

UNIVERSIDAD NACIONAL DEL OESTE

PROGRAMACIÓN CON OBJETOS I



Profesor:

Mag. Ing. Pablo Pandolfo

Introducción al Lenguaje Java

4 Principales razones para Aprender JAVA

1. Java actualmente es **usado por diferentes empresas y gobiernos** como por Ejemplo: Facebook, Amazon y Gobiernos de Países

2. Java **funciona en casi todo tipos de Dispositivos** como: Automóviles, Smartphones, SmartTV, Web, Desktop, Cajeros automáticos, etc.

3. Java cuenta con una **gran comunidad** de más de 20 años, esto permite a Java seguir creciendo.

4. Java es **altamente demandado** en el mercado tecnológico

Introducción al Lenguaje Java

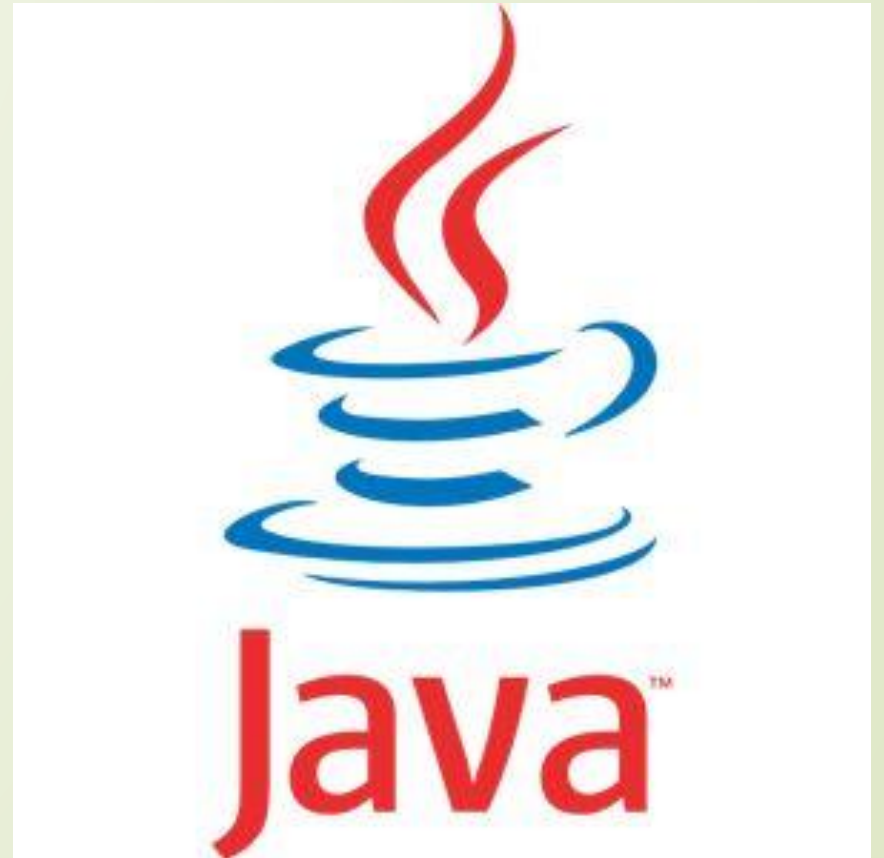
Lenguajes de Programación usados por Empresas Importantes

Google	JavaScript, C, C++, Python, Java, Go
Facebook	JavaScript, Hack, PHP, Python, C++, Java
Amazon	JavaScript, Java, C++, Perl
Wikipedia	JavaScript, PHP, Hack
eBay	JavaScript, Java, Scala
Twitter	JavaScript, C++, Java, Scala, Ruby on Rails
Youtube	JavaScript, C, C++, Python, Java, Go
Netflix	JavaScript, Java, Python
Yahoo	JavaScript, PHP
Linkedin	JavaScript, Java, Scala



Introducción al Lenguaje Java

- Java abarca **dos aspectos**:
 - Una plataforma.
 - Un lenguaje.





Introducción al Lenguaje Java

□ Una plataforma:

- Es un ambiente de software y/o hardware sobre el que se ejecuta un programa.
- Enorme biblioteca.
- Código reutilizable.
- Entorno de ejecución: seguridad, adaptabilidad, Garbage Collector.



Introducción al Lenguaje Java

□ Un lenguaje:

- Posibilita el desarrollo de aplicaciones seguras, robustas sobre múltiples plataformas en redes heterogéneas y distribuidas.
- Sintaxis agradable.
- Semántica comprensible.

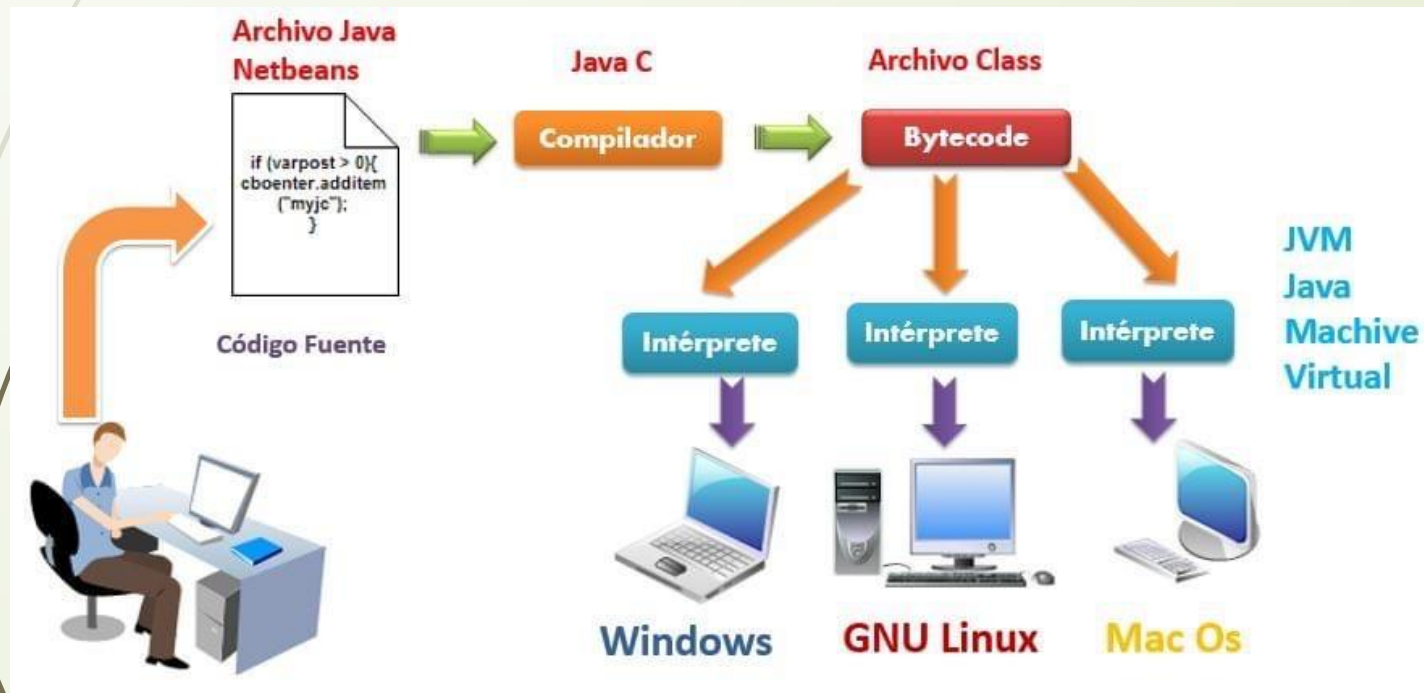


Introducción al Lenguaje Java

- La plataforma Java se compone de:
 - **La Java Virtual Machine (JVM):** Es la base de la Plataforma Java y puede ser incorporada en la mayoría de las plataformas (hardware y sistema operativo). Contiene el intérprete Java.
 - **La Java Application Programming Interface (Java API):** Es una colección de componentes de software que proveen una amplia gama de funcionalidades, como GUIs, I/O, Seguridad, etc. Está dividida en paquetes o librerías de componentes relacionadas.

Introducción al Lenguaje Java

- La Plataforma Java (la API y la JVM) independiza al programa Java del hardware.



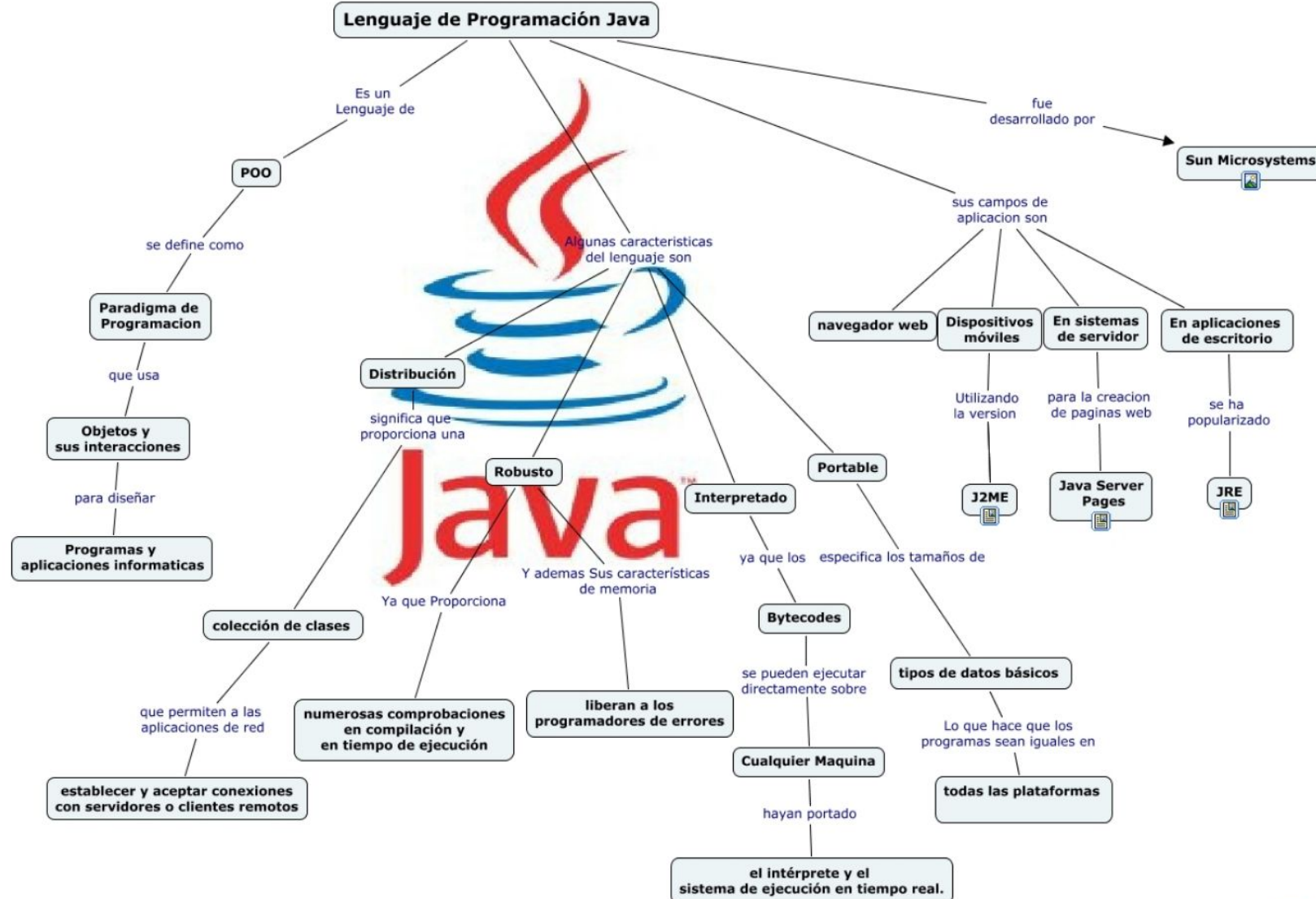


Introducción al Lenguaje Java

- ❑ Principales características del Lenguaje Java:
 - ❑ Simple y familiar.
 - ❑ Orientado a objetos.
 - ❑ Distribuido.
 - ❑ Robusto.
 - ❑ Seguro.
 - ❑ Arquitectura neutral.
 - ❑ Multithread.
 - ❑ Interpretado.
 - ❑ Dinámico.

Introducción al Lenguaje Java

¿Que es el Lenguaje de programación Java?





Introducción al Lenguaje Java



- Simple y Familiar:

- El objetivo de los diseñadores de Java fue crear un lenguaje de programación que fuera fácil de aprender. Java adoptó una sintaxis similar a la de C/C++ teniendo en cuenta su popularidad. Y eliminó aquellas características que son fuente de confusión.

- Características de C/C++ eliminadas en Java:

- Preprocesador, estructuras de datos, uniones, herencia múltiple de clases, GOTO, punteros.



Introducción al Lenguaje Java

- Orientado a Objetos:
 - Java posee todas las características de un lenguaje orientado a objetos: Polimorfismo, Encapsulamiento y Herencia.
 - Java implementa herencia simple de clases. Una clase puede ser subclase de una única clase. Todas las clases heredan de la clase Object. (Todo es un objeto)
 - Java enriquece el modelo de herencia simple de clases, implementando interfaces.



Introducción al Lenguaje Java



□ Distribuido:

- Con Java es posible usar protocolos como HTTP y FTP para leer o copiar archivos remotos de manera tan simple como si estuviesen en el File System local.
- El comportamiento distribuido de Java posibilita la colaboración y la distribución de carga de trabajo del sistema. Ejemplo: Applets.
- RMI (Java Remote Method Invocation) provee una forma simple y directa de manejar objetos distribuidos.



Introducción al Lenguaje Java



- Seguro:

- En Java el modelo de asignación de memoria es transparente al programador, ya que es controlado íntegramente por JVM.
- El intérprete Java chequea los archivos “.class” que vienen de la red, evaluando:
 - Que el código no falsifique punteros, no viole restricciones de acceso, no viole el acceso a los objetos usando casting.



Introducción al Lenguaje Java

- Arquitectura neutral:
 - Java fue diseñado para soportar aplicaciones que se ejecutan en ambientes de redes heterogéneos, independientemente de la plataforma de hardware y del sistema operativo.
 - La arquitectura neutral dada por los ByteCodes es el paso más importante hacia la portabilidad de los programas.



Introducción al Lenguaje Java



□ Multithreaded:

- Un Thread es un flujo de control secuencial dentro de un programa. Java provee múltiples threads en un programa, ejecutándose concurrentemente y llevando a cabo tareas distintas.
- La API Java contiene primitivas de sincronización.
- Los múltiples hilos (threads) de ejecución permiten mejorar la interactividad y la performance del sistema. (Mejor respuesta interactiva y comportamiento de tiempo real)



Introducción al Lenguaje Java

- Interpretado y dinámico:
 - El compilador Java genera ByteCodes para la JVM. El intérprete, incorporado en la JVM es el que permite ejecutar el programa.
 - Los ByteCodes de Java pueden ejecutarse en cualquier plataforma que tenga la JVM implementada.
 - Java es dinámicamente extensible ya que las clases se linkan a medida que se necesitan y pueden ser cargadas dinámicamente a través de la red.

Introducción al Lenguaje Java

¿QUÉ ES UN IDE?

Aplicación que provee un entorno de desarrollo de software con diversas herramientas integradas: editor de código con resaltador de sintaxis, compilador, depurador, etc. Algunos soportan varios lenguajes. Ejemplos: Eclipse, Netbeans, Qt Creator, Visual Studio.

¿Y UN LENGUAJE DE PROGRAMACIÓN?

Conjunto de instrucciones dirigidas a una computadora, con una sintaxis específica, para implementar algoritmos. Ejemplos: Java, C++, Python, Javascript, C, Go, Scala.

¿QUÉ ES UN FRAMEWORK?

Es una base sobre la cual se puede crear software. Provee una plataforma con funcionalidad predefinida, un “punto de partida” para facilitar tareas. Ejemplos: Qt, node, Net, react, Android.

Introducción al Lenguaje Java



Introducción al Lenguaje Java



Versión	Año	Nombre Clave	Características
JDK 1.0	1996		El lenguaje en sí
JDK 1.1	1997		Reestructuración de AWT, Clases internas, JavaBeans, JDBC, RMI
J2SE 1.2	1998	Playground	Reflection, SWING, Java Plug-in, Collections (Java 2) J2SE/EE/ME
J2SE 1.3	2000	Kestrel	RMI basado en CORBA, JNDI
J2SE 1.4	2002	Merlin	Aserciones, Expresiones Regulares, Logging
J2SE 5.0	2004	Tiger	Clases genéricas, for each, argumentos variables, enumeraciones, annotations
Java SE 6	2006	Mustang	Combinación de Java con lenguajes dinámicos como PHP, Python, Ruby y JavaScript.
Java SE 7	2011	Dolphin	Soporte para XML dentro del propio lenguaje. Un nuevo concepto de superpaquete. Soporte para closures. Introducción de anotaciones estándar para detectar fallos en el software.
Java SE 8	2014		Incorpora de forma completa la librería JavaFX. Diferentes mejoras en seguridad, concurrencia, integración de JavaScript Añade funcionalidad para programación funcional mediante expresiones Lambda. Nuevas API para manejo de fechas y tiempo (date - time).
Java SE 9	Planificado para Marzo 2017		
Java SE 10	Planificado para 2018		

Introducción al Lenguaje Java

¿QUÉ ES UN IDE?

Es un espacio para trabajar proyectos completos, con herramientas integradas. Suele tener emulador, controlador de versiones y terminales.



Introducción al Lenguaje Java




- Es un IDE abierto y extensible (plugins)
- Un IDE es un programa compuesto por un conjunto de herramientas útiles para un desarrollador.
- Software libre.
- Antecesor de Eclipse: VisualAge
- En 2001 IBM y Borland crearon la Fundación Eclipse
- Sitio web: www.eclipse.org

Versión	Año	Nombre
3.0	2004	Eclipse 3.0
3.1	2005	Eclipse 3.1
3.2	2006	Callisto
3.3	2007	Europa
3.4	2008	Ganymede
3.5	2009	Galileo
3.6	2010	Helio
3.7	2011	Indigo
4.2	2012	Juno
4.3	2013	Kepler
4.4	2014	Luna
4.5	2015	Mars
4.6	2016	Neon
4.7	2017	Oxygen
4.8	2018	Photon

Introducción al Lenguaje Java

 eclipse	 VERSUS	 IntelliJ IDEA
 Eclipse proporciona actualizaciones frecuentes.		 Proporciona actualizaciones mayores al IDE.
 Versión Enterprise muy completa y gratis.		 Versión Ultimate muy completa pero de paga.
 Ofrece diversidad de plug-ins.		 Ofrece diversidad de extensiones.
 Interfaz sencilla para sistemas de control de versiones.		 Mejor interfaz para sistemas de control de versiones.
 Buen autocompletado.		 Un mejor autocompletado, correcciones y tips de sintaxis.

Introducción al Lenguaje Java



Keyboard Shortcut Cheat Sheet

The most useful Eclipse shortcuts

Navigating the editor

<code>ctrl</code> + <code>page up</code> / <code>page dn</code>	Moves through editor tabs, going right/left
<code>ctrl</code> + <code>E</code>	Lists all open files, <i>then use arrow keys and enter to navigate</i>
<code>ctrl</code> + <code>Q</code>	Jump to the last edited location
<code>ctrl</code> + <code>shift</code> + <code>↓</code> / <code>↑</code>	Jump to previous/next method
<code>ctrl</code> + <code>shift</code> + <code>P</code>	Jump to opening/closing bracket, <i>when a bracket is already selected</i>
<code>alt</code> + <code>shift</code> + <code>N</code>	Create a new resource, <i>such as a Project, Class, Interface, Folder, etc</i>

Editing Text

<code>alt</code> + <code>↓</code> / <code>↑</code>	Move current line/selection up/down
<code>ctrl</code> + <code>alt</code> + <code>↓</code> / <code>↑</code>	Duplicate current line, move up/down
<code>alt</code> + <code>shift</code> + <code>←</code> / <code>→</code>	Expand selection by previous/next element
<code>ctrl</code> + <code>D</code>	Delete the current line
<code>ctrl</code> + <code>shift</code> + <code>enter</code>	Insert a line above the current line
<code>shift</code> + <code>ctrl</code> + <code>Y</code> / <code>X</code>	Change selection to lowercase/uppercase

Other Shortcuts

<code>ctrl</code> + <code>shift</code> + <code>O</code>	Automatically insert needed imports
<code>alt</code> + <code>shift</code> + <code>R</code>	Rename selected element and its reference
<code>ctrl</code> + <code>O</code>	Show the outline of the current class
<code>ctrl</code> + <code>space</code> + <code>R</code>	Opens the content assist window

Introducción al Lenguaje Java

"Hola Mundo" en distintos lenguajes

C++

```
#include <iostream>
int main() {
    cout<<"Hola, mundo"<<endl;
    return 0;
}
```

F#

```
printf "Hola, mundo\n"
```

C#

```
class HolaMundo {
    static void Main() {
        System.Console.WriteLine("Hola, mundo");
    }
}
```

JavaScript

```
console.log("Hola, mundo");
```

Java

```
class HolaMundo {
    static public void main(String args[]) {
        System.out.println("Hola, mundo");
    }
}
```

Kotlin

```
fun main(args : Array<String>) {
    println("Hola, mundo")
}
```

Visual Basic .Net

```
Class HolaMundo
    Public Shared Sub Main()
        WriteLine("Hola, mundo")
    End Sub
End Class
```

Python 3

```
print("Hola, mundo")
```

Ruby

```
puts "Hola, mundo"
```

Pascal

```
program HolaMundo;
begin
    writeln('Hola, mundo');
end.
```

C

```
#include <stdio.h>
int main() {
    printf("Hola, mundo");
    return 0;
}
```

**EL 99% DE PROGRAMADORES
NO LO NOTA**

```
main( ) {
    printf("hello, world");
}
```

**HASTA QUE EL CÓDIGO
DA ERROR**



Introducción al Lenguaje Java



```
public class HolaMundo {  
    public static void main(String[] args) {  
        System.out.println("Hola Mundo");  
    }  
}
```

```
// #NuncaTeDetengas
```



```
>>> javac HolaMundo.java
```

```
>>> java HolaMundo
```

```
>>> Hola Mundo
```

Introducción al Lenguaje Java

**A ESTO LE LLAMAS
PROGRAMAR**





Introducción al Lenguaje Java

Lo que te enseñaron en la universidad

```
public void holamundo(){  
    System.out.println("Hola mundo");  
}
```

Lo que el mundo laboral te pide

```
public String getName() {  
    return Optional.ofNullable(name).filter(f -> f.contains(".")).map(f->{  
        int index = name.indexOf(".");  
        int finallen = Math.abs(name.length()-index);  
        return f.substring(0, finallen);  
    }).orElse("");  
}
```




Introducción al Lenguaje Java

Comparativa: Java vs. C++ (I)

Java	C++
<ul style="list-style-type: none">• Interpretado• Portable• Todo memoria dinámica• Garbage collection• No existen punteros• Todos los métodos virtuales• No hay herencia múltiple• Información sobre clases en tiempo de ejecución.• Tratamiento genérico de tipos (wrappers).	<ul style="list-style-type: none">• Compilado• Aspectos no portables• Memoria automática y dinámica• Gestión del programador• Existen punteros• No todos los métodos virtuales• Hay herencia múltiple• Información muy escasa (dynamic_cast).• Punteros a void.

Introducción al Lenguaje Java



A comparison graphic titled 'Java VS python' on a dark blue background. The Java logo (a red flame over a blue cup) is on the left, and the Python logo (two interlocking snakes, one blue and one yellow) is on the right. Below the logos are two rounded rectangular boxes, one for Java and one for Python, each containing a list of characteristics. The Java box lists: 'Necesitas ";"', 'Es complejo, incluye una curva de aprendizaje media/alta', 'Uno de sus principales frameworks es Spring', 'Mayor robustez para operaciones con base de datos', 'Ideal para sistemas empresariales', and 'Es estrictamente tipado hasta la versión 8'. The Python box lists: 'No Necesitas ";"', 'Fácil de aprender y usar', 'Sus principales frameworks son Django y Flask', 'Menor robustez para operaciones con base de datos', 'Ideal para ciencia de datos, machine learning, inteligencia artificial', and 'Tipado dinámico'.

Java VS python

-  Necesitas ";"
-  Es complejo, incluye una curva de aprendizaje media/alta
-  Uno de sus principales frameworks es Spring
-  Mayor robustez para operaciones con base de datos
-  Ideal para sistemas empresariales
-  Es estrictamente tipado hasta la versión 8

-  No Necesitas ";"
-  Fácil de aprender y usar
-  Sus principales frameworks son Django y Flask
-  Menor robustez para operaciones con base de datos
-  Ideal para ciencia de datos, machine learning, inteligencia artificial
-  Tipado dinámico

Introducción al Lenguaje Java

 Java VS  Kotlin	
 <p>Ideal para aplicaciones empresariales.</p>	 <p>Muy usado para desarrollo Android.</p>
 <p>En ocasiones se debe repetir código o palabras reservadas.</p>	 <p>Menor redundancia de código</p>
 <p>Soporte de hilos.</p>	 <p>Soporte de Coroutines</p>
 <p>Usa Clases Simples.</p>	 <p>Usa Data Clases.</p>
 <p>No tiene soporte de programación funcional hasta Java 8.</p>	 <p>Soporta programación funcional, objetos y más.</p>



Ejercicios



- ❑ **1) FizzBuzz:** Imprimir por pantalla los números del 1 al 100 pero considerando lo siguiente:
 - ❑ Si el número es divisible por 3 se debe imprimir “Fizz”.
 - ❑ Si el número es divisible por 5 se debe imprimir “Buzz”.
 - ❑ Si el número es divisible por 3 y por 5 se debe imprimir “FizzBuzz”.

- ❑ **2) Sumados:** Sumar los 1000 primeros números naturales ($1+2+3+4+\dots+1000$), imprimiendo por cada suma el resultado parcial obtenido.