

UT02. 02-INSTALACIÓN Y USO DE ENTORNOS DE DESARROLLO

Entornos de Desarrollo

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

4.- Estructura de entornos de desarrollo

Componentes

- **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. Características de refactorización.
- **Depurador:** Botón de ejecución y traza, puntos de ruptura y seguimiento de variables. Opción de depurar en servicios remotos.
- **Generador automático de herramientas:** Para la visualización, creación y manipulación de componentes visuales y todo un arsenal de asistentes y utilidades de gestión y generación código.
- **Interfaz gráfica:** Nos brinda la oportunidad de programar en varios lenguajes con un mismo IDE. Es una interfaz agradable que puede acceder a innumerables bibliotecas y plugins, aumentando las opciones de nuestros programas

Estructura de entornos de desarrollo

- **Editor de texto**

- Escribir el código del programa (**código fuente**), compuesto por caracteres alfanuméricos y caracteres especiales como *, +, -, /, {, }, (,)! ...
- Ofrece funciones para el usuario como cortar, pegar, buscar...
- Además es capaz de reconocer, resaltar y cambiar los colores de las variables, las cadenas de caracteres el inicio y fin de los corchetes... Para que el código fuente sea mucho más **visual, cómodo** y podamos **reconocer errores** a simple vista.

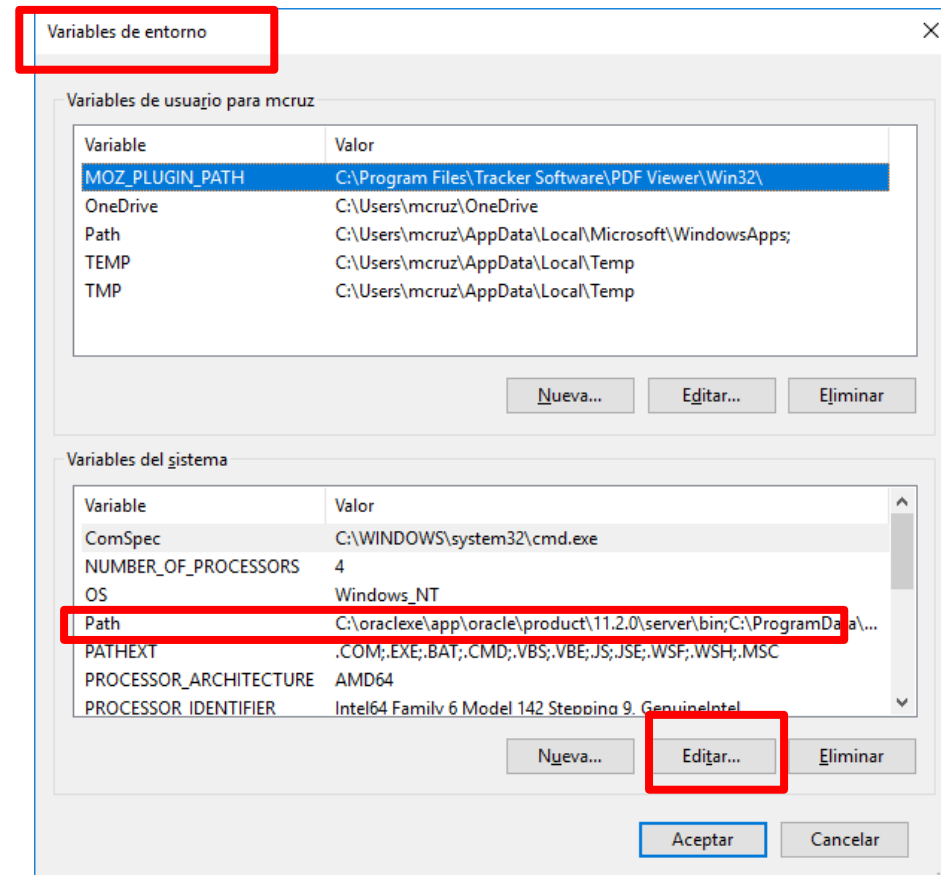
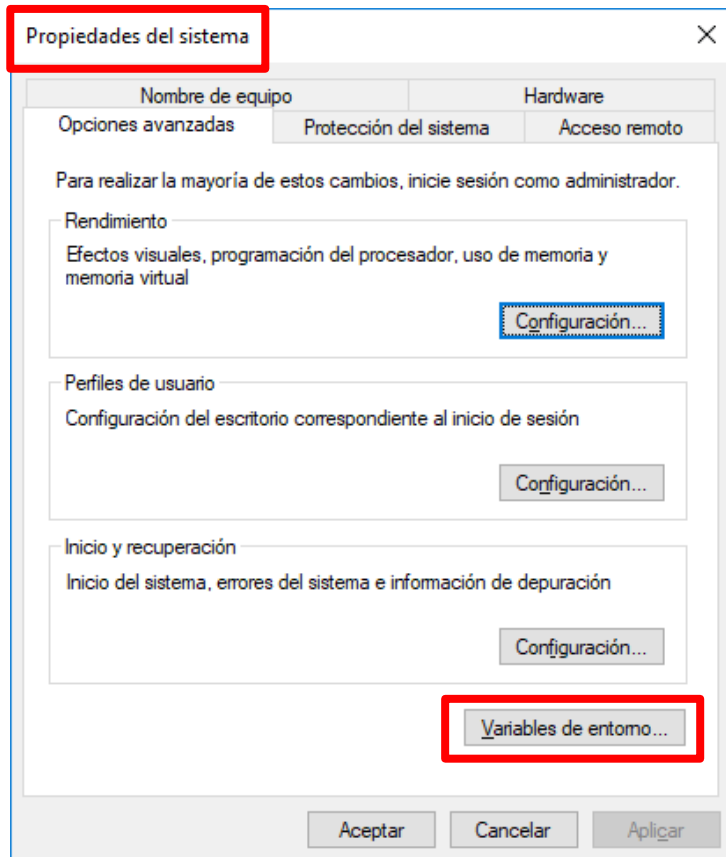
Estructura de entornos de desarrollo.

- **Compilador:**
 - Es el **traductor**
 - Es el encargado de traducir el código fuente, generando un programa equivalente que la máquina será capaz de interpretar.
 - El proceso de traducción se conoce como compilación.

Compilar y Ejecutar en Java

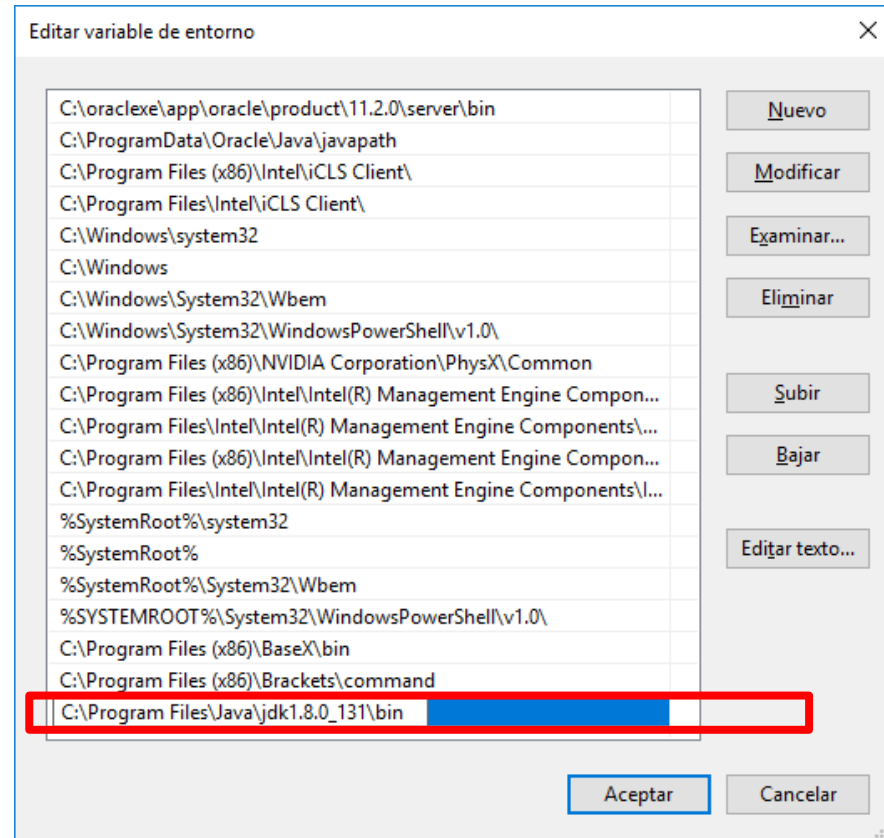
- Añadir la ruta (**classpath de java**)

- Este equipo>>Propiedades>>Configuración avanzada del sistema
- Propiedades del sistema>>Variables de entorno...>>En variables del sistema, seleccionamos Path y Botón Editar...



Compilar y Ejecutar en Java

- En la ventana Editar variable del entorno, pulsar **Nuevo...** y copiar la ruta de la carpeta bin de jdk de Java de mi equipo, por ejemplo **C:\Program Files\Java\jdk1.8.0_131\bin**



Compilar y Ejecutar en Java

- **Powershell/símbolo del sistema**

- Comandos que vamos a utilizar: **javac** para compilar y **java** para ejecutar
- Escribimos un programa en java en el bloc de notas que visualice por pantalla *“Hola, mundo”*.
- Guardamos con el nombre **uno.java**.
- Compilamos `javac uno.java`. Obtenemos el archivo **Uno.class**
- Ejecutamos `java Uno`

```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. Todos los derechos reservados.

PS C:\Users\mcruz> cd "D:\Portatil\Documents\Entornos de desarrollo\UT2\Ejemplos"
PS D:\Portatil\Documents\Entornos de desarrollo\UT2\Ejemplos> javac uno.java
PS D:\Portatil\Documents\Entornos de desarrollo\UT2\Ejemplos> java Uno
Hola, mundo
```


Compilar y Ejecutar en Java

- Si queremos redirigir la salida a un archivo usamos **>archivo .txt**
- Argumentos en el comando
 - Programa que sume visualice la suma de dos números pasados como argumentos. Puedes usar **Integer.parseInt()**

```
PS D:\Portatil\Documents\Entornos de desarrollo\UT2\Ejemplos> java Ejer1 5 6
Suma: 11
PS D:\Portatil\Documents\Entornos de desarrollo\UT2\Ejemplos> ■
```

Estructura de entornos de desarrollo.

- **Intérprete o interpretador:**

- Se encarga de **analizar** y **ejecutar** otros programas, escritos en un lenguaje de alto nivel.
- Los intérpretes se diferencian de los compiladores en que sólo realizan la traducción a medida que sea necesaria (instrucción por instrucción)
- Normalmente no guardan el resultado de dicha traducción.
- Suelen ser más lentos que los compilados debido a la necesidad de traducir el programa mientras se ejecuta
- Son más flexibles como entornos de programación y depuración.
- Ofrecen 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)

Estructura de entornos de desarrollo

- **Depurador (Debugger):**

- Permite depurar o **limpiar los errores** en el código fuente de otro programa informático.
- El depurador permite detener el programa en:
 - 1. Un punto determinado mediante un **punto de ruptura**.
 - 2. Un punto determinado bajo ciertas condiciones mediante un **punto de ruptura condicional**.
 - 3. Un momento determinado cuando se cumplan **ciertas condiciones**.
 - 4. Un momento determinado a **petición del usuario**.

- **Depurador (Debugger):**

- Durante esa interrupción, el usuario puede:
 - 1. Examinar y modificar la memoria y las variables del programa.
 - 2. Examinar el contenido de los registros del procesador.
 - 3. Examinar la pila de llamadas que han desembocado en la situación actual.
 - 4. Cambiar el punto de ejecución, de manera que el programa continúe su ejecución en un punto diferente al punto en el que fue detenido.
 - 5. Ejecutar instrucción a instrucción.
 - 6. Ejecutar partes determinadas del código, como el interior de una función, o el resto de código antes de salir de una función.

Estructura de entornos de desarrollo

- **Constructor de interfaz gráfica:**
 - Herramienta de programación que simplifica la creación de interfaces gráficas de usuario permitiendo al diseñador para organizar los widgets (window gadget) utilizando un editor WYSIWYG de arrastrar y soltar.

5. Instalación de entornos integrados de desarrollo

Instalación de JDK

- Sobre **Windows**

- NetBeans 12

- Requiere la instalación de Open JDK (Java Development Kit -- Kit de Desarrollo de Java).
 - Mirar apuntes **UT02-00-Instalación NetBeans 12**

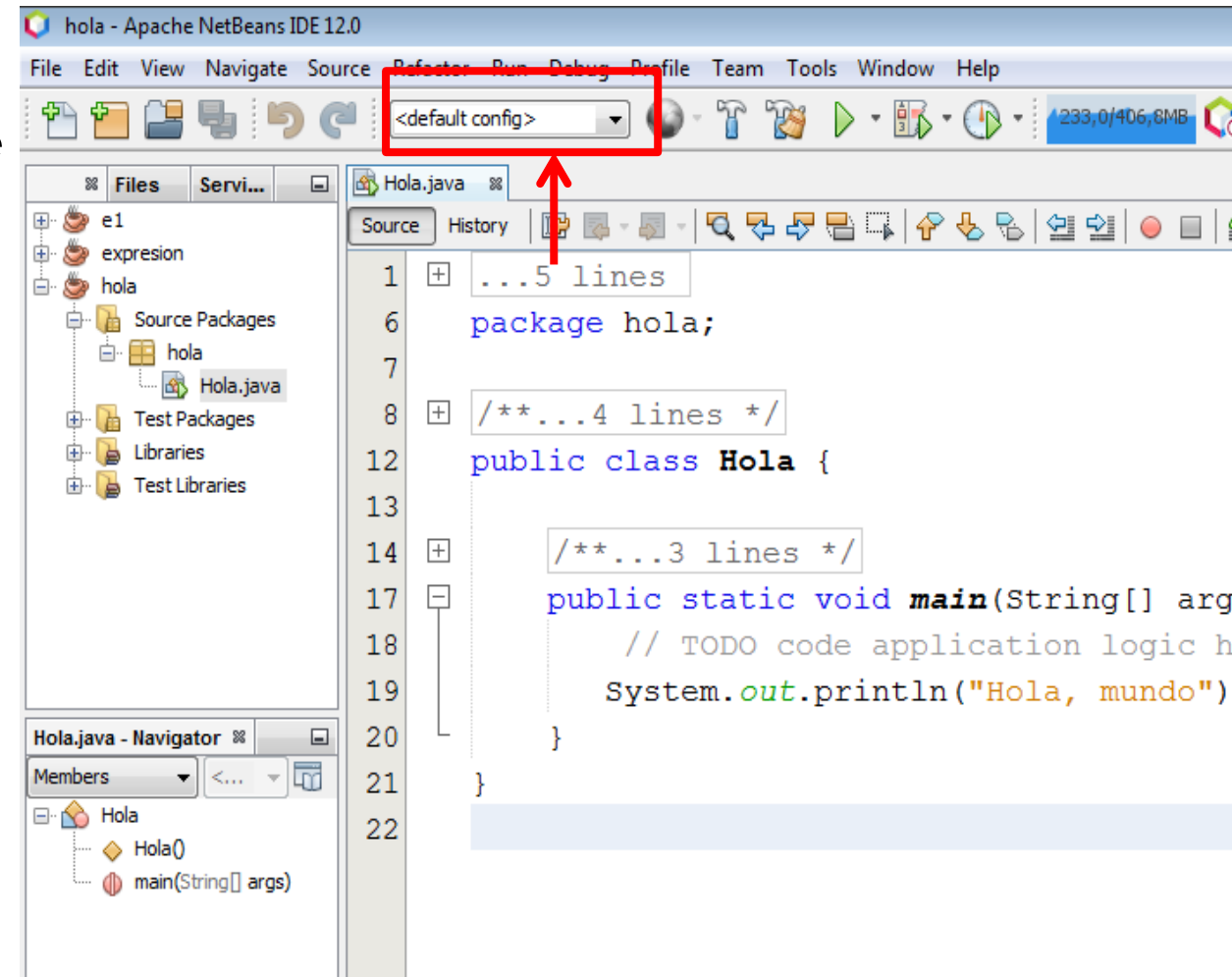
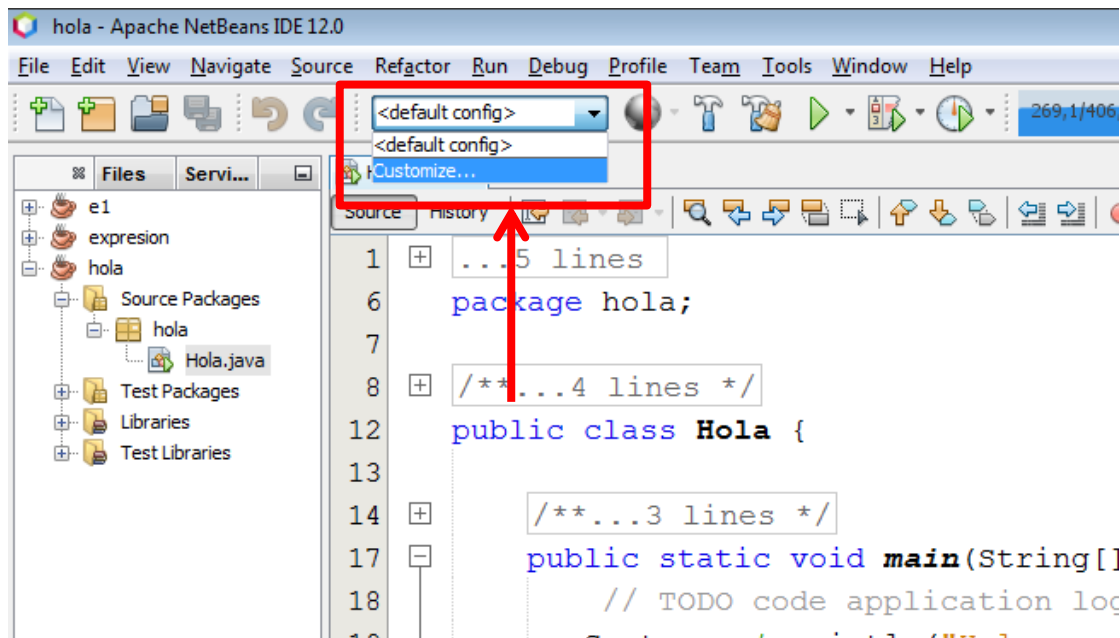
- Sobre **Linux**

- En la consola de comandos, un poco más compleja que en el caso del JDK.
 - Darle al archivo permiso de ejecución y ejecutarlo.
 - Instalación de **NetBeans** en Ubuntu

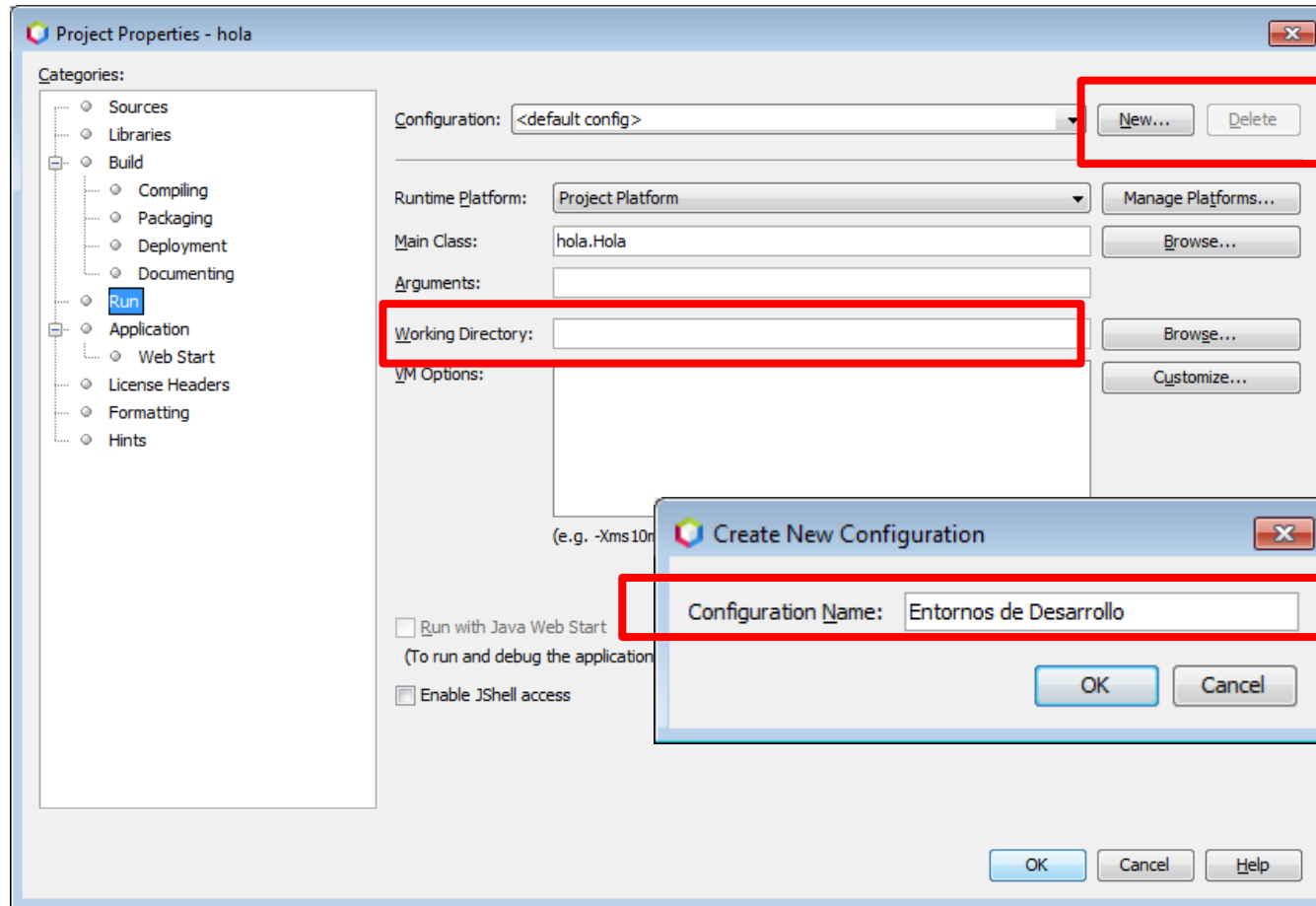
6. Configuración y personalización de entornos de desarrollo

Configuración y personalización de entornos de desarrollo

- Acceder a personalizar su configuración
- Crear un nuevo proyecto
- Seleccionar un desplegable con el nombre de “configuración”
- Desplegable>>Customice...



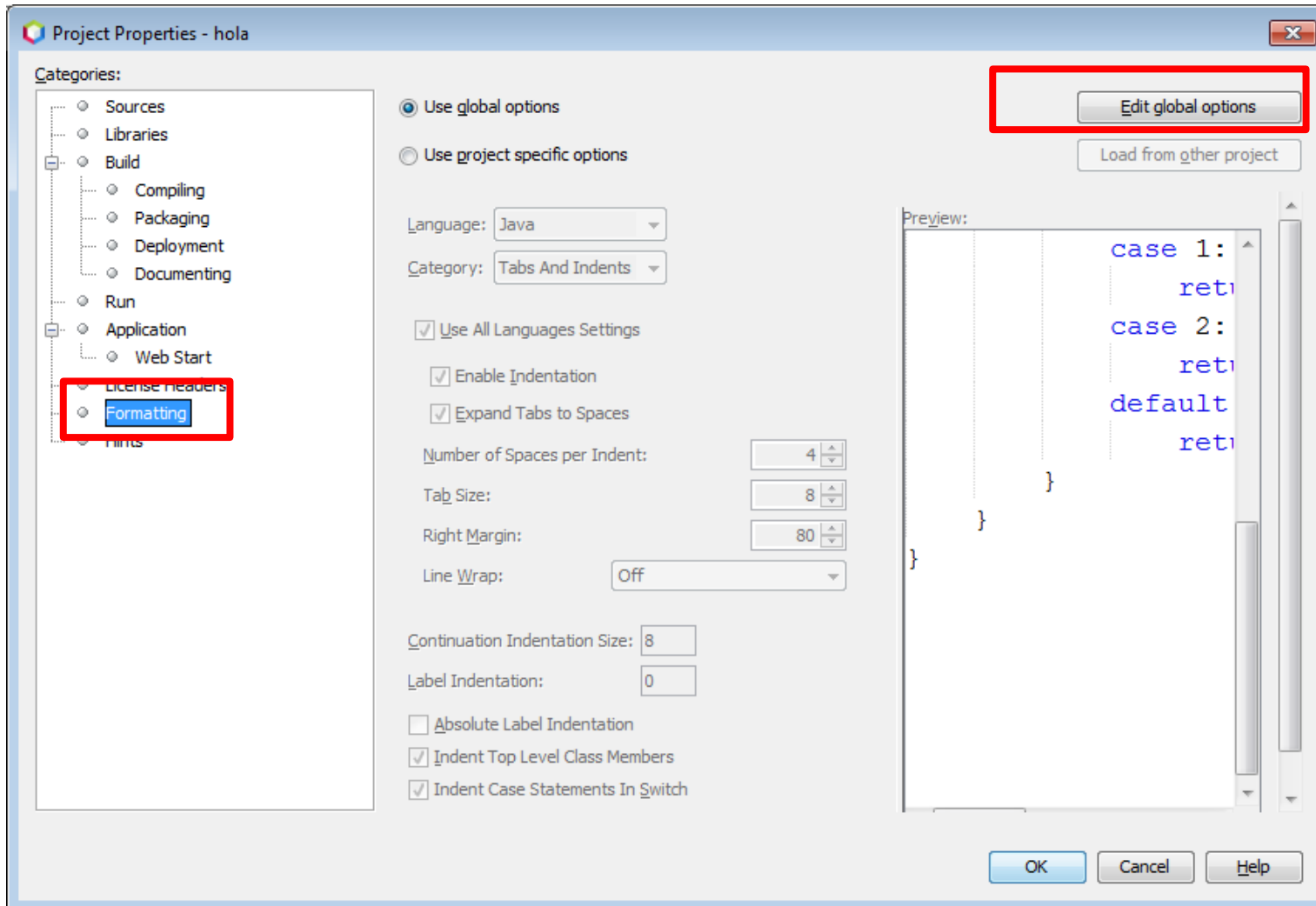
Configuración y personalización de entornos de desarrollo



- Botón “**New...**”, para una nueva configuración.
- Escribimos el nombre de la configuración
- Cuadro **Working directory** se especifica el directorio de trabajo

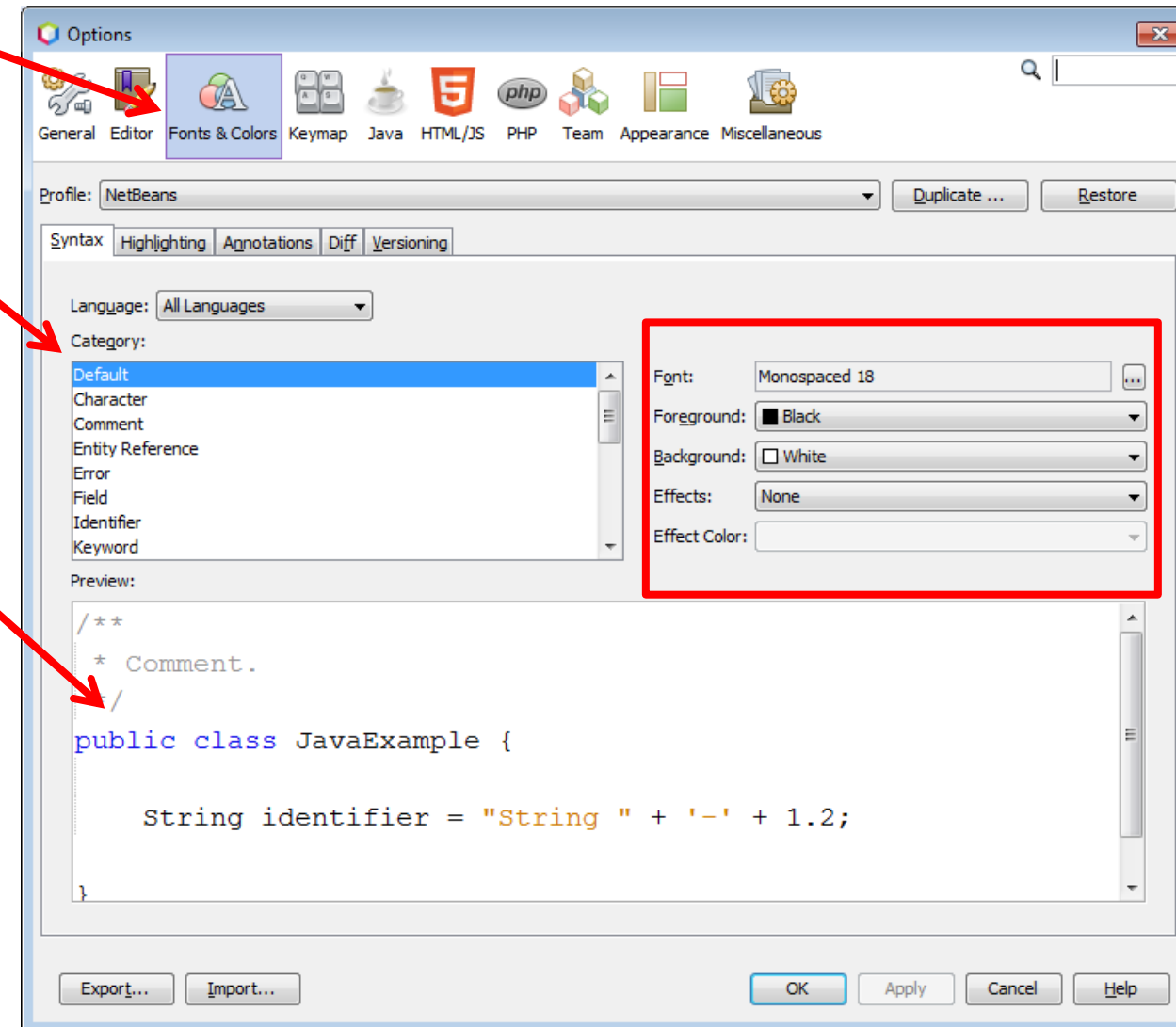
Configuración y personalización de entornos de desarrollo

- En **Formatting**, accedemos a las opciones de formato, en el botón **Edit Global options**



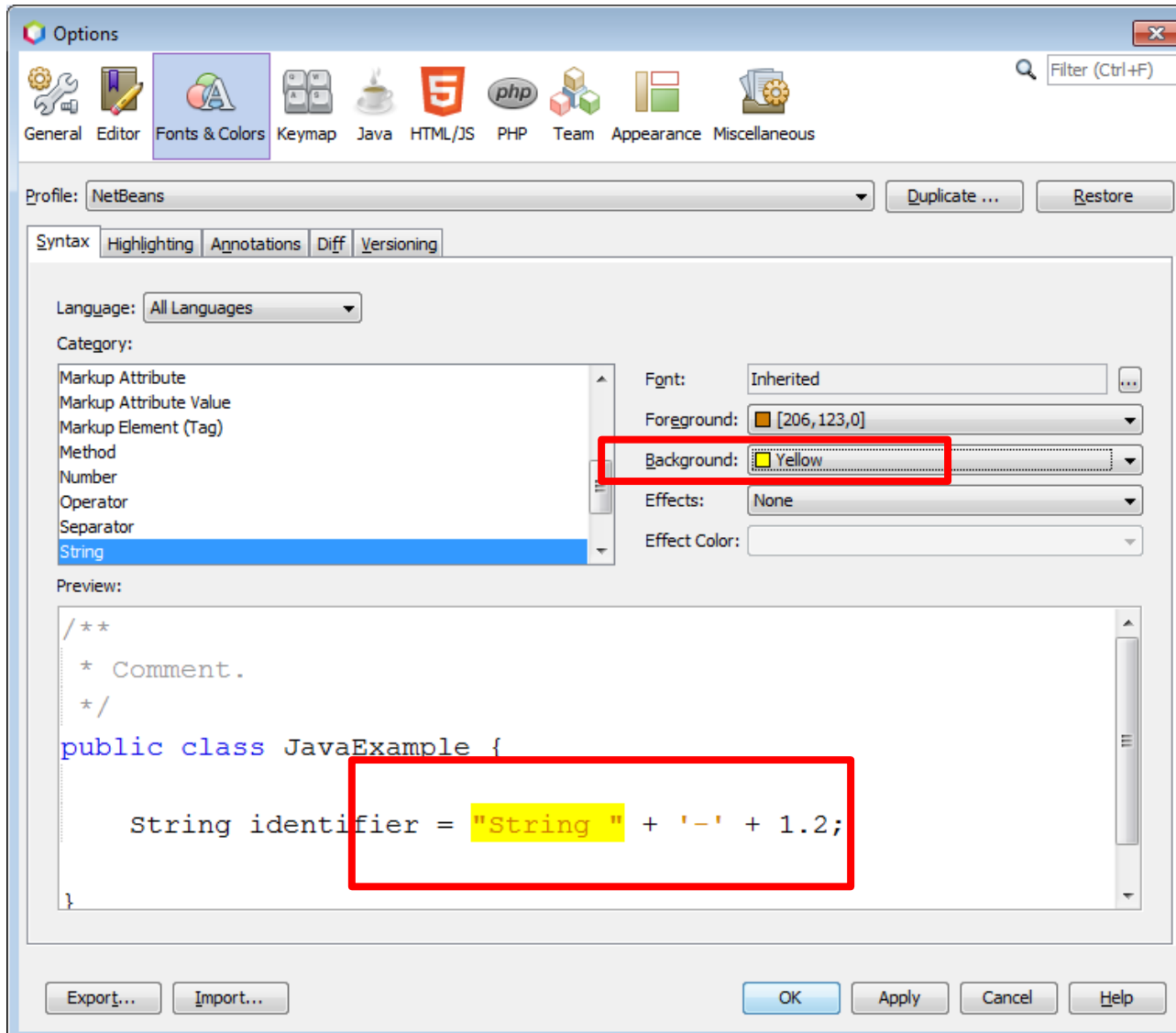
Configuración y personalización de entornos de desarrollo

- En “**Fonts & Colors**” se configura el aspecto del texto
- En **Category** se selecciona el tipo de texto que se va a modificar: errores, comentarios, palabras reservadas...
- Y se el resultado se podrá ver en el cuadro de previsualización



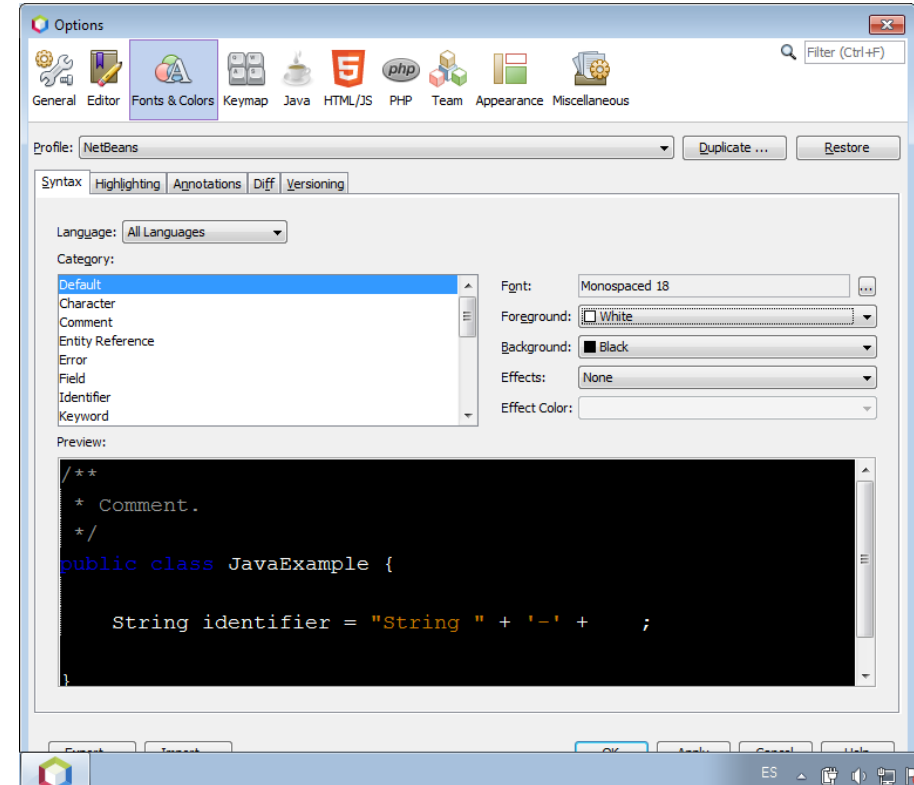
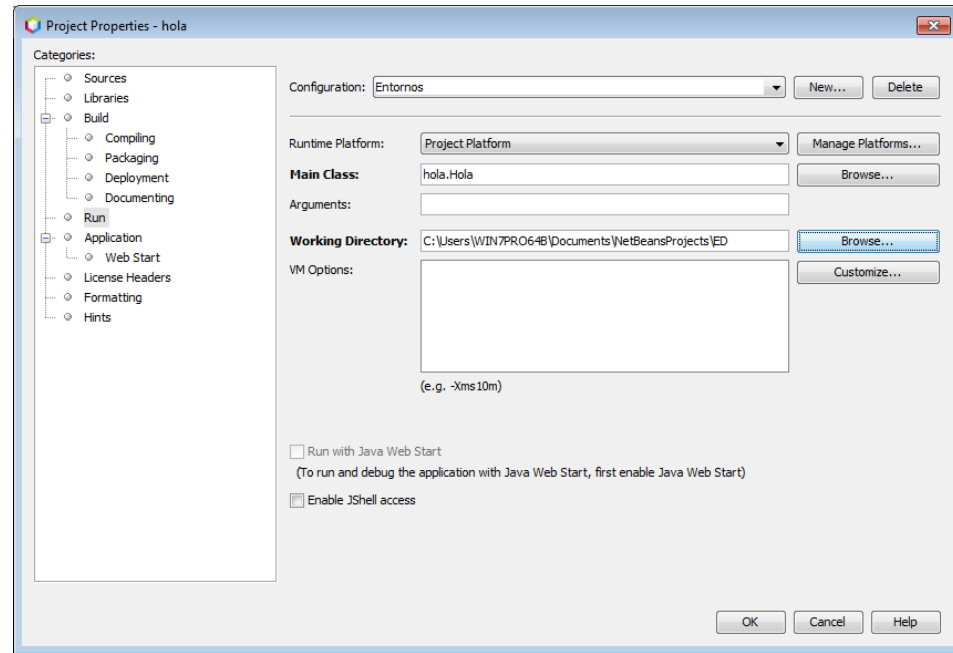
Configuración y personalización de entornos de desarrollo

- Se ha marcado los String con fondo amarillo y se puede observar en el cuadro de previsualización



Configuración y personalización de entornos de desarrollo

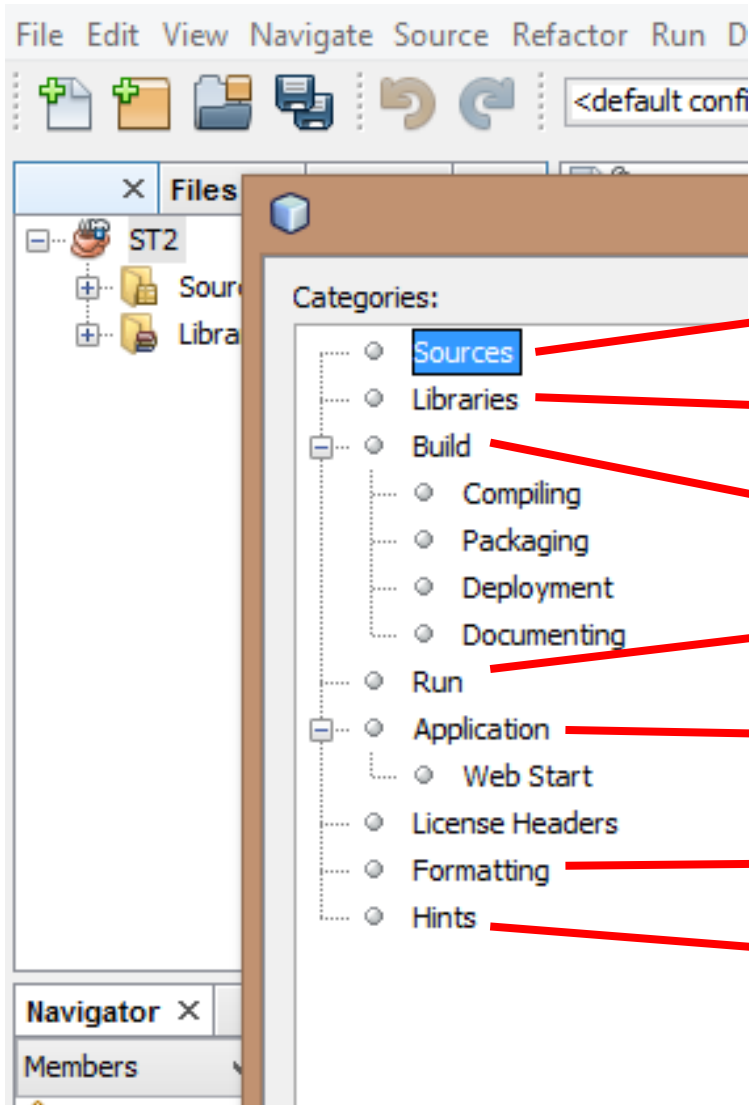
- Una vez creada la nueva configuración la podemos seleccionar en el desplegable de configuración



Configuración y personalización de entornos de desarrollo

- **Parámetros configurables del entorno**
 - Carpeta o carpetas donde se alojarán todos los archivos de los proyectos.
 - Carpetas de almacenamiento de paquetes fuente y paquetes prueba.
 - Administración de la plataforma del entorno de desarrollo.
 - Opciones de la compilación de los programas: compilar al grabar, generar información de depuración.
 - Opciones de empaquetado de la aplicación: nombre del archivo empaquetado (con extensión .jar) y momento del empaquetado.
 - Opciones de generación de documentación asociada al proyecto.
 - Descripción de los proyectos, para una mejor localización de los mismos.
 - Opciones globales de formato del editor: número de espaciados en las sangrías, color de errores de sintaxis, color de etiquetas, opción de autocompletado de código, propuestas de insertar automáticamente código.
 - Opciones de combinación de teclas en teclado.

Configuración de Netbeans



- Fuentes.
- Bibliotecas.
- Generación de código.
- Ejecución de código.
- Opciones de la aplicación.
- Formato del código en el editor de textos.
- Otros

Configuración de Netbeans

- **Fuentes**

- La carpeta que contendrá el proyecto
- La carpeta que almacenará los paquetes fuentes
- La carpeta que contendrá los paquetes prueba.

- **Bibliotecas:**

- Elegir la plataforma de la aplicación.
- Toma por defecto el JDK, pero se puede cambiar si se quiere, siempre y cuando sea compatible con la versión de NetBeans utilizada.
- Configurar el paquete de pruebas que se realizará al proyecto.

- **Generación de código – compilando**

- Compilar al grabar: al guardar un archivo se compilará automáticamente.
- Generar información de depuración: para obtener la documentación asociada.
- Enable annotation processing: permitir anotaciones durante el proceso.
- Agregar anotaciones concretas para el proceso de compilación y añadir opciones de proceso que, según las características del proyecto, puedan ser de interés para nosotros.

Configuración de Netbeans

- **Generación de código – empaquetando**

- Empaquetar aplicaciones antes de su distribución (tener un único archivo comprimido con todos los archivos de instalación y configuración).
- Modificar el lugar donde se generará el archivo resultante del empaquetado, así como si deseamos comprimirlo.
- Archivo empaquetado se construya tras la compilación, que es lo habitual (por eso esta opción aparece como predeterminada)

- **Generación de código – documentando**

- NetBeans permite obtener documentación de la fase de codificación de los programas de forma automática.

- **Ejecutando código**

- Define la clase principal, las carpetas de trabajo del proyecto y opciones de la máquina virtual.
- “Configurar el nombre” ➔ nombre que tendrá nuestra configuración personalizada. ➔ todas las opciones de configuración que seleccionemos que guardarán con ese nombre.

Configuración de Netbeans

- **Opciones de la aplicación**

- Descripción del proyecto, cambiar nombre, etc...
- Es conveniente hacerlo, ya que el nombre de los nuevos proyectos se generar automáticamente por NetBeans al inicio de la sesión.
- También podemos definir las opciones del inicio web.

- **Formato**

- Personalizar aspectos globales del formato del código fuente en la aplicación (para el proyecto actual o todos los proyectos)
- Se puede configurar los tamaños de los espaciados, pestañas, el tipo de letra y colores que prefiramos para el texto del código, color de los errores del código (por defecto, de color rojo), lo mismo con los números, espacios en blanco, métodos abreviados de teclado, etc.

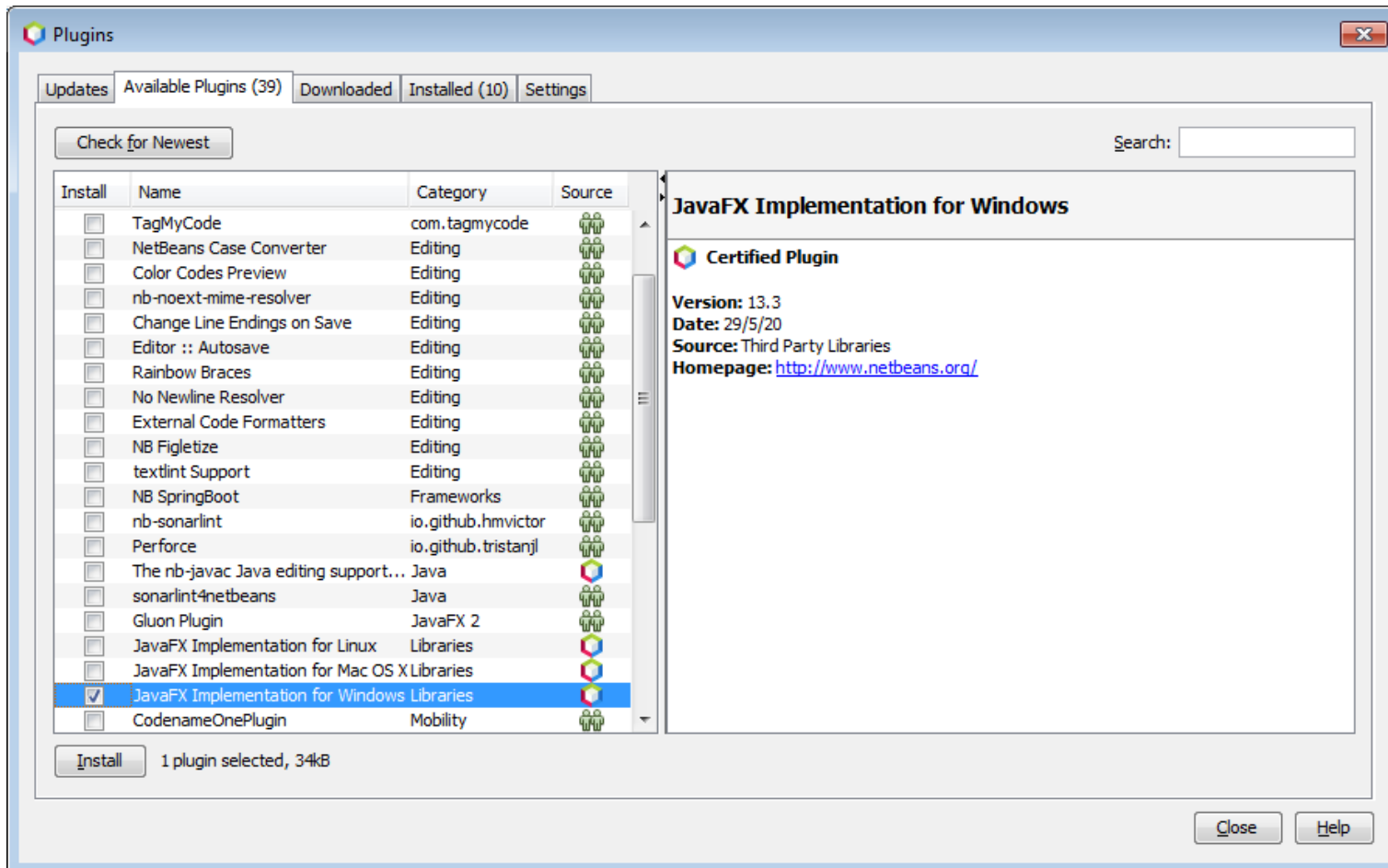
7. Gestión de módulos

Gestión de módulos. Plugins

- Nos permiten añadir nuevas funcionalidades al IDE
- Toda la información sobre plugins de NetBeans: **Tools>>Plugins**
- Instalarlo **on-line** en el entorno.
 - Instala complementos desde nuestro mismo IDE, sin descargarlos previamente.

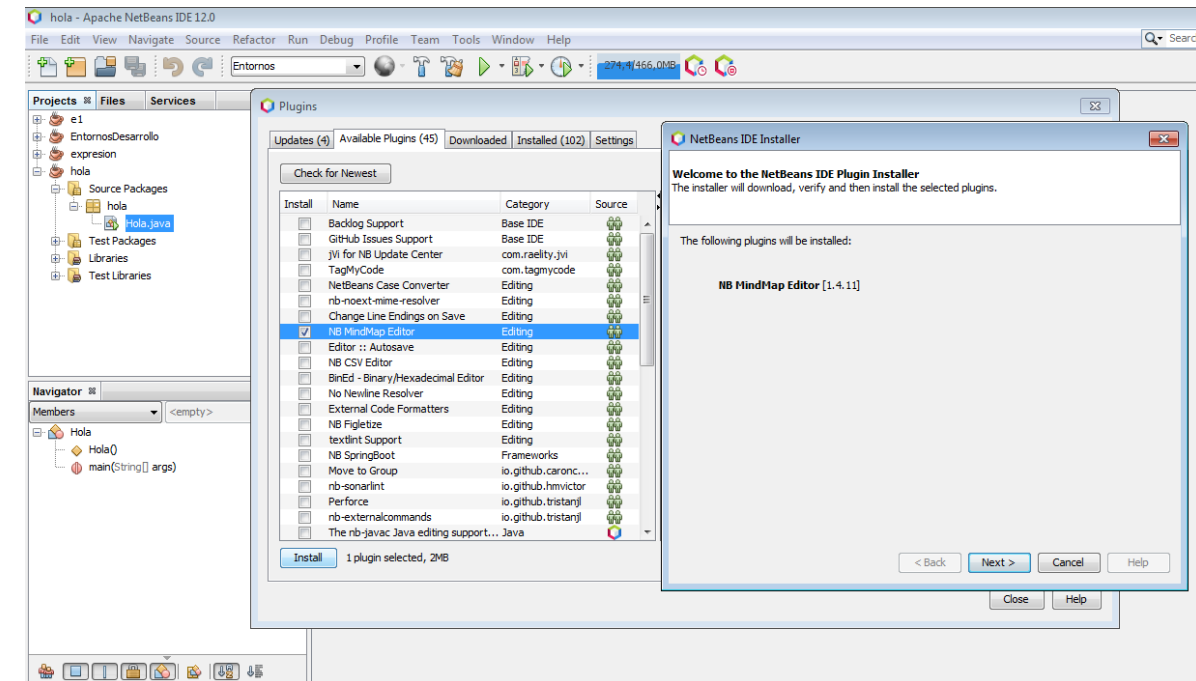
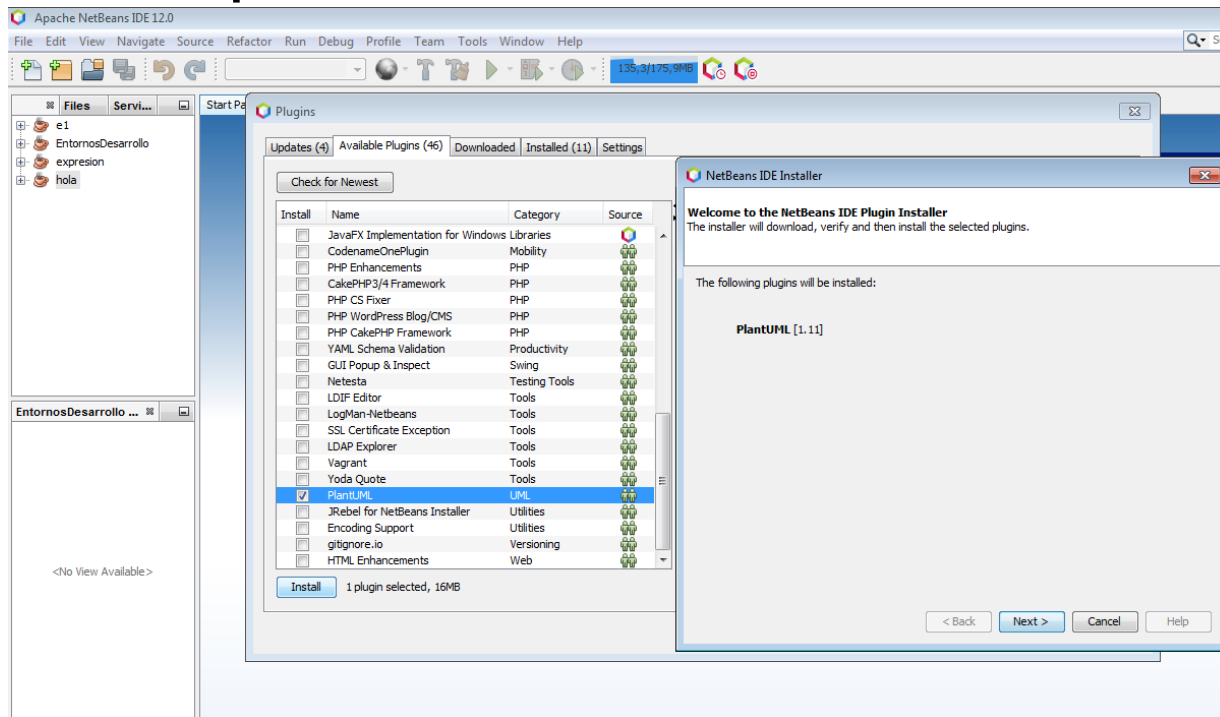
Gestión de módulos. Plugins online

- Pestaña **Available**
Plugins>>Seleccionar el
plugin a instalar>>Botón
Install



Gestión de módulos. Plugins online

- Ejercicio: instala el plugin PlantUML/NB MindMap, determina las características y pasos a realizar y comprueba qué funcionalidad aporta al IDE



Gestión de módulos. Plugins offline

- Para la instalación offline hay que acceder a la web oficial de Apache NetBeans y descargamos el plugins
 - Módulo descargado de un sitio web.
 - Formato .nbm
 - Se denomina **off-line**
- Descargar un plugin para NetBeans 12
 - <http://plugins.netbeans.org/PluginPortal/>

Gestión de módulos. Plugins offline

- Ejem. off-line
 - <http://plugins.netbeans.org/>
 - Zona catálogo
 - Escribir sudoku
 - Seleccionar el plugin, la carpeta dónde guardarlo
 - Entrar en NetBeans
 - Crear un nuevo proyecto → aplicación Java
 - Herramientas → Complementos
 - Descargado → Agregar plugins
 - Seleccionar sudoku y aceptar
 - Instalar plugin

Gestión de módulos. Plugins offline

- Ejem. off-line
 - Color Codes Preview: instala el plugin, determina las características y pasos a realizar y comprueba qué funcionalidad aporta al IDE

Gestión de módulos IV

- Eliminar plugin
 - Encontrar el módulo o plugin dentro de la lista de complementos instalados en el entorno.
 - User installed plugins (en la derecha aparecen los plugins instalados)
 - Opciones:
 - **Desactivarlo:** El módulo o plugin sigue instalado, pero en estado inactivo (no aparece en el entorno).
 - **Desinstalarlo:** El módulo o plugin se elimina físicamente del entorno de forma permanente.

Gestión de módulos

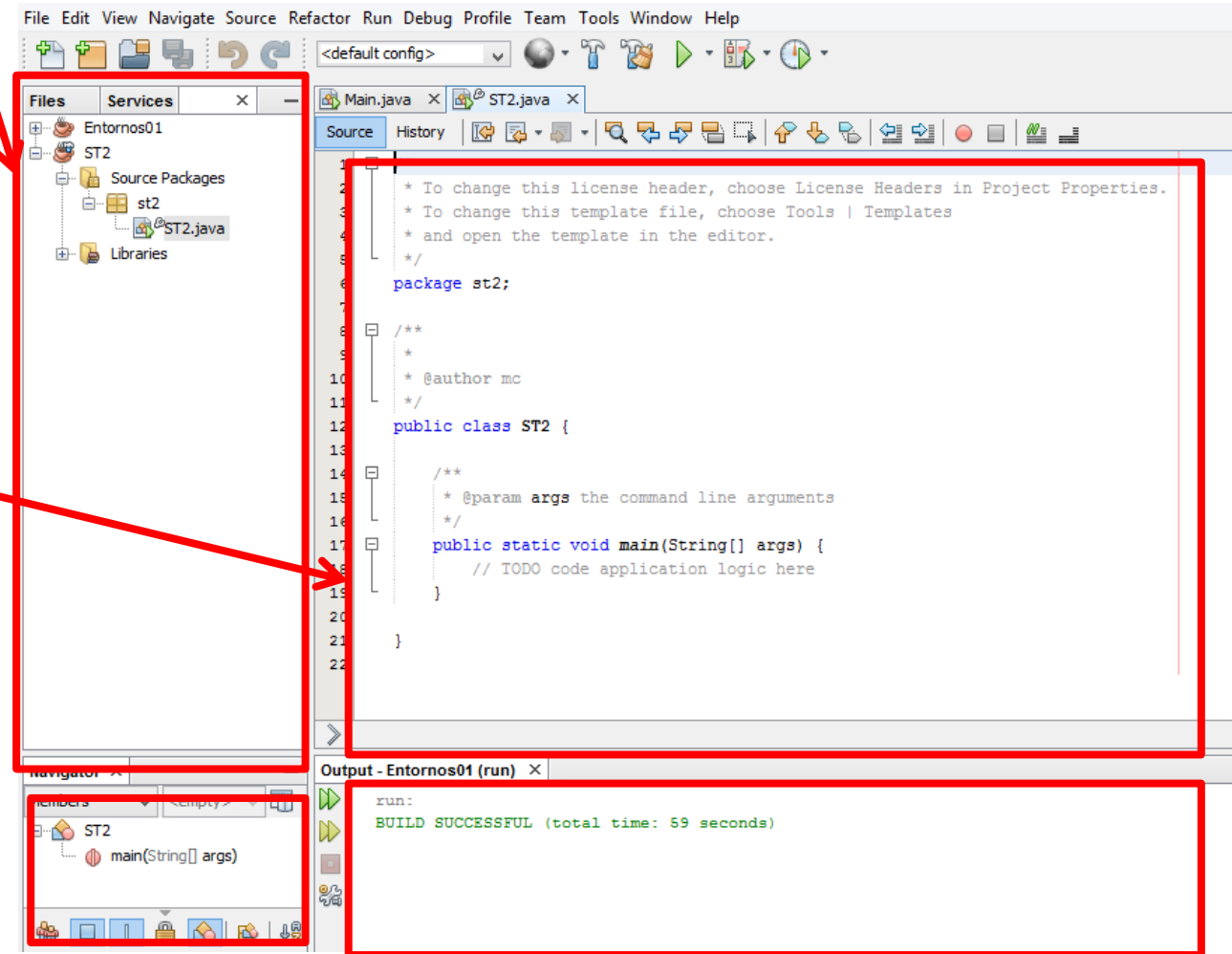
■ Funcionalidades

- **Construcción de código:** facilitan la labor de programación.
- **Bases de datos:** ofrecen nuevas funcionalidades para el mantenimiento de las aplicaciones.
- **Depuradores:** hacen más eficiente la depuración de programas.
- **Aplicaciones:** añaden nuevas aplicaciones que nos pueden ser útiles.
- **Edición:** hacen que los editores sean más precisos y más cómodos para el programador.
- **Documentación de aplicaciones:** para generar documentación de los proyectos en la manera deseada.
- **Interfaz gráfica de usuario:** para mejorar la forma de presentación de diversos aspectos del entorno al usuario.
- **Lenguajes de programación y bibliotecas:** para poder programar bajo un Lenguaje de Programación que, en principio, no soporte la plataforma.
- **Refactorización:** hacer pequeños cambios en el código para aumentar su legibilidad, sin alterar su función.
- **Aplicaciones web:** para introducir aplicaciones web integradas en el entorno.
- **Prueba:** para incorporar utilidades de pruebas al software.

8. Uso básico de entornos de desarrollo

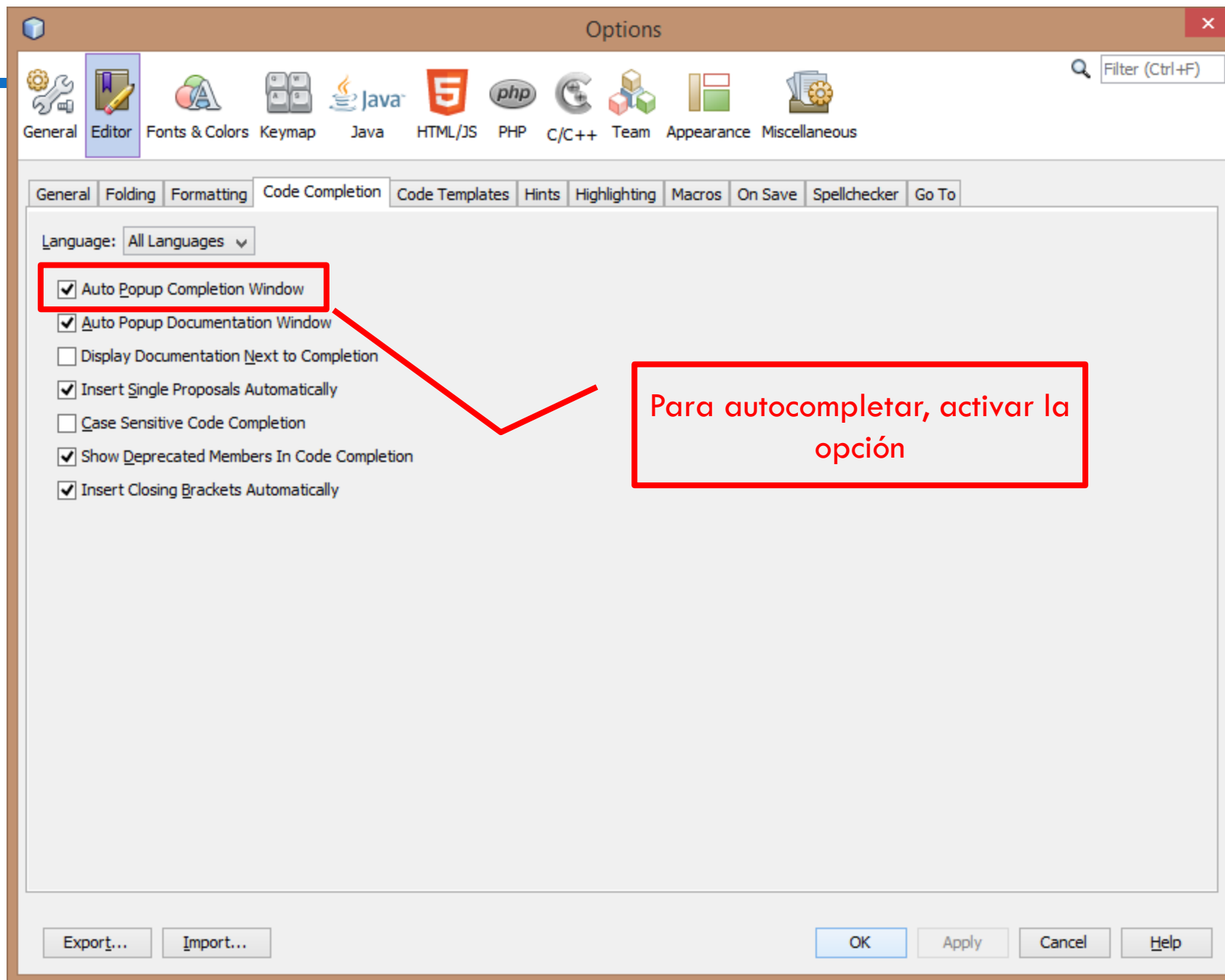
Uso básico de entornos de desarrollo

- **Ventana Izquierda:** ventana de proyectos.
 - Cada proyecto comprende una serie de archivos y bibliotecas que lo componen.
 - El principal archivo del proyecto Java es el llamado Main.java.
- **Ventana derecha:** espacio de escritura de los códigos de los proyectos.
- **Barra de herramientas:** Desde aquí podremos acceder a todas las opciones del IDE.



Edición de programas

- Crear programa de “hola, mundo”
 - Autocompletado de código.
 - Coloración de comandos.
 - Subrayado en rojo cuando hay algún error y posibilidad de depuración y corrección de forma visual, mediante un pequeño icono que aparece a la izquierda de la línea defectuosa.
 - Ampliar programa con JFrame
- <https://www.fdi.ucm.es/profesor/luis/fp/devtools/NetBeansUso.html>
- <https://dcodingames.wordpress.com/2015/11/02/guia-basica-para-usar-netbeans/>




```
public class HolaMundoGrafico {  
    public static void main(String[] args) {  
        //Creamos una ventana  
        JFrame miVentana = new JFrame();  
        miVentana.setTitle ("Hola Mundo Grafico");  
        //le ponemos título a nuestra ventana  
        //Creamos un campo de texto  
        JTextPane miCampoDeTexto = new JTextPane();  
        miCampoDeTexto.setText("HOLA MUNDO");  
        //ahora metemos el campo de texto en la ventana  
        miVentana.add(miCampoDeTexto);  
        //Detalles importantes  
        //1- le decimos a la ventana que cierre el programa si alguien  
        pulsa la X de cierre  
        miVentana.setDefaultCloseOperation (JFrame.DISPOSE_ON_CLOSE);  
        //2- ponemos como tamaño de ventana el mínimo necesario para que  
        se vea todo lo que contiene  
        miVentana.pack();  
        //3- decimos que la ventana sea visible  
        miVentana.setVisible(true);  
    }  
}
```

Generación de ejecutables

- Siguiendo pasos son:
 - Compilación
 - Depuración
 - Ejecución

9. Actualización y mantenimiento de entornos de desarrollo

Actualización y mantenimiento de entornos de desarrollo

- Tarea fundamental que requiere tener todos sus componentes periódicamente actualizados.
- Copias de seguridad sobre las bases de datos de nuestros proyectos por si ocurriera algún error o proceso defectuoso poder restaurarlos.
- El mantenimiento y las actualizaciones se hacen de forma on-line.
- NetBeans contamos con el complemento llamado Auto Update Services.
 - <http://plugins.netbeans.org/>
- Una vez instalado, nos permitirá realizar continuas revisiones del entorno y actualizaciones de todos los plugins