



**UT2- INSTALACIÓN Y USO DE ENTORNO DE
DESARROLLO**
Entornos de desarrollo (1º DAW)

Dpto. INFORMÁTICA





Curso: 2022-23

Unidad 2: Entornos de desarrollo

Contenido

1. INTRODUCCIÓN	2
2. EVOLUCIÓN HISTORICA	2
3. FUNCIONES DE UN ENTORNO DE DESARROLLO	2
4. COMPONENTES DE UN ENTORNO DE DESARROLLO	3
5. ENTORNOS DE DESARROLLO ACTUALES	4
6. ENTORNOS DE DESARROLLO ONLINE	6
7. ENTORNOS INTEGRADOS LIBRES Y PROPIETARIOS.....	6
7.1 Entornos Integrados Libres.....	6
7.2 Entornos Integrados Propietarios	6
8. INSTALACIÓN DE UN ENTORNO DE DESARROLLO	7
8.1 El compilador de Java	7
8.2 Depurar un programa	8
8.3 Profiler. Análisis de aplicaciones.....	8
8.4 Generación automática de documentación	8
8.5 Gestión de módulos	8

 <p>CIFP VIRGEN DE GRACIA</p>	<p>UT2- INSTALACIÓN Y USO DE ENTORNO DE DESARROLLO</p> <p>Entornos de desarrollo (1º DAW)</p>	<p>Dpto. INFORMÁTICA</p>  <p>Curso: 2022-23</p>
--	---	---

1. INTRODUCCIÓN

IDE son las siglas de *Integrated Development Enviroment*, en español *Entorno Integrado de Desarrollo*, es una aplicación informática compuesta por un conjunto de herramientas de programación que van a facilitar la tarea al programador y obtener mayor rapidez y eficacia en el desarrollo de las aplicaciones.

Normalmente, un IDE está dedicado a un determinado lenguaje de programación. No obstante, las últimas versiones de los IDEs tienden a ser compatibles con varios lenguajes (por ejemplo, Eclipse, NetBeans, Microsoft Visual Studio, IntelliJ...) mediante la instalación de plugins adicionales.

En la fase de desarrollo de software intervienen varias herramientas en cada una de las distintas etapas hasta llegar a obtener el resultado final, existen IDEs para todas las etapas del desarrollo.

2. EVOLUCIÓN HISTORICA

Los primeros Entornos de Desarrollo Integrados nacen a principios de los años 70, y se popularizan en la década de los 90. Tienen el objetivo de **ganar fiabilidad y tiempo** en los proyectos de software. Proporcionan al programador una serie de componentes con la misma interfaz gráfica, ganando comodidad, aumento de eficiencia y reducción de tiempo de codificación.


El primer lenguaje de programación que utiliza un IDE fue el BASIC. Éste primer IDE estaba basado en consola de comandos exclusivamente (obvio ya que hasta los 90 no existían SO 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.

3. FUNCIONES DE UN ENTORNO DE DESARROLLO

Los entornos de desarrollo están compuestos por una serie de herramientas software de programación, necesarias para la consecución de sus objetivos. Básicamente estas herramientas son:

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

 <p>CIFP VIRGEN DE GRACIA</p>	<p>UT2- INSTALACIÓN Y USO DE ENTORNO DE DESARROLLO</p> <p>Entornos de desarrollo (1º DAW)</p>	<p>Dpto. INFORMÁTICA</p>  <p>Curso: 2022-23</p>
--	---	---

Las funciones de un 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.

4. COMPONENTES DE UN ENTORNO DE DESARROLLO

Un IDE normalmente consiste en un editor de texto donde escribir el código con resaltado de sintaxis y corrector sintáctico, un compilador y/o intérprete, un depurador, un control de versiones, un constructor de interfaz gráfica (GUI), entre otras funcionalidades.

- **Editor de texto**: es la parte que nos permite escribir el código fuente del programa, ofrece funciones propias de la edición como copiar, cortar, pegar o buscar. Además, es capaz de reconocer, resaltar y cambiar los colores de las variables, las cadenas de caracteres, las palabras reservadas, las instrucciones, el inicio y el fin de los corchetes, de esta manera el código fuente será mucho más visual, cómodo y se podrán reconocer los errores a simple vista.
- **Compilador**: es el encargado de traducir el código fuente tecleado por el programador y escrito en un lenguaje de alto nivel, a un programa escrito en lenguaje de bajo nivel llamado lenguaje máquina, capaz de ser interpretado y ejecutado por el ordenador. El proceso de traducción se conoce como compilación.



UT2- INSTALACIÓN Y USO DE ENTORNO DE DESARROLLO

Entornos de desarrollo (1º DAW)

Dpto. INFORMÁTICA



Curso: 2022-23

- **Intérprete o interpretador:** como ya vimos, los intérpretes se diferencian de los compiladores en que solo realizan la traducción a medida que se va ejecutando la instrucción. Normalmente no guardan el resultado de dicha traducción. Son más lentos que los compiladores debido a la necesidad de traducir el programa mientras se ejecuta, pero a cambio son más flexibles como entornos de programación y depuración y permiten ofrecer al programa interpretado un entorno no dependiente de la máquina donde se ejecuta el intérprete, sino del propio intérprete (lo que se conoce comúnmente como máquina virtual).
- **Depurador (*Debugger*):** es el encargado de depurar y limpiar los errores en el código fuente de un programa. El depurador permite examinar paso a paso, instrucción a instrucción, la ejecución de un programa y examinar las distintas situaciones o cambios que se produzcan en las variables del programa o en los registros del procesador. El depurador va a permitir detener el programa en cualquier punto de ruptura para examinar la ejecución.
- **Constructor de interfaz gráfica:** esta herramienta de programación simplifica la creación de interfaces gráficas de usuario permitiendo al diseñador colocar los controles (botones, listas, menús y demás elementos) utilizando un editor *WYSIWYG* (es el acrónimo de *What You See Is What You Get*, en español, "*lo que ves es lo que obtienes*") de arrastrar y soltar. Algunos IDE incorporan estas herramientas con el plugin correspondiente.
- **Control de versiones:** estas aplicaciones permiten controlar los cambios que se realizan sobre las aplicaciones, de esta manera se obtendrán revisiones y versiones de las aplicaciones en un momento dado de su desarrollo.
- **Panel de configuración:** Todo entorno de desarrollo dispone de un panel en el que cambiar la configuración del mismo, podemos modificar el aspecto visual, conexiones de red, asignación de teclas, entre muchas otras funciones.
- **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.


5. ENTORNOS DE DESARROLLO ACTUALES

5.1 Xcode

Xcode es la herramienta para realizar aplicaciones (app) para dispositivos Apple. Con esta herramienta, podrán realizarse aplicaciones nativas para iOS y OSx.

Si desea descargarse una versión antes de que se encuentre disponible para todo el mundo, hay que hacerse desarrollador de Apple.

Actualmente, no cuesta nada darse de alta como desarrollador, es gratuito, lo que cuesta es subir una aplicación a la App Store.

 <p>CIFP VIRGEN DE GRACIA</p>	<p>UT2- INSTALACIÓN Y USO DE ENTORNO DE DESARROLLO</p> <p>Entornos de desarrollo (1º DAW)</p>	<p>Dpto. INFORMÁTICA</p>  <p>Curso: 2022-23</p>
--	---	---

Con las nuevas versiones, ya puede programarse en **Swift**, mientras que, con las versiones anteriores, solamente puede programarse con Objective C. que es un lenguaje parecido a Java/C/C++. Pero con una sintaxis algo diferente. Muy potente y orientado a objetos.

5.2 NetBeans

NetBeans está escrita en Java, lo que la convierte en una plataforma disponible para un gran número de sistemas operativos (Windows, Linux, Mac OS X). Se creó para desarrollar aplicaciones en Java, pero también puede programarse con ella en Python, PHP, HTML5 y C/C++.

Es *open source* lo que hace que muchos programadores se decanten por este IDE. De hecho, cuando sale una nueva versión al mercado, suele estar bastante probada.

Se base en el modularidad. Todas las funciones las realizan módulos, los cuales pueden ir añadiéndose según necesidades del programador. De hecho, cuando se descarga, tiene todos los módulos de Java incluidos por defecto. Muchas herramientas están basadas en NetBeans como Sun Studio, Sun Java Studio Creator y otros más.

Contiene una herramienta para crear interfaces de usuario (llamada al comienzo *Matisse*). *Estas herramientas permiten crear aplicaciones basadas en las librerías Swing.*

En el editor, puede programarse también en JavaScript, Ajax y CSS.

5.3 Eclipse

Es un IDE de código abierto. Es una plataforma potente con un buen editor, depurador y compilador. El JDT (java development toolkit) es de los mejores que existen en el mercado y tiene detrás una gran comunidad de usuarios que van añadiendo mejores al software.

Fue desarrollado por IBM como evolución de su VisualAge, pero ahora lo mantiene la función Eclipse, que es independientes y sin ánimo de lucro.

5.4 Visual Studio Code.

Es un potente editor de código disponible para diferentes plataformas. Lanzado en 2015 por Microsoft, VS Code está optimizado para construir y depurar aplicaciones web. Además, cuenta con soporte, nativo y por medios de complementos, para un gran número de lenguajes de programación, como JavaScript, TypeScript, Node.js, C++, C#, Java, Python, PHP, Go. .NET, entre otros.

5.5 IntelliJ:

IntelliJ: Es un entorno de desarrollo pensado para Java, pero que soporta muchos otros lenguajes. Se caracteriza por su versatilidad, rapidez y sencillez de uso

haciendo que otras soluciones como Netbeans o Eclipse parezcan obsoletas para muchos programadores y desarrolladores.

6. ENTORNOS DE DESARROLLO ONLINE

Los entornos de desarrollo online están extendiéndose cada vez más. Pues a la desventaja de la potencia, poseen muchas otras ventajas como el trabajo colaborativo, los repositorios comunes, el poder trabajar con cualquier dispositivo, etc.

Estas ventajas hacen que muchos desarrolladores y empresas de desarrollo opten por entornos en la nube.

Veamos como funciona dos IDE en la nube:

- Cloud 9
- CodAnywhere

7. ENTORNOS INTEGRADOS LIBRES Y PROPIETARIOS

7.1 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:

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.

7.2 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.

IDE	Lenguajes que soporta	Sistema Operative
Microsoft Visual Studio.	Basic, C/C++, C#.	Windows.
FlashBuilder.	ActionScript.	Windows, Mac OS X.
C++ Builder.	C/C++.	Windows.
JBuilder.	Java.	Windows.
Xcode.	C/C++, Java.	Mac OS X.

Sin embargo, hay otros Entornos de Desarrollo, con **intelliJ-IDEA** que tienen una versión comercial (propietaria) y otra de código abierto (libre).

En grandes organizaciones el IDE y por tanto el tipo de licencia que se elija para el desarrollo de un proyecto es una cuestión de importante a decidir.

Esta decisión la tomarán los supervisores del proyecto y la dirección de la empresa.

 <p>CIFP VIRGEN DE GRACIA</p>	<p>UT2- INSTALACIÓN Y USO DE ENTORNO DE DESARROLLO</p> <p>Entornos de desarrollo (1º DAW)</p>	<p>Dpto. INFORMÁTICA</p>  <p>Curso: 2022-23</p>
--	---	---

En el siguiente enlace encontrarás un documento muy interesante, en inglés, donde se detallan todos los entornos de desarrollo existentes en la actualidad con todas sus características: licencias, sistemas operativos donde pueden ser instalados y configurados, lenguajes que soporta, desarrolladores y última versión estable.

8. INSTALACIÓN DE UN ENTORNO DE DESARROLLO

8.1 El compilador de Java

El compilador de Java, también llamado javac, se encapsula dentro de un paquete de desarrollo que se llama JDK (*Java development Kit*).

Para programar en Java, se necesita el compilador y, por lo tanto, habrá que instalar un JDK en la máquina donde vaya a desarrollarse.

Para ejecutar los programas desarrollados en Java, el sistema donde se ejecute deberá tener un JRE, (Java runtime environment- entorno de ejecución Java), el cual contendrá una JVM (Java virtual Machine).

Java es multiplataforma, por lo tanto, no hay que compilar cada programa para cada sistema operativo, ya que, cuando se compila un programa, funcionará en cualquier sistema siempre y cuando tenga instalada la JVM correspondiente.

Téngase en cuenta que cada sistema operativo tendrá una JVM diferente.

- ¿Cómo sé si ya está instalado el JVM?



```
$ java -version
java version "1.8.0_91"
Java(TM) SE Runtime Environment (build 1.8.0_91-b15)
Java HotSpot(TM) Client VM (build 25.91-b15, mixed mode)
```

- ¿Cómo sé si ya está instalado el JDK?

```
$ javac -version
javac 1.8.0_05
```

- ¿Qué hay que hacer para instalar el JRE y el JDK?
- En Ubuntu, existe una versión de JRE y JDK en los repositorios. La instalación sería:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install default-jre
$ sudo apt-get install default-jdk
```


 <p>CIFP VIRGEN DE GRACIA</p>	<p>UT2- INSTALACIÓN Y USO DE ENTORNO DE DESARROLLO</p> <p>Entornos de desarrollo (1º DAW)</p>	<p>Dpto. INFORMÁTICA</p>  <p>Curso: 2022-23</p>
--	---	---

En Windows y en Ubuntu se pueden instalar descargando el JDK 17 o superior para el software que se va a utilizar.

8.2 Depurar un programa

Ningún programa suele funcionar a la primera ni será tal y como se diseñó en un primer momento. Siempre hay que depurar algunos fallos o simplemente verificar que lo que está haciendo lo hace de forma correcta.

8.3 Profiler. Análisis de aplicaciones

Muchas veces, cuando una aplicación está completamente desarrollada o en periodo de pruebas, es preciso analizar su rendimiento. NetBeans proporciona una herramienta para monitorizar los hilos de ejecución, el rendimiento de la CPU, el uso de memoria, etc.

8.4 Generación automática de documentación

Las aplicaciones o programas tienen que estar perfectamente documentados, pues, de lo contrario, sería muy difícil mantener el código. En Java, la documentación del código se escribe dentro del propio lenguaje, lo cual es verdaderamente útil. Java, además, tiene una herramienta que se llama Javadoc que extrae los textos y comentarios del código fuente y los transforma en páginas web (html).

8.5 Gestión de módulos

Los entornos como NetBeans aumentan su potencia gracias a la gestión de módulos o plugins.

Con esto módulos, pueden crearse informes, trabajar con otros lenguajes de programación que no sean Java, etc.