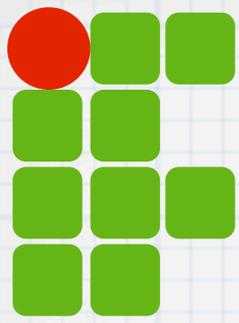


INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Programação de Computadores

Estruturas multi-dimensionais

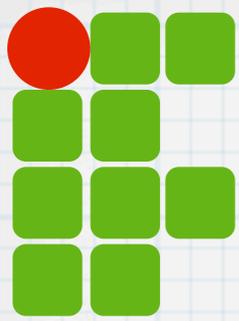
Copyright © 2013 IFRN



O que veremos hoje?

- * Revisão arrays
- * Arrays multidimensionais
- * Matrizes
- * Percorrer elementos da matriz
- * Exemplos
- * Exercícios

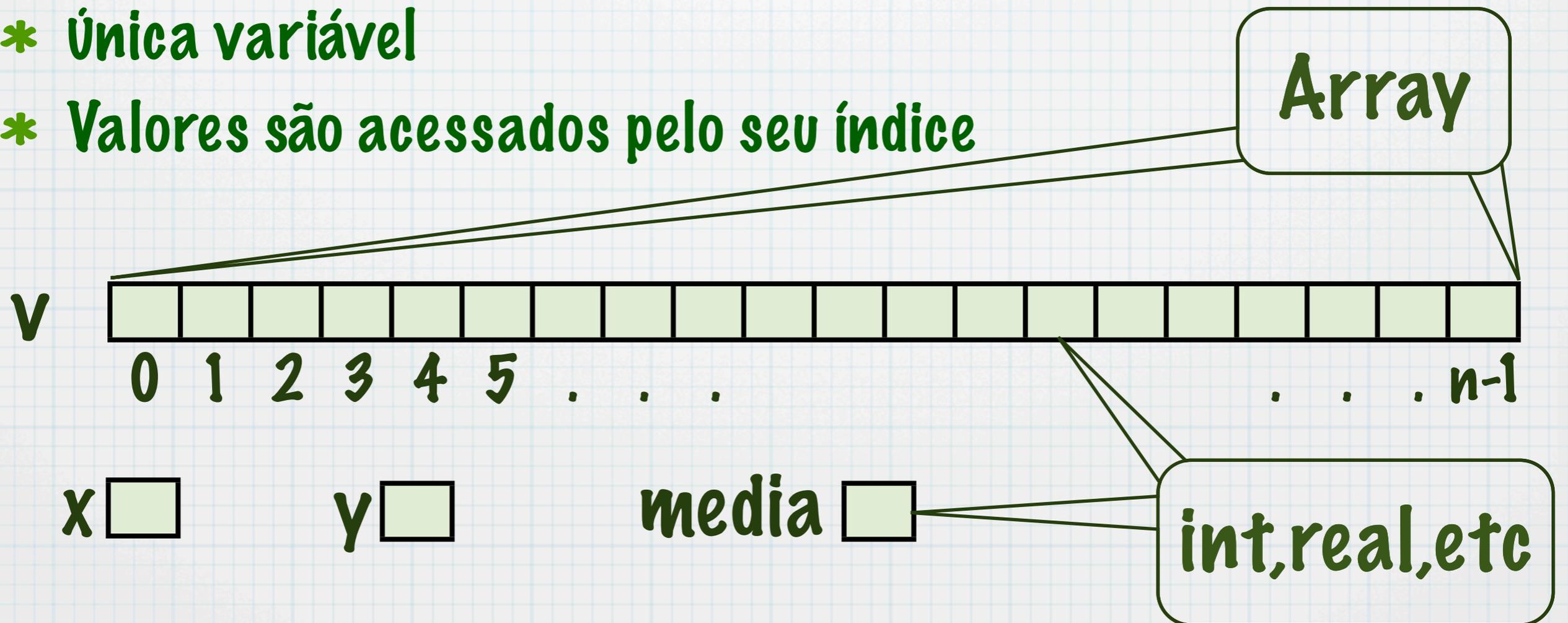


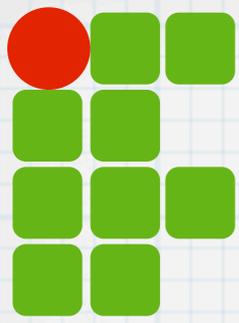


Arrays

* O que são arrays?

- * Um agregado de elementos
- * Capacidade de armazenar uma coleção de valores
- * Única variável
- * Valores são acessados pelo seu índice





Arrays

* O que são arrays?

* Um agregado de elementos

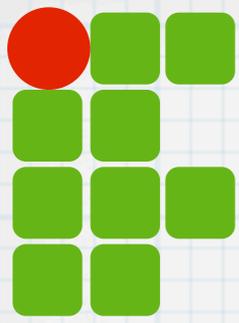
* Capacidade de armazenar uma coleção de valores

* Única variável

* Valores são acessados por seu índice

E se cada elemento do array for um array?



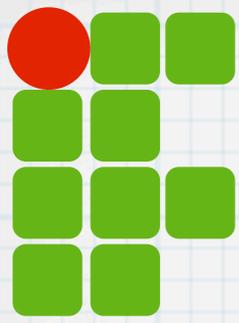


Matriz

* Arrays com 2 dimensões

- * Array de array
- * Linha x coluna

| | | |
|---|---|---|
| 2 | 7 | 6 |
| 9 | 5 | 1 |
| 4 | 3 | 8 |



Matriz

* Arrays com 2 dimensões

- * Array de array
- * Linha x coluna

Linha 0 →

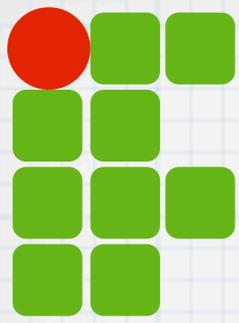
| | | |
|---|---|---|
| 2 | 7 | 6 |
|---|---|---|

Linha 1 →

| | | |
|---|---|---|
| 9 | 5 | 1 |
|---|---|---|

Linha 2 →

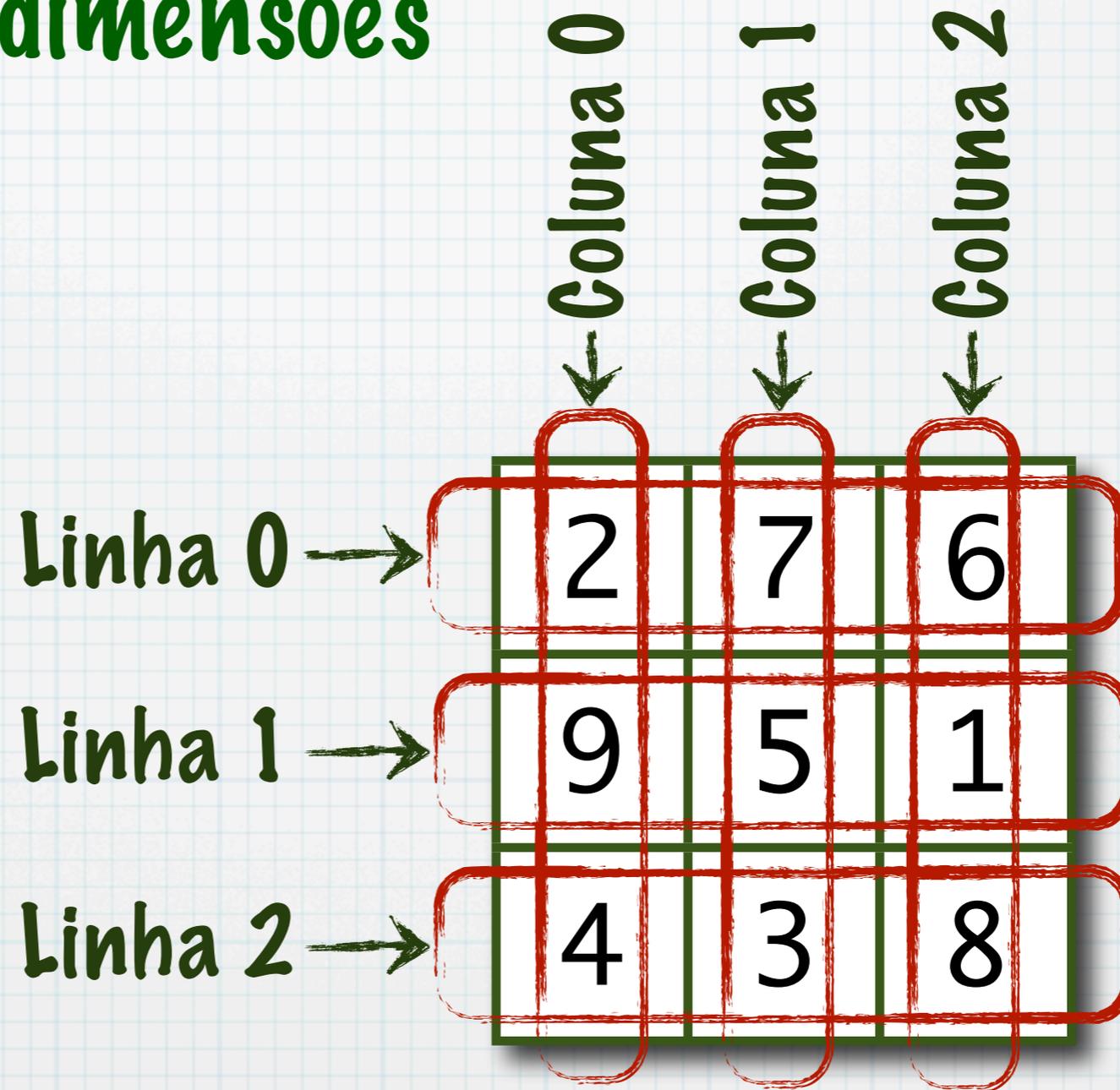
| | | |
|---|---|---|
| 4 | 3 | 8 |
|---|---|---|

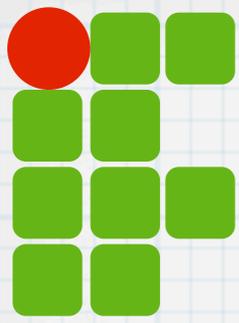


Matriz

* Arrays com 2 dimensões

- * Array de array
- * Linha x coluna





Matriz

* Definição

* Inicia Array

* $m = []$

* Cada elemento array é um array

* $m[0] = []$

* $m[1] = []$

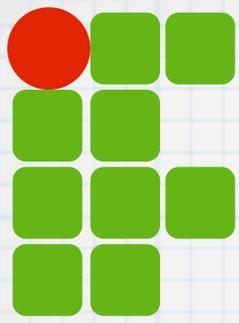
* ...

* Atribuição de elementos

* $m[0][0] = 2$

* $m[0][1] = 7$

| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 2 | 7 | 6 |
| 1 | 9 | 5 | 1 |
| 2 | 4 | 3 | 8 |



Matriz

* Definição

* Inicia Array

* $m = []$

* Cada elemento array é um array

* $m[0] = []$

* $m[1] = []$

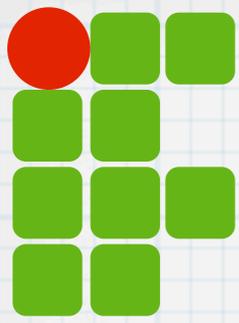
* ...

* Atribuição de elementos

* $m[0][0] = 2$

* $m[0][1] = 7$

| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 2 | 7 | 6 |
| 1 | 9 | 5 | 1 |
| 2 | 4 | 3 | 8 |



Matriz

* Definição

* Inicia Array

* $m = []$

* Cada elemento array é um array

* $m[0] = []$

* $m[1] = []$

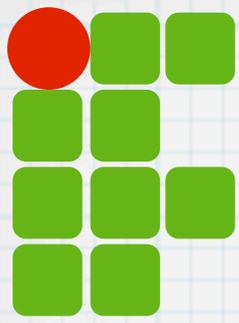
* ...

* Atribuição de elementos

* $m[0][0] = 2$

* $m[0][1] = 7$

| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 2 | 7 | 6 |
| 1 | 9 | 5 | 1 |
| 2 | 4 | 3 | 8 |



Percorrer matriz

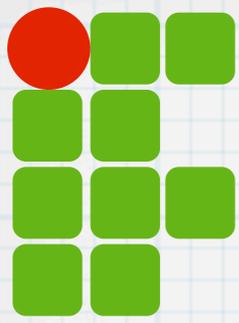
* Dois for

* um para as linhas

* um interno para as colunas de uma linha

```
matriz = []  
for i in 0..2 do  
  matriz[i]=[]  
end  
...  
for linha in matriz do  
  for elemento in linha do  
    print elemento  
    print " "  
  end  
  puts  
end
```

| | | |
|---|---|---|
| 2 | 7 | 6 |
| 9 | 5 | 1 |
| 4 | 3 | 8 |



Percorrer matriz

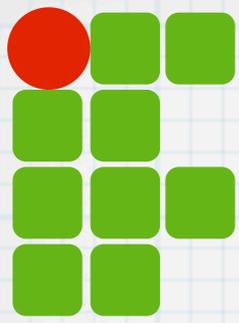
* Dois for

* um para as linhas

* um interno para as colunas de uma linha

```
m = []
for i in 0..2 do
  m[i]=[]
end
...
for l in 0..2 do
  for c in 0..2 do
    print m[l][c]
    print " "
  end
  puts
end
```

| | | |
|---|---|---|
| 2 | 7 | 6 |
| 9 | 5 | 1 |
| 4 | 3 | 8 |



Ler matriz

* Duas opções

* 1 valor por linha de entrada

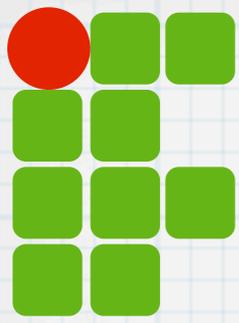
```
matriz = 3.times.map do
  3.times.map do
    gets.to_i
  end
end
```

* n valores por linha de entrada

```
matriz = 3.times.map do
  gets.chomp.split(" ")
end
```

* Retorno será string

* Pode ser necessário verificar dados de uma linha

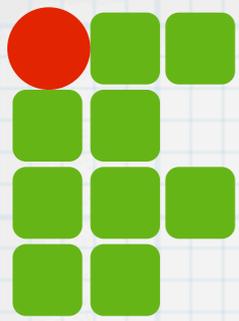


Somar elementos

```
matriz = 5.times.map do
  5.times.map do
    gets.to_i
  end
end
soma = 0
for linha in 0..4 do
  for coluna in 0..4 do
    soma = soma + matriz[linha][coluna]
  end
end
puts soma
```

entrada.txt

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
```



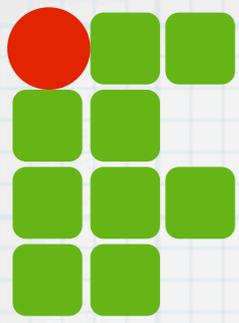
Somar elementos

```
matriz = 5.times.map do
  5.times.map do
    gets.to_i
  end
end
soma = 0
for linha in 0..4 do
  for coluna in 0..4 do
    soma = soma + matriz[linha][coluna]
  end
end
puts soma
```

| m | 0 | 1 | 2 | 3 | 4 |
|---|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 6 | 7 | 8 | 9 | 10 |
| 2 | 11 | 12 | 13 | 14 | 15 |
| 3 | 16 | 17 | 18 | 19 | 20 |
| 4 | 21 | 22 | 23 | 24 | 25 |

entrada.txt

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
```



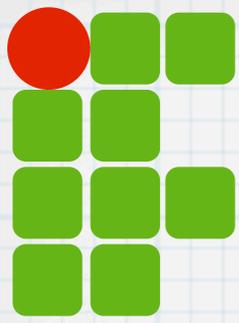
Somar elementos por coluna

```
soma = []  
for i in 0..4 do  
  soma[i]=0  
end  
for l in 0..4 do  
  for c in 0..4 do  
    soma[c]=soma[c]+m[l][c]  
  end  
end
```

| m | 0 | 1 | 2 | 3 | 4 |
|---|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 6 | 7 | 8 | 9 | 10 |
| 2 | 11 | 12 | 13 | 14 | 15 |
| 3 | 16 | 17 | 18 | 19 | 20 |
| 4 | 21 | 22 | 23 | 24 | 25 |

soma

| | | | | |
|---|---|---|---|---|
| ? | ? | ? | ? | ? |
|---|---|---|---|---|

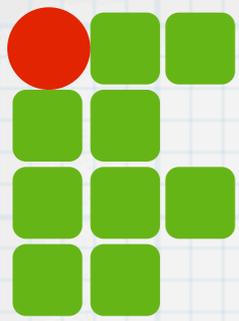


Somar elementos por linha

```
m = 5.times.map do
  gets.chomp.split(" ")
end
for l in 0..4 do
  for c in 0..4 do
    m[l][c]=m[l][c].to_i
  end
end

s = []
for l in 0..4 do
  s[l]=0
  for c in 0..4 do
    s[l]=s[l]+m[l][c]
  end
end
```

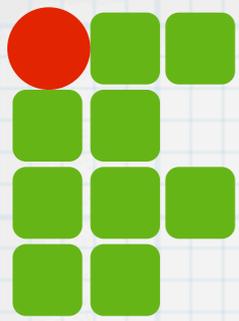
| m | 0 | 1 | 2 | 3 | 4 | soma |
|---|----|----|----|----|----|------|
| 0 | 1 | 2 | 3 | 4 | 5 | ? |
| 1 | 6 | 7 | 8 | 9 | 10 | ? |
| 2 | 11 | 12 | 13 | 14 | 15 | ? |
| 3 | 16 | 17 | 18 | 19 | 20 | ? |
| 4 | 21 | 22 | 23 | 24 | 25 | ? |



Contar linhas nulas

```
m = 5.times.map do
  gets.chomp.split(" ")
end
for l in 0..4 do
  for c in 0..4 do
    m[l][c]=m[c][l].to_i
  end
end
qtd_nulas = 0
for l in 0..4 do
  nula=true
  for c in 0..4 do
    if (m[l][c]!=0) then
      nula = false
    end
  end
  if (nula) then
    qtd_nulas = qtd_nulas + 1
  end
end
```

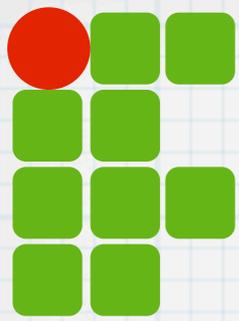
| m | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|----|---|
| 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 6 | 5 | 7 | 78 | 6 |
| 3 | 7 | 0 | 0 | 0 | 9 |
| 4 | 0 | 0 | 0 | 0 | 0 |



Contar linhas nulas

```
m = 5.times.map do
  gets.chomp.split(" ")
end
for l in 0..4 do
  for c in 0..4 do
    m[l][c]=m[c][l].to_i
  end
end
qtd_nulas = 0
for l in 0..4 do
  nula=true
  for c in 0..4 do
    if (m[l][c]!=0) then
      nula = false
    end
  end
  if (nula) then
    qtd_nulas = qtd_nulas + 1
  end
end
```

| m | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|----|---|
| 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 6 | 5 | 7 | 78 | 6 |
| 3 | 7 | 0 | 0 | 0 | 9 |
| 4 | 0 | 0 | 0 | 0 | 0 |



Matriz transposta

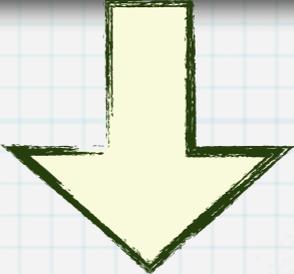
```
m = 2.times.map do
  3.times.map do
    gets.to_i
  end
end

mt = []
for i in 0..2 do
  mt[i] = []
end

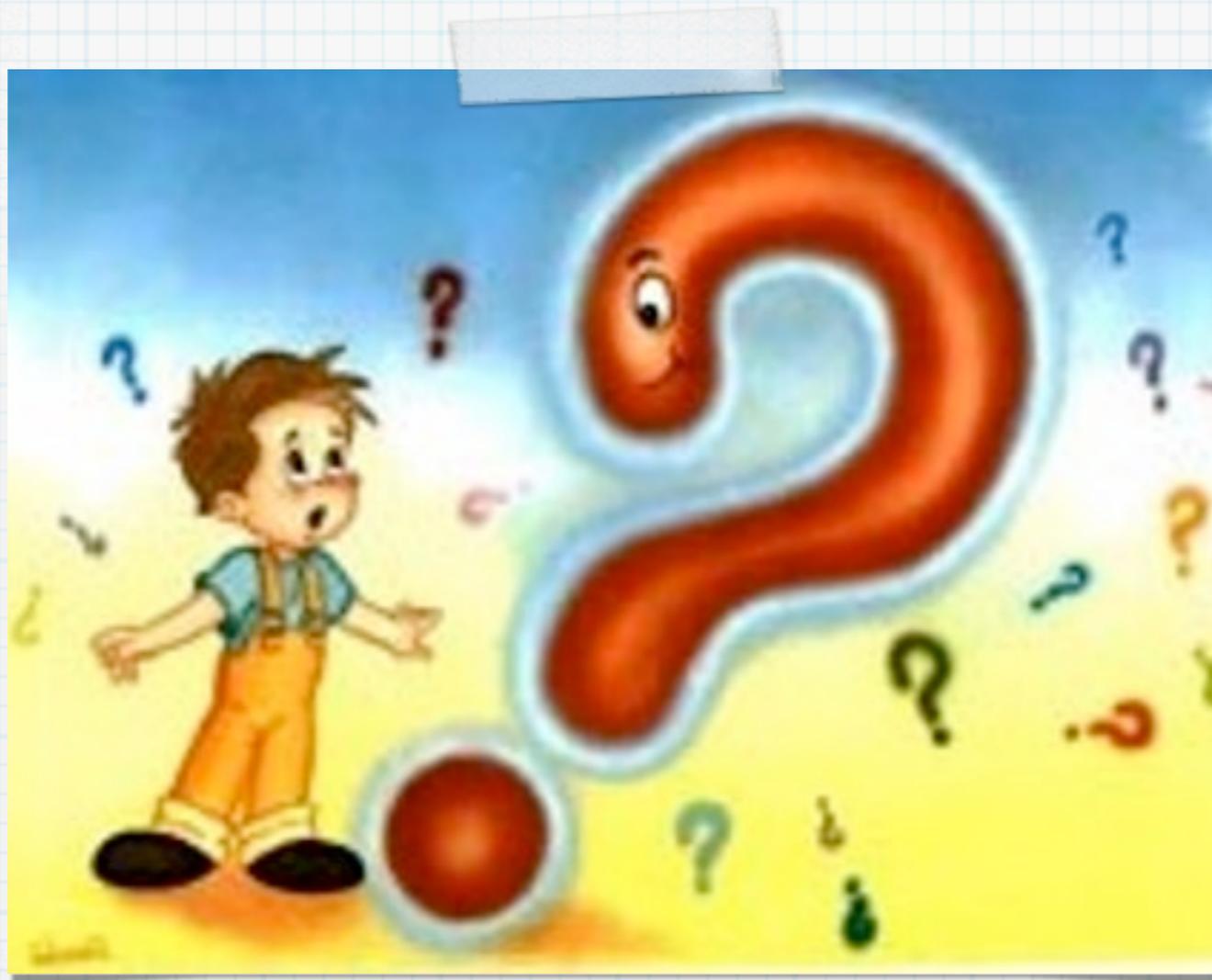
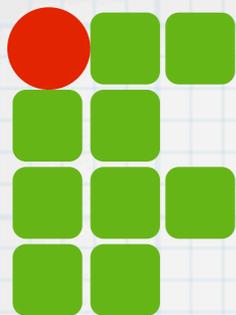
for c in 0..1 do
  for l in 0..2 do
    mt[l][c] = m[c][l]
  end
end
```

| | | |
|---|---|---|
| 2 | 7 | 6 |
| 9 | 5 | 1 |

Transposta



| | |
|---|---|
| 2 | 9 |
| 7 | 5 |
| 6 | 1 |



Dúvidas?