



UNIVERSIDAD DE ALMERÍA

# Grado en Ingeniería Informática

## Introducción a la Programación

### 2021-2022



# Tema 1. Fundamentos de Programación

- Algoritmos y programas
- Introducción a la programación
- Datos y tipos de datos
- Tipos de datos primitivos en Java
- Variables
- Expresiones y sentencias
- Programas
- Estructuras de control
- Uso de subprogramas
- Métodos

# Tema 1. Fundamentos de Programación



## Algoritmos y programas



☐ Algoritmo

☐ Programa

☐ Lenguajes de programación

☐ Traductores

☐ Herramientas de programación

☐ Ciclo de vida del software

# Tema 1. Fundamentos de Programación

## ➤ Algoritmo

Secuencia ordenada de pasos, bien definidos, que resuelve un problema concreto.

### Características

- ✓ Corrección (sin errores).
- ✓ Precisión (ausencia de ambigüedades).
- ✓ Repetitividad (solución genérica de un problema dado).
- ✓ Finitud (número finito de órdenes).
- ✓ Eficiencia (temporal [tiempo necesario] y espacial [memoria utilizada])

# Tema 1. Fundamentos de Programación

## ➤ Programa

Implementación de un algoritmo en un lenguaje de programación



# Tema 1. Fundamentos de Programación

## ➤ Programa

**Conjunto ordenado de instrucciones** que se dan al ordenador indicándole las operaciones o tareas que ha de realizar para resolver un problema.

Una **instrucción** es un conjunto de símbolos que representa una orden para el ordenador: la ejecución de una operación con datos

# Tema 1. Fundamentos de Programación

## ➤ Lenguajes de programación

Idiomas artificiales diseñado para que sea fácilmente entendible por un humano e interpretable por una máquina

Se forman con **símbolos** tomados de un determinado repertorio (componentes léxicos)

Se construyen siguiendo unas **reglas** precisas (sintaxis)

- Lenguaje máquina
- Lenguaje ensamblador
- Lenguajes de alto nivel

# Tema 1. Fundamentos de Programación

## ➤ Lenguajes de programación

### *Clasificación de los lenguajes de programación de alto nivel*

#### ❖ Lenguajes imperativos:

Los programas indican al ordenador de forma inequívoca los pasos a seguir para la resolución de un problema.

- Programación estructurada (C, Pascal, Fortran, ...)
- Programación orientada a objetos (Smalltalk, C++, Java, C#, ...)

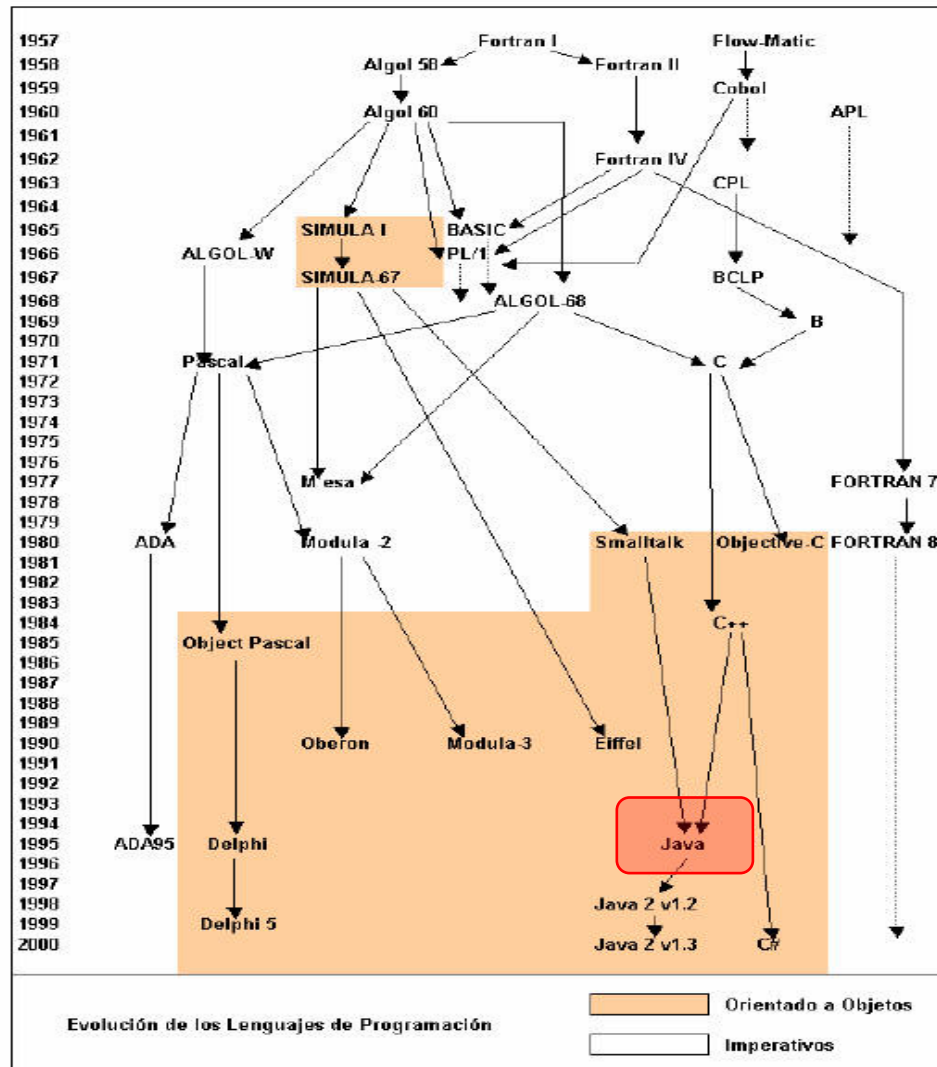
#### ❖ Lenguajes declarativos (funcionales y lógicos):

Los programas se implementan como conjuntos de funciones (lógicas) cuya evaluación nos dará el resultado deseado. Haskell, LISP, Erlang, Scala, Clojure, F#, Miranda, ...

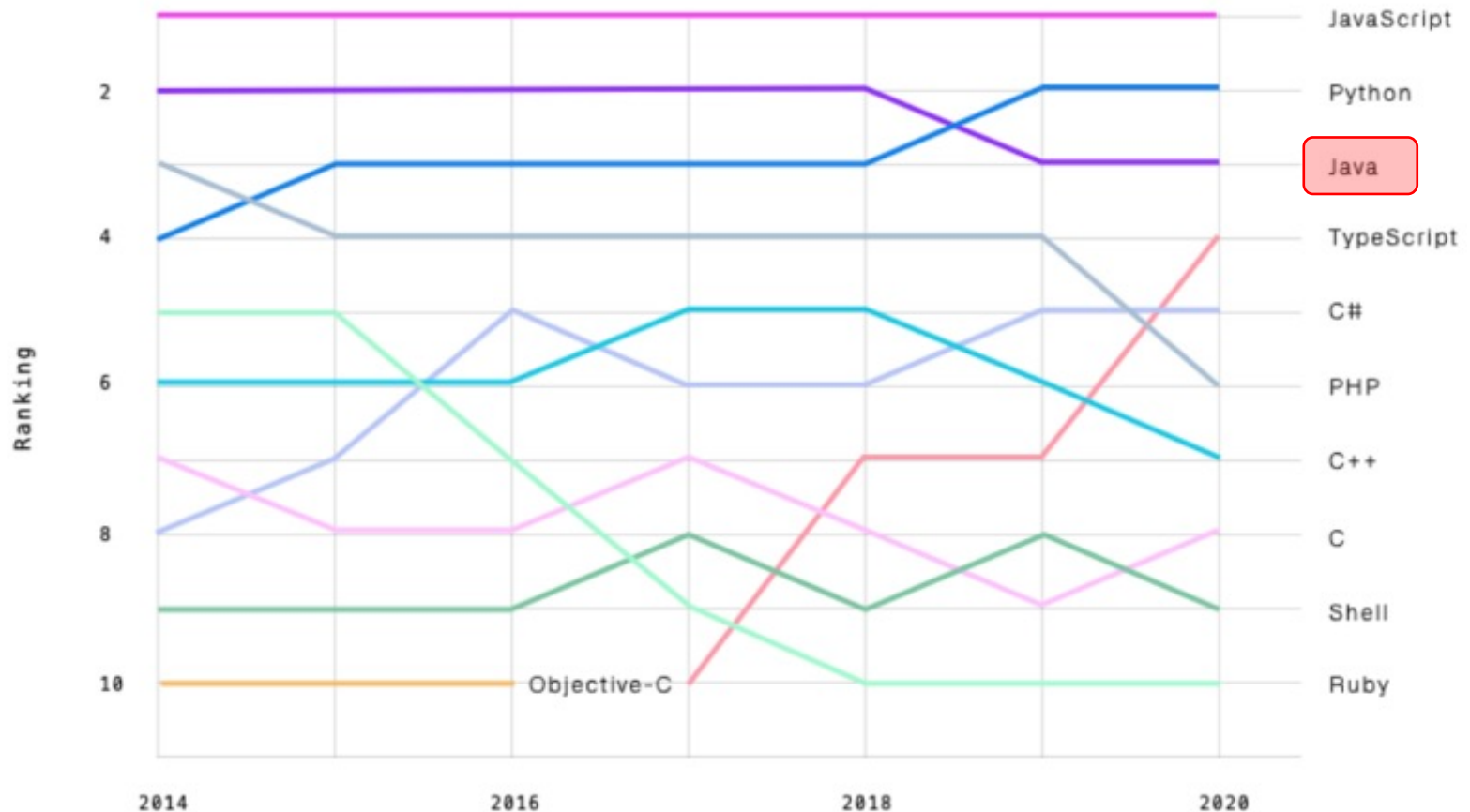
**Lógicos:** Prolog, Alf, Elf, Fish, Lambda Prolog, etc.



## Evolución inicial de los lenguajes de programación imperativos



## Los 10 lenguajes de programación más populares según **Github**

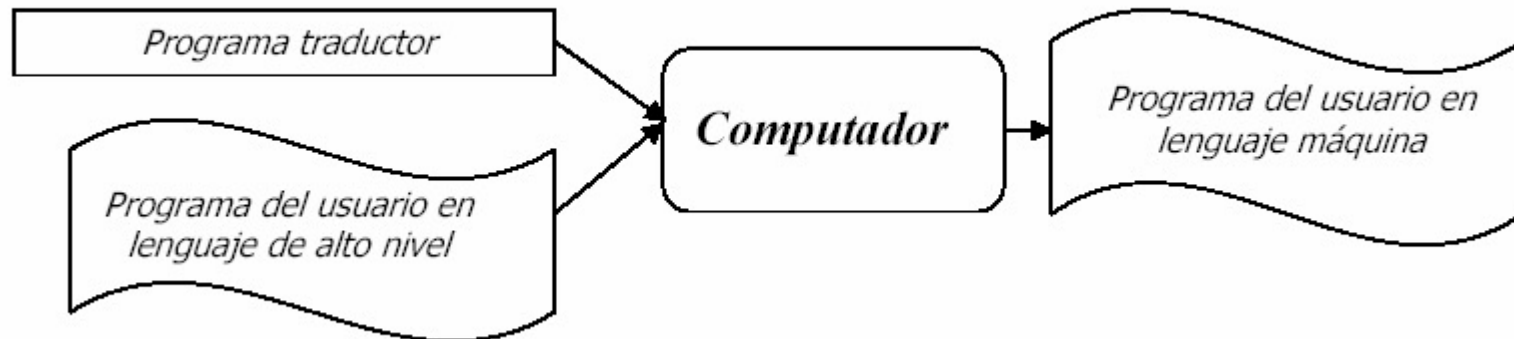


Otros lenguajes con impacto en la actualidad: Scala, Go, Swift, Groovy, Visual Basic, HTML, Perl, Kotlin, Objective-C, Rust, Ensamblador, R, ...

# Tema 1. Fundamentos de Programación

## ➤ Traductores

Los traductores transforman programas escritos en un lenguaje de alto nivel en programas escritos en código máquina



# Tema 1. Fundamentos de Programación

## ➤ Traductores

### *Tipos de traductores*

- **Compiladores**



Generan un programa ejecutable a partir del código fuente

- **Intérpretes**

Van analizando, traduciendo y ejecutando las instrucciones del programa una a una. No se traduce una instrucción hasta que la ejecución de la anterior haya finalizado.

# Tema 1. Fundamentos de Programación

## ➤ Herramientas de programación

Editores de código fuente, depuradores, herramientas de automatización de la compilación, ...

**IDEs** (entornos integrados de desarrollo)

*Ejemplos : Microsoft Visual Studio .NET, **Eclipse**, Netbeans, Visual Studio Code, IntelliJ Idea, BlueJ, jGRASP, Xcode, JDeveloper, Codenvy, etc.*

# Tema 1. Fundamentos de Programación

## ➤ Por qué utilizar los Java IDEs

Proporcionan plantillas de código predeterminada.

Validan automáticamente la sintaxis Java.

Son compatible con la refactorización de código.

Admiten la coloración de sintaxis.

Permiten establecer puntos de interrupción.

Ofrecen un depurador robusto.

Proporcionan un editor de Java robusto y potente.

# Tema 1. Fundamentos de Programación

## Herramientas de programación (IDE-Eclipse)

The screenshot displays the Eclipse IDE interface. The Package Explorer on the left shows a project structure with sessions 01 through 10. The main editor window shows the code for `Distancia.java`, which calculates the distance between two points (2, 1) and (0, 0) using the Pythagorean theorem. The console at the bottom shows the output: "La distancia del punto (2, 1) al punto (0, 0) es 2.23606797749979".

```
16
17 public class Distancia {
18     public static void main(String[] args) {
19
20         int x = 2;
21         int y = 1;
22
23         // calculo de la distancia a
24         double dist = Math.sqrt(x * x + y * y);
25
26         System.out.println();
27
28         // salida de la distancia
29         System.out.println("La distancia del punto (" + x + ", " + y + ") al punto (" + 0 + ", " + 0 + ") es " + dist);
30     }
31 }
32
```

La distancia del punto (2, 1) al punto (0, 0) es 2.23606797749979

eclipse ide 2018-09

GI PP2019EclipsePhoton

# Tema 1. Fundamentos de Programación

## ➤ Ciclo de vida del software

Comprende las siguientes etapas:

- **Planificación**
- **Análisis (¿qué?):** Análisis de requisitos
- **Diseño (¿cómo?):** Estudio de alternativas
- **Implementación**
- **Depuración y pruebas**
- **Explotación: Uso y mantenimiento**



# Tema 1. Fundamentos de Programación



## Introducción a la programación



- ☐ La plataforma de programación Java
- ☐ Aplicaciones y applets
- ☐ Características de Java

# Tema 1. Fundamentos de Programación

## ➤ Qué es Java y objetivos principales

**Java** es un lenguaje de programación basado en clases y orientado a objetos.

**Objetivo:** permitir que los desarrolladores de aplicaciones escriban el código una sola vez y que ese código pueda ser ejecutado en cualquier computadora, WORA (Write Once, Run Anywhere)

**Java** se puede combinar con otras muchas tecnologías como Spring, node js, Android, Hadoop, J2EE, etc. y nos permite crear aplicaciones completas, robustas, escalables, portátiles y distribuidas.

**Java** también promueve la integración continua (Jenkins) y el control de versiones del código (Git, SVN), a través de pruebas unitarias, utilizando herramientas de testing (Selenium).

# Tema 1. Fundamentos de Programación

## ➤ Por qué aprender programación con Java

**Java** es uno de los principales lenguajes de programación en la actualidad y lo seguirá siendo al menos una década.

Como **Java** es seguro y multiproceso, es perfecto para los servicios de gestión de transacciones y banca.

El sistema operativo móvil como **Android** utiliza la API (Application Programming Interfaces) de Java. Los algoritmos del mercado de valores se basan en Java. Y más recientemente, la gestión de muchos de los grandes volúmenes de datos (**Big Data**) se tratan de forma sencilla mediante Java (Hadoop-MapReduce).

# Tema 1. Fundamentos de Programación

## ➤ Papel de Java en el desarrollo de software

Desarrollo web front-end: JavaScript

Desarrollo web back-end: JavaScript, **Java**, Python, PHP, Ruby

Desarrollo móvil: **Java**, Swift, Kotlin, C#

Desarrollo de videojuegos: C++, C#, **Java**

Aplicaciones de escritorio: **Java**, C#, C++, Python

Aplicaciones financieras y banca: **Java**, C#, JavaScript, Cobol

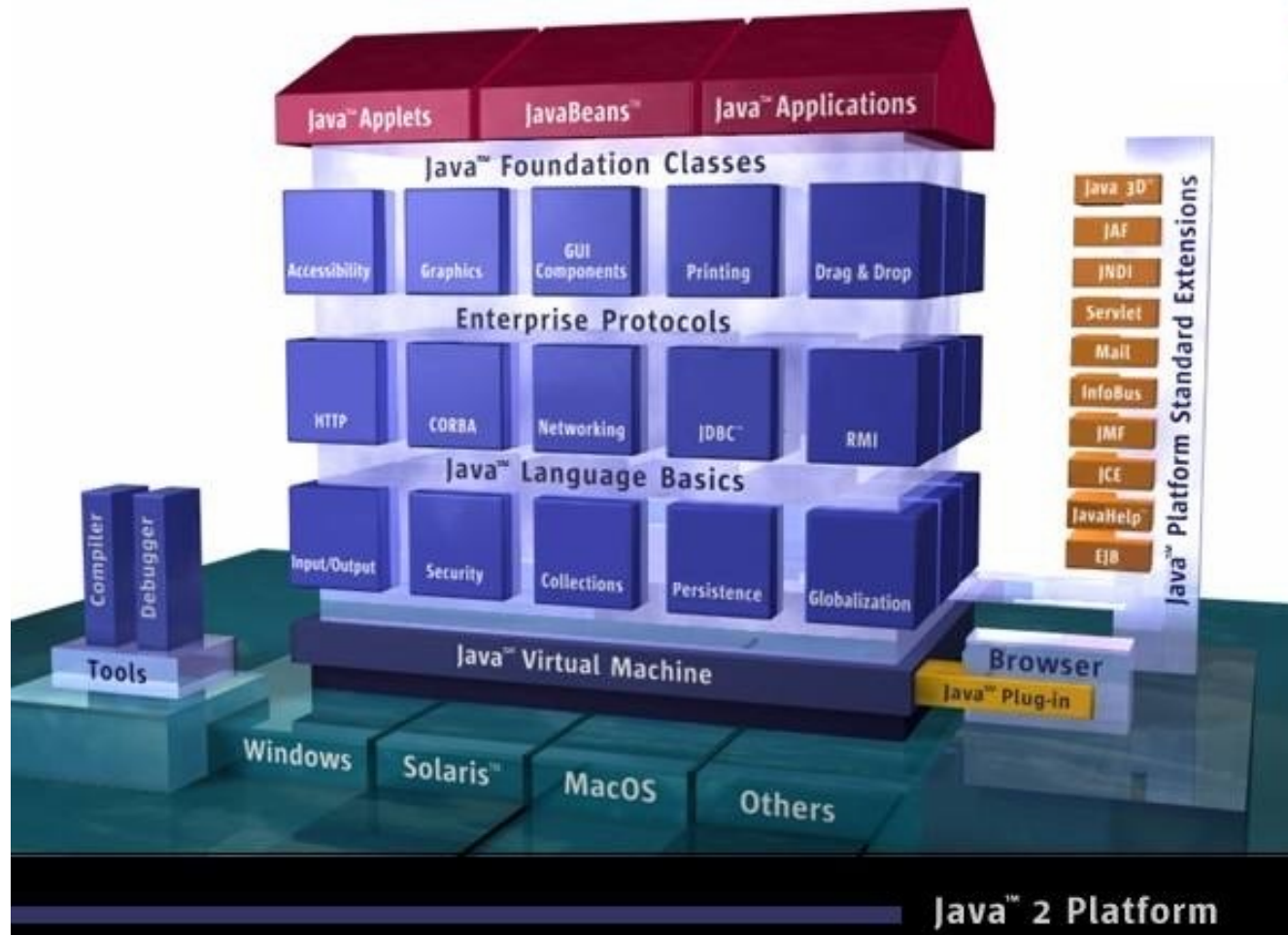
Desarrollo Big Data: Scala, **Java**, Python

Blockchain y Criptomonedas: C, C++, Solidity, Python

Programación de sistemas: C, Go, Rust

# Tema 1. Fundamentos de Programación

## ➤ La plataforma de programación Java



# Tema 1. Fundamentos de Programación

## ➤ La plataforma de programación Java

La **plataforma Java** es una plataforma de software

Los programas en Java pasan por la **Java Virtual Machine**, que convierte el código de bytes en código nativo, haciendo que el programa se ejecute en cualquier dispositivo. Esto significa que no necesita compiladores específicos por cada máquina donde se ejecute el código Java.

El lenguaje de programación **Java** (te permite desarrollar aplicaciones) es diferente de la plataforma Java.

La **plataforma Java** es una colección de programas y herramientas que facilita el desarrollo y ejecución de programas escritos en el lenguaje Java

# Tema 1. Fundamentos de Programación

## ➤ La plataforma de programación Java

Hay **cuatro plataformas Java** para el lenguaje de programación Java:

**Java SE** (Plataforma Java, Standard Edition)

**Java EE** (Plataforma Java, Enterprise Edition)

**Java FX** (Plataforma Java, Plataforma Gráfica)

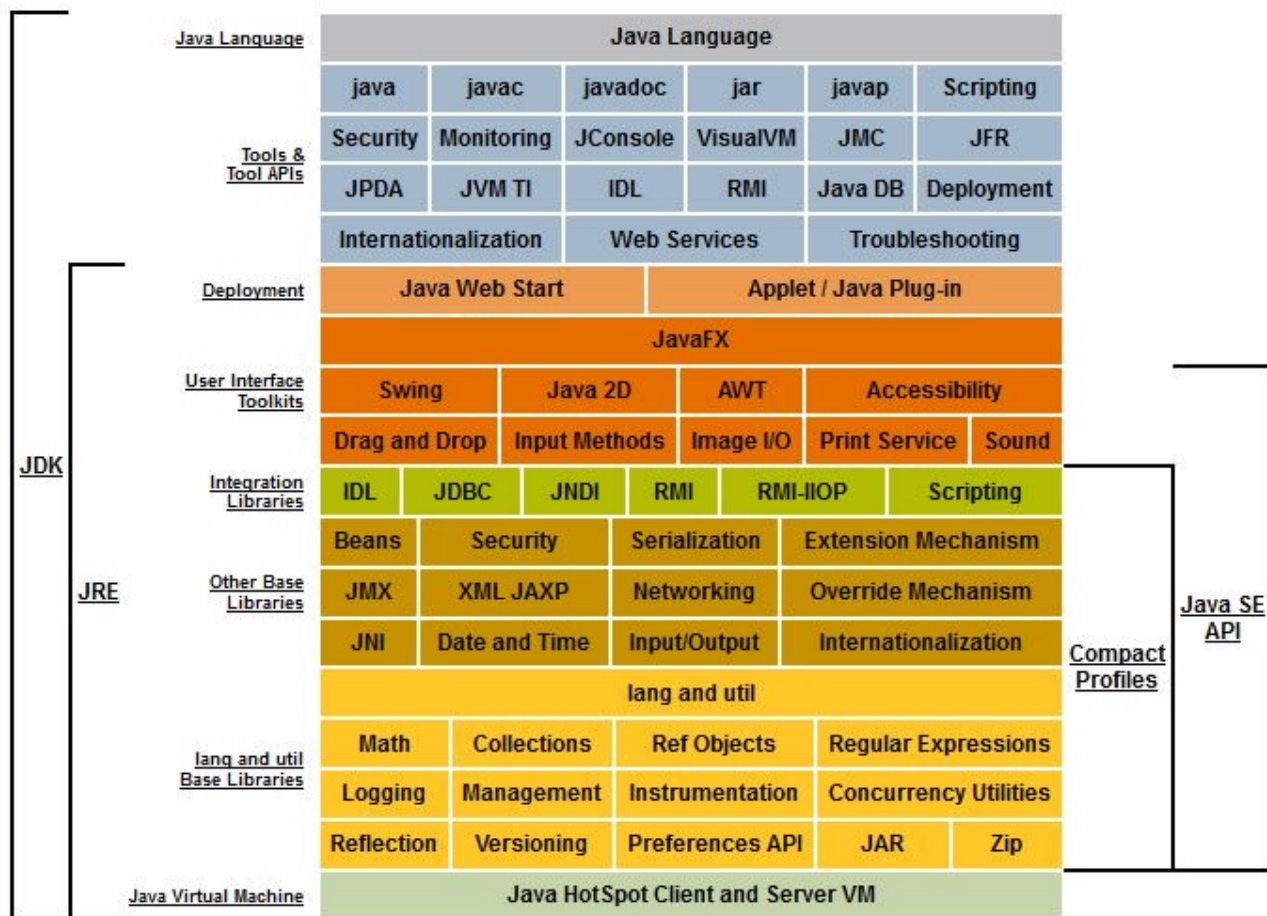
**Java ME** (Plataforma Java, Micro Edition)

Las aplicaciones independientes se pueden construir en la plataforma **Java SE**. La mayor parte de las aplicaciones empresariales depende de **Java EE**. **Java ME** es para aplicaciones que se ejecutan en dispositivos móviles e integrados.



# Tema 1. Fundamentos de Programación

## Tecnologías de Java SE





# Tema 1. Fundamentos de Programación

## ➤ Componentes de Java

Hay tres componentes principales de Java: **JVM**, **JDK** y **JRE**. **JDK** (Java Development Kit) es donde los desarrolladores escriben su código y lo ejecutan a través de **JRE** (Java Runtime Environment). Con la **JVM** (Java Virtual Machine), cualquier código escrito en Java se puede traducir al código de bytes de Java (Java Bytecode). Cualquier máquina puede implementar este código basado en el sistema operativo que tenga instalado. La **JVM** reside dentro del JRE junto con los paquetes Java (bibliotecas).

Para programar en Java debe instalar: JDK y JRE

# Tema 1. Fundamentos de Programación

## ➤ Componentes de Java

**JDK:** Herramientas de desarrollo JRE, compilador (javac), archivos jar (paquete y archivo), javadocs, etc.

**JVM:** La máquina abstracta donde se ejecutan los Bytecodes de Java. Consiste en un documento de especificación que describe la implementación de JVM, el programa de implementación real y la instancia de JVM (tiempo de ejecución) donde puede ejecutar su programa principal.

**JRE:** Implementación física (instancia de tiempo de ejecución) de JVM. Contiene los paquetes de la biblioteca y los archivos de soporte que JVM usa para ejecutar un programa.

# Tema 1. Fundamentos de Programación

## ➤ La plataforma de programación Java



**La máquina virtual Java (JVM: Java Virtual Machine)** Imprescindible para poder ejecutar aplicaciones Java.

**Las bibliotecas estándar de Java (Java Application Programming Interface = Java API)** Amplia colección de componentes.

**El lenguaje de programación Java.** Para escribir aplicaciones móviles, web, financieras, desarrollo de aplicaciones en la nube, Big Data, IoT, etc.

# Tema 1. Fundamentos de Programación

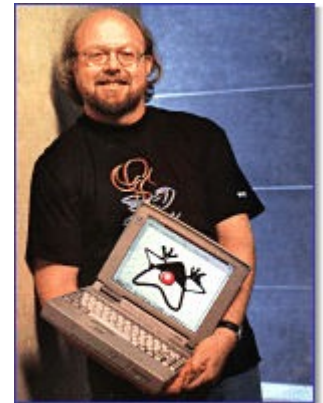
## ➤ La plataforma de programación Java

### *Historia de Java*

La versión más difundida sobre el origen es la que presenta a **Java** como un lenguaje pensado para pequeños electrodomésticos. **Sun Microsystems** decidió intentar introducirse en el mercado de la electrónica de consumo y desarrollar programas para pequeños dispositivos electrónicos.



James Gosling



# Tema 1. Fundamentos de Programación

## ➤ La plataforma de programación Java

### *Historia de Java*

El creador de Java es **James Gosling**, cuyo objetivo era implementar una maquina virtual y un lenguaje de programación con una estructura y sintaxis similar a C++, pero más sencillo de utilizar.

El nombre se debe a un tipo de café cultivado en la **isla de Java**.

La primera versión se lanzó en **1995** (Sun Microsystems).

Más tarde, **Oracle** adquirió Sun Microsystems, y a partir de ahí Oracle mantiene la versión principal.

La última versión de Java es **Java 16** lanzada en Marzo de 2021.

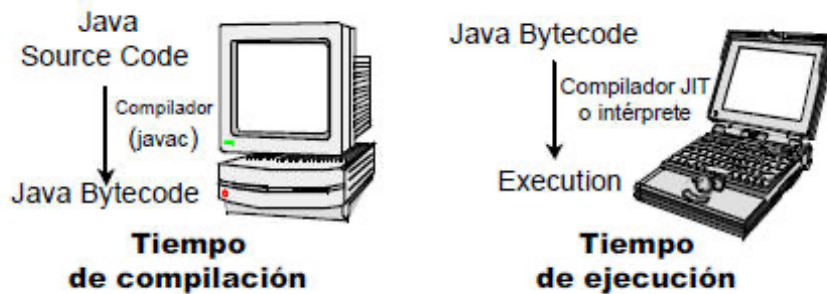
# Tema 1. Fundamentos de Programación

## ➤ La plataforma de programación Java



### *La máquina virtual de Java*

#### *La máquina virtual Java*



- El compilador de Java genera un código intermedio independiente de la plataforma (bytecodes).
- Los bytecodes pueden considerarse como el lenguaje máquina de una máquina virtual, la Máquina Virtual Java (JVM).
- Cuando queremos ejecutar una aplicación Java, al cargar el programa en memoria, podemos
  - a) Interpretar los bytecodes instrucción por instrucción
  - b) Compilar los bytecodes para obtener el código máquina necesario para ejecutar la aplicación en el ordenador (compilador JIT [Just In Time]).

**IMP!!**



Information Module Profile

De esta forma, podemos ejecutar un programa escrito en Java sobre distintos sistemas operativos (Windows, Solares, Linux...) sin tener que recompilarlo, como sucedería con programas escritos en lenguajes como C.

# Tema 1. Fundamentos de Programación

## ➤ La plataforma de programación Java



### *Uso típico de Java*

- Aplicaciones (programas independientes)
- Applets (“pequeñas aplicaciones”) son programas diseñados para ejecutarse como parte de una página web.

### *Herramientas de programación en Java*

- **Java SDK [Software Development Kit]** <https://java.com>
  - ✓ Compilación de aplicaciones Java: `javac`
  - ✓ Ejecución de aplicaciones Java: `java`
  - ✓ Ejecución de applets: `appletviewer`
  - ✓ Generación de documentación: `javadoc`
  - ✓ Creación de archivos de distribución JAR [Java ARchives]: `jar`
  - ✓ Depuración de aplicaciones Java: `jdb`
  - ✓ etc.
- **Entornos integrados de desarrollo: IDEs** → **Eclipse** (<https://eclipse.org>) y **Apache NetBeans** (<https://netbeans.apache.org/>)

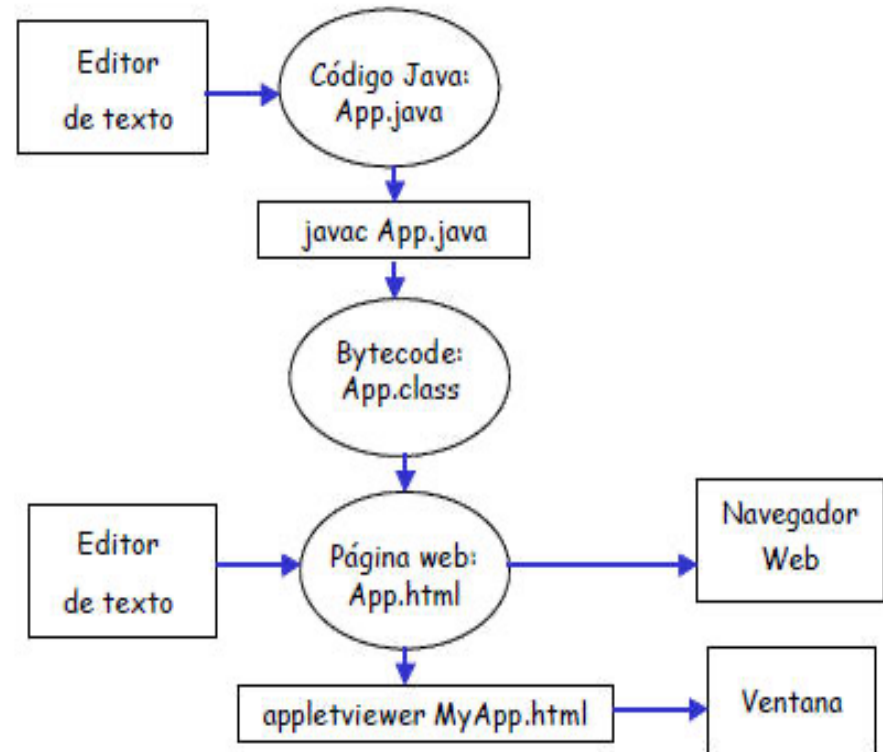
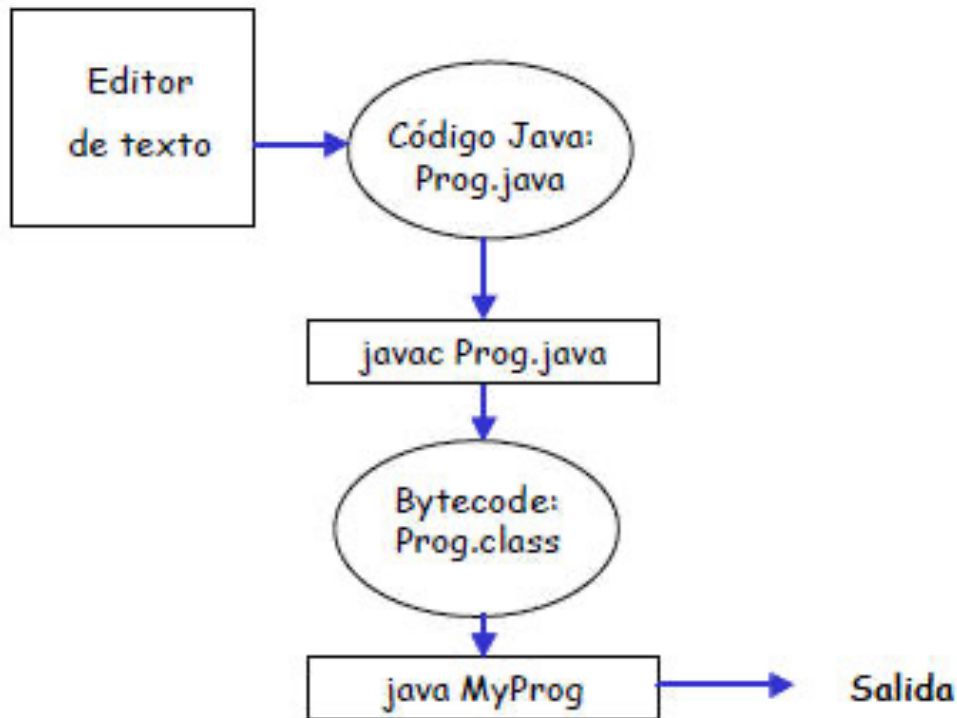


# Tema 1. Fundamentos de Programación

## ➤ Aplicaciones y applets

Creación y ejecución de aplicaciones Java

Creación y ejecución de applets





# Tema 1. Fundamentos de Programación



## Aplicaciones y applets

### *Fases en la creación y ejecución de programas en Java*

#### **Fase I: Editor**

- Se crea un programa con la ayuda de un editor. Se almacena en un fichero con extensión .java

#### **Fase II: Compilador**

- El compilador lee el código Java (fichero .java).
- Si se detectan errores sintácticos, el compilador nos informa de ello.
- Se generan los bytecodes, que se almacenan en ficheros .class

#### **Fase III: Cargador de clases**

- El cargador de clases lee los bytecodes (.class): Los bytecodes pasan de disco a memoria principal.

#### **Fase IV: Verificador de bytecodes**

- El verificador de bytecodes comprueba que los bytecodes son válidos y no violan las restricciones de seguridad de la máquina virtual Java.

#### **Fase V: Intérprete de bytecodes o compilador JIT**

- La máquina virtual Java (JVM) lee los bytecodes y los traduce al lenguaje que el ordenador entiende (código máquina).

# Tema 1. Fundamentos de Programación

## ➤ Aplicaciones y applets

### *Características clave de Java*

- Java es multiplataforma
- Java es seguro
- Java tiene un amplio conjunto de bibliotecas estándar que facilita la codificación o programación
- Java incluye una biblioteca portable para la creación de Interfaces Gráficas de Usuario (AWT y JFC/Swing)
- Java simplifica algunos aspectos al programar
- Permite recolección automática de basura (memoria)
- Admite multiprocesamiento y concurrencia

# Tema 1. Fundamentos de Programación



## Datos y tipos de datos



☐ Dato

☐ Tipo de dato

☐ Codificación de los datos en el ordenador

# Tema 1. Fundamentos de Programación

## ➤ Dato

Representación formal de hechos, conceptos o instrucciones adecuada para su comunicación, interpretación y procesamiento por seres humanos o medios automáticos.

## ➤ Tipo de dato

Especificación de un dominio (rango de valores) y de un conjunto válido de operaciones a los que normalmente los traductores asocian un esquema de representación interna propio.

# Tema 1. Fundamentos de Programación

## *Clasificación de los tipos de datos*

- En función de quién los define:
  - ✓ Tipos de datos estándar
  - ✓ Tipos de datos definidos por el usuario
  
- En función de su representación interna:
  - ✓ Tipos de datos escalares o simples
  - ✓ Tipos de datos estructurados

# Tema 1. Fundamentos de Programación

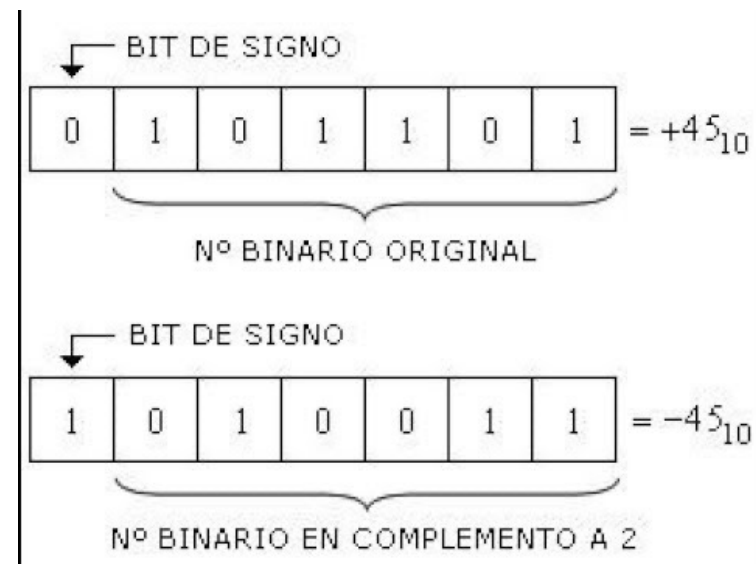
## ➤ Codificación de los datos en el ordenador

En el interior del ordenador, los datos se representan en binario.

Un bit nos permite representar 2 símbolos diferentes: 0 y 1

En general,

N	$2^N$
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	8192
14	16384
15	32768
16	65536



# Tema 1. Fundamentos de Programación

## ➤ Codificación de los datos en el ordenador

### NÚMEROS ENTEROS

*Ejemplo:* Si utilizamos 32 bits para representar números enteros, disponemos de  $2^{32}$  combinaciones diferentes de 0s y 1s:

4 294 967 296 valores.

Como tenemos que representar números negativos y el cero, el ordenador será capaz de representar

del  $-2\,147\,483\,648$  al  $+2\,147\,483\,647$ .

Con 32 bits no podremos representar números más grandes.

$2\,147\,483\,647 + 1 = -2\,147\,483\,648$  !!!

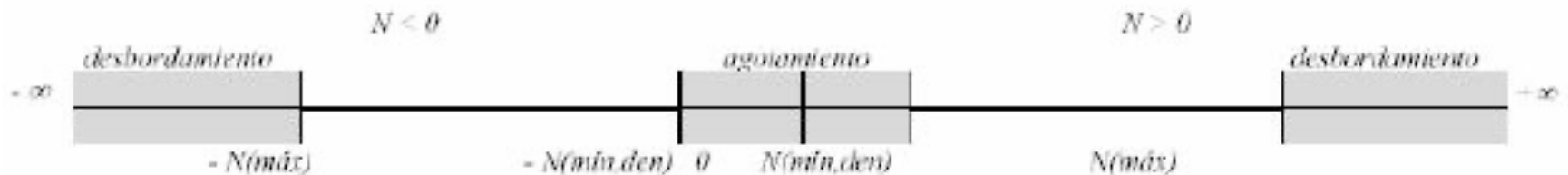
# Tema 1. Fundamentos de Programación

## ➤ Codificación de los datos en el ordenador

### NÚMEROS REALES (en notación científica)

(+ | −) mantisa  $\times 2^{\text{exponente}}$

- El ordenador sólo puede representar un subconjunto de los números reales (números en coma flotante)
- Las operaciones aritméticas con números en coma flotante están sujetas a errores de redondeo.





# Tema 1. Fundamentos de Programación

## Tipos primitivos en Java

Tipo	Representación / Valor	Tamaño (en bits)	Valor mínimo	Valor máximo	Valor por defecto
<b>boolean</b>	<b>true</b> o <b>false</b>	1	N.A.	N.A.	<b>false</b>
<b>char</b>	Carácter Unicode	16	\u0000	\uFFFF	\u0000
<b>byte</b>	Entero con signo	8	-128	128	0
<b>short</b>	Entero con signo	16	-32768	32767	0
<b>int</b>	Entero con signo	32	-2147483648	2147483647	0
<b>long</b>	Entero con signo	64	-9223372036854775808	9223372036854775807	0
<b>float</b>	Coma flotante de precisión simple Norma IEEE 754	32	$\pm 3.40282347\text{E}+38$	$\pm 1.40239846\text{E}-45$	0.0
<b>double</b>	Coma flotante de precisión doble Norma IEEE 754	64	$\pm 1.79769313486231570\text{E}+308$	$\pm 4.94065645841246544\text{E}-324$	0.0

# Tema 1. Fundamentos de Programación

## ➤ Codificación de los datos en el ordenador

### Representación de textos

Se escoge un conjunto de caracteres: alfabéticos, numéricos, especiales (separadores y signos de puntuación), gráficos y de control (por ejemplo, retorno de carro).

Se codifica ese conjunto de caracteres utilizando  $n$  bits.

Por tanto, se pueden representar hasta  **$2^n$  símbolos** distintos.

# Tema 1. Fundamentos de Programación

## Codificación de los datos en el ordenador

### Ejemplos de códigos normalizados

ASCII (American Standard Code for Information Interchange)

- ANSI X3.4-1968, 7 bits (128 símbolos)
- ISO 8859-1 = Latin-1, 8 bits (256 símbolos)

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
00	0	NUL	SOH	STX	ETX	EOT	ENO	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
10	16	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
20	32	SP	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
30	48	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	64	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	80	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	96	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	112	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
80	128																
90	144																
A0	160		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	®	¯	
B0	176	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	192	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	208	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	224	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	240	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

# Tema 1. Fundamentos de Programación

## ➤ Codificación de los datos en el ordenador

### Ejemplos de códigos normalizados

UNICODE, ISO/IEC 10646, 16 bits (65536 símbolos)

<i>Zona</i>	<i>Códigos</i>		<i>Símbolos codificados</i>	<i>Nº de caracteres</i>
<b>A</b>	0000	0000 00FF	Latin-1	256
			otros alfabetos	7.936
		2000	Símbolos generales y caracteres fonéticos chinos, japoneses y coreanos	8.192
<b>I</b>	4000		Ideogramas	24.576
<b>O</b>	A000		Pendiente de asignación	16.384
<b>R</b>	E000 FFFF		Caracteres locales y propios de los usuarios. Compatibilidad con otros códigos	8.192

¡MUCHAS GRACIAS!



UNIVERSIDAD DE ALMERÍA

