

Entornos de desarrollo

IDE(Entorno de Desarrollo Integrado)

Índice

1. INTRODUCCIÓN.....	3
2. DEFINICIÓN DE IDE	3
3. FUNCIONES DE UN IDE.....	4
4. COMPONENTES DE UN IDE	5
5. ENTORNOS INTEGRADOS LIBRES Y PROPIETARIOS	6
5.1. ENTORNOS INTEGRADOS LIBRES	6
5.2. ENTORNOS INTEGRADOS PROPIETARIOS.....	6

1. INTRODUCCIÓN.

En el tema anterior vimos las fases del ciclo de vida del desarrollo del software. Una de ellas era **la fase de codificación**, en la cual se hacía uso de algún lenguaje de programación para pasar todas las acciones que debía llevar a cabo la aplicación a algún lenguaje que la máquina fuera capaz de entender y ejecutar.

También se hizo alusión a herramientas de apoyo al proceso de programación. En esta unidad vamos a analizar, instalar y ejecutar estas herramientas para entender su acción y efecto.

Muchas personas aprenden a programar utilizando un editor de texto simple, compilador y depurador. Pero la mayoría, finalmente, terminan haciendo uso de algún entorno de desarrollo integrado para crear las aplicaciones.

2. DEFINICIÓN DE IDE

IDE, son las siglas de (Integrated Development Environment), en español **Entorno de Desarrollo Integrado**.

En la fase de desarrollo de software **intervienen varias herramientas** en cada una de las distintas etapas hasta llegar a obtener el resultado final (un editor de texto, depuradores, compiladores...).

Es por ello por lo que se crea el IDE, **un programa informático que agrupa diversas herramientas de programación para facilitar la tarea al programador y obtener mayor rapidez en el desarrollo**, en él se puede llevar a cabo todo el desarrollo completo de un programa puesto que proporciona componentes estrechamente unidos con interfaces de usuario similares.

El objetivo es abstraer la configuración necesaria para reconstruir las utilidades de línea de comandos en una unidad cohesiva, que en teoría reduce el tiempo para aprender un lenguaje y aumenta la productividad del desarrollador.

Este puede centrarse en un lenguaje o en varios, como es el caso de Eclipse que mediante plugins añade soporte a nuevos lenguajes.

Un IDE normalmente consiste en un editor de texto donde escribir el código con resaltado de sintaxis y corrector sintáctico, un compilador y/o interprete, un depurador, un control de versiones...

Actualmente, muchos entornos de desarrollo modernos también tienen:

- un navegador de clases
- un inspector de objetos
- un diagrama de jerarquía de clases
- un constructor de interfaz gráfica (GUI)

Debido a esto, un IDE **es un software de alta complejidad que necesita un periodo de adaptación y aprendizaje** por parte del programador para poder conseguir incrementar la productividad en el desarrollo. Estos entornos también pueden formar parte de un paquete

mayor, denominado **SDK** (*Software Development Kit*) que no es más ni menos que un IDE junto a documentación técnica y ejemplos de código orientados específicamente a un sistema.

No hay unos entornos de desarrollo más importantes que otros. La elección del IDE más adecuado dependerá del lenguaje de programación que vayamos a utilizar para la codificación de las aplicaciones y el tipo de licencia con la que queramos trabajar.

3. FUNCIONES DE UN IDE

Como sabemos, los entornos de desarrollo están compuestos por una serie de herramientas software de programación, necesarias para la consecución de sus objetivos. Estas herramientas son:

- Un editor de código fuente.
- Un compilador y/o un intérprete.
- Automatización de generación de herramientas.
- Un depurador.

Las funciones de los IDE son:

- Editor de código: coloración de la sintaxis.
- Autocompletado 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, hay mecanismo de autorrecuperación 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.

4. COMPONENTES DE UN IDE

A fin de poder llevar a cabo las funciones mencionadas antes, los entornos de desarrollo ya sean libres o propietarios, están formados por una serie de componentes software que determinan sus funciones. Estos componentes son:

- **Editor de textos:** Resalta y colorea la sintaxis, tiene la función de autocompletar 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:** Para la visualización, creación y manipulación de componentes visuales y todo un arsenal de asistentes y utilidades de gestión y generación código.
- **Interfaz gráfica:** Nos brinda la oportunidad de programar en varios lenguajes con un mismo IDE. Es una interfaz agradable que puede acceder a innumerables bibliotecas y plugins, aumentando las opciones de nuestros programas.

5. ENTORNOS INTEGRADOS LIBRES Y PROPIETARIOS

El aspecto de la licencia del IDE que se elija para el desarrollo de un proyecto es una cuestión de vital importancia. En su elección prevalecerá la decisión de los supervisores del proyecto y de la dirección de la empresa.

5.1. Entornos Integrados Libres

Son aquellos con licencia de uso público.

No hay que pagar por ellos, y aunque los más conocidos y utilizados son Eclipse y NetBeans, hay bastantes más.

Tipos de entornos de desarrollo libres más relevantes en la actualidad.		
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.

5.2. Entornos Integrados Propietarios

Son aquellos entornos integrados de desarrollo que necesitan licencia. No son free software, hay que pagar por ellos.

El más conocido y utilizado es Microsoft Visual Studio, que usa el framework .NET y es desarrollado por Microsoft.

Tipos de entornos de desarrollo propietarios más relevantes en la actualidad.		
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, Linux, Mac OS X.
JCreator.	Java.	Windows.
Xcode.	C/C++, Java.	Mac OS X.