

# 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. On the left, there is a code editor window titled "Main.java" containing the number "1". To the right of the code editor are several toolbars and buttons. The top toolbar includes a "+" button for creating new files, the title "Java Hello World", a "NEW" button, a "JAVA" dropdown menu, a pink "RUN ▶" button, and a three-dot menu icon. Below the toolbar, there are two input fields: "STDIN" and "Input for the program (Optional)". Underneath these fields is a large text area labeled "Output:" which 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 with the following details:

- Title Bar:** The title bar displays "lo.java" and "427p45t7h".
- 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 contains the input "2 3".
- Output Area:** The "Output" section displays the program's output:
  - "Ingrresa 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. The top bar includes tabs for "ScannerEx.java" and "427nv54rs", along with buttons for "NEW", "JAVA ▾", "RUN ▶", and a three-dot menu. The main area has two panes: "STDIN" on the left and "Output" on the right. In the "STDIN" pane, the user has typed "hola". In the "Output" pane, the program's response is displayed in a vertical list:

```
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 file tab labeled "IfElseWhileForExample.java". To its right is a plus sign icon. Further right is a code identifier "427nvkqmd". At the top right, there are several buttons: "NEW", "JAVA ▾", "RUN ▶", and a three-dot menu icon.

The main area consists of two panes. The left pane is titled "STDIN" and contains the input value "17". The right pane is titled "Output" and displays the following text:

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

# Strings

The screenshot shows a Java code editor interface. On the left, there is a file list with "StringExample.java" selected. In the main area, the code is as follows:

```
1 public class StringExample {  
2     public static void main(String[] args) {  
3         String str = "Hello World";  
4         System.out.println(str);  
5     }  
6 }
```

On the right, the execution environment is shown. It includes tabs for "NEW", "JAVA ▾", "RUN ▶", and a three-dot menu. The "JAVA" tab is currently active. The "STDIN" section contains the input "23". The "Output:" section shows the printed string "Hello World".

## String methods

# Arrays

The screenshot shows a Java code editor interface. The top bar includes tabs for "ArrayExample.java" and "+", a file identifier "427p4c9xn", and buttons for "NEW", "JAVA ▾", "RUN ▶", and a three-dot menu. The main area has a dark background. On the left, a vertical toolbar shows the number "1". The central workspace contains the following code:

```
1
```

The right side of the interface displays the execution environment. It shows "STDIN" and an optional input field containing "Input for the program ( Optional )". Below that, it shows "Output:" followed by the value "4".

# Métodos

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

# Métodos

The screenshot shows a Java development environment. On the left, there are two tabs: "Main.java" and "Greeter.java". Below the tabs is a line number 1. In the center, there is a placeholder text "427p5fbvj". At the top right, there are buttons for "NEW", "JAVA ▾", "RUN ▶", and a three-dot menu. To the right of the code area, there is a "STDIN" section with a placeholder "Input for the program ( Optional )". Below it, there is an "Output:" section containing the text "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 is divided into two vertical sections. The left section is a code editor with a single line of code: "1". The right section contains two panels: "STDIN" which has a placeholder "Input for the program ( Optional )", and "Output:" which displays the text "miauuuu".

# Herencia

The screenshot shows a Java code editor interface. On the left, there is a code editor window titled "Car.java" with a line number 1. To the right of the code editor are several tabs: "+", "427pqr5m", "NEW", "JAVA ▾", "RUN ►", and three vertical dots. The "JAVA ▾" tab is currently selected. Below the tabs, there are two sections: "STDIN" and "Output". The "STDIN" section contains the placeholder text "Input for the program ( Optional )". The "Output" section displays the following text:  
Output:  
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. On the left, there is a dark sidebar with a file icon and a search bar. The main workspace has tabs for "Main.java" and a "+" sign. The "Main.java" tab is active, showing the number "1" in a blue box. The code area is currently empty. To the right of the code area are several buttons: "NEW", "JAVA ▾", "RUN ►", and a three-dot menu. The status bar at the bottom displays the identifier "427pruj44".

Main.java + 427pruj44 NEW JAVA ▾ RUN ► :

1

STDIN

Input for the program ( Optional )

Output:

The pig says: wee wee  
Zzz

# Generics

The screenshot shows a Java code editor interface. On the left, there is a dark sidebar with a vertical scroll bar. In the main area, a file named "Main.java" is open, indicated by the number "1" in the top-left corner of the editor pane. To the right of the editor are several tabs: "+", "427psdcfw", "NEW", "JAVA ▾", "RUN ►", and three vertical dots. The "JAVA ▾" tab is currently selected. Below the tabs, there are two sections: "STDIN" and "Output:". The "STDIN" section contains the placeholder text "Input for the program ( Optional )". The "Output:" section displays 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.

