

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA U.A.B.C

FCITEC



Alumno: Almeraz Landeros Isai Magdaleno

Matricula: 02201503

Maestro: Emmanuel Gomez Gutierrez

Materia: Herramientas de Desarrollo de Software

Act: Reporte #2

Fecha: 27/ 08 / 2025

1. Introducción

Un **Entorno de Desarrollo Integrado** (IDE) es una aplicación de software que proporciona facilidades completas para el desarrollo de software, combinando herramientas como editor de código, compilador, debugger y gestor de proyectos en una única interfaz.

La **importancia de conocer diferentes IDEs** se basa en que cada uno ofrece características específicas para distintos tipos de desarrollo, tecnologías y filosofías de trabajo. En el ecosistema Java, la elección del IDE correcto puede impactar significativamente la productividad, la calidad del código y el tiempo de entrega de proyectos.

El **propósito de este reporte** es proporcionar un análisis completo de IntelliJ IDEA, evaluando sus características técnicas, ventajas competitivas y casos de uso empresariales. Este análisis ayudará a comprender por qué IntelliJ IDEA se ha posicionado como el IDE líder en el desarrollo Java profesional, con un uso del 84% en el mercado.

2. Antecedentes del IDE

JetBrains fue fundado en el año 2000 en Praga, República Checa, por tres desarrolladores rusos: Sergey Dmitriev, Valentin Kipyatkov y Eugene Belyaev. La empresa inició originalmente bajo el nombre "IntelliJ Software" con apenas 3 empleados y una visión clara: crear herramientas de desarrollo que hicieran la programación más productiva y placentera.

IntelliJ IDEA fue lanzado oficialmente en enero de 2001, estableciendo un nuevo estándar en IDEs Java al ser una de las primeras herramientas en ofrecer navegación avanzada de código y capacidades de refactoring integradas de manera nativa.

La evolución de JetBrains puede dividirse en cuatro etapas principales:

Inicios (2000-2005): Establecimiento de la base tecnológica con 7 empleados y 400 clientes iniciales. Durante estos años, IntelliJ IDEA se posicionó como una alternativa superior a las herramientas existentes.

Expansión y reconocimiento (2006-2010): En 2007 se lanzó IntelliJ IDEA 7.0 con soporte para frameworks empresariales como Spring e Hibernate. **El hit más significativo ocurrió en 2009** cuando JetBrains liberó el código fuente bajo la licencia Apache 2.0 y lanzó la Community Edition gratuita, lo que democratizó el acceso a herramientas de desarrollo de alta calidad.

Era de diversificación (2011-2015): JetBrains expandió su portafolio con el desarrollo de Kotlin (anunciado en 2011, código abierto en 2012) y estableció una relación estratégica con Google cuando Android Studio 1.0 se lanzó en 2014 basado en IntelliJ IDEA Community Edition.

Consolidación y era moderna (2016-2025): Kotlin 1.0 se lanzó en 2016, Google lo adoptó oficialmente para Android en 2017, y en 2019 se declaró como lenguaje preferido para el desarrollo Android. En 2025, JetBrains cuenta con 11,4 millones de usuarios activos y mantiene el liderazgo en herramientas de desarrollo.

3. Características principales

Lenguajes soportados:

IntelliJ IDEA ofrece soporte nativo para lenguajes JVM incluyendo Java (hasta Java 25 con Early Access), Kotlin (con modo K2 estable), Groovy y Scala. A través de plugins, soporta más de 15 lenguajes adicionales incluyendo JavaScript/TypeScript, HTML/CSS, SQL, Python, PHP, Go, Ruby y Rust.

Sistemas operativos compatibles:

IntelliJ IDEA es multiplataforma con soporte completo para:

- **Windows:** Windows 10 y Windows 11 con instaladores nativos
- **macOS:** Versiones recientes con soporte nativo para x86 y ARM64 (Apple Silicon)
- **Linux:** Distribuciones basadas en glibc 2.29+ (Ubuntu 20.04 o posterior), incluyendo soporte para WSL y Wayland desde 2024.2

Herramientas integradas:

Debugging y profiling: Debugger JVM avanzado con breakpoints inteligentes, soporte para debugging remoto, y profiler incorporado con Java Flight Recorder y Async Profiler que genera flame graphs con overhead mínimo (1-2%).

Control de versiones: Integración nativa con Git, Mercurial, Perforce y Subversion, incluyendo soporte para pull requests de GitHub/GitLab y Local History automático.

Build tools: Soporte completo para Maven (incluyendo Maven 4), Gradle con Kotlin DSL, Ant, sbt para Scala y Bazel desde 2025.2.

Base de datos: DataGrip embebido en Ultimate Edition con soporte para MySQL, PostgreSQL, Oracle, SQL Server y múltiples sistemas NoSQL

4. Análisis detallado del IDE

Facilidad de uso:

IntelliJ IDEA presenta una curva de aprendizaje empinada inicial pero con retorno rápido. El tiempo de adaptación típico es de 2-4 semanas para dominio básico, significativamente menor que Eclipse debido a su interfaz más intuitiva y menos configuración inicial requerida.

Productividad:

Refactoring inteligente: El sistema puede cambiar automáticamente signatures de métodos y actualizar todas las referencias en el proyecto. Por ejemplo, cambiar ***processUser(String name)*** a ***processUser(User user)*** con actualización automática de todas las llamadas.

Análisis en tiempo real: Más de 1000 inspecciones detectan problemas potenciales, sugieren optimizaciones y identifican código duplicado automáticamente. Los desarrolladores reportan 20-30% menos tiempo en tareas de refactoring.

Code completion inteligente: VA más allá de sugerencias sintácticas, proporcionando autocompletado contextual basado en análisis semántico profundo del código.

Depuración:

El debugger de IntelliJ incorpora "Evaluate Expression" con code completion completo, debugging visual con valores mostrados inline y soporte para debugging remoto en contenedores Docker. La funcionalidad de debugging avanzado permite modificar código durante la ejecución y continuar sin reiniciar la aplicación.

Extensibilidad:

El ecosistema cuenta con más de 8,300 plugins para Ultimate Edition y 7,600 para Community Edition. La calidad es superior al promedio del mercado debido a la curación activa del marketplace. Plugins destacados incluyen AWS Toolkit oficial, soporte completo para Minecraft Development y integración con herramientas de análisis como SonarQube.

Comunidad y soporte:

JetBrains mantiene documentación oficial integral con más de 100 tutoriales específicos, Features Trainer integrado en el IDE y recursos especializados por tecnología. La comunidad cuenta con más de 15,9 millones de desarrolladores globalmente y cuenta con reconocimiento como Gartner Customer's Choice 2024.

Recursos del sistema:

Los requisitos mínimos son 2 GB de RAM libre y 3.5 GB de espacio en disco, aunque se recomiendan 8 GB de RAM total y SSD para rendimiento óptimo. IntelliJ consume 29x más objetos que Eclipse según estudios de memoria, pero optimiza mejor el Garbage Collection con pausas promedio de 8ms.

Multilenguaje:

IntelliJ IDEA excela en desarrollo poliglota con soporte nativo para proyectos que combinan Java/Kotlin con JavaScript/TypeScript para frontend, SQL para persistencia, y Docker para deployment. La Ultimate Edition incluye funcionalidades completas de WebStorm para desarrollo web.

Casos de uso reales:

90 de las Fortune Global Top 100 companies utilizan herramientas de JetBrains. Casos documentados incluyen bancos con más de 10.000 desarrolladores usando IntelliJ para infraestructura crítica, y empresas como IT Manufactory que desarrollan plataformas automotrices usando Jakarta EE + React + tecnologías móviles.

5. Ventajas y desventajas

Ventajas

1. **Inteligencia de código superior:** Análisis semántico profundo que proporciona autocompletado contextual, refactoring confiable y detección proactiva de errores más avanzada que la competencia.
2. **Productividad medible:** Desarrolladores reportan 20-30% menos tiempo en refactoring y 40% más eficiencia en debugging según estudios de usuario, con 20% incremento en code merges documentado.
3. **Ecosistema integrado completo:** Única herramienta que combina IDE, herramientas de base de datos, profiler, y soporte enterprise sin necesidad de herramientas externas adicionales.

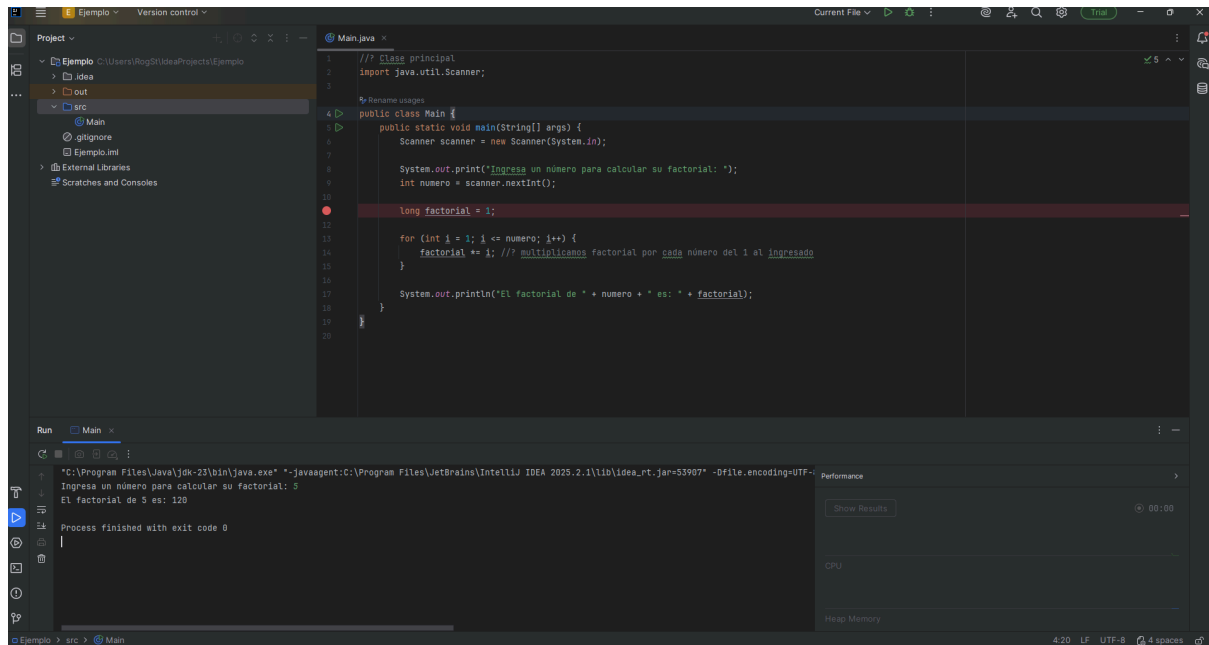
4. **Innovación continua:** Integración IA nativa, Code With Me para colaboración en tiempo real, y actualizaciones regulares con nuevas funcionalidades que mantienen el liderazgo tecnológico.
5. **Soporte empresarial excepcional:** base para Android Studio oficial, mejor soporte para frameworks modernos y adopción en el 90% de Fortune 100 companies.

Desventajas:

1. **Consumo elevado de recursos:** Utiliza 29x más objetos en memoria que Eclipse y requiere mayor RAM, especialmente problemático en hardware limitado o proyectos muy grandes.
2. **Costo de licencia:** Ultimate Edition cuesta \$169/año inicial, contra opciones gratuitas como Eclipse y NetBeans, representando una inversión significativa para equipos grandes.
3. **Curva de aprendizaje inicial:** Interfaz más compleja requiere 2-4 semanas de adaptación versus herramientas más simples como NetBeans que son inmediatamente accesibles.
4. **Dependencia de ecosistema propietario:** Mayor acoplamiento a herramientas JetBrains versus flexibilidad total de personalización que ofrece Eclipse.

6. Ejemplo trabajado en el IDE

- Captura de pantalla de un proyecto sencillo hecho en clase o de manera individual



Crear el proyecto y la clase

- Abrir IntelliJ IDEA y seleccionar New Project → Java → Next → Finish para crear un proyecto nuevo.
- Crear un archivo Java desde File → New → Java Class y nombrarlo FactorialExample.

Estructurar el programa

- Definir el método principal donde inicia la ejecución del programa.
- Configurar la captura de datos del usuario desde la consola.
- Inicializar la variable que almacenará el resultado del factorial.

Ingresar el número y calcular el factorial

- Solicitar al usuario un número entero y guardarlo en una variable.
- Definir otra variable inicializada en 1 para acumular el factorial.

- Usar un bucle que recorra los números desde 1 hasta el número ingresado, multiplicando la variable del factorial por cada valor sucesivo.

Mostrar el resultado

- Al finalizar el bucle, imprimir en la consola el resultado del factorial con un mensaje indicando el número ingresado y su factorial.

Ejecutar y probar

- Ejecutar el programa haciendo clic en Run (botón verde).
- Ingresar distintos números para verificar que el cálculo funciona correctamente.
- Cada ejecución es interactiva y permite probar números diferentes, y el uso de un tipo de dato adecuado (long) evita errores con factoriales grandes.

7. Defensa del IDE (argumento personal)

Datos de adopción de mercado muestran que IntelliJ IDEA creció del 71% al 84% en un año, siendo el único IDE que incrementó su participación mientras Eclipse descendió del 39% al 28%.

Inteligencia de código superior: Mientras Eclipse se enfoca en flexibilidad de personalización, IntelliJ prioriza inteligencia incorporada que funciona inmediatamente sin configuración extensiva.

Gestión de recursos optimizada: Aunque IntelliJ consume más memoria inicialmente, su sistema de Garbage Collection optimizado genera pausas menores (8ms promedio vs pausas más largas en Eclipse) y mejor throughput general una vez iniciado.

Conclusion

¿usaría este IDE en un proyecto real?

Sí lo usaría, especialmente en proyectos donde el enfoque principal sea Java, ya que IntelliJ IDEA ofrece muchas funciones, integraciones y ventajas que lo hacen muy completo y eficiente para trabajar en ese lenguaje. Pero para otros proyectos prefiero seguir utilizando VSCode, ya que lo tengo configurado a mi manera y se adapta muy bien a mis necesidades.

¿Qué se aprendió de esta práctica?

Se aprendió que IntelliJ IDEA es un IDE bastante completo y potente, utilizado en la mayoría del mercado gracias a sus múltiples características y ventajas frente a otras opciones.

Bibliografía

Blankenship, B. (2021, May 28). Why you should choose IntelliJ IDEA Ultimate. Medium.

<https://medium.com/nerd-for-tech/why-you-should-choose-intellij-idea-ultimate-84656cfefcf7>

Eclipse vs IntelliJ IDEA vs NetBeans for java development. (2021, January 13). GeeksforGeeks.

<https://www.geeksforgeeks.org/java/eclipse-vs-intellij-idea-vs-netbeans-for-java-development/>

IntelliJ IDEA. (n.d.). The JetBrains Blog; JetBrains. Retrieved August 28, 2025, from

<https://blog.jetbrains.com/idea/category/releases/>

IntelliJ IDEA overview. (n.d.). IntelliJ IDEA Help. Retrieved August 28, 2025, from <https://www.jetbrains.com/help/idea/discover-intellij-idea.html>

JetBrains Annual Highlights 2023. (n.d.). JetBrains: Developer Tools for Professionals and Teams. Retrieved August 28, 2025, from <https://www.jetbrains.com/lp/annualreport-2023/>

Obregon, A. (2024, April 28). IntelliJ IDEA Ultimate vs IntelliJ IDEA community edition — A comprehensive comparison. Medium. <https://medium.com/@AlexanderObregon/intellij-idea-ultimate-vs-intellij-idea-community-edition-a-comprehensive-comparison-6c02af27b183>

Sandu, B. (2024, October 20). NetBeans vs IntelliJ IDEA: Which IDE is better for java? TMS Outsource. <https://tms-outsource.com/blog/posts/netbeans-vs-intellij/>

Wikipedia contributors. (2025, August 8). IntelliJ IDEA. Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=IntelliJ_IDEA&oldid=1304889701

(N.d.-a). Capterra.com. Retrieved August 28, 2025, from <https://www.capterra.com/compare/136010-178681/IntelliJ-IDEA-vs-Apache-NetBeans>

(N.d.-b). Amazon.com. Retrieved August 28, 2025, from <https://aws.amazon.com/intellij/>