

UT02. INSTALACIÓN Y USO DE ENTORNOS DE DESARROLLO.

Entornos de Desarrollo

1 DAW – C.I.F.P. Carlos III - Cartagena

Índice

- 1.-Concepto de Entorno de desarrollo.
- 2.- Funciones de un Entorno de desarrollo.
- 3.- Entornos integrados libres y propietarios.
- 4.- Estructura de entornos de desarrollo.
- 5.- Instalación de entornos integrados de desarrollo.
- 6.- Configuración y personalización de entornos de desarrollo.
- 7.- Gestión de módulos:
 - 7.1.- Añadir
 - 7.2.- Eliminar
 - 7.3.- Funcionalidades
 - 7.4.- Herramientas concretas.
- 8.- Uso básico de entornos de desarrollo:
 - 8.1.- Edición de programas
 - 8.2.- Generación de ejecutables.
- 9.- Actualización y mantenimiento de entornos de desarrollo

1. Concepto de Entorno de desarrollo. Evolución histórica

Concepto de Entorno de desarrollo. Evolución histórica

- Codificación
 - Uso de un lenguaje de programación
 - Herramientas de apoyo
- Mínimo: editor + compilador + depurador ➔ migración a IDE
 - **IDE.-** Software compuesto por un conjunto de herramientas de programación
 - Editor de código de programación
 - Compilador
 - Intérprete
 - Depurador
 - Constructor de interfaz gráfico

Concepto de Entorno de Desarrollo

- Primeros entornos de desarrollo integrados nacieron a principios de los **años 70**, y se popularizaron en la década de los 90.
- **Objetivo:**
 - Ganar fiabilidad y tiempo en los proyectos de software.
 - Proporcionar al programador una serie de componentes con la misma interfaz gráfica, con la consiguiente comodidad, aumento de eficiencia y reducción de tiempo de codificación.
- Un **IDE** está dedicado a un determinado lenguaje de programación.
- **Tendencia:** IDE's compatibles con varios lenguajes (por ejemplo, Eclipse, NetBeans, Microsoft Visual Studio) mediante la instalación de plugins adicionales.

Evolución histórica

- En las décadas de tarjetas perforadas (medio de almacenamiento):
 - No existía IDE
 - Diagramas de flujo → introducción al sistema mediante Tarjetas Perforadas → Compilación
- Primer lenguaje que utilizó un IDE fue **BASIC**, primero en abandonar las tarjetas perforadas.
 - Basado en consola de comandos exclusivamente (no desarrollado interfaz gráfica).
 - Uso de la gestión de archivos, compilación y depuración 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 22.000 programadores en todo el mundo. Lideró este campo durante los años 70 y 80.
- El uso de los entornos integrados de desarrollo se ratifica y afianza en los 90
- Hoy en día contamos con infinidad de IDE, tanto de licencia libre como no.

2. Funciones de un entorno de desarrollo

Funciones de un entorno de desarrollo

- Los entornos de desarrollo están compuestos por una serie de herramientas software de programación
 - Un editor de código fuente
 - Un compilador y/o un intérprete
 - Automatización de generación de herramientas
 - Un depurador

Funciones de un entorno de desarrollo

- **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.

Funciones de un entorno de desarrollo

- Otras funciones adicionales
 - 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**.

3. Entornos integrados libres y propietarios

Entornos integrados libres y propietarios.

- Libres
 - Con licencia de uso público

	ID	Lenguajes	Sistema Operativo

Entornos integrados libres y propietarios

- Proprietarios.
 - Necesitan licencia

	IDE	Lenguajes que soporta	Sistema Operativo