

# Estructura de datos y algoritmos

Rodrigo Alvarez

[rodrigo.alvarez2@mail\\_udp.cl](mailto:rodrigo.alvarez2@mail_udp.cl)

# Que es java?

# Que es java?



# Que es java?

- Desarrollado por Sun Microsystems
- Lanzado en 1995
- Actualmente propiedad de Oracle
- Open Source



# Que es java?

- Lenguaje compilado a bytecode  
(corre encima de la jvm)
- Fuertemente orientado a objetos
- Sintaxis similar a C++
- Multiplataforma



Domina todos lenguajes de programación estudiando en EDteam

[ed.team/cursos](http://ed.team/cursos)



## MAJOR COMPANIES THAT USE



# Hello world

The screenshot shows a Java code editor interface. At the top, there's a header bar with tabs for "Main.java" and a "+" button, followed by the title "Java Hello World". On the right of the header are buttons for "NEW", "JAVA ▾", "RUN ►", and three vertical dots. The main area has a dark background. On the left, there's a vertical toolbar with a "1" icon. The main workspace shows the code for a Java application:

```
1 public class Main {  
2     public static void main(String[] args) {  
3         System.out.println("Hello World");  
4     }  
5 }
```

To the right of the code, there are sections for "STDIN" and "Output". The "STDIN" section contains placeholder text: "Input for the program ( Optional )". The "Output" section contains the instruction: "Click on RUN button to see the output".

| Tipo dato | Tamaño  | Descripción   |
|-----------|---------|---|
| byte      | 1 byte  | números del -128 al 127                                 |
| short     | 2 bytes | -32,768 al 32,767                                       |
| int       | 4 bytes | -2,147,483,648 al 2,147,483,647                         |
| long      | 8 bytes | -9,223,372,036,854,775,808 al 9,223,372,036,854,775,807 |
| float     | 4 bytes | números racionales, hasta 7 dígitos decimales           |
| double    | 8 bytes | hasta 15 dígitos decimales                              |
| boolean   | 1 bit   | true o false  |
| char      | 2 bytes | un único carácter                                       |

# Operadores aritméticos

| Operator | Name           | Description                            | Example |
|----------|----------------|--|---------|
| +        | Addition       | Adds together two values               | x + y   |
| -        | Subtraction    | Subtracts one value from another       | x - y   |
| *        | Multiplication | Multiplies two values                  | x * y   |
| /        | Division       | Divides one value by another           | x / y   |
| %        | Modulus        | Returns the division remainder         | x % y   |
| ++       | Increment      | Increases the value of a variable by 1 | ++x     |
| --       | Decrement      | Decreases the value of a variable by 1 | --x     |

# Operadores de asignación

| Operator | Example | Same As   |
|----------|---------|-----------|
| =        | x = 5   | x = 5     |
| +=       | x += 3  | x = x + 3 |
| -=       | x -= 3  | x = x - 3 |
| *=       | x *= 3  | x = x * 3 |
| /=       | x /= 3  | x = x / 3 |
| %=       | x %= 3  | x = x % 3 |

# Operadores de comparación

| Operator           | Name                     | Example                |
|--------------------|--------------------------|------------------------|
| <code>==</code>    | Equal to                 | <code>x == y</code>    |
| <code>!=</code>    | Not equal                | <code>x != y</code>    |
| <code>&gt;</code>  | Greater than             | <code>x &gt; y</code>  |
| <code>&lt;</code>  | Less than                | <code>x &lt; y</code>  |
| <code>&gt;=</code> | Greater than or equal to | <code>x &gt;= y</code> |
| <code>&lt;=</code> | Less than or equal to    | <code>x &lt;= y</code> |

# Operadores lógicos

| Operator | Name        | Description                           | Example                                       |
|----------|-------------|---------------------------------------|---|
| &&       | Logical and | true if both statements are true      | <code>x &lt; 5 &amp;&amp; x &lt; 10</code>    |
|          | Logical or  | true if one of the statements is true | <code>x &lt; 5    x &lt; 4</code>             |
| !        | Logical not | Reverse the result                    | <code>!(x &lt; 5 &amp;&amp; x &lt; 10)</code> |

# I/O: Input/Output en java

The screenshot shows a Java code editor interface. The top bar includes tabs for "lo.java" and "+", a file identifier "427p45t7h", and buttons for "NEW", "JAVA ▾", "RUN ▶", and a three-dot menu. The code area contains a single line of code: "1". To the right, under "STDIN", is the input "23". Below the code area, under "Output:", is the program's output:  
Ingresá dos ints:  
el primer numero es: 2  
el segundo numero es: 3  
La suma de los numeros es: 5

# Scanner class

La clase `Scanner` se utiliza para obtener la entrada del usuario, y es parte del paquete `java.util`.

| Method                     | Description                                      |
|----------------------------|--|
| <code>nextBoolean()</code> | Reads a <code>boolean</code> value from the user |
| <code>nextByte()</code>    | Reads a <code>byte</code> value from the user    |
| <code>nextDouble()</code>  | Reads a <code>double</code> value from the user  |
| <code>nextFloat()</code>   | Reads a <code>float</code> value from the user   |
| <code>nextInt()</code>     | Reads a <code>int</code> value from the user     |
| <code>nextLine()</code>    | Reads a <code>String</code> value from the user  |
| <code>nextLong()</code>    | Reads a <code>long</code> value from the user    |

# Scanner

The screenshot shows a Java code editor interface with the following details:

- Title Bar:** The title bar displays "ScannerEx.java" and "427nv54rs".
- Toolbar:** The toolbar includes buttons for "NEW", "JAVA ▾", "RUN ▶", and a three-dot menu.
- Code Area:** The code area contains a single line of code: "1".
- stdin Area:** The "STDIN" section shows the input provided to the program: "hola", "3", and "3.14".
- Output Area:** The "Output" section shows the program's response to the inputs:
  - "Ingrese una palabra"
  - "La palabra ingresada es: hola"
  - "Ingrese un numero"
  - "El numero ingresado es: 3"
  - "Ingrese un numero flotante"
  - "El numero flotante ingresado es: 3.14"

# if, else, while, for

The screenshot shows a Java code editor interface. On the left, there is a dark sidebar with a file icon and a search bar containing the text "IfElseWhileForExample.java". Below the search bar is a list of recent files: "427nvkqmd" (selected), "427nvkqmd", "427nvkqmd", and "427nvkqmd". To the right of the sidebar is a toolbar with buttons for "NEW", "JAVA ▾", "RUN ▶", and a three-dot menu. The main workspace has a dark background. In the top-left corner of the workspace, the number "1" is displayed. The code itself is not visible in the image.

427nvkqmd

NEW JAVA ▾ RUN ▶ :

1

STDIN

17

Output:

```
the number is odd
numeros pares:
2
4
6
8
10
12
14
16
divisibles por 3:
3
```

# Strings

The screenshot shows a Java code editor interface. The top bar includes tabs for "StringExample.java" and "+", a file identifier "427nu2xbm", and buttons for "NEW", "JAVA ▾", "RUN ▶", and a menu icon. The main area displays the following code:

```
1
```

The output window shows the following results:

STDIN  
23

---

Output:  
H  
e  
l  
l  
o  
  
W  
o  
r  
l  
d  
Hello World. Goodbye

## String methods

# Arrays

The screenshot shows a Java code editor interface. At the top, there are tabs for "ArrayExample.java" and a plus sign, with the identifier "427p4c9xn" next to it. On the right side, there are buttons for "NEW", "JAVA ▾", "RUN ▶", and three vertical dots. The main area has a dark background. On the left, a vertical bar has the number "1" at its top. The central area contains the following code:

```
public class ArrayExample {  
    public static void main(String[] args) {  
        int[] arr = {1, 2, 3, 4};  
        System.out.println(arr[3]);  
    }  
}
```

To the right of the code, under the heading "STDIN", is the placeholder text "Input for the program ( Optional )". Below the code, under the heading "Output:", is the number "4".

# Métodos

```
[modificador de acceso] [tipo_retorno] nombre_funcion(tipo_1 parametro_1, tipo_2 parametro_2, ...){  
    // instrucciones  
    return valor;  
}
```

# Métodos

Main.java      Greeter.java      +      427p5fbvj      NEW      JAVA ▾      RUN ►      :

1

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

STDIN  
Input for the program ( Optional )

---

Output:  
hi!

# Clases

The screenshot shows a Java code editor interface. At the top, there is a header bar with the following elements from left to right: "ClassExample.java", a "+" button, "427pmavjh", "NEW", "JAVA ▾", "RUN ▶", and a three-dot menu icon. The main area has a dark background. On the left, there is a vertical toolbar with a single item labeled "1". The main workspace contains the following code:

```
1 public class ClassExample {
```

To the right of the code, there are two sections: "STDIN" and "Output". The "STDIN" section contains the placeholder text "Input for the program ( Optional )". The "Output" section contains the text "miauuuu".

# Herencia

The screenshot shows a Java code editor interface. At the top, there's a header with "Car.java" and a "+" button on the left, and "427pqr5m" in the center. On the right, there are buttons for "NEW", "JAVA ▾", "RUN ►", and three vertical dots. The main area has a dark background. On the left, a code editor window shows the number "1" at the top left. On the right, there are two sections: "STDIN" which contains placeholder text "Input for the program ( Optional )", and "Output:" followed by the text "Tuut, tuut!" and "Ford 2023 4".

```
1
Tuut, tuut!
Ford 2023 4
```

## Interfaces

Las interfaces en Java son una colección de métodos abstractos y constantes que pueden ser implementados por cualquier clase que las implemente.

Una interfaz define un conjunto de métodos y su firma, pero no proporciona implementaciones de los métodos.

# Interfaces

The screenshot shows a Java code editor interface. At the top, there's a header with "Main.java" and a "+" button on the left, "427pruj44" in the center, and "NEW", "JAVA ▾", "RUN ▶", and a three-dot menu icon on the right. The main area has a dark background. On the left, a vertical bar has the number "1" at its top. The central workspace is empty. To the right, there's a sidebar with "STDIN" and "Input for the program (Optional)" fields, which are currently empty. Below this is a horizontal line. Underneath the line, the word "Output:" is followed by the text "The pig says: wee wee Zzz".

# Generics

The screenshot shows a Java code editor interface. On the left, there is a file tab labeled "Main.java" with a line number "1" above it. To the right of the tabs is a "+" button. The main workspace contains the following code:

```
1 public class Main {  
2     public static void main(String[] args) {  
3         System.out.println("Hello, World!");  
4     }  
5 }
```

At the top right of the editor, there are several buttons: "NEW", "JAVA ▾", "RUN ►", and a three-dot menu icon.

On the right side of the editor, there are two sections: "STDIN" and "Output". The "STDIN" section contains the placeholder text "Input for the program ( Optional )". The "Output" section contains the text "Output:" followed by the number "3".

- `ArrayList`: Implementa un **TDA List** en array manipulando el tamaño de forma dinámica.
- `LinkedList`: Implementa un **TDA List** sobre una lista de nodos.
- `Queue`: Implementa un **TDA FIFO** en una lista de nodos.
- `Stack`: Implementa un **TDA LIFO** en una lista de nodos.
- `Map`: Implementa una estructura **TDA key-value** en un árbol o una tabla hash.
- `Set`: Implementa una estructura **TDA unique value** en un árbol o una tabla hash.

