

1

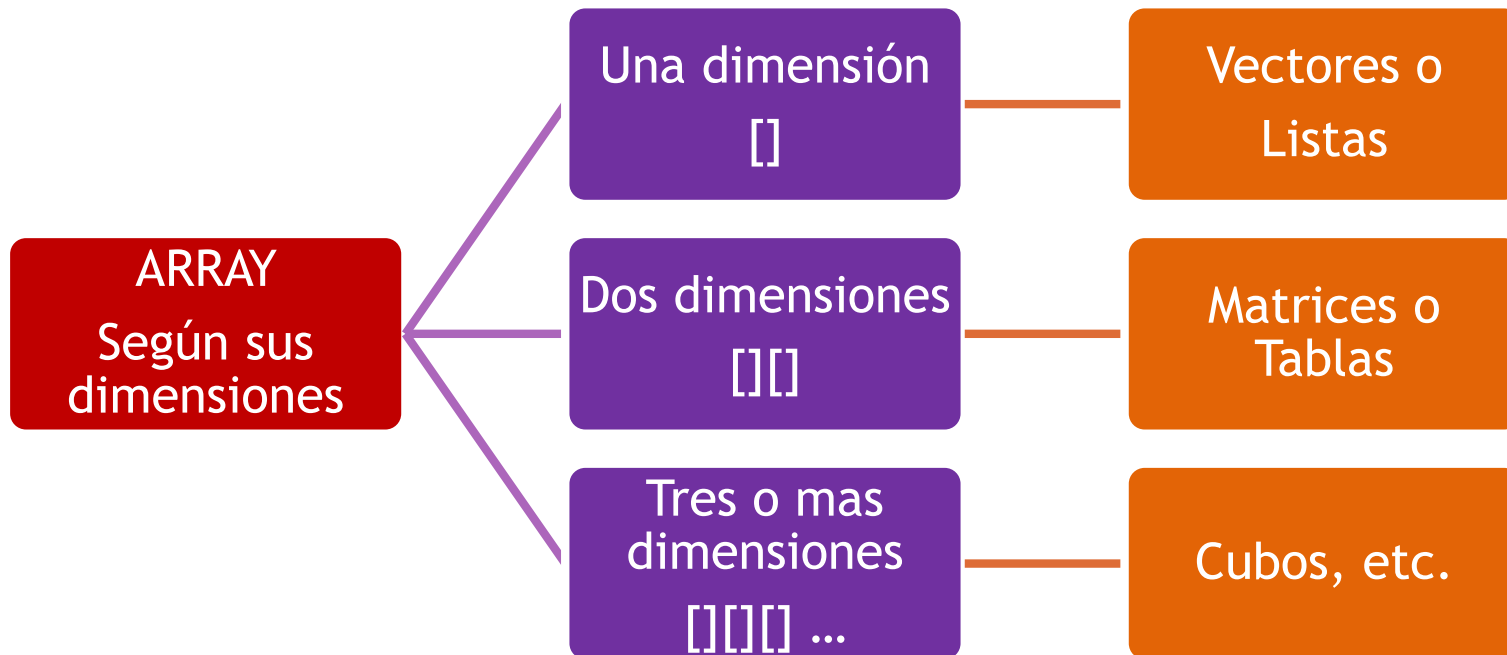
DIMENSIONES
→ PRIMITIVOS
→ DE CLASES
INSTANCIACION
CAPTURA
CONSULTA

ARRAY EN JAVA

www.slideshare.net/emergar

www.youtube.com/emergaray

ARRAY SEGÚN SU NUMERO DE DIMENSIONES



VECTORES O LISTAS

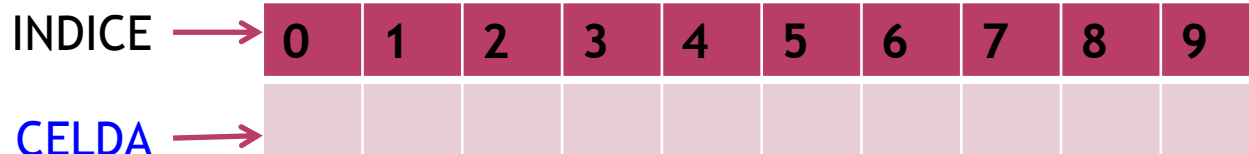
INDICE CELDA



A vertical table representing an array. The first column contains indices from 0 to 9. The second column contains empty cells. Red arrows point from the labels 'INDICE' and 'CELDA' to their respective columns.

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Representación
Grafica
Vertical



A horizontal table representing an array. The first row contains indices from 0 to 9. The second row contains empty cells. Red arrows point from the labels 'INDICE' and 'CELDA' to their respective rows.

INDICE	→	0	1	2	3	4	5	6	7	8	9
CELDA	→										

Representación
Grafica
Horizontal

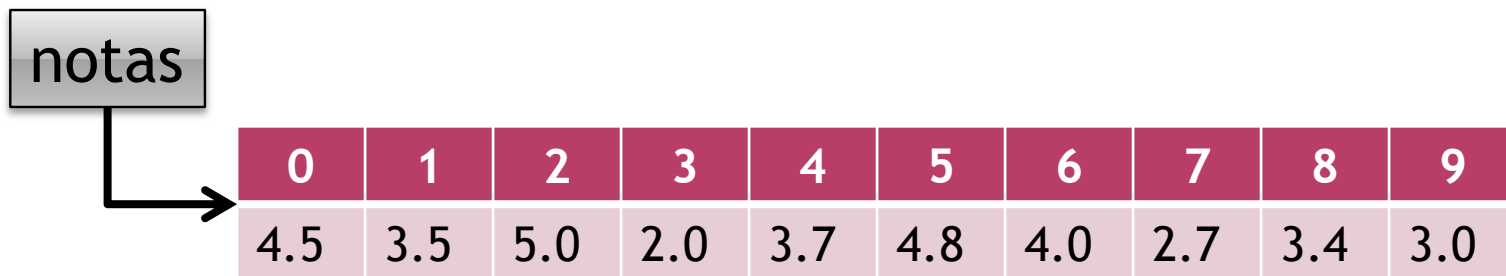
MATRICES O TABLAS

	Columna 0	Columna 1	Columna 2
Fila 0	X	O	O
Fila 1	O	X	O
Fila 2	X	O	X

Ejemplo de una Matriz de tamaño 3x3, es decir, 3 Filas x 3 Columnas y 9 elementos, con lo cual la cantidad de elementos de la matriz es el resultado de multiplicar El numero de filas por el de columnas

DEFINICION DE ARRAY

- ◉ Un array (arreglo) es una estructura de datos que contiene una colección de datos del mismo tipo, tiene la ventaja que permite definir de una sola vez, varias variables, es decir, si se necesitan almacenar 10 notas de los estudiantes de un curso, en vez de definir 10 variables, se puede definir un Array de 10 posiciones del tipo de dato que se quiere trabajar.



A diagram illustrating an array. A box labeled 'notas' has an arrow pointing to the first element of a 10-element array. The array is represented as a table with two rows: the first row contains indices from 0 to 9, and the second row contains the corresponding numerical values.

0	1	2	3	4	5	6	7	8	9
4.5	3.5	5.0	2.0	3.7	4.8	4.0	2.7	3.4	3.0

ARRAY: COMPARACIÓN

Variable

Dato

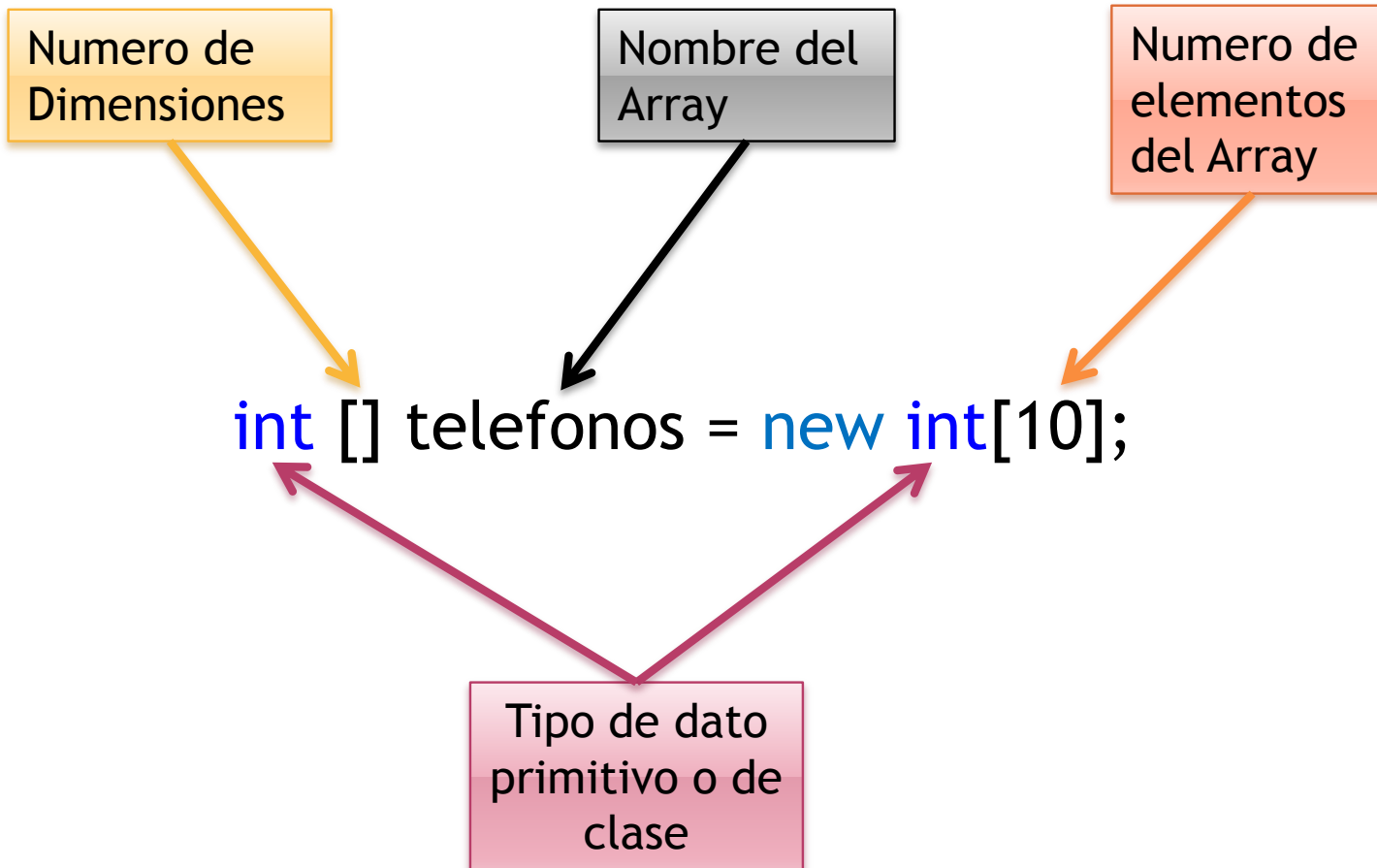
ARRAY Unidimensional o Vector

Dato	Dato	Dato	Dato	Dato
------	------	------	------	------

ARRAY Bidimensional o Matriz

Dato	Dato	Dato	Dato	Dato
Dato	Dato	Dato	Dato	Dato
Dato	Dato	Dato	Dato	Dato
Dato	Dato	Dato	Dato	Dato
Dato	Dato	Dato	Dato	Dato

DEFINIR UN ARRAY (ELEMENTOS)



DEFINIR UN ARRAY

//Array de tipos primitivos

```
int [] telefonos = new int[10];
```

```
double [] notas = new double[5];
```

```
long [] cedulas = new long[2000];
```

```
boolean [] estados = new boolean[100];
```

//Array de Clases

```
String [] nombres = new String[5];
```

```
Moto [] concesionario = new Moto[100];
```

```
Estudiante [] curso = new Estudiante[30];
```


INICIALIZAR UN ARRAY PRIMITIVO

//Inicializar un Array definido con anterioridad

notas[0] = 3.5; //Primera Posición

notas[1] = 4.5; //Segunda Posición

notas[2] = 3.0; //Tercera Posición

notas[3] = 3.7; //Cuarto Posición

notas[4] = 3.8; //Quinta Posición

//Definir un Array e Inicializar

double [] notas = { 3.5, 4.5, 3.0, 3.7, 3.8 };

//Array para 5 números reales

INICIALIZAR UN ARRAY DE CLASE(1)

//Inicializar un Array definido con anterioridad

nombres[0] = “Jesús”; //Primera Posición

nombres[1] = “María”; //Segunda Posición

nombres[2] = “José”; //Tercera Posición

nombres[3] = “Marcos”; //Cuarto Posición

nombres[4] = “Pedro”; //Quinta Posición

//Definir un Array e Inicializar

String [] nombres = { “Jesús”, “María”, “José”,
“Marcos”, “Pedro” };

//Array para 5 Cadenas de caracteres

INICIALIZAR UN ARRAY DE CLASE(2)

//Inicializar un Array elemento a elemento

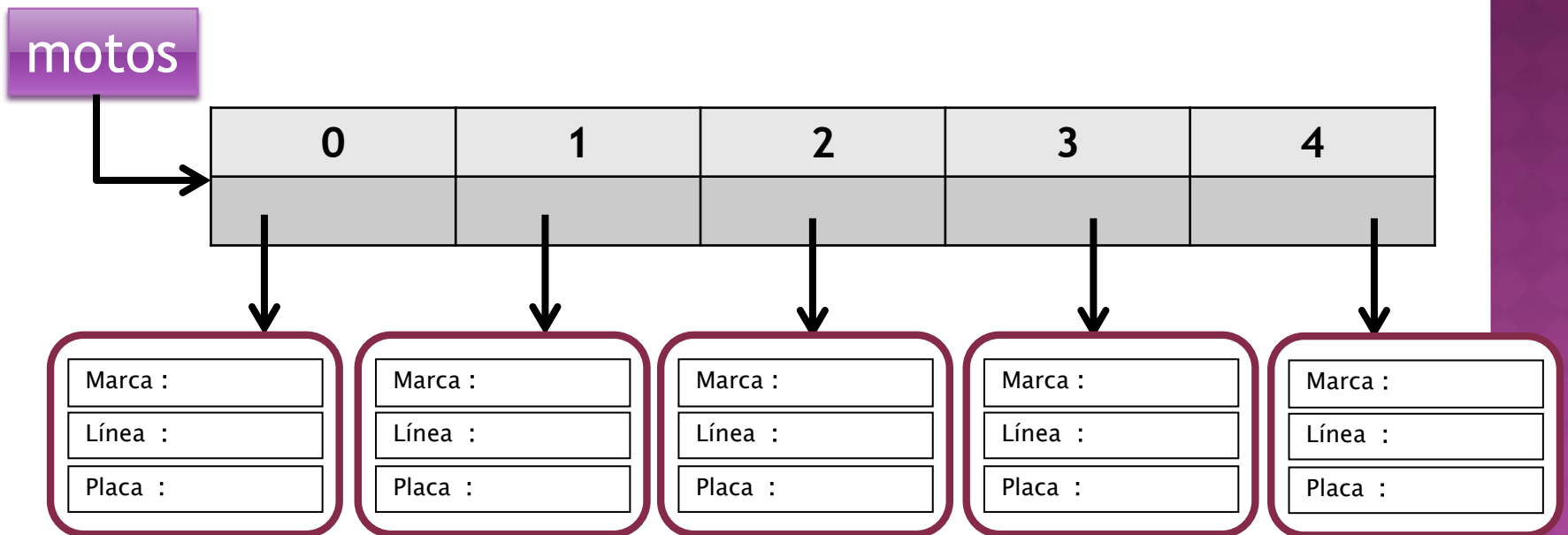
```
Moto [] motos;  
motos[0] = new Moto();  
motos[1] = new Moto();  
motos[2] = new Moto();  
motos[3] = new Moto();  
motos[4] = new Moto();
```

//Definir un Array e Inicializar inmediatamente

```
Moto [] motos = { new Moto(), new Moto(), new  
Moto(), new Moto(), new Moto() };
```

//Array para 5 objetos de tipo Moto

INICIALIZAR UN ARRAY DE CLASE(2)



INICIALIZAR UN ARRAY DE CLASE(3)

//Inicializar un Array elemento a elemento con datos

```
Moto [] motos;
```

```
motos[0] = new Moto("Honda", "CBS 125", "AGT-254");
```

```
motos[1] = new Moto("Susuki", "Viva 125", "ILJ-112");
```

```
motos[2] = new Moto("Yamaha", "FZ16", "JJY-211");
```

```
motos[3] = new Moto("Honda", "XTS", "ETR-296");
```

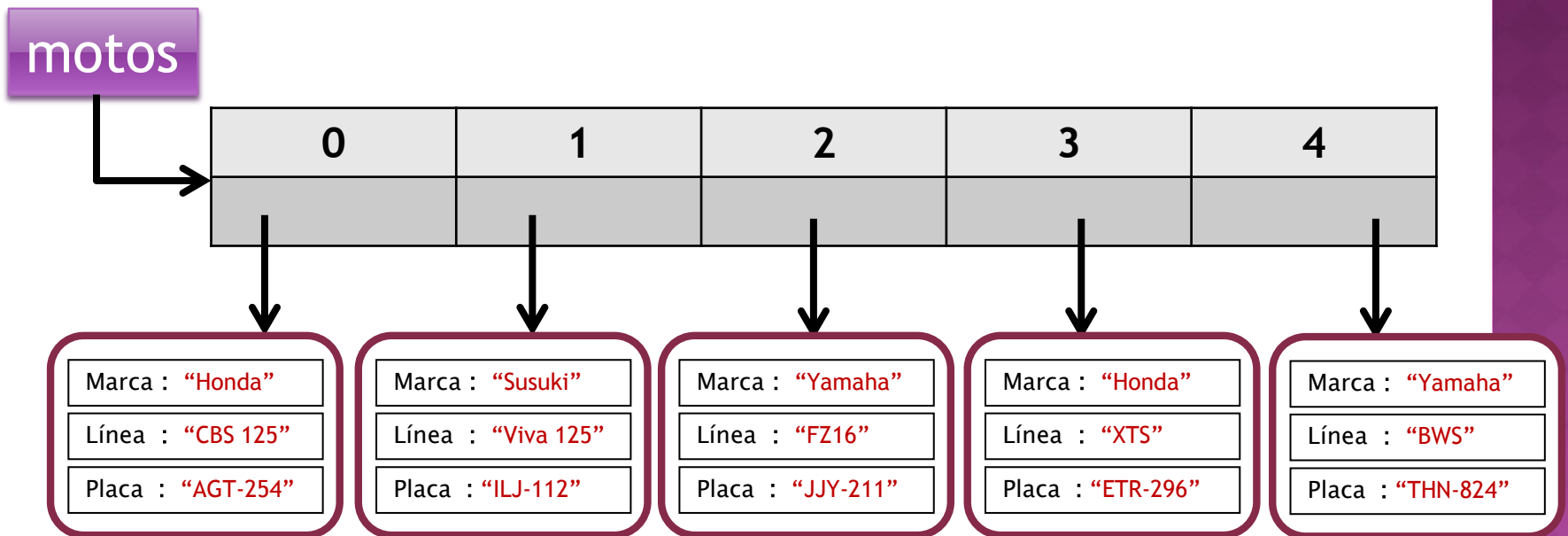
```
motos[4] = new Moto("Yamaha", "BWS", "THN-824");
```

//Definir un Array e Inicializar con datos

```
Moto [] motos = { new Moto("Honda", "CBS 125", "AGT-254"), new  
Moto("Susuki", "Viva 125", "ILJ-112"), new Moto("Yamaha", "FZ16",  
"JJY-211"), new Moto("Honda", "XTS", "ETR-296"), new  
Moto("Yamaha", "BWS", "THN-824") };
```

//Array para 5 objetos de tipo Moto

INICIALIZAR UN ARRAY DE CLASE(3)



RECORRER UN ARRAY

```
for(int k = 0; k < vector.length; k++){  
    //Acciones con el elemento sub-k  
    //es decir: vector[k]  
}
```

LLENAR UN ARRAY POR CONSOLA

```
Scanner teclado = new Scanner( System.in );  
double [] notas = new double[5];  
for(int k = 0; k < notas.length; k++){  
    System.out.println( “Digite la nota # ” + (k+1) );  
    notas[k] = teclado.nextFloat();  
}
```


LLENAR UN ARRAY CON JOPTIONPANE

```
String [] nombres = new String[5];  
for(int k = 0; k < nombres.length; k++){  
    nombres[k] = JOptionPane.showInputDialog(  
        "Digite el nombre # " + (k+1) );  
}
```

MOSTRAR UN ARRAY POR CONSOLA

```
for(int k = 0; k < notas.length; k++){  
System.out.println( "Nota # " + (k+1) + notas[k] );  
}
```

MOSTRAR UN ARRAY CON JOPTIONPANE

```
for(int k = 0; k < nombres.length; k++){  
    JOptionPane.showMessageDialog(  
        "Nombre # " + (k+1) + nombres[k] );  
}
```

HACER UNA COPIA DE UN ARRAY A OTRO

```
Scanner teclado = new Scanner( System.in );
double [] notas = new double[30];
double [] notas2 = new double[30];
for(int k = 0; k < notas.length; k++){
    System.out.println( "Digite la nota # " + (k+1) );
    notas[k] = teclado.nextFloat();
}
for(int k = 0; k < notas2.length; k++){
    notas2[k] = notas[k];
}
//Ahora notas y notas2 tienen los mismo datos en
//las misma posiciones
```

CREAR UN PROYECTO JAVA

- ◉ Crear un proyecto java en NetBeans
- ◉ Adicionar una clase java que usted imagine del mundo real, debe tener al menos 4 atributos, con los métodos: constructores, getters, setters y toString.
- ◉ Agregar la clase **Main** para definir un Array de una dimensión con 5 elementos para capturar 5 objetos de la clase definida anteriormente, para luego mostrar los datos capturados, sea por consola o con el JOptionPane.