

# 1. ¿Qué es un IDE?

Un **IDE (Entorno de Desarrollo Integrado)** es una aplicación diseñada para facilitar el desarrollo de software reuniendo en un mismo lugar todas las herramientas necesarias para programar.

## Componentes principales:

- **Editor de código:**  
Es donde el programador escribe el código fuente. Incluye resaltado de sintaxis, autocompletado, detección de errores y herramientas de formato.
- **Compilador / Intérprete:**  
Transforma el código en un programa ejecutable (compiladores) o lo ejecuta línea a línea (intérpretes).
- **Depurador (Debugger):**  
Permite ejecutar el programa paso a paso, ver valores de variables, identificar fallos lógicos y corregir errores.
- **Gestor de proyectos:**  
Organiza carpetas, archivos y configuraciones del programa. Facilita la navegación dentro del proyecto.
- **Extensiones o plugins:**  
Añaden nuevas funciones al IDE, como soporte para nuevos lenguajes, temas visuales, herramientas de análisis, etc.

## Objetivo principal:

**Hacer que el trabajo del programador sea más rápido, fácil y ordenado**, evitando usar varias herramientas externas.

## ★ 2. Principales IDEs (versión ampliada)

### ◇ 2.1 Code::Blocks

- Diseñado específicamente para desarrollo en **C y C++**.

- Muy adecuado para estudiantes por su interfaz simple.
- Permite instalar diversas versiones con **MinGW**, un compilador de C/C++.
- Es ligero, rápido y funciona en Windows, Linux y macOS.
- Soporta proyectos grandes y plugins, aunque su apariencia es más clásica.

## ◇ 2.2 Visual Studio Code (VS Code)

- Un editor avanzado que funciona como IDE gracias a las extensiones.
- Gratuito y multiplataforma (Windows, Linux, macOS).
- Posee integración completa con **Git**, lo que permite controlar versiones del código.
- Extremadamente personalizable: temas, extensiones, lenguajes, debuggers.
- Ideal para lenguajes como JavaScript, Python, Java, C/C++, PHP, entre otros.
- Uno de los entornos más utilizados actualmente por su rapidez y su ecosistema enorme.

## ◇ 2.3 Visual Studio

- IDE profesional creado por Microsoft.
- Muy usado en empresas para desarrollar aplicaciones .NET, videojuegos (Unity), aplicaciones de escritorio y más.
- Incluye herramientas avanzadas:
  - Analizadores de código
  - Diagnóstico de rendimiento
  - Diseñadores visuales
  - Depuradores avanzados
- Solo funciona en **Windows**.
- La versión **Community** es gratuita y muy potente para estudiantes y desarrolladores independientes.

## ◇ 2.4 Eclipse

- Uno de los IDEs más longevos y utilizados para **Java**.

- Totalmente gratuito y de código abierto.
- Basado en plugins, lo que permite ampliar sus capacidades fácilmente.
- Tiene versiones especializadas para Java, C/C++, desarrollo web, etc.
- Popular en entornos académicos debido a su estabilidad y soporte.
- Puede sentirse algo pesado en comparación con otros IDEs modernos.

## ◇ 2.5 NetBeans

- Otro IDE muy utilizado para Java, mantenido actualmente por **Apache**.
- Interfaz intuitiva y más visual que Eclipse.
- Compatible con varios lenguajes: Java, PHP, C/C++, HTML5, etc.
- Más pesado en rendimiento, pero muy eficiente para proyectos Java estructurados.
- Facilita la creación de interfaces gráficas con herramientas visuales.

## ◇ 2.6 IntelliJ IDEA

- IDE avanzado desarrollado por JetBrains.
- Considerado uno de los mejores para Java y Kotlin.
- Ofrece autocompletado inteligente, análisis profundo del código y sugerencias contextuales.
- Perfecto para trabajar con frameworks modernos como **Spring**, **Hibernate**, **JavaFX**, etc.
- Tiene licencia Community (gratuita) y Ultimate (de pago con herramientas extra).
- Recomendado para desarrollos profesionales y proyectos complejos.

# 3. Entornos de desarrollo online

Los **IDEs online** permiten programar desde un navegador web sin instalar nada.

### Ventajas:

- Acceso desde cualquier lugar.

- No dependen del hardware del usuario.
- Fomentan el trabajo colaborativo en tiempo real.
- Útiles para tareas rápidas, pruebas, enseñanza o compartir código.

### Características:

- Suelen parecerse visualmente a Visual Studio Code.
- Permiten ejecutar código directamente en la nube.
- Ofrecen entornos preconfigurados para múltiples lenguajes.

### Ejemplos:

- **StackBlitz** → muy fuerte en JavaScript, Angular, React.
- **Replit** → soporta más de 50 lenguajes, muy popular en educación.
- **Codeanywhere** → orientado a desarrollo profesional en la nube.

## 4. Instalación de entornos de desarrollo

### ◇ Para C/C++

Se recomienda:

- **VS Code**, si quieres flexibilidad y soporte para varios lenguajes.
- **Code::Blocks**, si buscas la opción más directa y simple.

### ✓ Instalación de Code::Blocks:

1. Ir a [codeblocks.org](https://codeblocks.org).
2. Descargar versión con **MinGW incluido**.
3. Instalar y abrir el IDE.
4. Crear un proyecto tipo "Console Application".

## ◇ Para Java

Se requieren dos componentes:

### 1) Instalar el JDK (Java Development Kit)

- Permite compilar y ejecutar programas Java.
- Debe estar configurado en el sistema.

### 2) Instalar un IDE (Eclipse, IntelliJ...)

- Eclipse es recomendado en entornos educativos.
- IntelliJ es ideal para proyectos profesionales.

### ✓ Instalación de Eclipse:

1. Descargar e instalar el JDK.
2. Acceder a *eclipse.org*.
3. Instalar la versión “Eclipse IDE for Java Developers”.
4. Crear un **workspace** y luego un proyecto Java.

## 5. Ideas clave para memorizar rápido

✓ IDE = muchas herramientas en una sola aplicación.

✓ **VS Code** es el más usado hoy por su flexibilidad.

✓ **Visual Studio** es profesional, potente y solo para Windows.

✓ **Eclipse, NetBeans e IntelliJ** dominan el desarrollo en Java.

✓ Los **IDEs online** permiten programar desde cualquier lugar sin instalar nada.

✓ Para programar en C/C++ es necesario tener un **compilador** como MinGW.

✓ Para Java, el **JDK es obligatorio** antes de instalar cualquier IDE.