

Entornos de Desarrollo Integrados (IDEs) y otros editores

- Entornos de Desarrollo Integrados (IDEs) y otros editores
 - Tipos de Editores de Código
 - Editores de Texto
 - IDEs
 - Herramientas que Componen un IDE
 - Diferencias entre compiladores, intérpretes y máquinas virtuales
 - Características Comunes en los IDEs
 - IntelliSense
 - Multicursor
 - Ejecución Paso a Paso
 - IDEs Especializados por Lenguaje
 - Licencias Educativas de JetBrains
 - Servicios Educativos de Otras Plataformas

Tipos de Editores de Código

Los editores de código se dividen en dos categorías principales: **editores de texto** y **IDEs**. Los editores de texto son más livianos y sencillos, pero no integran de forma nativa todas las herramientas necesarias para la ejecución de código. Los IDEs, en cambio, integran en un solo programa tanto el editor de texto, como las herramientas de compilación y depuración de programas. Sin embargo, son más pesados y menos versátiles precisamente debido a esto. A continuación, vamos a repasar una serie de editores e IDEs:

Editores de Texto

1. Notepad++

- **Características:** Ligero, soporte para múltiples lenguajes, resaltado de sintaxis, y una interfaz simple.
- **Limitaciones:** Funcionalidades básicas; carece de herramientas avanzadas como depuración.
- **Licencia:** Gratuito y de código abierto.

2. Visual Studio Code

- **Características:** Extensible mediante plugins, integración con Git, terminal integrado y soporte para múltiples lenguajes. Con las extensiones adecuadas, se comporta casi como si fuera un IDE. Tanto es así que en Mac OSX el IDE Visual Studio for Mac ha sido descontinuado en favor de este editor, con la extensión C# Dev Kit.
- **Limitaciones:** Requiere configuración adicional para funciones avanzadas.
- **Licencia:** Gratuito.

3. Sublime Text

- **Características:** Rápido, interfaz minimalista, soporte para múltiples lenguajes.
- **Limitaciones:** Versión de pago después de un periodo de evaluación; algunas características son limitadas.
- **Licencia:** Gratuito en versión de evaluación; pago para uso continuado.

IDEs

1. IntelliJ IDEA

- **Características:** IDE potente para Java, con soporte para Kotlin, Groovy y otros lenguajes. Ofrece integración con sistemas de control de versiones, herramientas de refactorización, y una interfaz intuitiva.
- **Limitaciones:** Requiere hardware medianamente potente.
- **Licencia:** Community Edition es gratuito y de código abierto; Ultimate Edition es de pago, con licencia educativa disponible.

2. Eclipse

- **Características:** IDE versátil principalmente para Java, con soporte para plugins que permiten el desarrollo en otros lenguajes. Ofrece herramientas de depuración y una gran comunidad.
- **Limitaciones:** La interfaz no es anticuada y puede no ser todo lo eficiente que uno desearía.
- **Licencia:** Gratuito y de código abierto.

3. NetBeans

- **Características:** IDE oficial para Java SE, pero también admite PHP y C/C++. Es fácil de usar y configurar.
- **Limitaciones:** Menos extensible que otros IDEs; a veces más lento en comparación con IntelliJ.
- **Licencia:** Gratuito y de código abierto.

4. Visual Studio

- **Características:** IDE completo para C#, .NET y desarrollo web, herramientas de depuración avanzadas, integración con Azure. Muy usado también para C++.
- **Limitaciones:** Solo disponible en Windows;

- **Licencia:** Community Edition es gratuito; Professional y Enterprise son de pago, con licencias educativas disponibles.

5. **Godot** (Editor de Juegos)

- **Características:** Herramientas para desarrollo de juegos 2D y 3D, scripting en GDScript, interfaz amigable.
- **Limitaciones:** Menos soporte para lenguajes convencionales; no es un IDE generalista aunque se pueden realizar aplicaciones con él.
- **Licencia:** Gratuito y de código abierto.

6. **Unreal Engine** (Editor de Juegos)

- **Características:** Motor para desarrollo de juegos 3D, herramientas visuales de creación de contenido y programación con Blueprints. Sirve también para crear contenido audiovisual en 3D y ha sido usado en series como *The Mandalorian*.
- **Limitaciones:** Requiere hardware potente; curva de aprendizaje pronunciada.
- **Licencia:** Gratuito, con royalties aplicables a los ingresos por juegos. Se pueden comprar licencias alternativas.

Herramientas que Componen un IDE

Los IDEs están compuestos por varias herramientas que facilitan el desarrollo de software. A continuación se presentan ejemplos de herramientas comunes:

1. **Editor de Código:** Donde se escribe el código fuente. Ejemplos incluyen el editor de IntelliJ y el editor de Visual Studio.
2. **Compilador/Intérprete:** Convierte el código fuente en un formato ejecutable. Por ejemplo, el compilador de Java en IntelliJ o el compilador de C# en Visual Studio.
3. **Herramientas de Depuración:** Permiten encontrar y corregir errores en el código. Tanto IntelliJ como Visual Studio cuentan con potentes herramientas de depuración integradas.
4. **Gestión de Proyectos:** Facilita la organización de archivos y recursos. Los IDEs permiten gestionar múltiples proyectos desde una única interfaz.

Diferencias entre compiladores, intérpretes y máquinas virtuales

Compiladores: La Traducción Completa

Un **compilador** actúa como un traductor completo de un idioma a otro. Toma el código fuente escrito en un lenguaje de programación de alto nivel (como C++ o C) y lo convierte, de una sola vez,

en código máquina o en un formato intermedio que la computadora puede entender directamente. Este proceso de compilación genera un archivo ejecutable independiente.

Las ventajas de la compilación radican en la velocidad de ejecución: una vez compilado, el programa se ejecuta de forma muy eficiente. Sin embargo, cualquier cambio en el código fuente requiere una nueva compilación de todo el programa. Ejemplos clásicos incluyen el compilador GCC para C/C++ y el compilador de Java que genera bytecode.

Intérpretes: La Ejecución Línea a Línea

A diferencia de los compiladores, un **intérprete** no traduce el código fuente por completo antes de la ejecución. En su lugar, lee y ejecuta el código línea por línea, o instrucción por instrucción, en tiempo real. Esto significa que el código fuente se traduce y ejecuta simultáneamente.

Los intérpretes ofrecen mayor flexibilidad y son ideales para el desarrollo rápido, ya que los cambios en el código se pueden probar de inmediato sin una fase de compilación. Sin embargo, la ejecución suele ser más lenta que la de los programas compilados debido al proceso de traducción en cada ejecución. Python, JavaScript y Ruby son ejemplos prominentes de lenguajes que suelen ser interpretados.

Máquinas Virtuales: Un Entorno de Ejecución Abstracción

Las **máquinas virtuales (VM)** añaden otra capa de abstracción. No son ni compiladores ni intérpretes en sí mismas, sino un entorno de software que emula un sistema informático completo o una plataforma de hardware específica. Su objetivo es proporcionar un entorno aislado y consistente para la ejecución de programas, independientemente del hardware o sistema operativo subyacente.

En el contexto de la ejecución de código, muchas máquinas virtuales trabajan en conjunto con un compilador o un intérprete. Por ejemplo, la Máquina Virtual de Java (JVM) toma el bytecode compilado de Java y lo interpreta o lo compila "justo a tiempo" (JIT) en código máquina para su ejecución. Esto permite que los programas Java sean "escritos una vez, ejecutados en cualquier lugar". Otro ejemplo son las máquinas virtuales de sistemas operativos (como VMware o VirtualBox) que permiten ejecutar múltiples sistemas operativos en un solo equipo físico.

Actividad

Busca 3 lenguajes compilados (que no sean C y C++), tres lenguajes interpretados (que no sean python, Javascript ni Ruby) y otros lenguajes que funcionen con un sistema de máquina virtual similar al de Java.

Actividad

Busca qué significa el término *transpilación*

Características Comunes en los IDEs

IntelliSense

- **Descripción:** Autocompletado de código y sugerencias contextuales.
- **Ejemplo:** En **Visual Studio Code**, al escribir una función en JavaScript, el IDE sugiere automáticamente los parámetros y funciones disponibles. En **IntelliJ**, la funcionalidad es similar para Java y otros lenguajes, mejorando la productividad del desarrollador.
- En Visual Studio Code, Intellisense se instala por separado para cada lenguaje de programación.

Multicursor

- **Descripción:** Permite editar múltiples líneas de código simultáneamente.
- **Ejemplo:** En **Visual Studio Code**, puedes colocar cursores en diferentes líneas y editar a la vez, lo que acelera cambios repetitivos. IntelliJ también ofrece esta funcionalidad, facilitando la edición de múltiples instancias de una variable o función.

Ejecución Paso a Paso

- **Descripción:** Herramientas de depuración que permiten seguir el flujo del programa línea por línea.
- **Ejemplo:** En **IntelliJ**, puedes establecer puntos de ruptura y seguir la ejecución de tu código para identificar errores. **Visual Studio** ofrece una funcionalidad similar, permitiendo a los desarrolladores depurar aplicaciones complejas.

IDEs Especializados por Lenguaje

Los IDEs y editores se especializan en diferentes lenguajes de programación. A continuación, se presentan algunos ejemplos destacados:

1. Java:

- **IntelliJ IDEA:** Community Edition es gratuito; Ultimate Edition de pago, pero dispone de gratuidad con la versión educativa.
- **Eclipse:** Gratuito y de código abierto.

Actividad

Instala IntelliJ IDEA y ejecuta un hola mundo.

2. C/C++:

- **Visual Studio:** Community Edition es gratuito; versiones Professional y Enterprise son de pago con licencias educativas.
- **Code::Blocks:** Gratuito y de código abierto.
- **CLion:** IDE de JetBrains, gratuito para uso no comercial.

Actividad

Instala CLION y ejecuta un hola mundo en C (usando printf) y otro en C++ (usando cout)

3. C#:

- **Visual Studio:** Community Edition es gratuito; versiones de pago con licencias educativas.
- **Visual Studio Code + C# Dev Kit:** Gratuito; ideal para desarrollo de aplicaciones C# multiplataforma.
- **Rider:** IDE de JetBrains, gratuito para uso no comercial

Actividad

Instala Rider y ejecuta un hola mundo.

4. Python:

- **PyCharm:** Community Edition es gratuito; Professional Edition de pago con licencia educativa.
- **Visual Studio Code:** Gratuito; soporte para Python mediante extensiones. Necesitas instalar [Python 3](#).

Actividad

Instala Pycharm y también Python. Ejecuta el mismo hola mundo en Pycharm y en VS Code usando Python3.

5. JavaScript:

- **Visual Studio Code:** Gratuito; muy popular para desarrollo web.
- **WebStorm:** De pago, con licencia educativa disponible.

6. SQL:

- **DataGrip:** De pago, con licencia educativa.
- **DBeaver:** Gratuito y de código abierto.

Licencias Educativas de JetBrains

Para obtener la **licencia gratuita para estudiantes de JetBrains**, sigue estos pasos:

1. Visita el [sitio web de JetBrains](#).
2. Selecciona el producto que deseas utilizar (IntelliJ, PyCharm, etc.).
3. Regístrate con tu cuenta de correo electrónico institucional (por ejemplo, [@alu.edu.gva.es](#)).
4. Completa el formulario de solicitud y sigue las instrucciones para activar tu licencia.

Puedes usar tu identidad digital (@alu.edu.gva.es) para crear una cuenta en JetBrains y aplicar para solicitar la licencia educativa. Dicha licencia es renovable mientras dure tu educación.

Ten en cuenta que tu identidad digital no será accesible cuando acabes tu etapa en el instituto, pero puedes transferir o añadir más correos electrónicos a tu cuenta para poder seguir accediendo a JetBrains.

Cuando acaba tu periodo educativo, dispones de un descuento del 40% para comprar una licencia anual, tanto del pack completo de IDEs de JetBrains como los IDEs en específico que más emplees. Esta licencia ya no tiene las limitaciones de las licencias educativas y la puedes usar para tu día a día.

Con la licencia anual, JetBrains ofrece una [licencia de reserva perpetua](#). Esta licencia te permite usar para toda la vida la versión del IDE más actual en el momento de la compra.

Aunque con los IDEs community edition o gratuitos sin uso comercial nos es suficiente en la mayoría de casos, el poder acceder a IntelliJ IDEA Ultimate de forma gratuita nos permite, por ejemplo, configurar proyectos con Jakarta EE.

Actividad

Solicita tu licencia educativa de JetBrains usando tu carnet de estudiante. Cuando lo tengas, instala desde su aplicación JetBrains Toolbox todos los IDEs que necesites, especialmente IntelliJ IDEA Ultimate Edition.

Servicios Educativos de Otras Plataformas

Además de JetBrains, los estudiantes pueden acceder a otros servicios gratuitos o con licencia educativa:

- **Microsoft Azure:** Acceso gratuito a servicios en la nube, ideal para aprender sobre desarrollo y administración de sistemas.
- **GitHub:** Licencias educativas que ofrecen herramientas premium de desarrollo colaborativo.
- **AWS Educate:** Acceso a recursos y herramientas de Amazon Web Services para aprender sobre computación en la nube.
- **Google Cloud Services:** Ofrece créditos gratuitos para estudiantes que deseen experimentar con servicios de Google.