

UT 2. Instalación y uso de entornos de desarrollo.

Un entorno de desarrollo o IDE (Integrated Development Environment) es una aplicación informática que está compuesta por un conjunto de herramientas que facilitan la tarea al programador de forma que ayuda a desarrollar aplicaciones con más rapidez.

Los entornos de desarrollo son utilizados en la fase de codificación del ciclo de vida de software independientemente del modelo que se utilice.

Existen entornos de desarrollo para un solo lenguaje o para múltiples lenguajes.

2.1. Evolución Histórica

En las décadas de utilización de la tarjeta perforada como sistema de almacenamiento el concepto de Entorno de Desarrollo Integrado sencillamente no tenía sentido. 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 utiliza un IDE fue el BASIC (que fue el primero en abandonar también las tarjetas perforadas o las cintas de papel). Éste primer IDE estaba basado en consola de comandos exclusivamente (normal por otro lado, si tenemos en cuenta que hasta la década de los 90 no entran en el mercado los sistemas operativos con interfaz gráfica). Sin embargo, el uso que hace de la gestión de archivos, compilación, depuración... es perfectamente compatible con los IDE actuales.

A nivel popular, el primer IDE puede considerarse que fue el IDE llamado Maestro. Nació a principios de los 70 y fue instalado por unos 22000 programadores en todo el mundo. Lideró el campo durante los años 70 y 80. El uso de los entornos integrados de desarrollo se ratifica y afianza en los 90 y hoy en día contamos con infinidad de IDE, tanto de licencia libre como no.

Hoy en día los IDE más utilizados son:

- NetBeans
- Eclipse
- Microsoft Visual Studio.

2.2. Las funciones de los IDE son:

- Editor de código: coloración de la sintaxis.
- 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 en el código que facilitan su legibilidad sin alterar su funcionalidad (por ejemplo, cambiar el nombre a una variable).
- Permite introducir automáticamente tabulaciones y espaciados para aumentar la legibilidad.
- Depuración: seguimiento de variables, puntos de ruptura y mensajes de error del intérprete.
- Aumento de funcionalidades a través de la gestión de sus módulos y plugins.
- Administración de las interfaces de usuario (menús y barras de herramientas).
- Administración de las configuraciones del usuario.

2.3. Los elementos de un IDE

Los elementos fundamentales que suelen encontrarse en un entorno integrado de desarrollo de software son los siguientes:

- **Editor de código:** Proporciona una interfaz para escribir y editar el código fuente. Suele incluir características como resaltado de sintaxis, autocompletado, indentación automática y otras utilidades para facilitar la escritura del código.
- **Compilador/Intérprete:** Permite compilar o interpretar el código fuente en un formato ejecutable o en bytecode, dependiendo del lenguaje de programación utilizado. Algunos IDEs también ofrecen la capacidad de ejecutar el código directamente desde el entorno.
- **Depurador:** Es una herramienta que ayuda a identificar y corregir errores en el código. Permite establecer puntos de interrupción, examinar el estado de las variables, ejecutar el código paso a paso y realizar otras operaciones para analizar y solucionar problemas en el programa.
- **Gestión de proyectos o ficheros:** Permite crear, organizar y administrar proyectos de desarrollo de software. Esto incluye la capacidad de crear estructuras de directorios, agregar o eliminar archivos, gestionar dependencias y realizar otras tareas relacionadas con la organización del proyecto.
- **Control de versiones:** Algunos IDEs incluyen integración con sistemas de control de versiones como Git, que permiten realizar seguimiento de cambios en el código, realizar confirmaciones (commits), fusionar (merge) ramas y otras operaciones relacionadas con la gestión del código fuente.
- **Herramientas de construcción:** Algunos IDEs proporcionan herramientas para automatizar el proceso de construcción del software, como la generación de archivos de configuración, la compilación, el empaquetado y otras tareas relacionadas con la construcción del proyecto.

- **Integración con herramientas externas:** Los IDEs suelen ofrecer integración con otras herramientas y servicios externos, como sistemas de gestión de bases de datos, terminal, servidores web, frameworks, bibliotecas, entre otros, para facilitar el desarrollo y la integración con otros componentes del sistema.
- **Plugins:** son complemento que se relacionan con otras herramientas para agregarle una nueva función y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal.