

### Declaraciones

#### Clase

```
[public] [abstract] [final] class Nombre [extends Padre] {[atributos] [métodos]}
```

#### Variables

```
[public | private | protected] [static] [final] tipo nombre [= valor];
```

#### Métodos

```
[public | private | protected] [static] [abstract] [final] tipo nombre ([parámetros]) {[código]}
```

### Tipos de Datos

Dato	Bytes	Tipo	Ejemplo
byte	1	entero	24
short	2	entero	32
int	4	entero	105
long	8	entero	102459
float	4	decimal	12.5f
double	8	decimal	17.84
char	2	caracter	'A'
boolean	1	true/false	true
String	dinámico	cadena	"Hola"

### Conversión de Datos

De	A	Código
String	int	<code>int i = Integer.parseInt(str);</code>
String	float	<code>float f = Float.parseFloat(str);</code>
String	double	<code>double d = Double.parseDouble(str);</code>
Cualquiera	String	<code>String s = String.valueOf();</code>
Número	Número	<code>(nuevoTipo) var = (nuevoTipo) valor;</code>

### Métodos ArrayList

Uso: `ArrayList<tipo> lista = new ArrayList<tipo>();`

<code>add(elem)</code>	añadir elemento
<code>get(n)</code>	obtener <i>n</i> -ésimo elemento
<code>size()</code>	obtener el tamaño de la lista
<code>remove(n)</code>	eliminar el <i>n</i> -ésimo
<code>set(n, elem)</code>	establecer el <i>n</i> -ésimo elemento a <i>elem</i>

### Métodos HashMap

Uso: `HashMap<clave, valor> mapa = new HashMap<clave, valor>();`

<code>put(clave, elem)</code>	añade <i>elem</i> con la clave <i>clave</i>
<code>get(clave)</code>	obtener el elemento con <i>clave</i>
<code>containsKey(clave)</code>	true si contiene <i>clave</i> ; false si no

### Programa de Ejemplo

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Guardar como `HelloWorld.java`

Compilar: `javac HelloWorld.java`

Ejecutar: `java HelloWorld`



### Operadores

<code>x + y</code>	Suma	<code>x &gt; y</code>	Mayor
<code>x - y</code>	Resta	<code>x &lt; y</code>	Menor
<code>x * y</code>	Multiplic.	<code>x &gt;= y</code>	Mayor o igual
<code>x / y</code>	División	<code>x &lt;= y</code>	Menor o igual
<code>x % y</code>	Módulo	<code>x == y</code>	Igual
<code>x++</code>	Increment.	<code>x != y</code>	Diferente
<code>x--</code>	Decrem.	<code>!x</code>	No
<code>x    y</code>	Ó	<code>x &amp;&amp; y</code>	Y

### Métodos de Strings

Uso: `string.método(parámetros)`

<code>length()</code>	longitud de <i>s</i>
<code>charAt(n)</code>	saca el <i>n</i> -ésimo carácter
<code>substring(n, m)</code>	saca la cadena entre <i>n</i> y <i>m</i>
<code>indexOf(c)</code>	primera posición de <i>c</i>
<code>split(regex)</code>	divide por <i>regex</i>
<code>replace(a, b)</code>	reemplaza <i>a</i> por <i>b</i>
<code>trim()</code>	recorta espacios al principio y al final
<code>equals(s)</code>	compara con otra string: true si son iguales, false si no
<code>toUpperCase()</code>	la pasa a mayúsculas
<code>toLowerCase()</code>	la pasa a minúsculas

### Control de Flujo

#### Sentencia If

```
if (expresión) {código}
else if (expresión) {código} ...
else {código}
```

#### Bucle While

```
while (expresión) {código}
```

#### Bucle Do-While

```
do {código} while (expresión);
```

#### Bucle For

```
for (int i = 0; i < max; i++) {código}
```

#### Bucle For-Each

```
for(tipo variable : colección){código}
```

#### Switch

```
switch (variable) {
case valor: código; break;
case valor2: código; break; ...
default: código;}
```

#### Bloque Try-Catch

```
try {código peligroso}
catch (TipoExcepción e) {código}
catch (TipoExcepción2 e) {código}
finally {código}
```



By [Paulo2468](#)  
[cheatography.com/paulo2468/](https://cheatography.com/paulo2468/)

Not published yet.  
Last updated 24th February, 2025.  
Page 3 of 2.

Sponsored by [ApolloPad.com](#)  
Everyone has a novel in them. Finish Yours!  
<https://apollopad.com>