int [3][4];
```

```
char ch [][] = new char [2][3];
```


Matrizes

❑ Inicialização:

- Armazenar valores

❑ Sintaxe:

<nome>[<linha>] [<coluna>] = <valor>;

❑ Exemplos:

ch[0][0] = 'A';

ch[0][1] = 'B';

ch[0][2] = 'C';

ch[1][0] = 'D';

ch[1][1] = 'E';

ch[1][2] = 'F';



A	B	C
D	E	F

Matrizes

❑ Consulta:

- Recuperação de valores

❑ Sintaxe:

<nome>[<linha>] [<coluna>]

❑ Exemplos:

```
int it2 = it[1][2] / 2;  
char ch2 = ch[0][1];
```

54	35	47	92
19	74	58	42
75	66	81	13

A	B	C
D	E	F

Matrizes

❑ Declaração, instanciação e inicialização:

❑ Sintaxe:

```
<tipo>[][] <nome> = {  
    {<vlrLin0Col0>,<vlrLin0Col1>, ... ,<vlrLin0ColN>},  
    {<vlrLin1Col0>,<vlrLin1Col1>, ... ,<vlrLin1ColN>},  
    {<vlrLin2Col0>,<vlrLin2Col1>, ... ,<vlrLin2ColN>},  
    {<vlrLinNCol0>,<vlrLinNCol1>, ... ,<vlrLinNColN>},  
};
```

❑ Exemplos:

```
int[][] it = {{54,35,47,92},{19,74,58,42},{75,66,81,13}};
```

```
char[][] ch = {{'A','B','C'},{'D','E','F'}};
```

54	35	47	92
19	74	58	42
75	66	81	13

A	B	C
D	E	F

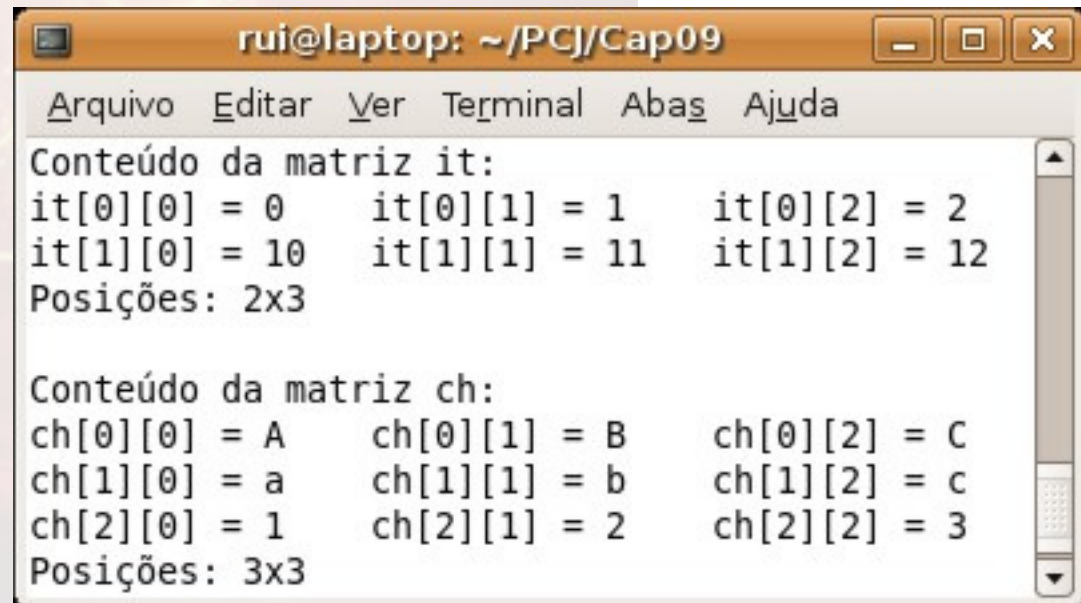
Matrizes

❑ Código 9.6 – Matriz.java

- **Declaração, instanciação, inicialização e consulta a matriz de números inteiros e de caracteres.**
- **Uso do laço for para percorrer o matriz de caracteres.**

0	1	2
10	11	12

A	B	C
a	b	c
1	2	3



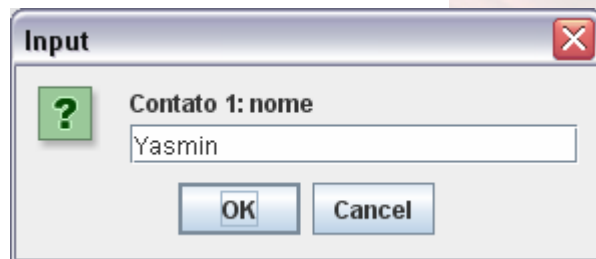
```
rui@laptop: ~/PCJ/Cap09
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
Conteúdo da matriz it:
it[0][0] = 0      it[0][1] = 1      it[0][2] = 2
it[1][0] = 10     it[1][1] = 11     it[1][2] = 12
Posições: 2x3

Conteúdo da matriz ch:
ch[0][0] = A      ch[0][1] = B      ch[0][2] = C
ch[1][0] = a      ch[1][1] = b      ch[1][2] = c
ch[2][0] = 1      ch[2][1] = 2      ch[2][2] = 3
Posições: 3x3
```

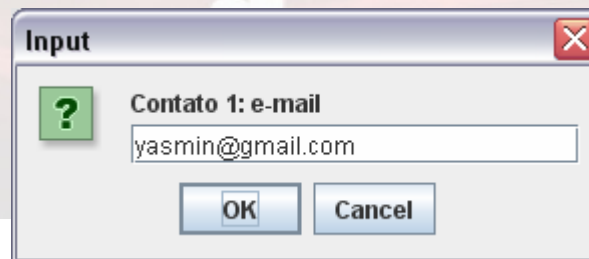
Matrizes

❑ Código 9.7 – Contatos.java

- **Crie uma matriz para armazenar o nome e o e-mail de contatos.**
- **O usuário poderá indicar quantos contatos deseja registrar.**
 - ❖ **Este número deve ser maior que zero.**
 - ❖ **O usuário poderá registrar um número menor de contatos.**
- **Utilize um laço for para captar e armazenar os dados.**
- **Utilize laços while para captar o nome e o e-mail.**
- **Se um dado informado não for válido, solicite-o novamente:**
 - ❖ **Nome: de 2 a 50 caracteres**
 - ❖ **E-mail: de 5 a 50 caracteres**
- **Utilize um laço for para recuperar e exibir a lista de contatos.**



Input dialog box titled "Contato 1: nome". It contains a text input field with the value "Yasmin" and "OK" and "Cancel" buttons.



Input dialog box titled "Contato 1: e-mail". It contains a text input field with the value "yasmin@gmail.com" and "OK" and "Cancel" buttons.



Message dialog box titled "Lista de contatos:". It displays the following text: "Yasmin - yasmin@gmail.com" and "Tayla - tayla@gmail.com". It has an "OK" button.

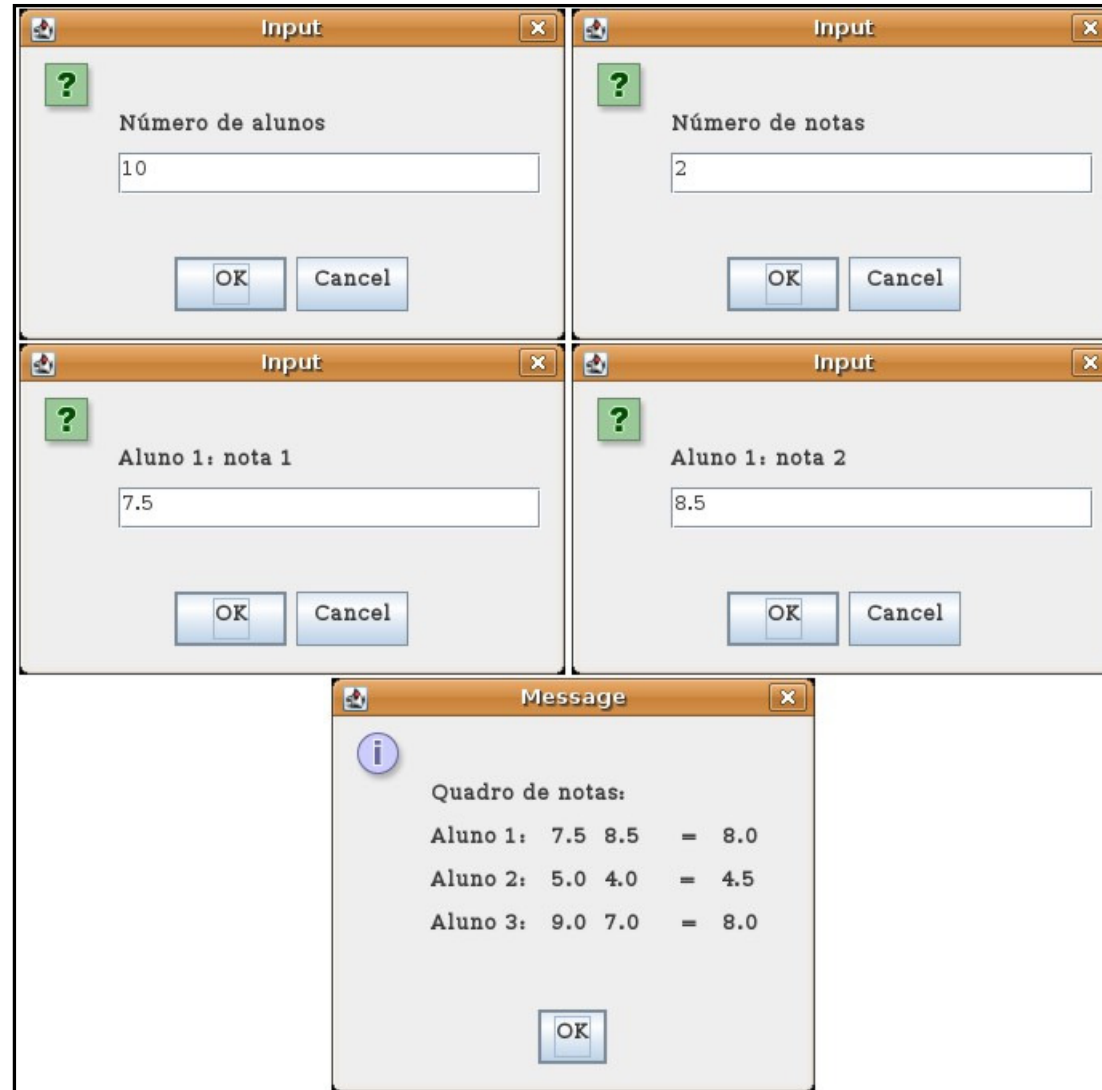
Matrizes

❑ Código 9.8 – Notas.java

- **Crie uma matriz para armazenar as notas de alunos.**
- **O usuário poderá indicar:**
 - ❖ **Quantos alunos terão suas notas registradas.**
 - ❖ **Quantas notas serão registradas para cada aluno.**
- **Utilize dois laços for aninhados para captar e armazenar as notas de cada um dos alunos.**
- **Utilize um laço while para captar cada nota.**
 - ❖ **A nota deve ser um número entre 0,0 e 10,0.**
 - ❖ **Se a nota for inválida, deve ser solicitada novamente.**
- **Utilize dois laços for aninhados para recuperar e exibir as notas de cada aluno e a sua média.**

Matrizes

❑ Código 9.8 – Notas.java



Exercício 1

- ❑ **Crie um novo aplicativo, chamado Exercicio0901, que gere uma senha provisória para o usuário. A senha a ser gerada deve ter 8 caracteres e todos eles devem ser dígitos numéricos. Ao final, o aplicativo deve exibir uma mensagem com a senha gerada.**



Exercício 2

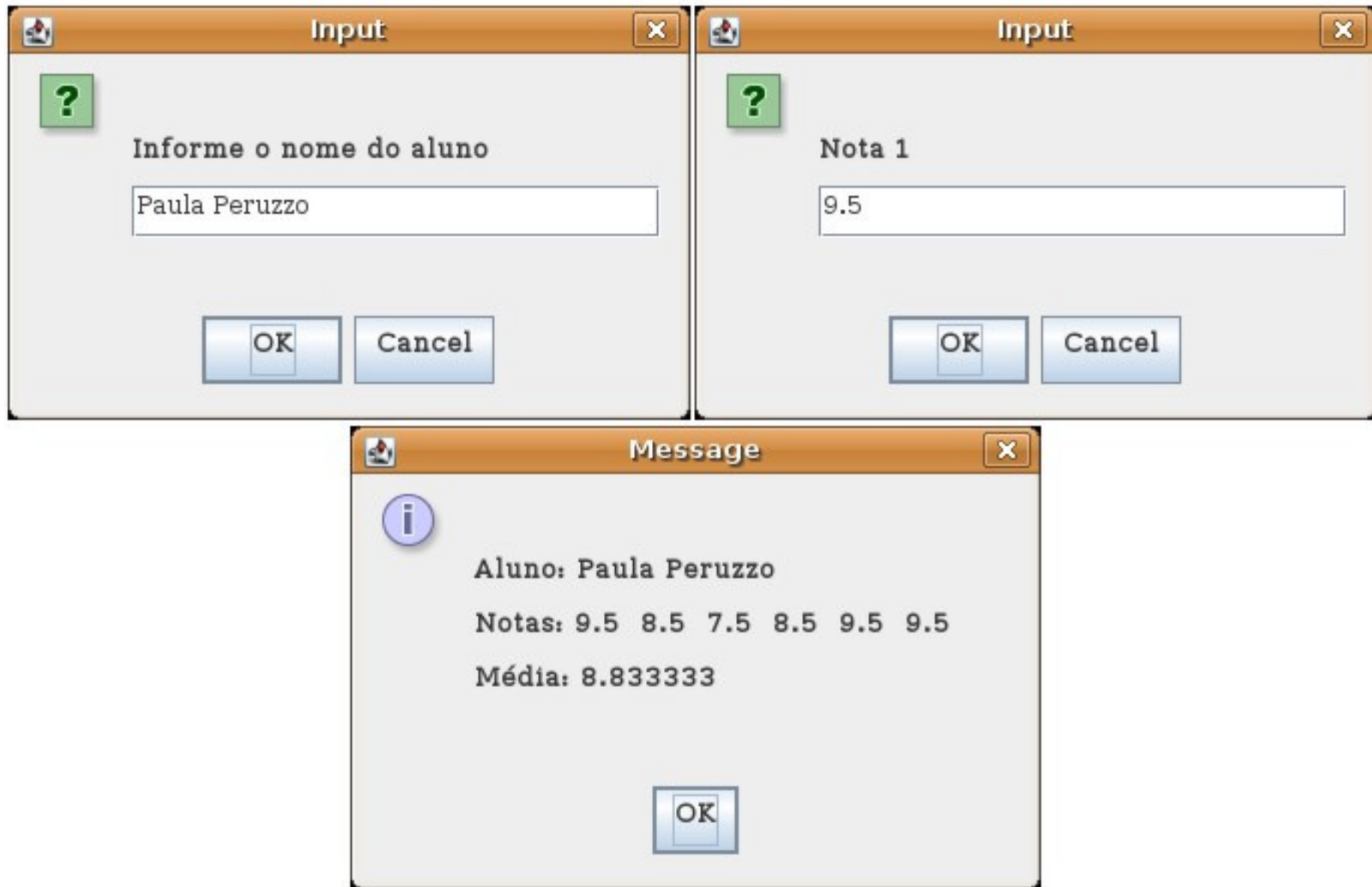
- ❑ **Crie um novo aplicativo, chamado Exercicio0902, que capte o nome de cinco pessoas e grave estes nomes em um vetor. Depois disso, este aplicativo deve sortear um dos nomes informados e, então, deve exibi-lo através de uma mensagem.**



Exercício 3

- Crie um novo aplicativo, chamado Exercicio0903, que capte o nome de um aluno e as suas notas e calcule a sua média final.**
- O nome do aluno deve ter de 2 a 50 caracteres. Se o nome informado for inválido, exiba uma mensagem de erro e solicite-o novamente. Se este diálogo for cancelado, o aplicativo deve ser encerrado.**
- Este aplicativo deve permitir que sejam registradas até 10 notas para o aluno. Cada nota deve ser um número fracionário entre 0,0 e 10,0.**
- Todas as notas informadas devem ser somadas e o resultado deve ser dividido pelo número de notas informadas para se obter a média do aluno.**
- Ao final, o aplicativo deve exibir uma mensagem contendo o nome do aluno, todas as suas notas e a sua média.**

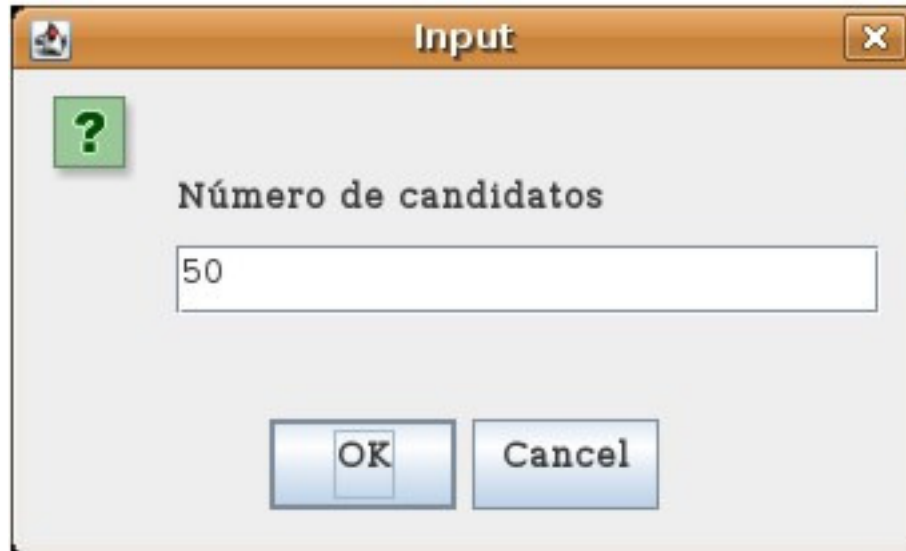
Exercício 3



Exercício 4

- Crie um novo aplicativo, chamado Exercicio0904, que capte o nome de candidatos a um concurso qualquer e gere uma senha alfanumérica para cada um deles.**
- O aplicativo deve permitir que você indique o número de candidatos que deseja registrar, desde que este número seja superior a zero.**
- Armazene tanto os nomes dos candidatos registrados quanto a senha gerada para cada um deles em uma matriz contendo duas colunas e o número de linhas informadas.**
- A senha de cada candidato deve ser gerada imediatamente após seu nome ter sido registrado e deve ter oito caracteres.**
- Ao final, o aplicativo deve exibir uma mensagem contendo o número, o nome e a senha de cada candidato.**

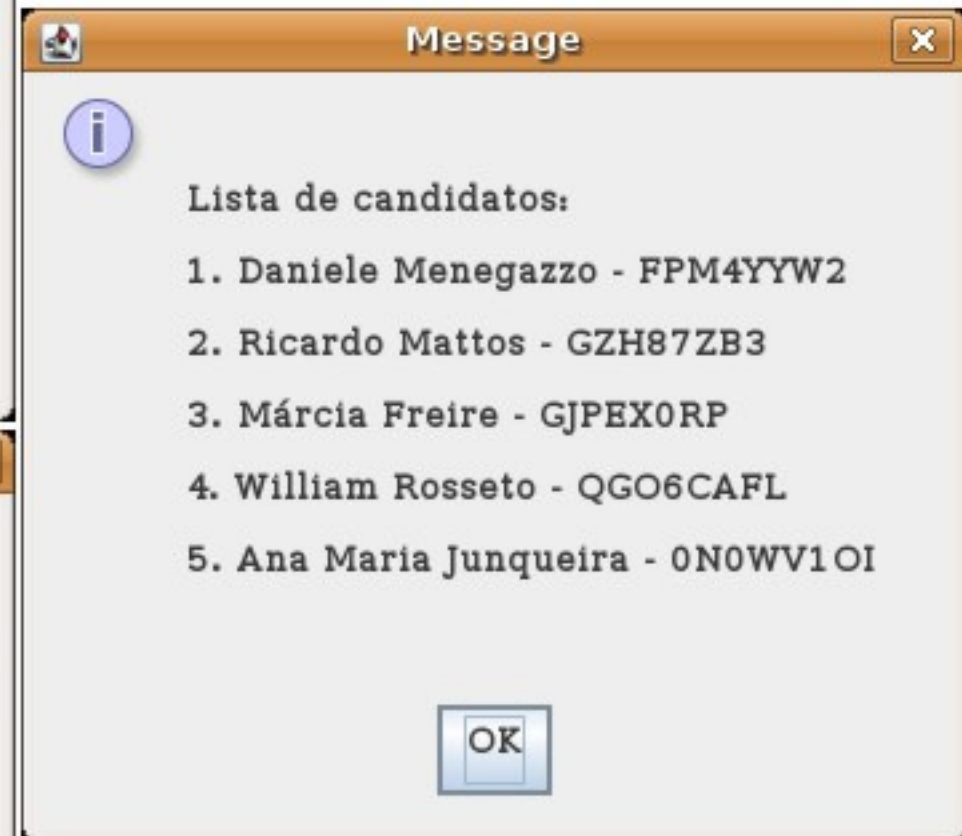
Exercício 4



Input dialog box titled "Input" with a question mark icon. The text "Número de candidatos" is displayed above a text input field containing the value "50". Below the input field are two buttons: "OK" and "Cancel".



Input dialog box titled "Input" with a question mark icon. The text "Candidato 1" is displayed above a text input field containing the value "Daniele Menegazzo". Below the input field are two buttons: "OK" and "Cancel".



Message dialog box titled "Message" with an information icon. The text "Lista de candidatos:" is displayed above a list of five candidates. Below the list is an "OK" button.

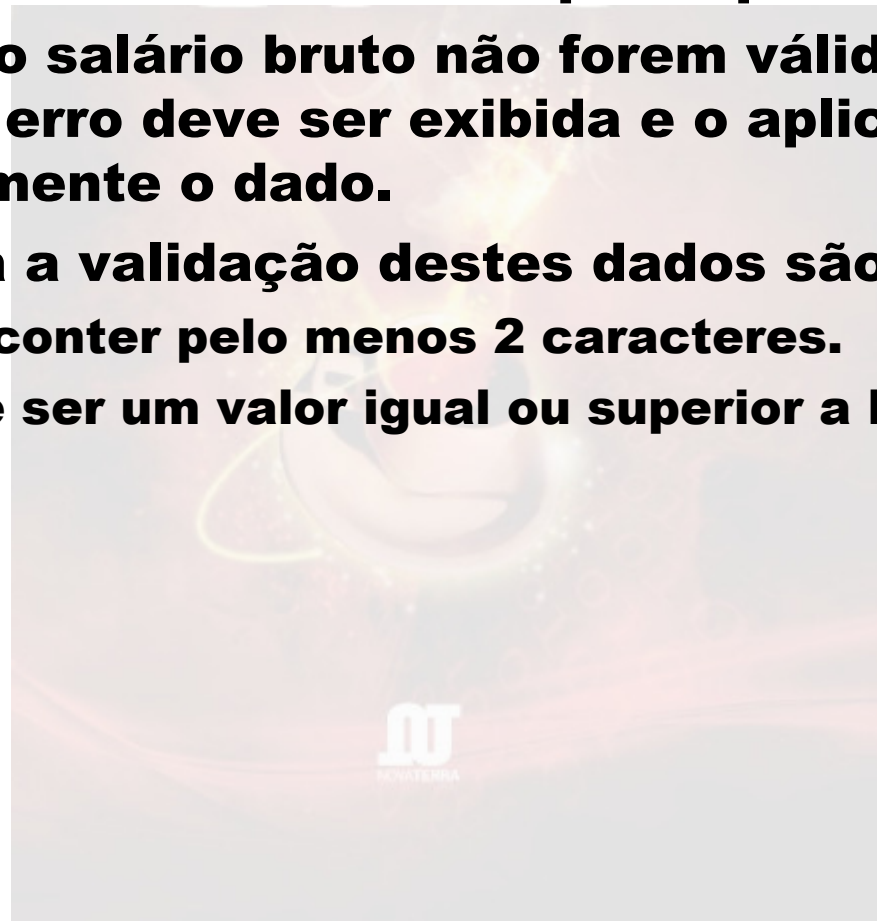
- 1. Daniele Menegazzo - FPM4YYW2
- 2. Ricardo Mattos - GZH87ZB3
- 3. Márcia Freire - GJPEX0RP
- 4. William Rosseto - QGO6CAFL
- 5. Ana Maria Junqueira - 0N0WV1OI

Exercício 5

- ❑ **Crie um novo aplicativo, chamado Exercicio0905, que simule a emissão de um relatório simplificado de folha de pagamento. O aplicativo deve utilizar diálogos de entrada de dados para solicitar o nome e o salário de cada funcionário.**
- ❑ **O aplicativo deve permitir que você indique o número de funcionários que deseja registrar, desde que este número seja superior a zero. Se este diálogo for cancelado, o aplicativo deve ser encerrado imediatamente.**
- ❑ **Armazene os nomes dos funcionários em um vetor.**
- ❑ **Utilize uma matriz que contenha quatro colunas para armazenar os seguintes dados de cada funcionário:**
 - **Salário bruto**
 - **INSS**
 - **Imposto de Renda**
 - **Salário líquido**

Exercício 5

- ❑ **O nome e o salário bruto devem ser informados e os descontos devem ser calculados pelo aplicativo.**
- ❑ **Se o nome ou o salário bruto não forem válidos, uma mensagem de erro deve ser exibida e o aplicativo deve solicitar novamente o dado.**
- ❑ **As regras para a validação destes dados são as seguintes:**
 - **Nome: deve conter pelo menos 2 caracteres.**
 - **Salário: deve ser um valor igual ou superior a R\$ 465,00.**



Exercício 5

- ❑ **O teto para recolhimento de INSS é de R\$ 354,07. Esse é o valor máximo que pode ser descontado do funcionário.**
- ❑ **Para cada funcionário, o aplicativo deve calcular os valores a serem descontados de seu salário a título de INSS e de Imposto de Renda. Para calcular o valor do INSS a ser descontado, utilize a tabela abaixo:**

Salário	Alíquota
Até R\$ 965,67	8,0
De R\$ 965,68 a R\$ 1.609,45	9,0
De R\$ 1.609,46 a R\$ 3.218,90	11,0

Exercício 5

- ❑ Para calcular o valor do Imposto de Renda, utilize a tabela abaixo.

Salário	Alíquota
Até R\$ 1.434,00	0,0
De R\$ 1.434,01 a R\$ 2.150,00	7,5
De R\$ 2.150,01 a R\$ 2.886,00	15,0
De R\$ 2.886,01 a R\$ 3.582,00	22,5
Acima de 3.582,00	27,5

Exercício 5

- ❑ **O aplicativo também deve calcular o salário líquido a ser pago a cada funcionário, que representa o valor do salário decrescido dos valores do INSS e do Imposto de Renda.**
- ❑ **Quando você cancelar um dos diálogos de entrada de dados, este aplicativo deve exibir uma mensagem contendo o resumo da folha de pagamento.**
- ❑ **Este resumo deve ter o nome e o salário bruto de cada funcionário registrado, os valores do INSS e do Imposto de Renda que serão descontados e o salário líquido a ser pago.**
- ❑ **Ao final deste resumo, devem ser indicadas algumas totalizações:**
 - **A soma do valor de todos os salários brutos**
 - **A soma dos descontos de INSS**
 - **A soma dos descontos de Imposto de Renda**
 - **A soma de todos os salários líquidos**

Exercício 5



Contato

Com o autor:

Rui Rossi dos Santos

E-mail: livros@ruirossi.pro.br

Web Site: <http://www.ruirossi.pro.br>

Com a editora:

Editora NovaTerra

Telefone: (21) 2218-5314

Web Site: <http://www.editoranovatterra.com.br>

