

INSTITUTO TECNOLÓGICO SUPERIOR DEL SUR DE GUANAJUATO



Framework Web Java.

Materia:

Programación Web II.

Elaborado por:

Chelsea Andrea Garcia Telles.

America Citlalli López Lemus.

Lizet Guadalupe López Medina.

Jesús Rosiles González.

Paola Montserrat Ruiz Carmen.

Docente:

Gustavo Iván Vega Olvera.

IDE idóneo para crear un proyecto web con Java

El desarrollo de aplicaciones web con Java requiere de herramientas que faciliten la programación, organización y despliegue de los proyectos. Para ello, los Entornos de Desarrollo Integrado (IDE) se han convertido en piezas fundamentales, ya que reúnen en un solo lugar un editor de código, depurador, gestor de dependencias e integración con servidores web.

En el ecosistema de Java destacan principalmente tres IDE: IntelliJ IDEA, Eclipse y NetBeans, cada uno con sus características, ventajas y enfoques. Conocerlos permite elegir la herramienta más adecuada según el contexto: ya sea educativo, profesional o empresarial.

1. IntelliJ IDEA (JetBrains)

IntelliJ IDEA es uno de los IDEs líderes para el desarrollo de Java, especialmente en proyectos web. Su popularidad se debe a su autocompletado inteligente, herramientas de refactorización y soporte integrado para frameworks como Spring, Hibernate, Maven y Gradle, lo que lo convierte en una de las opciones preferidas en entornos profesionales y empresariales.

- Soporte Extenso: Incluye compatibilidad nativa con frameworks y tecnologías clave como Spring, Jakarta EE (Java EE), Maven, Gradle y JavaScript/TypeScript, sin necesidad de instalar plugins adicionales.
- Herramientas de Productividad: Ofrece asistentes de código inteligentes, refactorización automática, depuración avanzada y análisis estático para escribir código más limpio y confiable.
- Experiencia de Usuario: Su interfaz es intuitiva y altamente personalizable, lo que facilita adaptar el entorno de trabajo a las necesidades del desarrollador.

La versión Ultimate es de pago y está orientada a desarrollo web y empresarial, mientras que la edición Community es gratuita pero con funciones limitadas.

2. Eclipse IDE

Eclipse es un IDE de código abierto y gratuito, ampliamente utilizado en entornos empresariales y académicos. Nació como un proyecto apoyado por IBM y hoy en día cuenta con una gran comunidad que lo mantiene y expande mediante plugins. Se distingue por su flexibilidad y capacidad de adaptarse a distintos tipos de proyectos Java, incluyendo aplicaciones web, de escritorio y móviles.

Ventajas:

- Gratuito y de código abierto.
- Amplio soporte para Java EE/Jakarta EE y posibilidad de añadir plugins para frameworks web.
- Integración con servidores como Tomcat y JBoss/WildFly.
- Comunidad muy activa y una gran variedad de extensiones disponibles.

Desventaja

- Su interfaz puede resultar compleja y menos moderna en comparación con otros IDEs como IntelliJ.

3. NetBeans (Apache NetBeans)

NetBeans es un IDE de código abierto con una larga trayectoria, actualmente gestionado por la Apache Foundation. Fue impulsado inicialmente por Sun Microsystems y después por Oracle. Se caracteriza por ser intuitivo, fácil de usar y

estable, lo que lo convierte en una excelente opción para estudiantes y desarrolladores que comienzan con Java.

Ventajas:

- Gratuito y respaldado oficialmente por la comunidad Apache.
- Soporte nativo para Java EE, HTML, CSS y JavaScript sin necesidad de plugins adicionales.
- Fácil de configurar con servidores como Tomcat, GlassFish y herramientas como Maven.
- Ideal para aprendizaje y proyectos educativos.

Desventaja

- Dispone de menos plugins y soporte avanzado en comparación con IntelliJ o Eclipse.

CARACTERÍSTICAS:

Las cualidades inherentes de Java lo han posicionado como un pilar en el desarrollo de software empresarial, web, móvil y de sistemas embebidos.

- **Orientado a Objetos (OOP):** Java se adhiere estrictamente al paradigma de la programación orientada a objetos. Esto significa que todo en Java se organiza alrededor de objetos, que son instancias de clases. Este enfoque promueve la modularidad, la reutilización de código y un diseño de software más intuitivo y escalable. Conceptos como la herencia, el polimorfismo, el encapsulamiento y la abstracción son pilares fundamentales de su estructura.
- **Independencia de la Plataforma:** Como se mencionó, esta es su característica más definitoria. La compilación del código Java a bytecode y su posterior ejecución en la JVM elimina la dependencia del sistema operativo. Esto permite a los desarrolladores crear aplicaciones que se distribuyen y ejecutan en una amplia gama de dispositivos y sistemas sin necesidad de re-compilar el código para cada plataforma.
- **Robusto y Confiable:** Java fue diseñado con la fiabilidad en mente. Su robustez se manifiesta en varios aspectos. Primero, el sistema de manejo de errores en tiempo de ejecución (excepciones) obliga al desarrollador a anticipar y gestionar posibles fallos del programa. Segundo, la gestión automática de la memoria a través de la recolección de basura (garbage collection) libera al programador de la tarea de liberar manualmente la memoria, reduciendo drásticamente los errores de fugas de memoria y punteros inválidos, comunes en lenguajes como C++.
- **Multihilo:** Java incluye soporte nativo para multihilo, lo que permite la ejecución simultánea de múltiples tareas dentro de un mismo programa. Esto es crucial para el desarrollo de aplicaciones de alto rendimiento, como servidores de aplicaciones, videojuegos y aplicaciones de interfaz gráfica de usuario (GUI) que deben responder a eventos mientras realizan operaciones en segundo plano.
- **Seguridad:** La JVM proporciona un entorno de ejecución seguro al restringir las operaciones que un programa puede realizar. El bytecode verifier examina el código compilado antes de la ejecución para asegurarse de que no viola ninguna regla de seguridad. Además, Java no utiliza punteros explícitos, lo que elimina una fuente común de vulnerabilidades de seguridad y acceso no autorizado a la memoria.

LICENCIAMIENTO:

El modelo de licenciamiento de Java ha evolucionado de un modelo de uso libre y abierto a una dualidad que requiere una cuidadosa consideración por parte de los desarrolladores y las empresas.

- **OpenJDK: La Implementación de Código Abierto:** OpenJDK (Open Java Development Kit) es la implementación de referencia y la base de código abierto del lenguaje Java. Se distribuye bajo la licencia GNU General Public License, versión 2 (GPLv2), con una excepción de vinculación conocida como Classpath Exception. Esta excepción es crucial, ya que permite a los desarrolladores enlazar su propio código (que puede ser propietario y comercial) con las bibliotecas de OpenJDK sin la obligación de liberar su código bajo la licencia GPL. Esto ha convertido a OpenJDK en la elección preferida y estándar de la industria, ya que es completamente gratuito para uso comercial y recibe contribuciones de una vasta comunidad y empresas como Red Hat, Adoptium, Microsoft y Amazon.
- **Oracle JDK: La Oferta Comercial:** Oracle JDK es la implementación comercial de Oracle. Si bien las versiones anteriores (hasta Java 8) se distribuían bajo una licencia que permitía un uso más liberal, el modelo de negocio cambió drásticamente a partir de la versión 11. Desde entonces, Oracle JDK se distribuye bajo un modelo de suscripción de pago para uso comercial. Esto significa que las empresas que utilizan Oracle JDK en entornos de producción deben adquirir una licencia de pago de Oracle. No obstante, Oracle ha mantenido la disponibilidad de su JDK bajo la licencia "No-Fee Terms and Conditions", que permite el uso gratuito para fines personales, de desarrollo y de prueba. La principal ventaja de Oracle JDK es el acceso a soporte técnico oficial, parches de seguridad garantizados y actualizaciones de rendimiento proporcionadas directamente por Oracle.

FRAMEWORK:

Spring es el **framework** de código abierto más popular al utilizar Java. Surgió gracias a Rod Johnson en 2002; como una alternativa más ligera y flexible a Java EE.

Se centra en un contenedor de inversión de control (IoC). Su trabajo es instanciar, inicializar y conectar objetos de la aplicación, los cuales se manejan como **beans** (clases, servicios, repositorios).

Beneficios:

- **Orientado a aspectos (AOP).** Separa la lógica de la aplicación del manejo de errores y seguridad.
- **Desarrollo rápido y eficiente.** Elimina configuraciones extensas, atención plena a la lógica de la aplicación.
- **Escalabilidad.**
- **Mantenimiento a largo plazo.**

Spring dio paso en 2003 a **Spring Boot** que reduce las clases de configuración, con menor cantidad de XML. Además, tiene dependencias de módulo listas para usarse.

Otro punto central es la inyección de dependencias ubicadas fácilmente por **Spring Starters**. Esto facilita el marco modular y que para que el desarrollador sea posible experimentar en un entorno inmersivo.

Los desarrolladores pueden crear:

- **APIs REST** para aplicaciones móviles y web.
- **Arquitecturas SOA**
- **Microservicios en la nube**

Spring Boot es usado en aplicaciones web con el patrón Modelo-Vista-Controlador, bases de datos, autenticación de usuarios, almacenamiento en la nube.

Ejemplos:

- **Spring Core.** Base del framework y manejo de inyección.
- **Spring MVC.** Para aplicaciones web tradicionales.
- **Spring Data.** Simplifica el acceso a bases de datos.
- **Spring Security.** Autenticación y autorización.

Referencias:

- The Leading IDE for Professional Java and Kotlin Development. (2021, 1 junio). JetBrains. <https://www.jetbrains.com/es-es/idea/features/>
- IntelliJ IDEA overview | IntelliJ IDEA. (s. f.). IntelliJ IDEA Help. <https://www.jetbrains.com/help/idea/discover-intellij-idea.html>
- EducaciónIT. (2025, 18 marzo). Eclipse IDE: Principales características - EducaciónIT. EducaciónIT. <https://blog.educacionit.com/eclipse-ide-principales-caracteristicas/>
- Calendamaia. (2014, 10 enero). Eclipse IDE. Genbeta. <https://www.genbeta.com/desarrollo/eclipse-ide>
- López, M. (2022, 13 abril). ¿Qué es NetBeans? Ventajas y usos - IMMUNE Institute. IMMUNE Technology Institute. <https://immune.institute/blog/que-es-netbeans/>
- NetBeans IDE. (s. f.). <https://www.oracle.com/mx/application-development/netbeans/>
- Oracle. (s.f.). *Java SE Development Kit*. Recuperado de <https://www.oracle.com/java/technologies/downloads/>
- Oracle. (2020). *Oracle Java SE Licensing FAQ*. Recuperado de <https://www.oracle.com/java/technologies/javase/jdk-fags.html>
- Wikipedia. (2025, 12 de agosto). *Spring Framework*. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Spring_Framework
- Álvarez Caules, C. (2024, 9 de septiembre). *Spring Boot: ¿Qué es y cómo funciona?* Arquitectura Java. <https://www.arquitecturajava.com/spring-boot-que-es/>
- EBIS Education. (2024, 21 de diciembre). *Spring Framework: Qué es, para qué sirve y cómo usarlo*. EBIS Education. <https://www.ebiseducation.com/spring-framework-que-es-para-que-sirve-y-como-usarlo>