



HERRAMIENTAS DE DESARROLLO PARA JAVA POO

Herramientas de desarrollo – Objetos





Java





¿Qué es Java?

Además de ser un lenguaje de programación de propósito general, orientado a objetos. Es también una tecnología pensada para el desarrollo de aplicaciones de gran envergadura, altamente escalables de gran integración con otras tecnologías y sumamente robustas. Desarrollado originalmente por Sun Microsystems, actualmente pertenece a Oracle.



Características principales

- **Orientado a objetos:** Facilita la modularidad y el mantenimiento del código.
- **Multiplataforma** (*"Write Once, Run Anywhere"*): Gracias a la **JVM (Java Virtual Machine)**, el bytecode compilado puede ejecutarse en diferentes sistemas operativos sin cambios en el código.
- **Tipado estático:** El tipo de dato de cada variable se define en tiempo de compilación.
- **Recolección de basura (Garbage Collector):** El programador no gestiona manualmente la memoria, lo que reduce errores.
- **Robusto y seguro:** Mecanismos de control de excepciones y un fuerte sistema de tipos.



Características principales

- **Entornos empresariales:** Java es uno de los lenguajes más populares para backend en grandes corporaciones.
- **Desarrollo de Android:** Aunque Kotlin ha ganado terreno, Java sigue siendo un lenguaje muy utilizado para el desarrollo en Android.
- **Aplicaciones de escritorio:** Con JavaFX o Swing, aunque hoy en día es menos común que su uso en servidores.
- **Big Data:** Tecnologías como Apache Hadoop o Apache Spark están escritas en Java o en la JVM.



Ventajas de utilizar Java para la POO

- **Multiplataforma (Write Once, Run Anywhere):** Java se ejecuta sobre la Máquina Virtual de Java (JVM), lo que permite que el mismo código pueda correr en distintos sistemas operativos (Windows, Linux, macOS) sin cambios.
- **Modelo de POO Maduro y Consistente:** Java implementa un modelo de objetos clásico (clases, herencia, interfaces), lo que facilita el aprendizaje y la aplicación de principios de diseño orientados a objetos.
- **Amplio Ecosistema de Librerías y Frameworks:** Existe una gran variedad de librerías (como las de Apache, Google, etc.) y frameworks (Spring, Java EE, Jakarta EE, Hibernate) que ofrecen soluciones robustas para casi cualquier necesidad de desarrollo.



Ventajas de utilizar Java para la POO

- **Comunidad y Soporte Extenso:** Al ser uno de los lenguajes más populares, cuenta con una comunidad muy activa, foros y recursos de documentación. Esto significa más tutoriales, ejemplos y soporte cuando te enfrentas a problemas.
- **Estabilidad y Longevidad:** Java lleva más de dos décadas en el mercado. Empresas y organizaciones grandes confían en él para proyectos de larga duración, lo que asegura estabilidad y continuidad en su desarrollo.
- **Herramientas de Desarrollo Avanzadas:** IDEs como IntelliJ IDEA, Eclipse o NetBeans ofrecen autocompletado, depuración, refactorización y otras características que aceleran la productividad.



Ventajas de utilizar Java para la POO

- **Gestión Automática de Memoria (Garbage Collector):** Java simplifica la gestión de memoria, evitando problemas comunes de lenguajes como C++ en cuanto a fugas de memoria o uso indebido de punteros.
- **Seguridad Incorporada:** La JVM proporciona un entorno de ejecución seguro y aislado, con verificaciones de bytecode y un sistema de permisos configurable, útil para entornos corporativos.
- **Escalabilidad y Rendimiento Sólido:** Con la optimización Just-In-Time (JIT) de la JVM, las aplicaciones Java pueden alcanzar un rendimiento cercano al de lenguajes compilados nativos, especialmente en entornos de larga ejecución (por ejemplo, servidores).
- **Portafolio de Aplicaciones Amplio:** Java se utiliza en una gran variedad de dominios: desde aplicaciones de escritorio y servidores empresariales hasta aplicaciones móviles (Android) y grandes sistemas distribuidos (Big Data con Hadoop, Spark, etc.).

Herramientas de desarrollo – Objetos



Frameworks y librerías



¿Qué es un framework?

Un framework es un esqueleto o estructura sobre el cual se desarrolla una aplicación. Aporta un conjunto de componentes y reglas de arquitectura, y establece el flujo de trabajo principal, de modo que el desarrollador encaja su código dentro de dicho marco.

El framework a menudo llama al código del usuario (inversión de control), en lugar de que sea el programador quien invoque directamente las partes del framework.



Características de un framework

- **Inversión de control (IoC):** El framework gestiona el flujo de la aplicación y “llama” al código del desarrollador en determinados puntos (por ejemplo, eventos, ciclos de vida, etc.).
- **Estructura y convenciones:** Provee reglas y guías claras sobre dónde y cómo escribir el código, lo que promueve la organización y la escalabilidad.
- **Amplio alcance:** Generalmente aborda múltiples aspectos de la aplicación (configuración, seguridad, persistencia, etc.).



¿Qué es una librería?

Una librería (library) o “biblioteca” es un conjunto de funciones o clases reutilizables que resuelven problemas comunes o brindan funcionalidades específicas (por ejemplo, manejo de fechas, operaciones matemáticas, acceso a bases de datos, etc.).

El programador es quien decide cuándo y cómo utilizar las funciones de la librería dentro de su propio código.



Características de una librería

- **Reusabilidad de código:** Ahorra tiempo y esfuerzo al no tener que reescribir lógica ya existente.
- **Control en manos del desarrollador:** El programador llama a la librería cuando lo considera oportuno.
- **Enfoque específico:** Suelen resolver un conjunto de problemas concretos (por ejemplo, criptografía, generación de PDFs, etc.).



Diferencias entre librerías y frameworks

Flujo de control:

- Librería: El programador llama a las funciones de la librería cuando lo necesita.
- Framework: El framework establece la estructura y, en muchos casos, llama al código del programador (Inversión de control).

Alcance:

- Librería: Suele ser más específica, enfocada en resolver un problema o conjunto de problemas concretos.
- Framework: Ofrece un entorno de desarrollo completo, abarcando arquitectura, ciclos de vida, gestión de dependencias, etc.

Forma de Integración:

- Librería: Se integra de forma puntual en las partes donde se requiere su funcionalidad.
- Framework: Impone una manera de organizar el proyecto y el código.

Herramientas de desarrollo – Objetos



Spring Framework

Spring es un framework de código abierto para el desarrollo de aplicaciones empresariales en **Java**. Su objetivo es simplificar la creación de aplicaciones robustas y eficientes, facilitando la inyección de dependencias, la programación orientada a aspectos y la integración con diversas tecnologías.

Características:

- Inyección de Dependencias (DI): Reduce el acoplamiento entre componentes.
- Inversión de Control (IoC): Permite delegar la creación de objetos al contenedor de Spring.
- Programación Orientada a Aspectos (AOP): Separa la lógica de negocio de aspectos transversales como seguridad, logging y transacciones.
- Modularidad: Incluye diversos módulos como Spring MVC, Spring Data, Spring Security, etc.



Herramientas de desarrollo – Objetos



Arquitectura de Spring Framework

Spring se compone de varios módulos principales:

1. **Core Container:** Implementa IoC y DI.
2. **Spring AOP:** Soporta la programación orientada a aspectos.
3. **Spring MVC:** Facilita la creación de aplicaciones web.
4. **Spring Data:** Simplifica el acceso a bases de datos.
5. **Spring Security:** Proporciona autenticación y autorización.
6. **Spring Boot:** Automatiza la configuración de Spring para acelerar el desarrollo.



Spring Boot

Spring Boot es una extensión de Spring que proporciona configuraciones predeterminadas y herramientas para facilitar la creación de aplicaciones con Spring.

Ventajas de Spring Boot:

- Configuración automática (Auto-Configuration).
- Embutido con servidores web como Tomcat y Jetty.
- Integración con Maven/Gradle: Maneja dependencias y empaquetado de forma sencilla.
- Fácil integración con bases de datos y servicios REST.
- Menos configuración XML y más anotaciones.
- Microservicios listos para producción.





Lombok

Lombok es una librería que reduce la cantidad de código boilerplate en Java, eliminando la necesidad de escribir manualmente getters, setters, constructores y métodos como equals() y hashCode().

Ventajas de Lombok:

- Reduce código repetitivo.
- Mejora la legibilidad del código.
- Integración con IDEs populares como IntelliJ IDEA y VS Code.



Thymeleaf

Thymeleaf es un motor de plantillas para Java que permite renderizar HTML con datos dinámicos.

Ventajas de Thymeleaf:

- Integración con Spring Boot.
- Permite trabajar con HTML estático y dinámico.
- Sintaxis sencilla basada en atributos HTML.
- Permite inyectar variables de modelo, iterar sobre listas, mostrar datos, etc. en las vistas.



Visual Studio Code



Visual Studio
Code



Visual Studio Code (VSC)

Es un editor de código fuente gratuito, multiplataforma y altamente personalizable desarrollado por **Microsoft**. Es ampliamente utilizado en el desarrollo de aplicaciones web y en muchos otros lenguajes de programación.

Tiene una interfaz intuitiva, su diseño es limpio y fácil de usar. Dispone de una gran variedad de extensiones que facilitan la escritura y depuración del código.

Integración con Git: Permite manejar el control de versiones directamente desde el editor. **Soporte Multilenguaje:** No solo es útil para JavaScript, sino para muchos otros lenguajes (*Python, Java, C++, etc.*).

Se descarga desde <https://code.visualstudio.com/>



Visual Studio Code (VSC)

Un **IDE** (*Integrated Development Environment*) es un entorno de desarrollo que reúne herramientas como edición de código, depuración, compilación y ejecución en una sola aplicación.

Visual Studio Code se clasifica principalmente como un editor de código.

Pero gracias a sus extensiones, se puede transformar en un entorno muy similar a un **IDE**.



Visual Studio Code para Java

¿Por qué usar VSC con Java?

- **Ligereza y velocidad:** Es un editor relativamente liviano comparado con IDEs tradicionales (Eclipse, IntelliJ).
- **Multiplataforma:** Disponible en Windows, Linux y macOS.
- **Extensiones:** La comunidad ha desarrollado numerosos complementos que facilitan el desarrollo en Java.



Extension Pack for Java (by Microsoft)

- **Language Support for Java (by Red Hat):** Proporciona autocompletado, resaltado de sintaxis, validación de código y sugerencias para Java.
- **Debugger for Java:** Permite depurar aplicaciones Java directamente en VS Code (poner breakpoints, inspeccionar variables, etc.).
- **Java Test Runner:** Facilita la ejecución y depuración de pruebas unitarias con JUnit o TestNG.
- **Maven for Java:** Maneja proyectos que usan Maven para la gestión de dependencias y el ciclo de vida de compilación.
- **Project Manager for Java:** Ofrece una vista para navegar y administrar proyectos Java (creación de proyectos, importación, etc.).
- **Visual Studio IntelliCode:** Proporciona sugerencias inteligentes de autocompletado basadas en IA.