

# INSTALACIÓN Y USO DE ENTORNOS DE DESARROLLO

## 1. CONCEPTO DE ENTORNO DE DESARROLLO. EVOLUCIÓN HISTÓRICA

Un IDE se compone de:

- **Editor** de código de programación
- **Compilador**
- **Intérprete**
- **Depurador**
- **Constructor** de interfaz gráfico

Los primeros IDE **nacieron a principios de los 70** y se popularizaron en los 90.

Normalmente los IDE estaban dedicados a un **único lenguaje**, no obstante las últimas versiones tienden a ser compatibles con **varios lenguajes** mediante la instalación de **plugins** adicionales.

Los programas estaban escritos con **diagramas de flujo** y entraban al sistema a través de las **tarjetas perforadas**. Posteriormente, eran compilados.

El primer lenguaje de programación que utilizó un IDE fue el **BASIC** (que fue el primero en abandonar también las tarjetas perforadas o las cintas de papel). Estaba basado en **consola de comandos** exclusivamente. Sin embargo su uso de gestión de archivos, compilación y depuración es compatible con los IDE actuales.

El primer IDE fue llamado **Maestro**, nacido principios de los **70** e instalado por unos 22.000 programadores en todo el mundo, lideró el campo en los 70 y 80.

El uso de IDEs se ratifica y afianza en los 90, hoy en día contamos con infinidad de IDE.

**Tabla de los IDE más relevantes hoy en día: Entorno de desarrollo Lenguajes que soporta**  
**Tipo de licencia**

Entorno de desarrollo	Lenguajes que soporta	Tipo de licencia
NetBeans	C/C++, Java, JavaScript, PHP, Python	De uso público
Eclipse	Ada, C/C++, Java, JavaScript, PHP	De uso público
Microsoft Visual Studio	Basic, C/C++, C#	Propietario
C++ Builder	C/C++	Propietario
JBuilder	Java	Propietario

## 2. FUNCIONES DE UN ENTORNO DE DESARROLLO

Los IDE están compuestos por una **serie de herramientas** de software de programación:

- **Editor** de código fuente
- **Compilador** y/o intérprete
- **Automatización** de generación de herramientas
- Un **depurador**

Las **funciones** de los IDE son:

- **Editor** de código
- **Auto-completado** de código, atributos y métodos de clases.
- **Identificación** automática de código.
- Herramientas de **concepción visual** para crear y manipular componentes visuales.
- **Asistentes** y utilidades de gestión y generación de código.
- **Archivos fuente** en unas carpetas y compilados a otras.
- **Compilación** de proyectos complejos en un solo paso.
- **Control de versiones**: tener un único almacén de archivos compartido por todos los colaboradores de un proyecto. Ante un error, mecanismo de auto-recuperación a un estado anterior estable.
- **Soporta cambios** de varios usuarios de manera simultánea.
- **Generador de documentación** integrado.
- **Detección de errores** de sintaxis en tiempo real.

Otras **funciones** importantes son:

- Ofrece **refactorización** de código: cambios menores
- Permite introducir automáticamente **tabulaciones y espaciados**
- **Depuración**: seguimientos de variables, puntos de ruptura y mensajes de error del intérprete

- Aumento de funcionalidades a través de la gestión de **módulos y plugins**.
- **Administración de las interfaces** de usuario
- **Administración de las configuraciones** del usuario

### 3. ENTORNOS INTEGRADOS LIBRES Y PROPIETARIOS

#### Entornos Integrados Libres

Son aquellos con **licencia de uso público**, no hay que pagar por ellos.

IDE	Lenguajes que soporta	Sistema Operativo
NetBeans	C/C++, Java, JavaScript, PHP, Python	Windows, Linux, Mac OS X
Eclipse	Ada, C/C++, Java, JavaScript, PHP	Windows, Linux, Mac OS X
Gambas	Basic	Linux
Anjuta	C/C++, Python, Javascript	Linux
Geany	C/C++, Java	Windows, Linux, Mac OS X
GNAT Studio	Fortran	Windows, Linux, Mac OS X

#### Entornos Integrados Propietarios

Son aquellos que **necesitan licencia**, no son free software, hay que pagar por ellos.

IDE	Lenguajes que soporta	Sistema Operativo
Microsoft Visual Studio	Basic, C/C++, C#	Windows
FlashBuilder	ActionScript	Windows, Mac OS X
C++ Builder	C/C++	Windows
Turbo C++ profesional	C/C++	Windows
JBuilder	Java	Windows
JCreator	Java	Windows, Linux, Mac OS X
Xcode	C/C++, Java	Mac OS X

### 4. ESTRUCTURA DE ENTORNOS DE DESARROLLO

Los IDE están formados por una **serie de componentes software** que determinan sus funciones:

- **Editor de textos:** Se resalta y colorea la sintaxis, autocompleta código, ayuda y listado de parámetros de funciones y métodos de clase. Inserción automática de paréntesis, corchetes, tabulaciones y espaciados.

- **Compilador/intérprete:** Detección de errores de sintaxis en tiempo real. Características de refactorización.
- **Depurador:** Botón de ejecución y traza, puntos de ruptura y seguimiento de variables. Opción de depurar en servidores remotos.
- **Generador automático de herramientas:** Visualización, creación y manipulación de componentes visuales. Todo un arsenal de asistentes y utilidades de gestión y generación código.
- **Interfaz gráfica:** Programar en varios lenguajes con un mismo IDE. Interfaz agradable que puede acceder a innumerables bibliotecas y plugins, aumentando las opciones.

A partir del punto 5 se aplica las actividades de instalación y configuración, recomiendo estudiar de ahí.