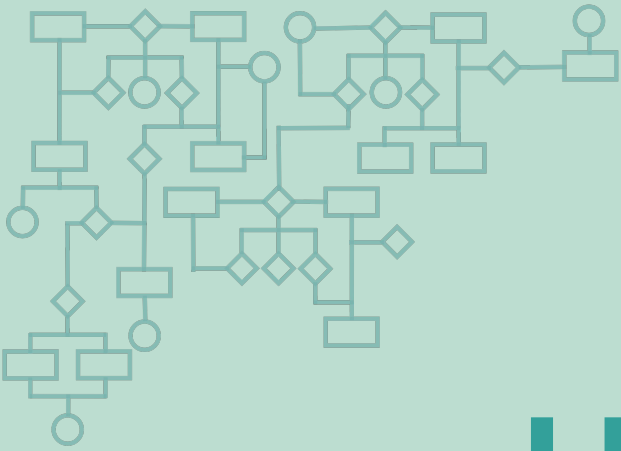


Clase N°11

Arreglos Multidimensionales

Taller de Programación
Computines





Hackerrank

www.hackerrank.com/c12-arreglos-multidimensionales



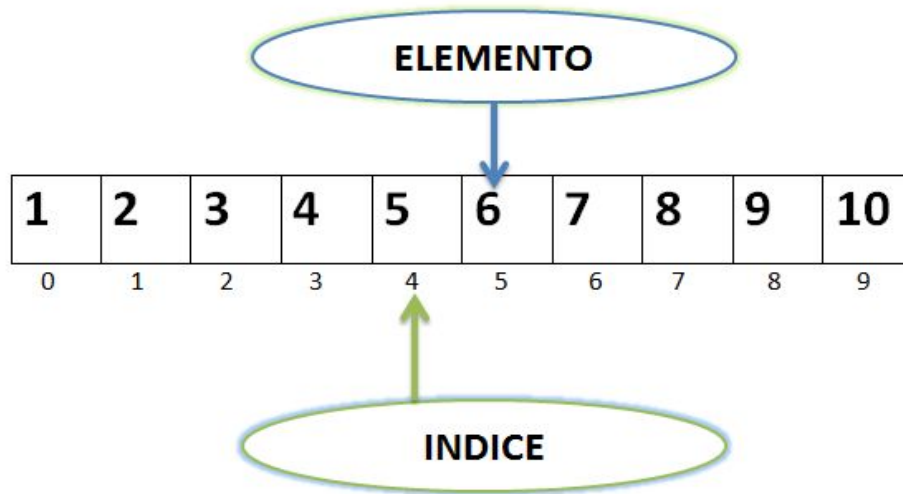
Repaso Arreglos

¿Qué es un
arreglo?

¿Qué es un arreglo?

Una arreglo es una colección de datos de largo fijo, en que todos los datos deben ser del mismo tipo

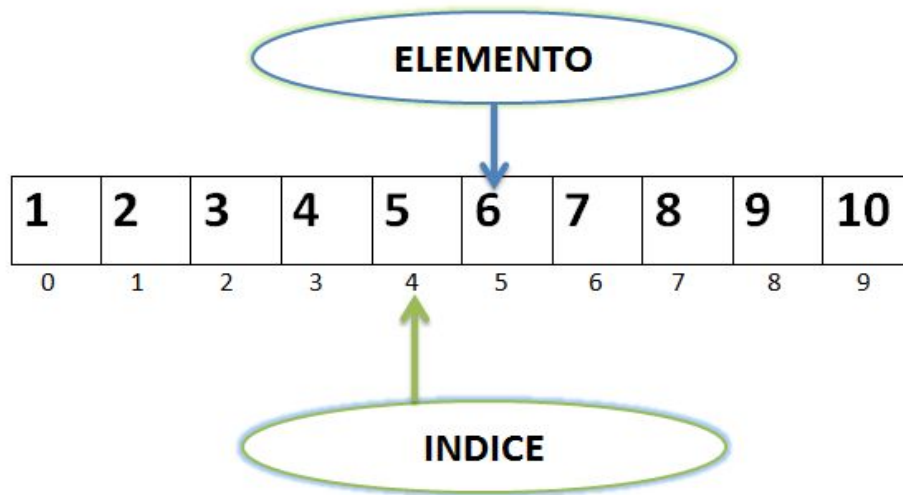
Cada elemento ocupa una posición:



¿Qué es un arreglo?

Una arreglo es una colección de datos de largo fijo, en que todos los datos deben ser del mismo tipo

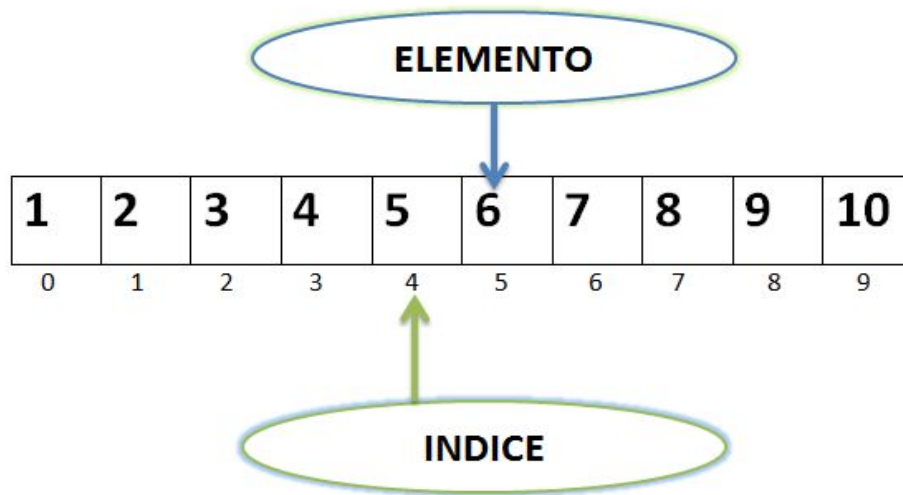
Cada elemento ocupa una posición:



¿Qué es un arreglo?

Una arreglo es una colección de datos de largo fijo, en que todos los datos deben ser del mismo tipo

Cada elemento ocupa una posición:



- Integer
- Double
- Boolean
- String
- etc.

¿Como crear
un arreglo?

¿Cómo crear un arreglo?



- **Declarar arreglo:**
 - `TipoDeDato[] arreglo;`
 - Ej: `int[] MiArregloDeInts;`

¿Cómo crear un arreglo?



- **Declarar arreglo:**
 - `TipoDeDato[] arreglo;`
 - Ej: `int[] MiArregloDeInts;`
- **Crear arreglo de datos con su valor nulo:**
 - `TipoDeDato[] arreglo = new TipoDeDato[Tamaño];`
 - Ej: `int[] MiSegundoArreglo = new int[10];`

¿Cómo crear un arreglo?



- **Declarar arreglo:**
 - `TipoDeDato[] arreglo;`
 - Ej: `int[] MiArregloDeInts;`
- **Crear arreglo de datos con su valor nulo:**
 - `TipoDeDato[] arreglo = new TipoDeDato[Tamaño];`
 - Ej: `int[] MiSegundoArreglo = new int[10];`
- **Crear y poblar un arreglo:**
 - `int[] miArreglo = {1, 10, -9, 0};`

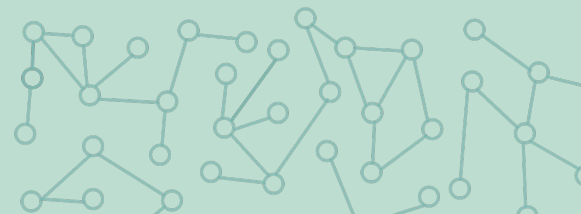
Operaciones en Arreglos

Operaciones en Arreglos

- **Para acceder a un elemento en particular del arreglo:**
 - `MiArreglo[índice];`
- **Para modificar un elemento de un arreglo:**
 - `MiArreglo[índice] = nuevoValor;`
- **Para obtener el largo de un arreglo:**
 - `MiArreglo.length`

Arreglos Multi- dimensionales

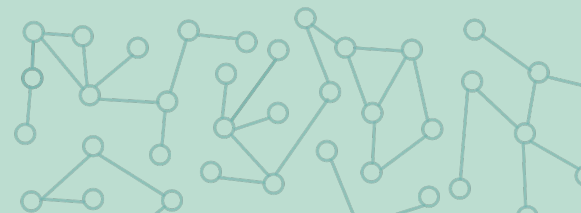
Arreglos Multidimensionales



Un arreglo se utiliza para guardar una secuencia de datos



Arreglos Multidimensionales



Un arreglo se utiliza para guardar una secuencia de datos

Dato 0	Dato 1	Dato 2	Dato 3	Dato 4	Dato 5	Dato 6
--------	--------	--------	--------	--------	--------	--------

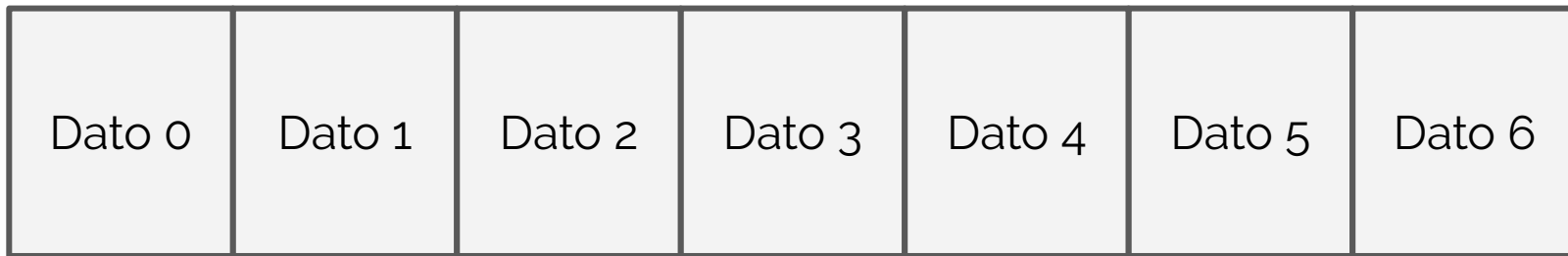
Integers:

21	5	13	17	8	9	6
----	---	----	----	---	---	---

Arreglos Multidimensionales



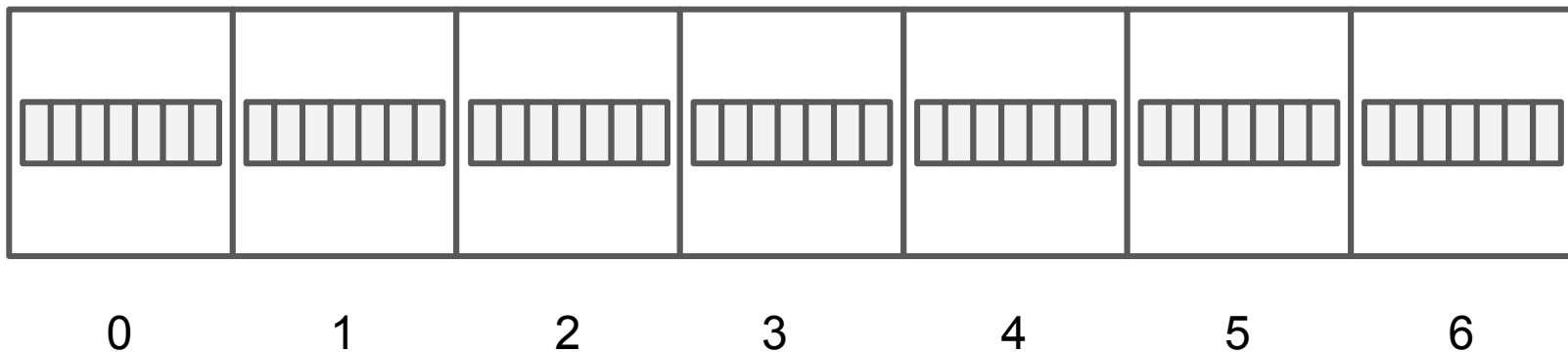
Los arreglos multidimensionales se utilizan para guardar secuencias de arreglos



Arreglos Multidimensionales



Los arreglos multidimensionales se utilizan para guardar secuencias de arreglos



Arreglos Multidimensionales



Otra manera de mostrar lo mismo:

0	0	1	2	3	4	5	6
1							
2							
3							

Arreglos Multidimensionales

Ejemplo:



Arreglos Multidimensionales

Ejemplo:

Arreglo 0 →

Arreglo 1 →

Arreglo 2 →

Arreglo 3 →

Arreglo 4 →

Arreglo 5 →

Arreglo 6 →

Arreglo 7 →



Como Crear Arreglos Multi- dimensionales

Cómo crear un arreglo multidimensional

Para crear un arreglo con 2 dimensiones:

- `TipoDeDato[][] arreglo;`

Cómo crear un arreglo multidimensional

Para crear un arreglo con 2 dimensiones:

- `TipoDeDato[][] arreglo;`
- `TipoDeDato[][] arreglo = new TipoDeDato[n][m];`

Cómo crear un arreglo multidimensional

Para crear un arreglo con 2 dimensiones:

- `TipoDeDato[][] arreglo;`
- `TipoDeDato[][] arreglo = new TipoDeDato[n][m];`
- `TipoDeDato[][] arreglo = {{...},{...},{...}};`

Cómo crear un arreglo multidimensional

Ejemplo:

```
bool[][] arreglo1;  
double[][] arreglo2 = new double[2][3];  
int[][] arreglo3 = {{6,4,3,5},{1,2,6,8},{6,3,4,1}};
```

Cómo crear un arreglo multidimensional

```
double[][] arreglo2 = new double[2][3];
```

Cómo crear un arreglo multidimensional

```
double[][] arreglo2 = new double[2][3];
```

0.0	0.0	0.0
0.0	0.0	0.0

Cómo crear un arreglo multidimensional

```
int[][] arreglo3 = {{6,4,3,5},{1,2,6,8},{6,3,4,1}};
```

Cómo crear un arreglo multidimensional

```
int[][] arreglo3 = {{6,4,3,5},{1,2,6,8},{6,3,4,1}};
```

6	4	3	5
1	2	6	8
6	3	4	1

Cómo Acceder a un Arreglo Multi- dimensional

Cómo acceder a un arreglo multidimensional

En un arreglo unidimensional utilizamos `arreglo[i]` para acceder al i-ésimo elemento del arreglo

En un arreglo multidimensional utilizamos la sentencia `arreglo[i][j]` para acceder al j-ésimo elemento del i-ésimo arreglo.

Cómo acceder a un arreglo multidimensional

```
int[][] arreglo = {{6,4,3,5},{1,2,6,8},{6,3,4,1}};
```

6	4	3	5
1	2	6	8
6	3	4	1

Arreglo[1] es igual a [1,2,6,8]

Arreglo[1][2] es igual a 6

Cómo acceder a un arreglo multidimensional

0	0	1	2	3	4	5	6
1	0	1	2	3	4	5	6
2	0	1	2	3	4	5	6
3	0	1	2	3	4	5	6

Cómo modificar un arreglo multidimensional

Para acceder a un elemento:

```
arreglo[i][j]
```


Para modificar un elemento:

```
arreglo[i][j] = nuevo_valor
```

Cómo modificar un arreglo multidimensional

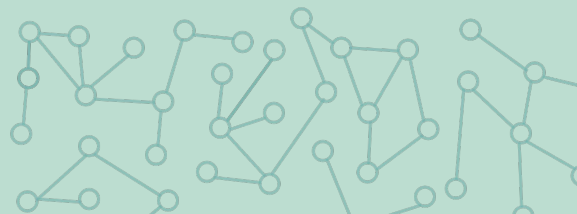
```
int[][] arreglo = {{6,4,3,5},{1,2,6,8},{6,3,4,1}};  
arreglo[1][2] = 7
```

6	4	3	5
1	2	6	8
6	3	4	1



6	4	3	5
1	2	7	8
6	3	4	1

Cómo obtener el largo



`arreglo.length` entrega el largo de la primera dimensión

`arreglo[indice].length` entrega el largo del arreglo en la otra dimensión

Ejemplo: `int[][] arreglo = {{1,2},{5,1,2},{9,6,2,7},{0}};`

`arreglo.length` entrega 4

`arreglo[1].length` entrega 3

`arreglo[3].length` entrega 1

Ejemplo: Imprimir arreglo en consola

```
public class Main {  
    public static void main(String[] args) {  
        int[][] x = {{1, 2}, {5, 1, 2}, {9, 6, 2, 7}, {0}};  
        for(int i = 0; i < x.length; i++){  
            for(int j = 0; j < x[i].length; j++){  
                System.out.print(x[i][j] + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```

Ejemplo: Imprimir arreglo en consola

```
public class Main {  
    public static void main(String[] args) {  
        int[][] x = {{1, 2}, {5, 1, 2}, {9, 6, 2, 7}, {0}};  
        for(int i = 0; i < x.length; i++){  
            for(int j = 0; j < x[i].length; j++){  
                System.out.print(x[i][j] + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```

Esto imprime:

```
1 2  
5 1 2  
9 6 2 7  
0
```

Ejemplo: Comparar dos arreglos

```
public class Main {  
    public static void main(String[] args) {  
        int[][] arreglo1 = {{1,2},{5,1,2},{9,6,2,7},{0}};  
        int[][] arreglo2 = {{1,2},{5,1,2},{9,2,2,7},{0}};  
        for(int i = 0; i < arreglo1.length; i++){  
            for(int j = 0; j < arreglo1[i].length; j++){  
                if(arreglo1[i][j] != arreglo2[i][j]){  
                    System.out.println("No son iguales");  
                    return;  
                }  
            }  
        }  
        System.out.println("Son Iguales!");  
    }  
}
```



```

public class Main {
    public static void main(String[] args) {
        int[][] arreglo1 = {{1,2},{5,1,2},{9,6,2,7},{0}};
        int[][] arreglo2 = {{1,2},{5,1,2},{9,2,2,7},{0}};
        if(arreglo1.length!=arreglo2.length){
            System.out.println("No son iguales");
            return;
        }
        for(int i = 0;i<arreglo1.length;i++){
            if(arreglo1[i].length!=arreglo2[i].length){
                System.out.println("No son iguales");
                return;
            }
            for(int j = 0; j < arreglo1[i].length; j++){
                if(arreglo1[i][j]!=arreglo2[i][j]){
                    System.out.println("No son iguales");
                    return;
                }
            }
        }
        System.out.println("Son Iguales!");
    }
}

```

Cuando quiero comparar
 los arreglos pero no sé
 que las dimensiones son
 iguales...