

# EVALUACIÓN DE IDEs

Entornos de  
desarrollo

# Evaluación de IDEs

**¿Qué características comunes comparten los IDEs en términos de edición de código, depuración y control de versiones.**

# Evaluación de IDEs

## Edición de código

- Autocompletado de código, haciéndote sugerencias.
- Resaltado de sintaxis
- Ambos IDEs te permiten refactorizar el código
- Los dos te permiten trabajar con linters y formateadores, aunque en el caso de VSC hay que usar extensiones (ESLint y Prettier) e IntelliJ lo tiene integrado.
- VSC permite bastantes lenguajes de programación instalando las extensiones correspondientes, IntelliJ también acepta muchos lenguajes, aunque acepta mejor lenguajes como Java, Kotlin y Python.



# Evaluación de IDEs



## Depuracion

### VS Code

Depuración básica con extensiones.

### Ambos

Ambos proporcionan herramientas de depuración con puntos de interrupción, inspección de variables y pilas de llamadas.

### Intellij

Herramientas de depuración avanzadas, integración profunda con lenguajes.



# Evaluación de IDEs

## Control de versiones

- Los dos funcionan perfectamente con Git permitiendo acciones como clonar repositorios, hacer commits, push, pull, etc. directamente desde los IDEs.
- Además permiten ejecutar comandos de Git directamente desde el terminal



# Evaluación de IDEs

**¿Qué diferencias notaron en la forma en que los IDEs manejan módulos, personalización y generación de ejecutables?**

# Evaluación de IDEs

## Manejo de módulos

- IntelliJ IDEA tiene un sistema para proyectos con múltiples módulos, especialmente en Java/Kotlin
- VSCode no tiene un sistema tan estructurado.





# Evaluación de IDEs

## Personalización

- IntelliJ ofrece personalización más limitada, pero optimizada para lenguajes específicos.
- Visual Code Studio se puede personalizar completamente.





# Evaluación de IDEs

## Generacion de ejecutables

### VS CODE

Requiere configuraciones externas o extensiones

### INTELLIJ

Generación nativa de ejecutables (JAR, WAR, etc.)

# Evaluación de IDEs

**¿Cuál de los IDEs elegidos consideran que es más adecuado para proyectos de desarrollo específicos y por qué?**

# Evaluación de IDEs

- Es mejor los IDEs propietarios porque tienen herramientas específicas para un lenguaje en concreto. Según el proyecto que necesites realizar. Si necesitas trabajar un proyecto en python, sería mejor utilizar pycharm, si necesitas un proyecto en java o kotlin, sería más adecuado en IntelliJ IDEA.





# Evaluación de IDEs

**¿Qué IDE recomendarían a un equipo de desarrollo que trabaja en proyectos de Python, Java o Kotlin y por qué?**

# Evaluación de IDEs

- IntelliJ IDEA es el más adecuado para Kotlin y Java debido a las herramientas y frameworks que contiene específicamente para ellos, aunque para Python no está tan especializado como otros IDEs.



# Evaluación de IDEs

**Tabla comparativa**



# Evaluación de IDEs

Característica	Visual Studio Code (VSCode)	IntelliJ IDEA
Edición de código	Resaltado de sintaxis, autocompletado, refactorización	Resaltado de sintaxis, autocompletado, refactorización
Depuración	Depuración básica con extensiones	Herramientas de depuración avanzadas, integración profunda con lenguajes
Control de versiones	Integración con Git a través de extensiones	Integración nativa con Git, SVN, Mercurial
Manejo de módulos	Soporte básico para proyectos con múltiples carpetas	Soporte avanzado para proyectos multi-módulo (Java/Kotlin)
Personalización	Alta personalización mediante extensiones	Personalización limitada pero optimizada para lenguajes específicos
Generación de ejecutables	Requiere configuraciones externas o extensiones	Generación nativa de ejecutables (JAR, WAR, etc.)
Soporte de lenguajes	Amplio soporte mediante extensiones (Python, JS, etc.)	Soporte nativo para Java, Kotlin y otros lenguajes JVM
Ideal para	Proyectos ligeros, desarrollo web, Python, JavaScript	Proyectos grandes en Java, Kotlin, y ecosistemas JVM
Integración con herramientas externas	Requiere configuración manual para herramientas externas	Integración automática con herramientas como Maven, Gradle, Spring
Velocidad y uso de recursos	Ligero y rápido, ideal para proyectos pequeños	Más pesado, adecuado para proyectos grandes y complejos
Precio	Gratuito	Pago (aunque tiene una versión Community gratuita)