

Clase N°9

# Arreglos

Taller de Programación  
Computines



# Repaso de Strings

# Strings

Los Strings son cadenas de caracteres que sirven para representar texto

Cada carácter ocupa una posición:

M	e	t	a	p	o	d
---	---	---	---	---	---	---

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

# Strings

M	e	t	a	p	o	d
---	---	---	---	---	---	---

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

**¡Ojo!** El largo del String es 7, pues tiene 7 letras, pero las posiciones van del 0 al 6.

# Strings

Los Strings se asignan de la siguiente forma:

```
String s = "Metapod";  
String s2 = "Bad Bunny"; // Este es un solo String
```

Y no se comparan con ==, se comparan con la función equals

```
boolean strings_iguales = s.equals(s2);
```

# Strings

**Ojo!** El String “Metapod” es distinto al String “metapod”

# Funciones de Strings

length permite obtener el largo del String

```
String s = "Metapod";  
int largo = s.length();
```

Y para obtener un carácter en específico lo obtenemos con charAt

```
char c = s.charAt(2); // El caracter vale t
```

# Arreglo



# Arreglos

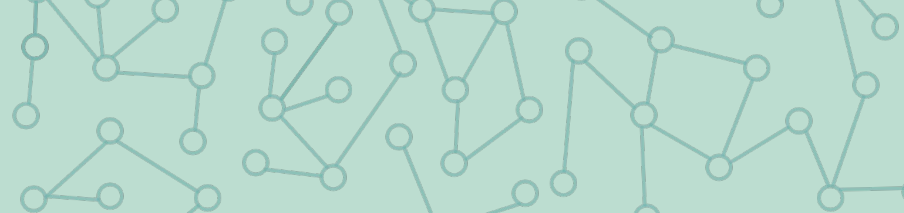
Una arreglo es una colección de datos de largo fijo, en que todos los datos deben ser del mismo tipo

Cada elemento ocupa una posición:

7	0	3	42	2	3	10
---	---	---	----	---	---	----

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

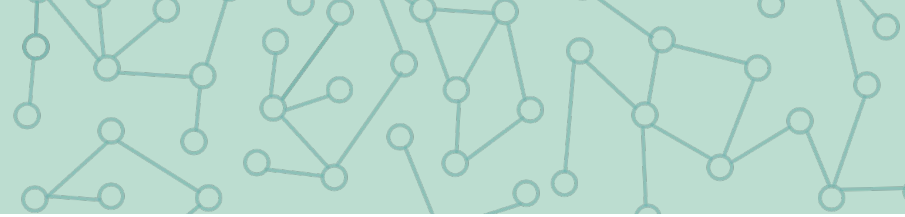
# ¿Por qué arreglos?



A veces necesitamos arreglos para ahorrarnos variables:

```
int a1;  
int a2;  
int a3;  
...
```

# ¿Por qué arreglos?



A veces no sabemos cuántas variables vamos a necesitar!

Imaginemos que el usuario va a ingresar una cantidad desconocida de números, y quiere que se los entreguemos ordenados

Para resolver la tarea anterior necesitamos utilizar arreglos

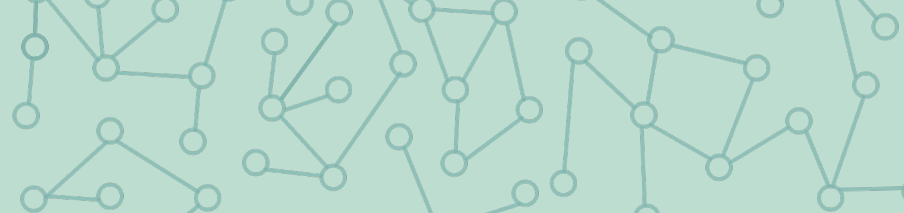
# Cómo crear un arreglo

# Declarar arreglos

Para declarar un arreglo:

```
TipoDeDato[] arreglo;  
int[] arreglo2;  
double[] arreglo3;  
boolean[] arreglo4;  
String[] arreglo5;
```

# Creando arreglos



Para crear un arreglo:

```
TipoDeDato[] arreglo = new TipoDeDato[TamañoDelArray];
```

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

```
String[] miArregloDeStrings;
```

```
miArregloDeStrings = new String[20];
```

# Creando arreglos

Para crear un arreglo:

```
int[] miArreglo = {1, 10, -9, 0};
```

```
String[] miArregloDeStrings;  
miArregloDeStrings = {"Bad Bunny", "Brytiago", "Darrell"};
```

# Operaciones Básicas



# Acceder a un elemento

Para acceder a un elemento en particular del arreglo:

```
String[] miArreglo = {"Perro", "Gato", "Pato", "Rana"};
```

```
String elemento = miArreglo[2];  
System.out.println(elemento)
```

```
// El programa imprime Pato
```

# Modificar un elemento

Para modificar un elemento en particular del arreglo:

```
String[] miArreglo = {"Perro", "Gato", "Pato", "Rana"};
```

```
miArreglo[2] = "Conejo Malo";  
System.out.println(miArreglo[2])
```

```
// El programa imprime Conejo Malo
```

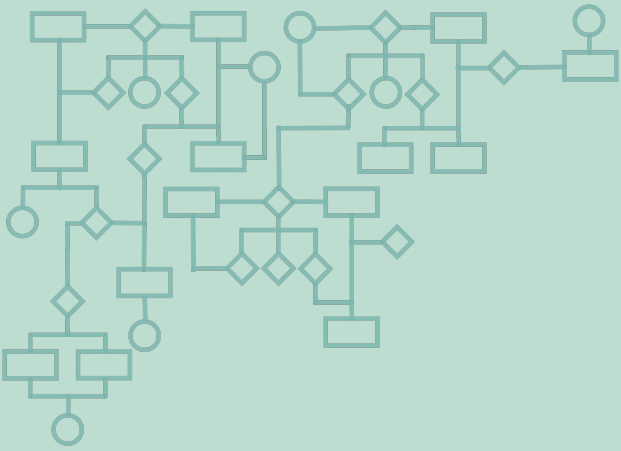
# Largo de un arreglo

Para conocer el largo de un arreglo debemos usar la propiedad `length`

```
String[] miArreglo = {"Perro", "Gato", "Pato", "Rana"};
```

```
int largo = miArreglo.length; // Ojo! No lleva paréntesis  
System.out.println(largo)
```

```
// El programa imprime 4
```



# ¡Ejemplo en vivo!



# Más sobre arreglos

# Iterar sobre un arreglo



# Iterar sobre un arreglo

```
public class ArregloDePrueba {  
  
    public static void main(String[] args) {  
        int[] miLista = {1955, 1978, 2018};  
  
        // Imprime todos los elementos del arreglo.  
        for (int i = 0; i < miLista.length; i++) {  
            System.out.println(miLista[i] + " ");  
        }  
    }  
}
```

# Recibir un arreglo por consola





# Recibir un arreglo por consola

```
public class ArregloDePrueba {  
  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        // Recibe el largo del arreglo y lo inicializa.  
        int n = scanner.nextInt();  
        int[] arreglo = new int[n];  
  
        // Recibe los elementos del arreglo y los guarda.  
        for (int i = 0; i < n; i++) {  
            arreglo[i] = scanner.nextInt();  
        }  
  
        scanner.close(); // Cierra el scanner.  
    }  
}
```

# Buscar un elemento en un arreglo



# Buscar un elemento en un arreglo

```
public class ArregloDePrueba {  
    public static void main(String[] args) {  
        int[] miLista = {3, 1, 4, 1, 5, 9, 2};  
        int buscado = 5;  
        boolean encontrado = false;  
  
        // Revisa si 5 está en el arreglo.  
        for (int i = 0; i < miLista.length; i++) {  
            if (miLista[i] == buscado) {  
                encontrado = true;  
                break;  
            }  
        }  
  
        if (encontrado) {  
            System.out.println("El dato se encuentra en el arreglo. :)))");  
        } else {  
            System.out.println("Raio: el dato no se encuentra en el arreglo.");  
        }  
    }  
}
```

# Comparar dos arreglos



# Comparar dos arreglos

```
public class ArregloDePrueba {
    public static void main(String[] args) {
        double[] milistaUno = {1.9, 2.9, 3.4, 3.5};
        double[] milistaDos = {1.9, 2.9, 3.4, 3.6};
        boolean iguales = true;

        // Primero, revisamos si son del mismo largo.
        if (milistaUno.length != milistaDos.length) {
            iguales = false;
        }

        // Luego, revisamos dato por dato.
        for (int i = 0; i < milistaUno.length; i++) {
            if (milistaUno[i] != milistaDos[i]) {
                iguales = false;
                break;
            }
        }

        if (iguales) {
            System.out.println("Son iguales.");
        } else {
            System.out.println("Noo, noo, noo, noooo... no son iguales.");
        }
    }
}
```

# Hackerrank

[www.hackerrank.com/c10-arreglos](https://www.hackerrank.com/c10-arreglos)

