

Java desde cero

Alejandro Traver Fos

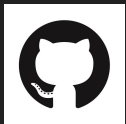


KEEPCODING

Tech School

1 | Intro: ¿Quién soy?

- Alejandro Traver Fos (Ale)
- Ingeniero Informático
- Android Developer
- Mis Apps destacadas: Sabadell, CaixaBank, La Liga Sports TV...



Y ME GUSTAN LOS MEMES



1 | Intro: ¿Y vosotros?

- Nombre?
- Hobbie?
- Background?
- Androide o Manzanito?
- Contadnos lo que queráis y **nominad a otro!**



¿Qué aprenderemos en este curso?

- T0: Introducción
- T1: IDE y entorno
- T2: Java básico: primeros pasos
- T3: Objetos
- T4: Estructura de datos
- T5: Programación funcional
- T6: Ficheros
- T7: Patrones para el capitán
- T8: Mayordomos: gestión de hilos



2 semanas...
a tope!

T0: Introducción



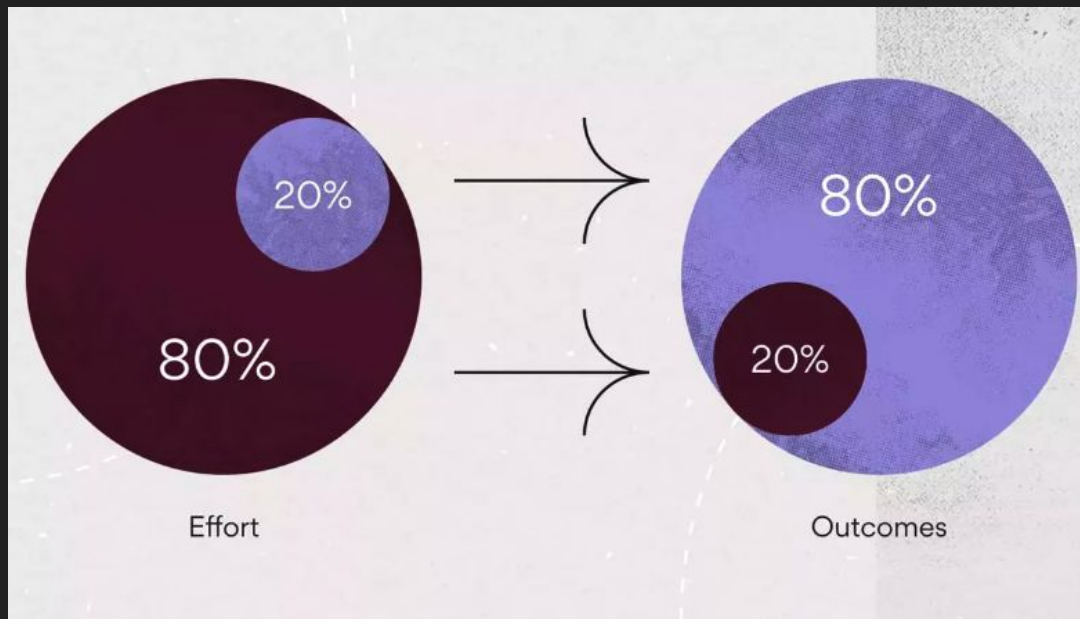
- ¿Por qué JAVA?
- Java en el mercado
- Conceptos básicos
- Entender las tecnologías del entorno
Java

Fundamental: Aprender a aprender un lenguaje nuevo.



- ¿Qué es Java?
- ¿Qué se puede hacer con Java?
- ¿Qué es la programación orientada a objetos?
- ¿Qué es una clase y un objeto?
- ¿Qué es un hilo de ejecución?

Aproximadamente el 80% de los efectos proviene del 20% de las causas



- Software
- Idiomas
- Riqueza
- Testing
- [...]
- ¿Qué más?

- Es un lenguaje de propósito general
- Creado por Sun Microsystems
- Popularizado por **hacer la web más interactiva** a mediados de los 90's
- Desarrollo de webs dinámicas: Applets y Servlets

- WRITE ONCE RUN ANYWHERE
- Se popularizó rápidamente
- Rápidamente se situó como el lenguaje más utilizado del mundo
- Evolución de los lenguajes de programación



- Actualmente, interactuamos con tecnología Java a diario
- ¿Quién utiliza tecnologías Java? Android, Netflix, LinkedIn, twitter
- Tendencia de uso a la baja, aunque sigue teniendo una importante cuota de mercado

- **Popularidad Relativa:** Java ha experimentado una disminución en su popularidad, en comparación con otros, como Python, JavaScript, y Kotlin.
- **Relevancia Continua:**
 - Base de Código: Amplia presencia en aplicaciones empresariales y financieras.
 - Ecosistema y Herramientas: Robusto ecosistema de bibliotecas y frameworks (Spring, IntelliJ IDEA, Eclipse).
 - Innovaciones Continuas: Nuevas versiones del JDK y proyectos.

- ¿Cómo conseguimos la portabilidad? La JVM (Java Virtual Machine).
- ¿Qué aporta java?
 - Seguridad y robustez.
 - Gestión de los hilos.
 - Gran comunidad y recursos disponibles.

- **Scala:**

- Lenguaje de programación que combina paradigmas orientados a objetos y funcionales.
- Compatible con el ecosistema Java y ejecutado sobre la JVM.
- Sintaxis concisa y expresiva.
- Soporte para programación funcional.
- Alta interoperabilidad con Java.

- **Kotlin:**

- Lenguaje de programación moderno desarrollado por JetBrains.
- Totalmente interoperable con Java y ejecutado sobre la JVM.
- Sintaxis más concisa y segura que Java.
- Características avanzadas como null safety y coroutines.
- Adoptado oficialmente por Google para el desarrollo de Android.

- **Ejecución:** El código fuente se ejecuta directamente por un intérprete
- **Ejecución Directa:** El código se ejecuta línea por línea.
- **Portabilidad:** Se pueden ejecutar en cualquier sistema (dep intérprete)



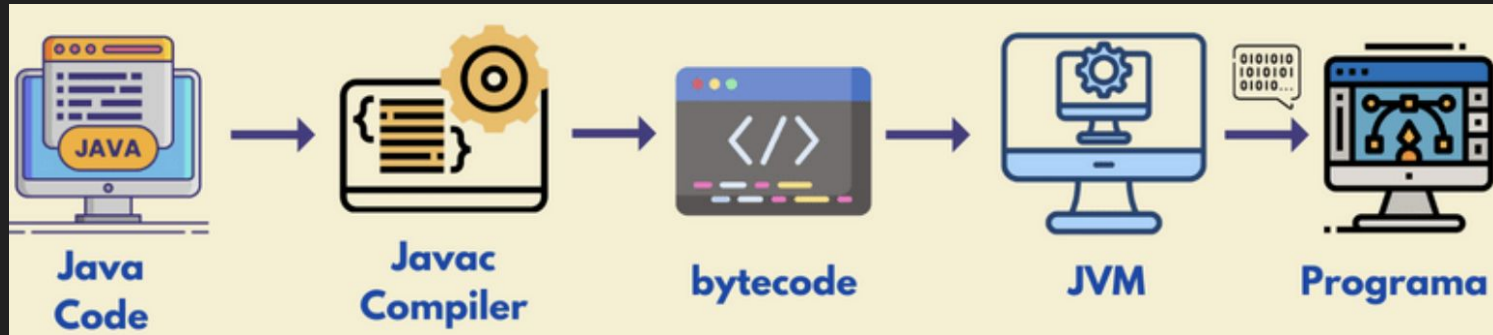
Ventajas	Desventajas
Flexibilidad y velocidad de desarrollo: scripting y prototipado	Errores en ejecución
Tipado dinámico: Los errores se pueden identificar y corregir fácilmente.	Rendimiento: Sobrecarga por el coste de interpretar línea a línea
Independencia del Sistema: Facilidad para el desarrollo multiplataforma.	

- Código fuente se traduce a código máquina mediante **antes** de su ejecución.
- **Compilación**: El código fuente se convierte a un ejecutable independiente.
- **Ejecución**: El código ya está en lenguaje máquina
- **Verificación**: Los errores de sintaxis y semántica se detectan durante la compilación.



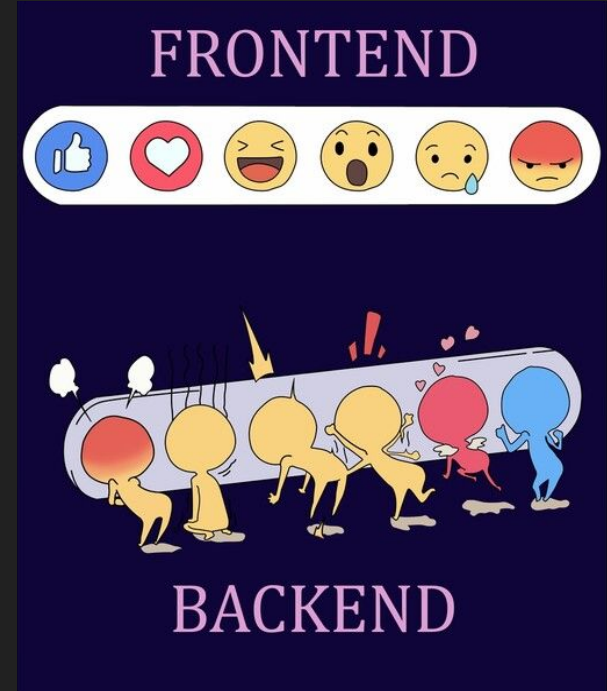
Ventajas	Desventajas
Rendimiento : Ejecutan más rápido que los lenguajes interpretados.	Tiempos de desarrollo : Requiere tiempo de compilación
Eficiencia : El código optimizado puede aprovechar mejor los recursos del sistema.	Portabilidad : Requiere tiempo de compilación
Distribución : Los ejecutables pueden distribuirse sin el código fuente ni del compilador.	Gestión de las versiones : Asociado a la distribución hay coste de gestión extra

- 1. Compilación a Bytecode:** El código fuente Java (.java) se escribe y luego se compila utilizando **javac**. El compilador javac convierte el código fuente en bytecode (.class).
- 2. Bytecode Intermedio:** El bytecode es un formato binario independiente de la plataforma. Es entendido y ejecutado por la JVM (Java Virtual Machine).
- 3. Ejecución por la JVM:** La JVM interpreta el bytecode o lo compila Just-In-Time (JIT) a código máquina específico del sistema operativo. Permite que el mismo bytecode se ejecute en cualquier plataforma con una JVM.



Se encarga de desarrollar las interfaces de usuario y experiencia visual como animaciones

- **Aplicaciones Web:** navegador
- **Aplicaciones de escritorio:** Mac y Windows
- **Aplicaciones móviles:** Android y iOS
- **JavaSwing:** Biblioteca para crear interfaces gráficas en aplicaciones de escritorio



Se encarga de desarrollar servicios en la nube y gestionar el almacenamiento y mantenimiento de los datos

- Desarrolla servidores y bases de datos
- Artificial Intelligence
- Big Data
- Frameworks:
 - Spring Boot
 - Quarkus

The power of backend

* Programmers know the real power of back-end





KEEPCODING

Tech School

Madrid | Barcelona | Bogotá

Datos de contacto

Alejandro Traver Fos
alextrfos@gmail.com