

# UD2 Instalación y uso de entornos de desarrollo

---

## Índice

- 1. Introducción a los entornos de desarrollo
  - EDI libres y propietarios
- 2. Componentes de un entorno de desarrollo
- 3. Instalación y uso de entornos de desarrollo.
- 4. Módulos.

## 1. Introducción a los entornos de desarrollo

Un **Entorno de Desarrollo Integrado** (EDI o IDE en inglés, Integrated Development Environment), es un tipo de software compuesto por un conjunto de herramientas de programación. El principal objetivo de los Entornos de Desarrollo es optimizar el proceso de programación.

Las principales tareas que nos facilita el uso de un EDI son:

- La ejecución en modo depuración, para corregir errores.
- La ejecución automática de pruebas.
- La documentación de forma sencilla.
- La refactorización u optimización de código.
- La integración con plataformas de control de versiones.

En la actividad 4 del tema anterior hemos visto el proceso para crear y ejecutar un programa en Java sin usar un EDI.

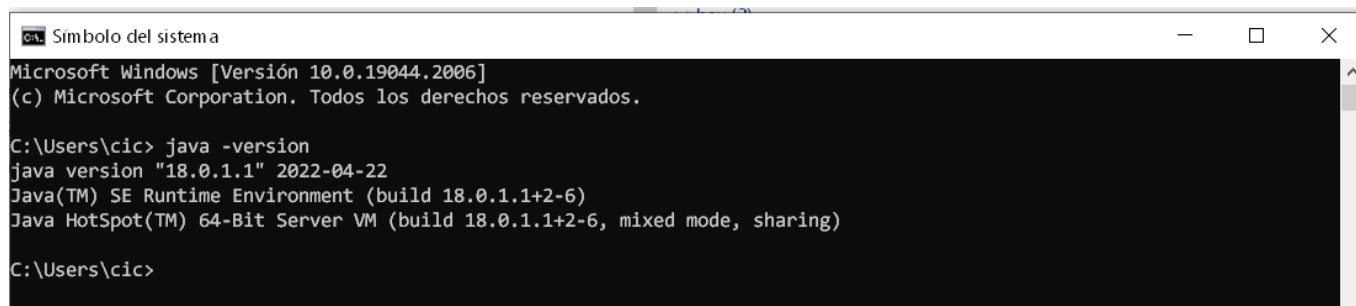
Para crear y ejecutar programas en Java necesitamos tener instalado el JDK y el JRE, usemos o no EDI.

- El **JDK (Java Development Kit)** es un software que provee herramientas de desarrollo para la creación de programas en Java.
- El **JRE (Java Runtime Environment)** es un conjunto de utilidades que permite la ejecución de programas Java. Está conformado por una Máquina Virtual de Java o JVM, un conjunto de bibliotecas y otros componentes necesarios para que una aplicación escrita en lenguaje Java pueda ser ejecutada. El JRE actúa como un "intermediario" entre el sistema operativo y Java.

El JDK y el JRE se instalan conjuntamente desde el siguiente enlace:

<https://www.oracle.com/java/technologies/javase/jdk18-archive-downloads.html>. En este caso instalaríamos el JDK 18, que es la última versión.

Para comprobar la versión del JDK y el JRE que tienes instalada en Windows debes acceder al CMD y ejecutar el comando `java -version`



```
Microsoft Windows [Versión 10.0.19044.2006]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\cic> java -version
java version "18.0.1.1" 2022-04-22
Java(TM) SE Runtime Environment (build 18.0.1.1+2-6)
Java HotSpot(TM) 64-Bit Server VM (build 18.0.1.1+2-6, mixed mode, sharing)

C:\Users\cic>
```

Normalmente, un IDE está dedicado a un determinado lenguaje de programación, aunque las últimas versiones de los IDE tienden a ser compatibles con varios lenguajes mediante la instalación de plugins adicionales.

## EDI libres y propietarios

Al crear un EDI los equipos de desarrollo pueden conceder u otorgar distintos tipos de licencias, pueden sólo autorizar su uso, su modificación, instalación, etc...

Según su licencia de uso podemos clasificar los entornos de desarrollo en dos grupos, libres y propietarios.

- **EDI libres:** son aquellos con licencia de uso público. El autor deja libertad a los usuarios, por lo que el programa puede ser usado, copiado, estudiado, modificado y redistribuido libremente sin ser obligatoriamente gratis. Podemos encontrarnos programas bajo esta licencia que son de pago, aunque suelen ser muy económicos.
- **EDI propietarios:** Necesitan licencia para ser usado, copiado, estudiado, modificado o redistribuido, es lo contrario a software libre.

### Actividad 1

## 2. Componentes de un entorno de desarrollo

Generalizando, IDE se compone de las siguientes herramientas:

- **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 y algunos EDI incluyen funciones de compilación y ejecución propias, sin necesidad de un JDK.
- **Depurador:** Botón de ejecución y traza, puntos de ruptura y seguimiento de variables.
- **Interfaz gráfica:** Normalmente es una paleta de componentes que me personalizar los componentes gráficos a un panel en el que vamos a trabajar. Puede acceder a innumerables bibliotecas y plugins aumentando las opciones de nuestros programas.
- **Control de versiones:** permite almacenar cambios en nuestro código y compartirlo con otros desarrolladores.

## 3. Instalación y uso de entornos de desarrollo.

Vamos a ver de forma práctica la instalación y uso de dos entornos de desarrollo, **NetBeans** y **Eclipse**.

### Actividad 2: Instalación y uso de NetBeans

## 4. Módulos.

Un módulo es un componente software que contiene clases de Java que pueden interactuar con las API del entorno de desarrollo y el manifest file, que es un archivo especial que lo identifica como módulo.

Normalmente se conoce como plugin a un conjunto de módulos.

¿Crees que la posibilidad de poder añadir módulos y plugins a un IDE contribuye a su éxito?

Actividad 4: instalación de módulos en NetBeans

Actividad 5: instalación de módulos en y Eclipse