1. **TEMA 2**
   1. **Entornos de desarrollo, IDE (Integrated Development Environment)**

Un entorno de desarrollo integrado, IDE, es un programa informático compuesto por un conjunto de herramientas que permiten completar todo el proceso de elaboración de una aplicación.

Un IDE cuenta con los siguientes elementos:

* un editor de código
* un compilador y/o intérprete
* un depurador (debugger).
* un conjunto de herramientas de ayuda para el desarrollo de software y su distribución.

Hay IDE que sólo permiten el desarrollo de aplicaciones para un determinado lenguaje de programación. Este sería el caso del **JDK** que proporciona las herramientas necesarias para la creación y distribución de aplicaciones JAVA (en realidad el JDK no es un IDE propiamente dicho) o del IDE **AndroidStudio** que permite la creación y distribución de app para sistemas Android.

Pero muchos IDE permiten hacer el desarrollo de aplicaciones para diferentes lenguajes. Este es el caso de los IDE **Eclipse** o **NetBeans**.

**Editor de código**. El editor de código de un IDE no es más que un editor de texto avanzado, en cuanto a que no sólo incluye las herramientas habituales de edición como puede ser copiar, pegar o buscar, sino que incluye otras orientadas a la asistencia en las labores de programación en uno o varios lenguajes de programación concretos.

Estas herramientas especializadas en la generación de código fuente, pueden entre otras, reconocer las instrucciones procedimentales, así como variables y métodos o funciones definidas, en función de ese reconocimiento son capaces de resaltar e indizar los distintos elementos para facilitar su lectura y depuración.

**Compilador e Intérprete**. El **compilador** es el encargado de traducir el código fuente creado por el programador en un lenguaje de alto nivel al lenguaje máquina que será capaz de interpretar la máquina.

En el caso del **interprete**, dicha traducción se realiza a medida que se ejecutan las instrucciones del programa, esto hace que sean más lentos en su ejecución, dado que precisan una traducción al tiempo que se ejecutan las instrucciones programadas.

**Debugger**. El **depurador o debugger**, es la herramienta encargada de depurar y limpiar los errores en el código del programa informático. El depurador permite examinar paso a paso, instrucción a instrucción, la ejecución del programa, así como examinar las distintas situaciones y cambios que se producen en las variables del programa o en los registros utilizados en su ejecución.

Otras herramientas de ayuda para el desarrollo del software, como puede ser el **Constructor de interface gráfico**, **Herramientas de diseño de diagramas**, etc.

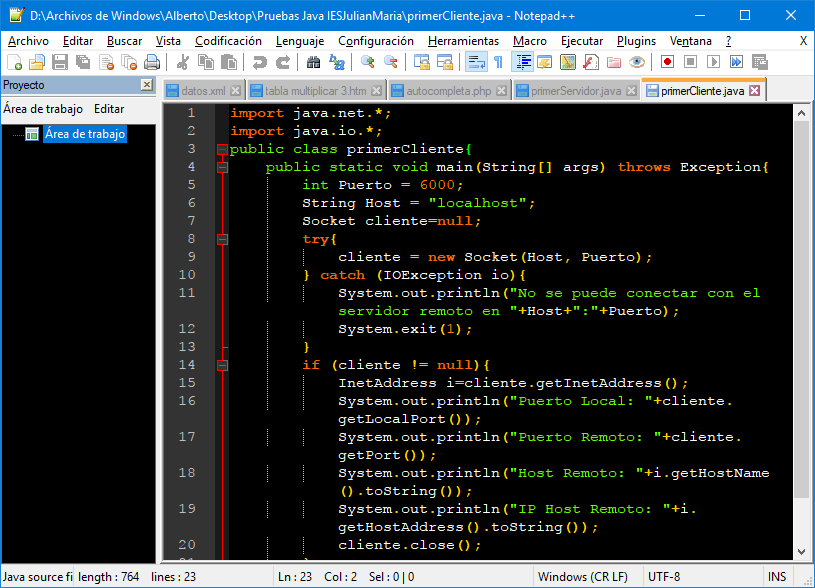
**Control de versiones**. Son herramientas que permiten controlar los cambios realizados sobre las aplicaciones, lo cual es muy ventajoso a la hora de determinar ciertos tipos de errores que pueden sobrevenir en el proceso de creación del software.

* 1. **Editores avanzados**

Existen editores avanzados con un enfoque claro hacia el desarrollo del software, son herramientas de edición de código que, a veces, logran incluso superar a las herramientas de edición de los propios IDE. Muchos de dichos editores son gratuitos o incluso libres u open source.

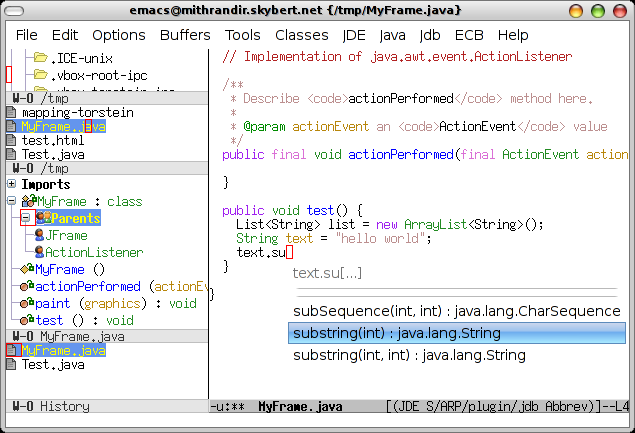
**Notepad++**: se trata de un editor de texto que nace como alternativa mejorada del notepad de Windows, sin embargo, en su evolución terminó por enfocarse en la edición de código fuente, incluyendo, actualmente, el reconocimiento de la mayor parte de lenguajes actuales, tanto de alto como, incluso, de bajo nivel.

Actualmente es uno de los editores más utilizado en entornos Windows, se trata de software libre bajo la licencia GPL (*GNU Public License*) que permite a los usuarios finales su modificación y distribución, pero impide su apropiación obligando a mantener el tipo de licencia en aquellos desarrollos basados en el presente.

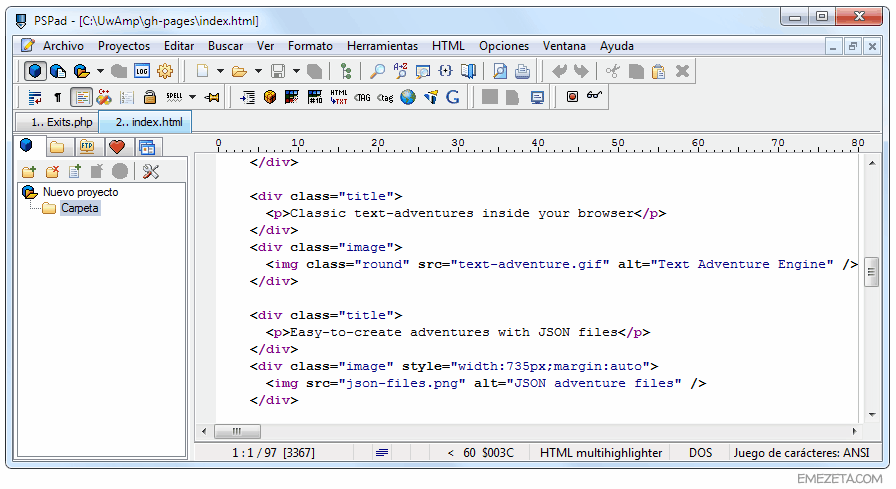


Hay otros editores avanzados con políticas similares a la de Notepad++: Emacs, PSPad, Geany, Sublime Text, Nano, VIM, Komodo Edit, Edit Rocket, etc.

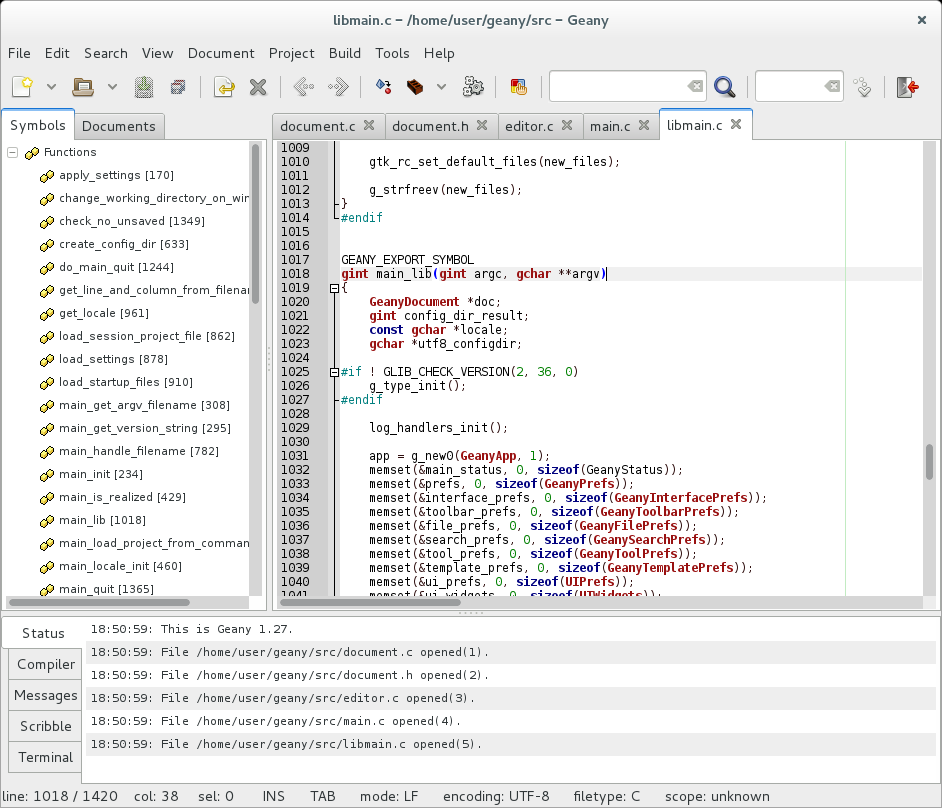
**Emacs**: se trata de un editor de texto avanzado, que nace y se desarrolla con el movimiento GNU a partir de los años 70 y hasta la actualidad, está disponible en múltiples sistemas y es uno de los dos principales contendientes en la guerra de editores del mundo Unix (el otro es el VI), sin embargo, aunque es un editor muy poderoso, ocurre que, como con el VI, su dominio requiere elevado conocimiento, no siendo tan intuitivo como otros editores actuales.



**PSPad**: se trata de un editor de texto con ciertas similitudes al Notepad++, también enfocado en el mundo Windows y con orientación hacia la edición de código fuente. Sin embargo en este caso se trata de software Freeware con una licencia más laxa que la GPL.

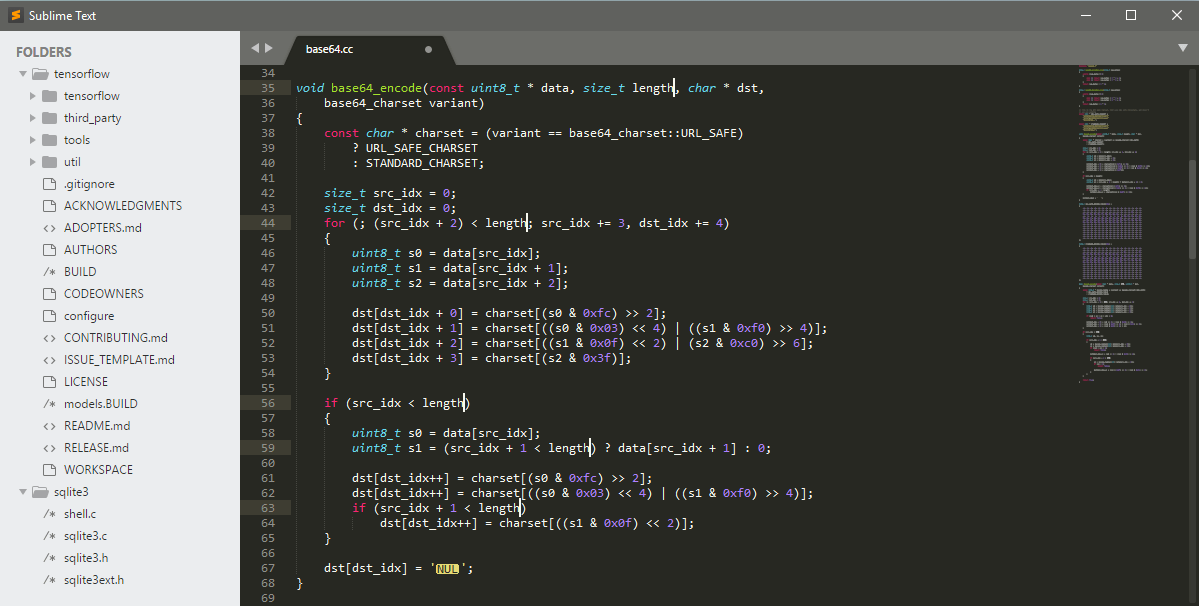


**Geany**: se trata de un editor de texto con las características de un IDE ligero. Se encuentra disponible para las plataformas más utilizadas (Linux, MacOS, BSD, Solaris y Windows) y con soporte para múltiples lenguajes de programación. Se distribuye como software libre bajo la licencia GPL.

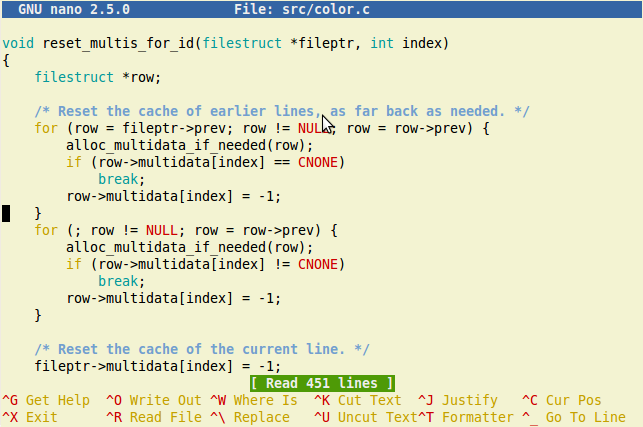


**Sublime text**: se trata de un software comercial, que sin embargo puede descargarse de forma gratuita en versión de evaluación sin fecha de caducidad, aunque se distribuye con la condición de adquisición de licencia para uso continuado. Se puede encontrar en las plataformas Linux, MacOS y Windows, y tiene soporte para la mayoría de lenguajes de programación actuales.

Dispone de algunas características interesantes como es la múltiple selección y edicción.



**Nano**: se trata de un editor con licencia GPL incluido en la mayoría de distribuciones de Linux, es un editor de línea de comandos, pero con características de editor avanzado.



* 1. **JDK**

Para poder ejecutar aplicaciones JAVA en un ordenador en él debe instalarse la **Máquina Virtual JAVA** (JVM).

Oracle proporciona el denominado **Java Runtime Environment** (JRE) que incluye la máquina virtual JAVA.

Hay diferentes JRE dependiendo de la plataforma (Linux, Solaris, Windows) (ver la página <http://www.oracle.com/technetwork/es/java/javase/downloads/index.html>).

Por otra parte, el **Java Developer`s Kit** (JDK) es una herramienta de desarrollo de aplicaciones JAVA de libre distribución que se puede descargar de la página [www.Oracle.com](http://www.Oracle.com) y que incluye el JRE.

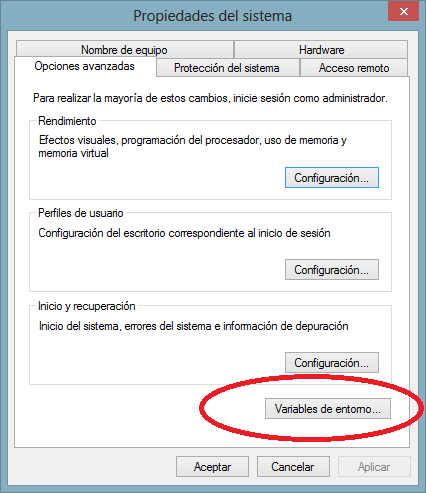
En el caso de las aplicaciones Java, el código fuente, se compila en código ejecutable, pero dicho ejecutable está constituido por un microcódigo que no corresponde al de la CPU del hardware sobre el que se ejecuta, sino al de una maquina virtual java que es preciso se ejecute de forma intermedia y que realice la traducción de dicho microcódigo al código máquina del hardware correspondiente. Esta forma de operar, hace que java sea multiplataforma y, en principio, una aplicación compilada de java, puede ejecutarse en cualquier plataforma que disponga de maquina virtual java, consiguiendo un rendimiento a medio camino entre los lenguajes interpretados y los lenguajes compilados.

Sin embargo, actualmente, el trabajo conjunto entre los desarrolladores de hardware y los desarrolladores de la maquina virtual java (Oracle) consiguen que el rendimiento, sea cada vez más cercano al de una aplicación nativa del hardware. Esto toma especial importancia en los dispositivos móviles, en particular en los smartphones Android.

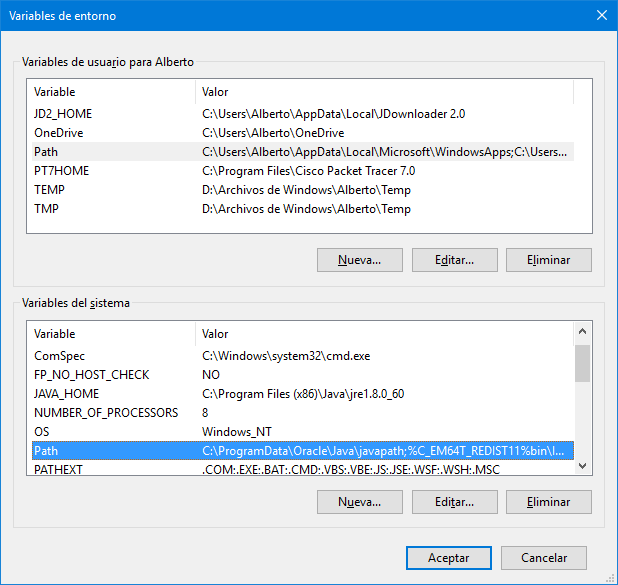
* + 1. **Instalación de JDK (sistemas Windows).**

Una vez descargado el JDK, procedemos a su instalación, ejecutando el archivo instalable. Una vez instalado, puede ser necesario modificar el path del sistema para que las herramientas del JDK estén disponibles desde la línea de comandos, dado que lo más habitual es que en el sistema ya esté instalado un JRE y por tanto las variables de sistema que apuntan al path de java lo hagan al JRE y no al JDK.

La manera más sencilla de establecer variables de entorno es abrir el explorador de Windows, seleccionar Equipo y en el menú contextual elegir Propiedades. Después seleccionar Configuración Avanzada del sistema y por último pinchar en el botón Variables de entorno.



Se pueden definir variables de entorno sólo válidas para el usuario que las define o para cualquier usuario del sistema:



Si JDK está instalado en **C:\Archivos de programa\Java\jdkxxx**, el valor añadido a la variable path debe ser **C:\ Archivos de programa\Java\jdkxxx\bin** ya que es en la carpeta BIN donde se encuentran los programas javac.exe y java.exe que sirven, respectivamente, para compilar y ejecutar las aplicaciones Java

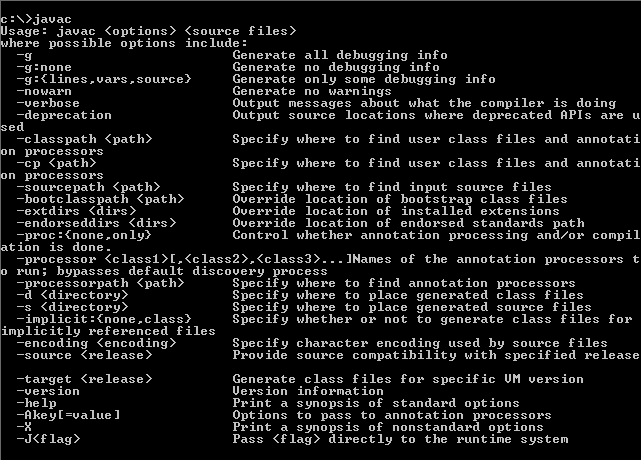
De la misma manera que en el sistema operativo Windows se define y utiliza la variable path para que Windows sepa dónde debe buscar los archivos ejecutables solicitados por el usuario, la JVM utiliza el valor de la variable de entorno Java **classpath** para saber dónde debe buscar los archivos compilados **.class** cuya ejecución solicita el usuario mediante el progranma java.exe .

El valor de la variable classpath (como el de la variable path) es una lista de caminos separados por el caracter ; . Para indicar que busque en el directorio desde el que el usuario pide la ejecución de un programa , en la variable classpath se incluye como camino posible un punto .

* + 1. **Herramientas que incluye el Kit de Desarrollo de Java (Java Development Kit)**

El Kit de desarrollo de Java (JDK) contiene las herramientas y librerías necesarias para crear y ejecutar aplicaciones en Java. De todas las herramientas que proporciona el JDK las más importantes son:

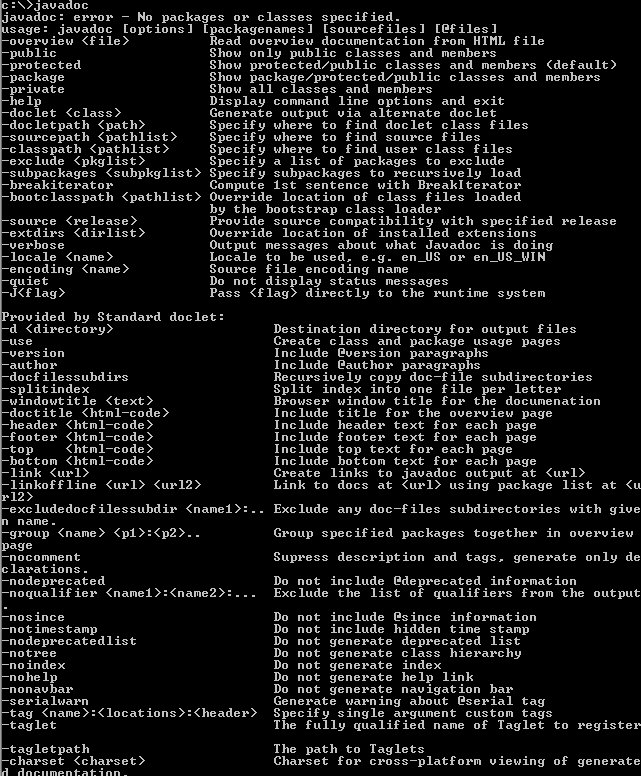
• **javac.exe**: Es el compilador de Java. Se encarga de convertir el código fuente escrito en Java a bytecode (microcódigo).



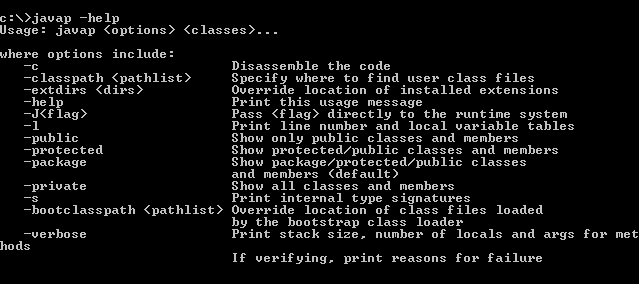
• **java.exe**: Se encarga de importar las librerías necesarias y ejecuta la aplicación en base a la información de los archivos .class creados en la fase de compilación y los archivos .class importados.



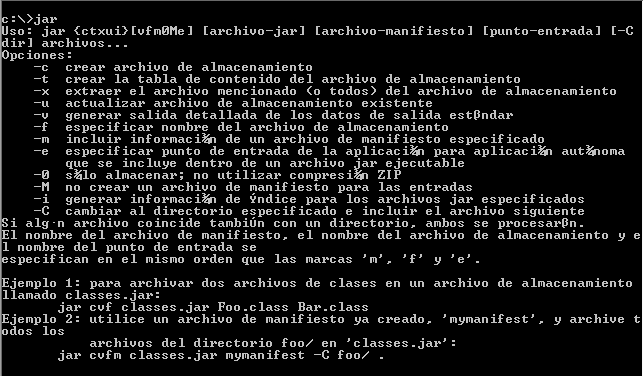
• **javadoc**: Se utiliza para generar la documentación de la aplicación en formato HTML a partir del código fuente Java. Para ello el programador ha debido introducir en el archivo fuente determinadas marcas y/o comentarios.



• **javap**: Es un desensamblador de Java. A partir de un fichero en bytecodes (.class) esta herramienta puede mostrarnos información sobre la clase, sus atributos y sus métodos. Puede ser útil para ayudar a reescribir el código fuente cuando éste se ha perdido y no se cuenta con documentación.



• **jar**: Es una herramienta que permite comprimir y descomprimir paquetes de clases y/o archivos en general.

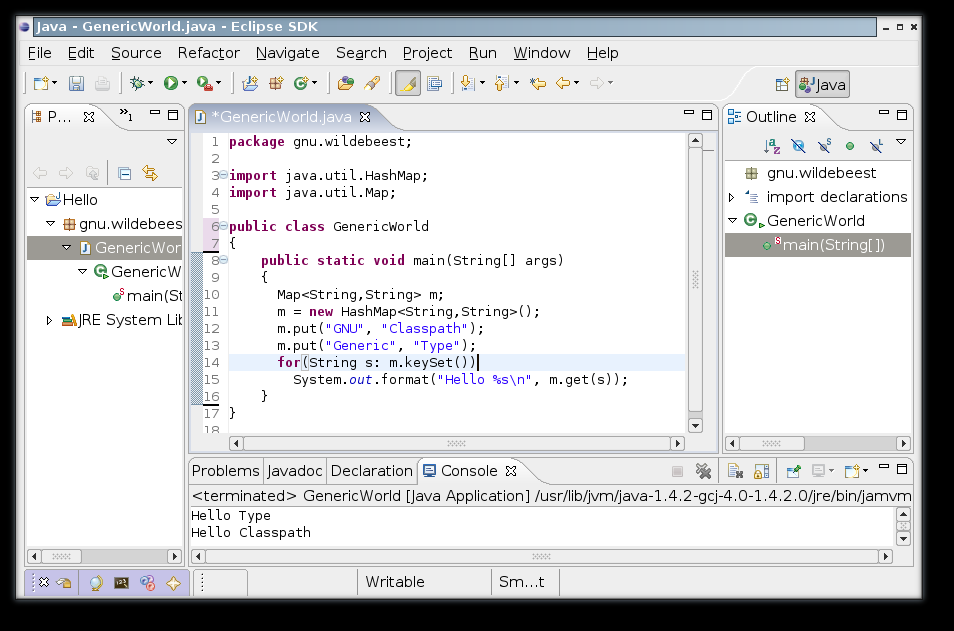


* 1. **Eclipse**

Eclipse es un IDE desarrollado inicialmente para el desarrollo de aplicaciones en el lenguaje java, pero que sin embargo se ha convertido en uno de los IDE multiplataforma y multilenguaje más extendidos actualmente, la comunidad que está detrás de este desarrollo es enorme, pudiéndose encontrar plugins prácticamente para cualquier funcionalidad que podamos pensar.

Eclipse se libera como software libre mediante una licencia propia *Eclipse Public License*, que está aprobada por la Open Source Iniciativa y que dispone de algunas características más laxas que la licencia GPL.

**Instalación de Eclipse**, para la instalación de eclipse basta con realizar la descarga de la versión requerida y descomprimir el archivo comprimido en el disco duro, para a continuación configurar los entornos de trabajo, así como la localización de los distintos ficheros necesarios. En el caso de Windows, actualmente se dispone de un instalador que realiza por nosotros algunas de dichas tareas de configuración.

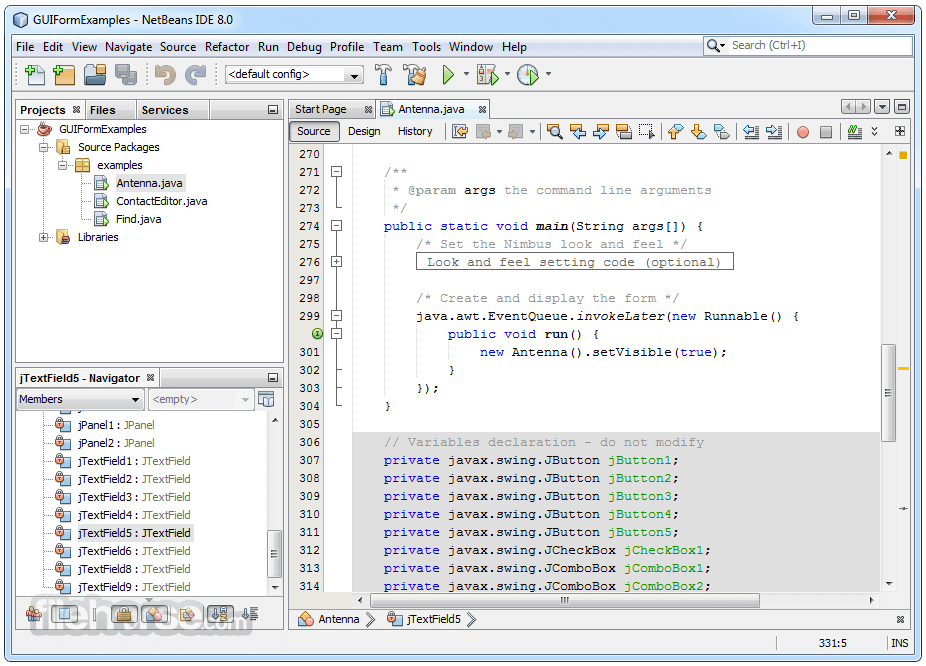


**Instalación de plugins**, desde el menú de ayuda y la opción de instalación de nuevo software se puede realizar la instalación de diferentes plugins como por ejemplo:

* El plugin **WindowsBuilder** que nos proporciona una interface para la creación de interfaces gráficas de una forma visual.
* el plugin **ERMaster** que nos permite realizar diagrams ER desde el propio IDE.
  1. **NetBeans.**

Se trata de un entorno de desarrollo realizado por Sun Microsystems (actualmente propiedad de Oracle Corporation) enfocado en el lenguaje de programación Java y con distribuciones para los principales sistemas operativos actuales. Actualmente NetBeans cuenta con soporte multilenguaje, soportándose los lenguajes de programación más extendidos.

NetBeans se distribuye de forma libre, actualmente, con licencia dual: mediante *Common Development and Distribution License* (CDDL) y mediante GPL2.



**Instalación de NetBeans**, la instalación de NetBeans es realmente sencilla, en el caso de Windows, únicamente es necesario descargar el instalador de la versión requerida y su ejecución para proceder a la instalación del entorno.

**Instalación de plugins**, NetBeans soporta la instalación de plugins, y se pueden encontrar plugins para multiples capacidades extra, como por ejemplo el plugin PlantUML que permite el diseño de diagramas UML desde el propio IDE.

* 1. **Otros IDE**

Hay otros IDE multilenguaje, pero están menos extendido, sin embargo los IDE enfocados a un lenguaje concreto son numerosos, una pequeña muestra:

BlueJ, Android Developer Studio, Microsoft Visual Studio, IBM RAD (Rational Application Developer), JDeveloper, SQLDeveloper, MySQLWorkbench, PHPMyAdmin, etc