CENTRE ESPECÍFIC
D´EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA

UD 02

INSTALACIÓN Y USO DE ENTORNOS IDE

ENTORNOS DE DESARROLLO 20/21
CFGS DAW

PARTE 1 DE 3: INTEGRATED DEVELOPMENT ENVIRONMENT (IDE)

Autor: Sergio Badal

sergio.badal@ceedcv.es

Fecha: 15-10-2020

1. ¿QUÉ ES UN ENTORNO DE DESARROLLO INTEGRADO (IDE)?

Se entiende por **Entorno de Desarrollo Integrado** o Integrated Development Environment (IDE) "~~un programa informático~~ una aplicación informática que tiene el objetivo de asistir al programador en la tarea de ~~diseñar y~~ codificar un software mediante la inclusión de múltiples herramientas destinadas para dicha tarea" (Casado, 2012:46). En definitiva, es ~~un programa~~ una aplicación informática que ayuda a programar.

Los elementos básicos de un IDE son:

- Un **editor de código**: para escribir el código fuente
- Un **compilador**: para generar el código objeto y el ejecutable de lenguajes no interpretados
- Un **intérprete**: para traducir el código
- Un **depurador**: que ayuda a corregir errores

Sus características principales son

- Pueden ser para uno o más lenguajes de programación
- Ayudan en la visualización del código fuente
- Permite moverse rápidamente entre los ficheros de la aplicación

Usar IDE, por tanto, supone una ayuda a la programación. Sin embargo, como se debe acostumbrarse a su funcionamiento acaban generando dependencia. Además, consumen más recursos y algunos son de pago.

Hoy en día existen editores de texto que realizan dichas funciones básicas (Notepad++, Texpad, KLite…). Todos ellos identifican las palabras reservadas y los elementos clave coloreándolos.

2. ¿POR QUÉ USAR UN IDE?

Sin embargo, un IDE ofrece esto y mucho más. Principalmente, el **autocompletado de código**: cada vez que comenzamos a escribir una palabra reservada aparece una ayuda contextual indicando las opciones con las que seguir. También permite crear estructuras de clases o instrucciones de bucles automáticamente. Además, ofrecen herramientas para refactorizar, para ejecutar depurando más opciones que ayudan a la programación y que hacen más corto el ciclo de vida del software.

Otra ventaja de usar IDEs es que son "altamente configurables". Esto quiere decir que cada programador puede configurarlo como más le agrade: con más o menos barras de herramientas, con atajos de teclado, con comandos personalizados, con distintas ubicaciones física de los distintos paneles, etc. Pero también permite configurar la depuración y la compilación de proyectos, convirtiéndose así en una herramienta cómoda y útil.

Además, como editores de programas que son, permiten la virtualización de la ejecución del código, de modo que se puede probar la funcionalidad del programa sin tener que desplegarla realmente.

Finalmente, los IDE permiten el **desarrollo colaborativo**. Esto es, desarrollar el software de manera descentralizada y distribuida, a través de la integración con el control de versiones. Así, varios programadores pueden trabajar sobre el mismo código mejorándolo, tal y como ocurre en el software libre. El control de versiones se lleva a cabo creando repositorios para que los clientes puedan descargar y subir código. Así, se trabaja sobre la versión anterior, y no sobre el proyecto inicial que se ha subido al repositorio. Los IDE permiten un fácil manejo de las versiones y una rápida sincronización.

En la siguiente figura podemos ver algunas de las funciones más destacadas:

Depurador	•Analiza elementos del programa y revisa errores
GUI	•(Graphical User Interface) Crea ventanas, botones...
Control de versiones	•Guarda el código fuente sin perder versiones anteriores
Plantillas	•Crea programas en base a uno creado como modelo
Prueba	•Prueba programas
Multiplataforma	•Uso en distintos sistemas operativos
Multilenguaje	•Uso con distintos lenguajes de programación
Libre/pago	•Permite la opción de uso pagando/sin pagar
Plugins	•Instala nuevas funciones
Bases de datos	•Iinteractúa con bases de datos
Tomcat	•Puede incluir el servidor de aplicaciones web interactivas Tomcat
UML	•Realizar gráficos con el lenguaje de modelado UML
Documentación	•Documentar el código
Refactorización	•Optimiza el código
Formateo de código	•Organiza el código fuente para que sea más legible

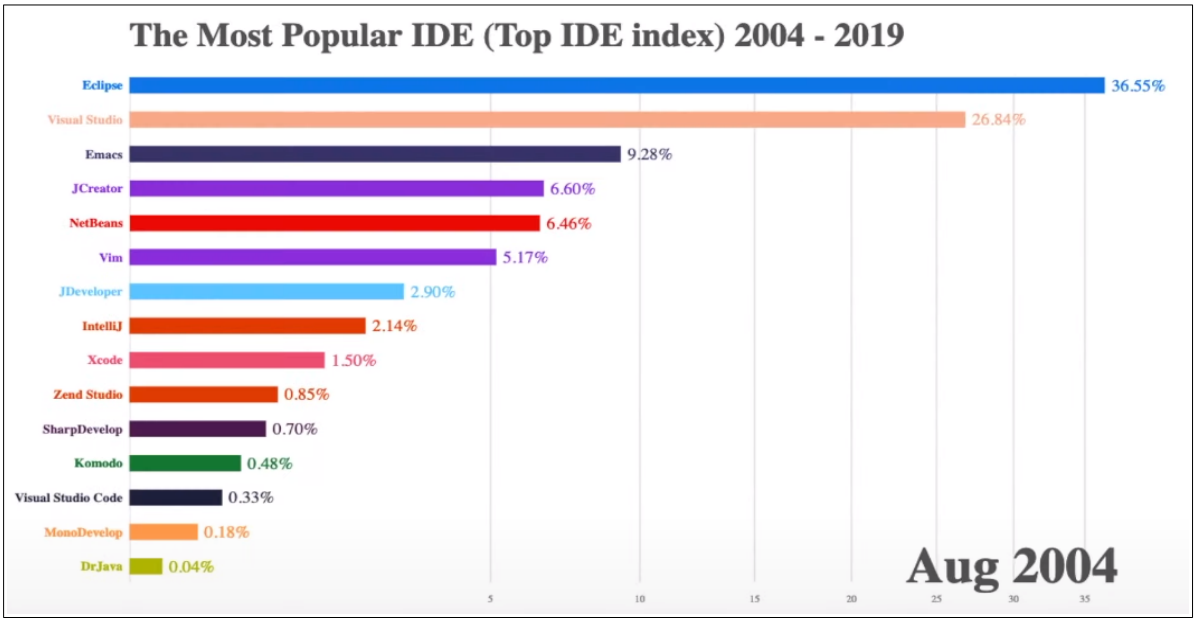
A la hora de elegir un IDE debemos tener en cuenta varias cosas:

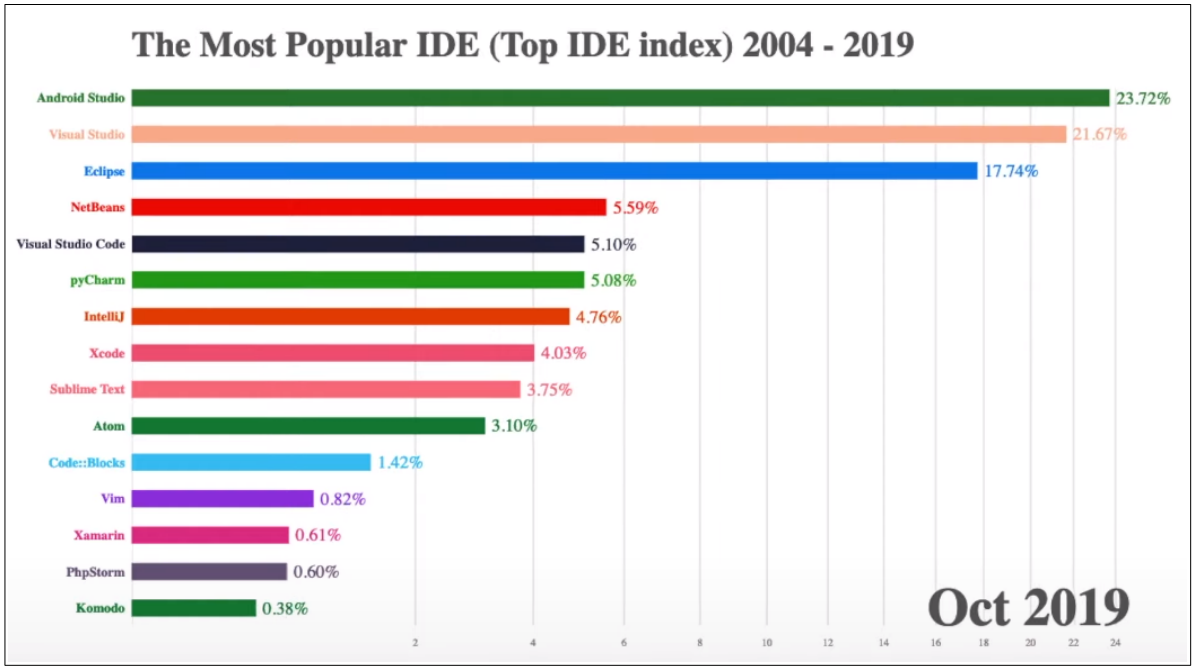
- **Sistema operativo**: Hoy en día la gran mayoría de IDEs son multiplataforma.
- **Lenguaje de programación**: un IDE puede soportar uno o varios lenguajes de programación.
- **framework**: la plataforma de trabajo también puede variar (por ejemplo, si vamos a desarrollar con Visual Basic con la plataforma .NET en Linux, deberíamos usar Mono Develop en vez de VB).
- **Herramientas**: también se elige un IDE por el surtido de herramientas que posee. No todos los IDE tienen las mismas funciones.

3. EVOLUCIÓN DE LOS IDE

Existen cientos de IDE casi todos especializados en uno o varios lenguajes. Para Java, los más usados son NetBeans y Eclipse (los que veremos en ese módulo) pero existen muchos más. En este vídeo puedes ver su evolución:

- https://www.youtube.com/watch?v=3CwJW5_C6R4





4. BIBLIOGRAFÍA

i. Escobedo, J. G. (2012, 20 agosto). *Depuración de programas con NetBeans*. Sitio Web de Javier García Escobedo (javiergarciaescobedo.es). https://javiergarciaescobedo.es/programacion-en-java/28-programacion-estructurada/114-depuracion-de-programas-con-netbeans

ii. Iglesias, C. C. (2020). Entornos de Desarrollo (GRADO SUPERIOR). RA-MA S.A. Editorial y Publicaciones.

iii. Aldarias, F. (2012): Apuntes de Entornos de Desarrollo, CEEDCV