

UT02. INSTALACIÓN Y USO DE ENTORNOS DE DESARROLLO.

Índice

- 1.-Comenzar a programar en Java.
JDK.
- 2.-Concepto de Entorno de desarrollo.
- 3.- Funciones de un Entorno de desarrollo.
- 4.- Entornos integrados libres y propietarios.
- 5.- Estructura de entornos de desarrollo.
- 6.- Instalación de entornos integrados de desarrollo.
- 7.- Uso básico de entornos de desarrollo:
 - 7.1.- Creación de Proyectos
 - 7.2.- Edición de programas
 - 7.3.- Generación de ejecutables
- 8.- Configuración y personalización de entornos de desarrollo.
- 9.- Gestión de módulos:
 - 9.1.- Añadir
 - 9.2.- Eliminar
 - 9.3.- Funcionalidades
- 10.- Actualización y mantenimiento de entornos de desarrollo

1. Comenzar a programar en Java. JDK

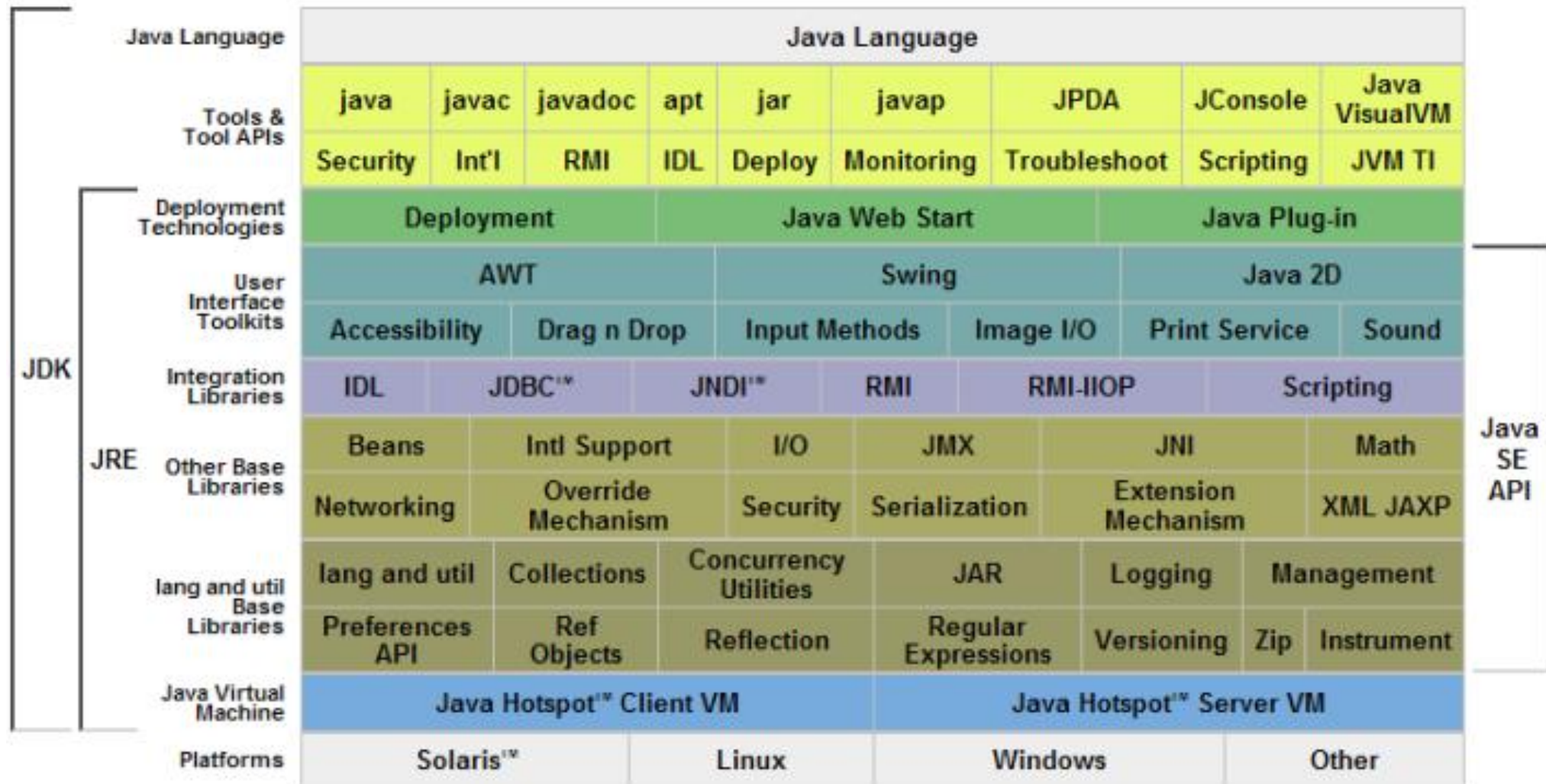
Comenzar a programar en Java: Plataforma Java



Herramientas que nos permiten desarrollar programas en java.

Contiene la JVM y API (librerías de clases)

Comenzar a programar en Java: Plataforma Java

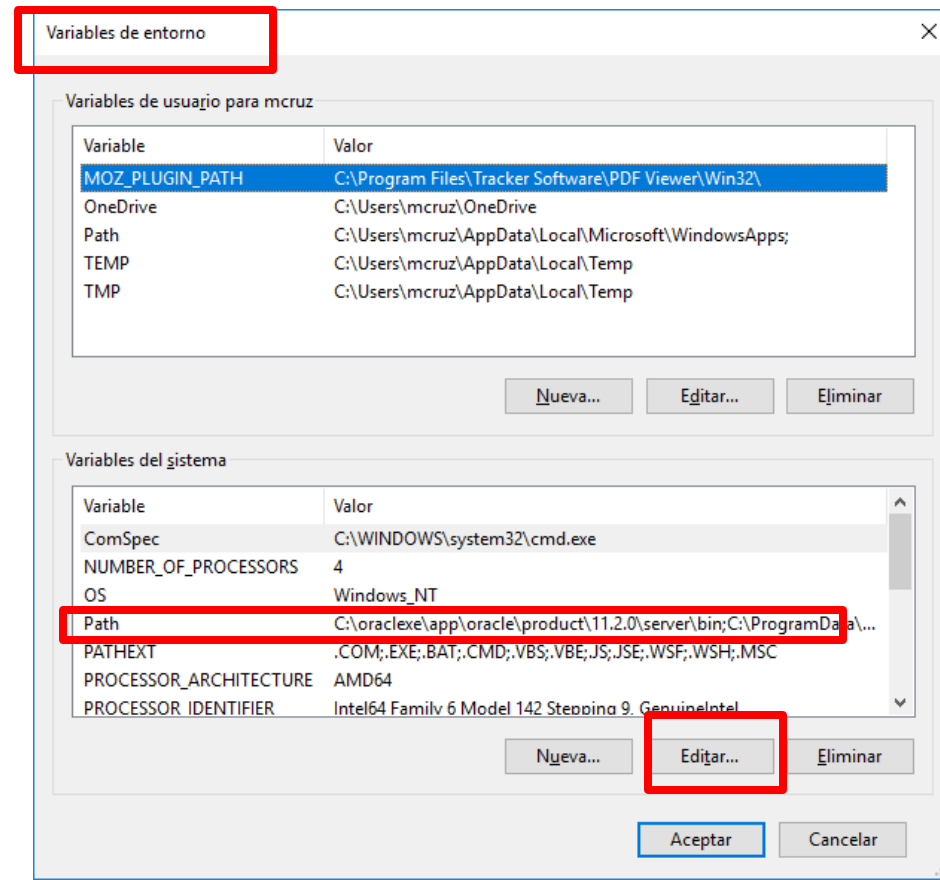
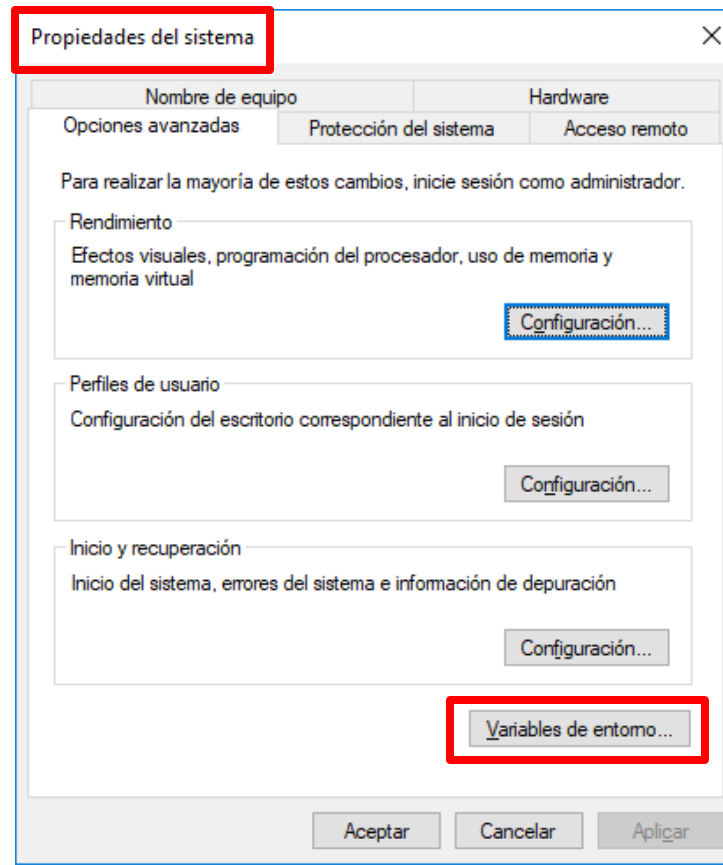


Comenzar a programar en Java: JDK

- ❑ Descargar e instalar JDK (Java Development Kit)
 - ▣ JDK Oracle vs OpenJDK
 - ▣ <https://www.oracle.com/java/technologies/downloads> -> Descargamos JDK (última versión)
- ❑ Modificar variables de entorno:
 - ▣ Variable PATH: Apunta donde está situado el directorio bin del JDK (donde están los ejecutables javac.exe, java.exe ...)
 - Añadir en PATH: C:\Archivos de Programa\Java\Jdk-14.0.2\bin\

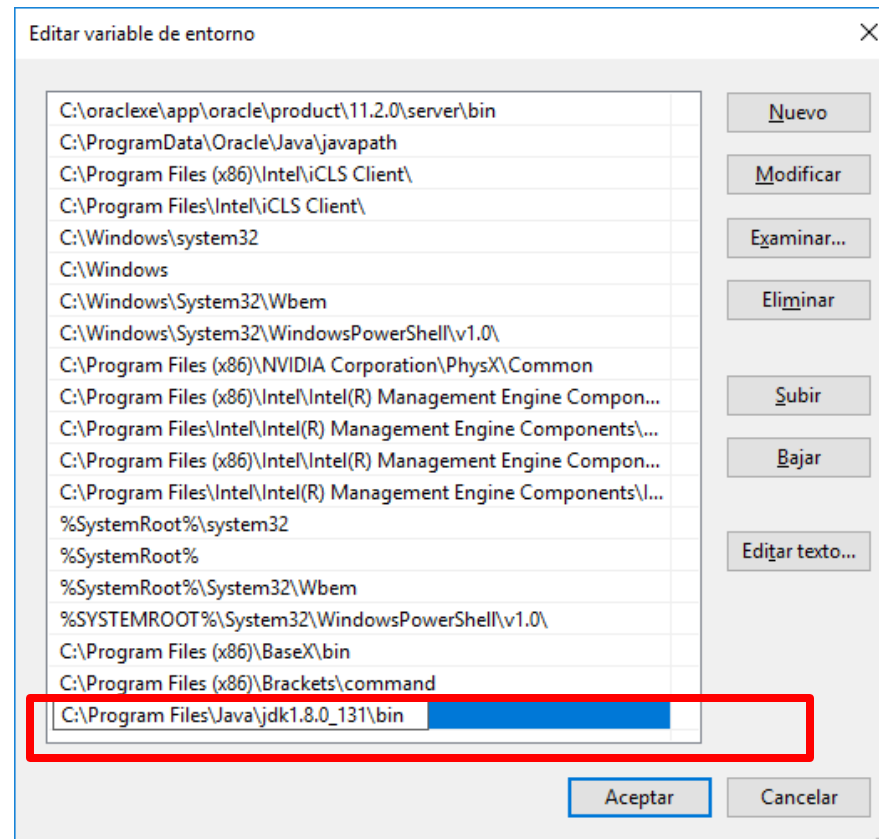
Comenzar a programar en Java: JDK

- ❑ Modificar la variable de entorno PATH:
 - ▣ Este equipo>>Propiedades>>Configuración avanzada del sistema
 - ▣ Propiedades del sistema>>Variables de entorno...>>En variables del sistema, seleccionamos Path y Botón Editar...



Comenzar a programar en Java: JDK

- 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\jdk-14.0.2\bin

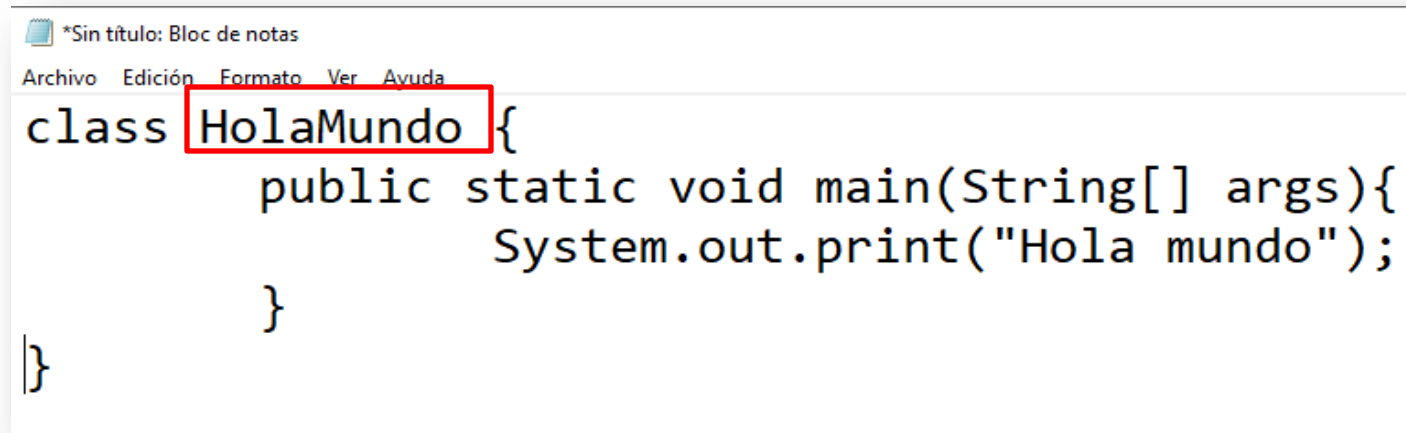


Comenzar a programar en Java: compilar y ejecutar

- Desde el PoweShell o Intérprete de comandos, utilizaremos los comandos siguientes:
 - Para conocer la versión de java instalada: `java --version`
 - **Compilar** un programa: `javac fichero.java` → genera el fichero `fichero.class` que contiene los bytecodes ejecutables por la máquina virtual
 - **Ejecutar** un programa : `java fichero`
 - **Nota:** Tanto para compilar como para ejecutar programas con los comandos anteriores deberemos situarnos en la carpeta donde estén situados los ficheros.
- Para empezar a programar editaremos nuestros programas en el Bloc de notas y los guardaremos con extensión `*.java`.

Compilar y Ejecutar en Java

- Escribimos nuestro primer programa en java en el bloc de notas que visualice por pantalla *"Hola mundo"*.

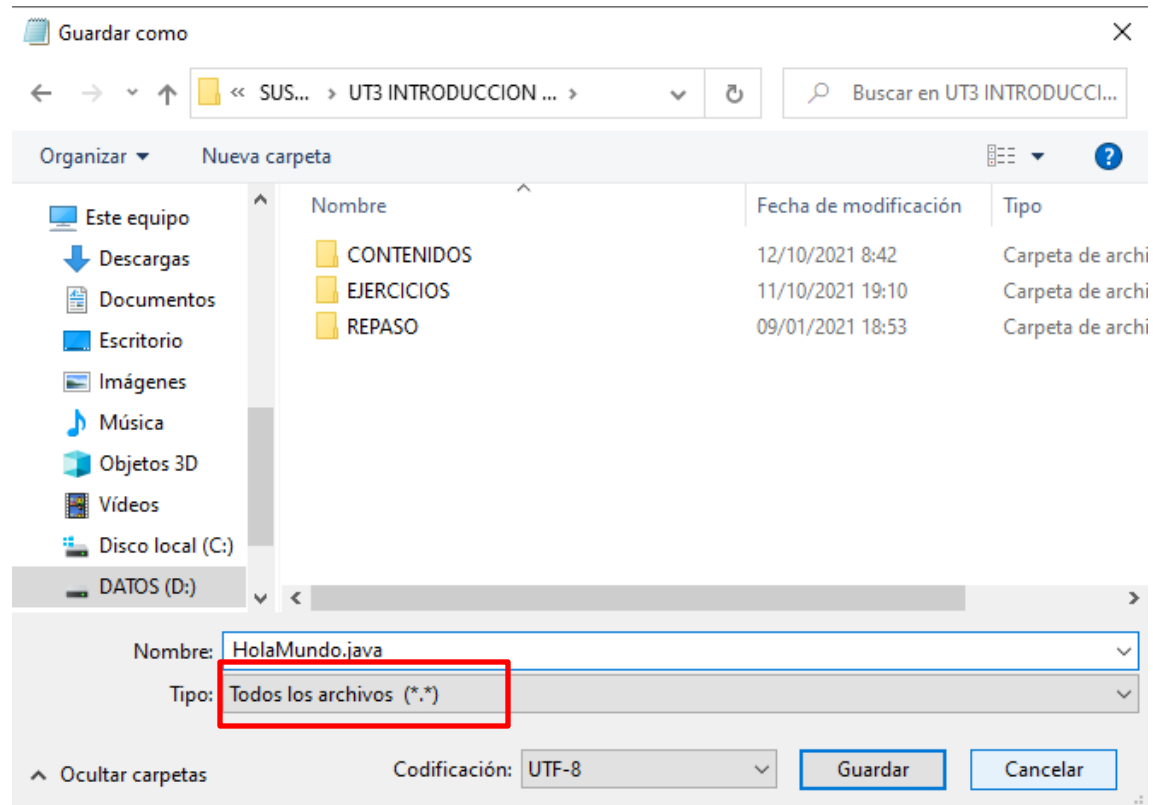
A screenshot of a Windows Notepad application window. The title bar reads "*Sin título: Bloc de notas". The menu bar includes "Archivo", "Edición", "Formato", "Ver", and "Ayuda". The text area contains the following Java code:

```
class HolaMundo {  
    public static void main(String[] args){  
        System.out.print("Hola mundo");  
    }  
}
```

The word "HolaMundo" in the first line is highlighted with a red rectangular box. The cursor is positioned at the end of the final closing brace on the last line.

Compilar y Ejecutar en Java

- Guardamos con el nombre **HolaMundo.java**.



Compilar y Ejecutar en Java

- Compilamos `javac HolaMundo.java` → Obtenemos `HolaMundo.class` (¡ojo!
Debemos ubicarnos en la ruta donde está el código fuente para compilar)
- Ejecutamos `java HolaMundo` ó `java -cp ruta HolaMundo`

```
Símbolo del sistema
C:\Users\SUSANA>java --version
java 11.0.10 2021-01-19 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.10+8-LTS-162)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.10+8-LTS-162, mixed mode)

C:\Users\SUSANA>javac --version
javac 11.0.10

C:\Users\SUSANA>d:
d:\>cd Ejemplo
d:\Ejemplo>javac HolaMundo.java
d:\Ejemplo>java HolaMundo
Hola mundo.

d:\Ejemplo>c:
C:\Users\SUSANA>java -cp d:\Ejemplo HolaMundo
Hola mundo.

C:\Users\SUSANA>
```

Consultamos la versión de Java (JVM o JRE) y la versión del JDK

En este ejemplo guardamos el código fuente (`HolaMundo.java`) en la carpeta `d:\Ejemplo`

Compilamos el código fuente
Ejecutamos el código objeto

Ejecutamos el código desde otra carpeta donde no se encuentra el código objeto

Compilar y Ejecutar en Java

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

```
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. Todos los derechos reservados.

PS C:\Users\admin> cd c:\java
PS C:\java> javac Ejercicio2.java
PS C:\java> java Ejercicio2 5 6
La suma es 11
PS C:\java> java Ejercicio2 5 6 > salida.txt
PS C:\java>
```

```
class Ejercicio2 {
    public static void main(String[] args){

        if(args.length > 1){
            int suma = Integer.parseInt(args[0]) + Integer.parseInt(args[1]);
            System.out.println("La suma es " + suma);
        }
    }
}
```

2. Concepto de Entorno de desarrollo.

Concepto de Entorno de desarrollo.

- Etapa de Codificación:
 - Uso de un lenguaje de programación
 - Herramientas de apoyo
- Mínimo: editor de código+ compilador + depurador
- **IDE** (Integrated **D**evelopment **E**nvironment): conjunto de herramientas que facilitan la tarea de programación:
 - Editor de código.
 - Compilador.
 - Intérprete.
 - Depurador.
 - Constructor de interfaz gráfico.
 - ...

Concepto de Entorno de Desarrollo

- Primeros entornos de desarrollo integrados nacieron a principios de los años 70, y se popularizaron en la década de los 90.
- Objetivo:
 - ▣ Ganar fiabilidad y tiempo en el desarrollo de proyectos de software.
 - ▣ Proporcionar al programador una serie de componentes con la misma interfaz gráfica, con la consiguiente comodidad, aumento de eficiencia y reducción de tiempo de codificación.
- Un IDE está dedicado a un determinado lenguaje de programación.
- Tendencia actual: ser compatibles con varios lenguajes (por ejemplo, Eclipse, NetBeans, Microsoft Visual Studio) mediante la instalación de plugins adicionales.

3. Funciones de un entorno de desarrollo

Funciones de un entorno de desarrollo

□ **Las funciones de los 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.

Funciones de un entorno de desarrollo

- Otras funciones adicionales
 - 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**.

Funciones de un entorno de desarrollo

□ Pregunta:

Un entorno integrado de desarrollo está compuesto por:

- ▣ Editor de código y traductor.
- ▣ Editor de código, compilador e interfaz de comandos.
- ▣ Editor de código, compilador, intérprete, depurador e interfaz gráfica.
- ▣ Interfaz gráfica, editor de código y depurador.

Funciones de un entorno de desarrollo







- Un entorno integrado de desarrollo está compuesto por:
 - ▣ Editor de código y traductor.
 - ▣ Editor de código, compilador e interfaz de comandos.
 - ▣ Editor de código, compilador, intérprete, depurador e interfaz gráfica.
 - ▣ Interfaz gráfica, editor de código y depurador.

4. Entornos integrados libres y propietarios.

Entornos integrados libres y propietarios.

□ Libres.


▣ Con licencia de uso público

	ID	Lenguajes	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, Java, Vala.	Linux.
	Geany.	C/C++, Java.	Windows, Linux, Mac OS X.
	GNAT Studio.	Ada.	Windows, Linux, Mac OS X.

Entornos integrados libres y propietarios.

□ Proprietarios.

▣ Necesitan licencia

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

5.- Estructura de entornos de desarrollo.

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.
- ▣ Herramientas para la automatización de las tareas de compilación:
Ant, Maven y Gradle
 - Usaremos Maven

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.

Estructura de entornos de desarrollo.

□ **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.

□ Ejercicio de relacionar.

Entorno de desarrollo.	Relación	Características.
Microsoft Visual Studio.		1. Libre. Soporta C/C++, Java, PHP, Javascript, Python.
NetBeans.		2. Propietario. Soporta Basic, C/C++, C#.
C++ Builder.		3. Propietario. Soporta C/C++.

□ Solución

Entorno de desarrollo.	Relación	Características.
Microsoft Visual Studio.	2	1. Libre. Soporta C/C++, Java, PHP, Javascript, Python.
NetBeans.	1	2. Propietario. Soporta Basic, C/C++, C#.
C++ Builder.	3	3. Propietario. Soporta C/C++.

6. Instalación de entornos integrados de desarrollo.

IDE para Java. NetBeans

- ❑ NetBeans es un IDE Open Source hecho principalmente para Java, aunque cuenta con un gran número de módulos que extienden su funcionalidad permitiendo, entre otras cosas, utilizarlo para otros lenguajes como PHP, JavaScript, HTML, CSS ...
- ❑ 1999-2009 fue patrocinado por **Sun Microsystems**.
- ❑ 2010-2015 es adquirido por **Oracle**.
- ❑ 2016-hoy forma parte de **Apache Software Foundation** tras ser donado por Oracle.
- ❑ NetBeans permite el desarrollo de todos los tipos de aplicación Java (J2SE, web, aplicaciones móviles,...).
- ❑ NetBeans trabaja por proyectos.
- ❑ Un proyecto incluye todos los recursos necesarios para construir un programa (archivos de código, bibliotecas externas, imágenes, etc.)
- ❑ Físicamente un proyecto es un directorio con una organización especial.
- ❑ La versión NetBeans 12 es LTS (Long Term Support o con soporte a largo plazo)
- ❑ A partir de NetBeans 11 se soporta JDK 14
- ❑ NetBeans no es compatible con OpenJDK, por tanto deberemos instalar JDK de Oracle.

IDE para Java. NetBeans

- Entre sus características se encuentra un sistema de proyectos basado en Ant, Maven o Gradle, control de versiones y refactoring.
- Ant, Maven y Gradle, son herramientas de automatización de la construcción de proyectos
 - <https://www.baeldung.com/ant-maven-gradle>
 - https://es.wikipedia.org/wiki/Apache_Ant
 - <https://es.wikipedia.org/wiki/Maven>
 - <https://es.wikipedia.org/wiki/Gradle>

Instalación de IDE NetBeans

□ Sobre Windows

▣ NetBeans 12

- Requiere la instalación de JDK (Java Development Kit -- Kit de Desarrollo de Java).

- Descarga de JDK

<https://www.oracle.com/java/technologies/java-se-glance.html>

- Descargar NetBeans 12.0

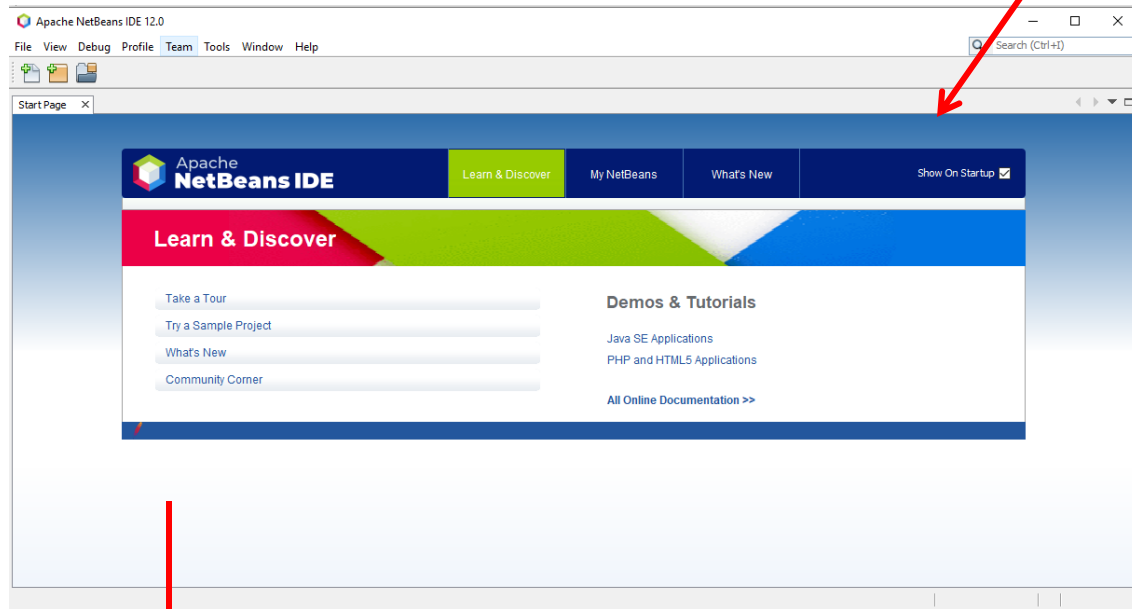
<https://netbeans.apache.org/download/>

□ Sobre Linux

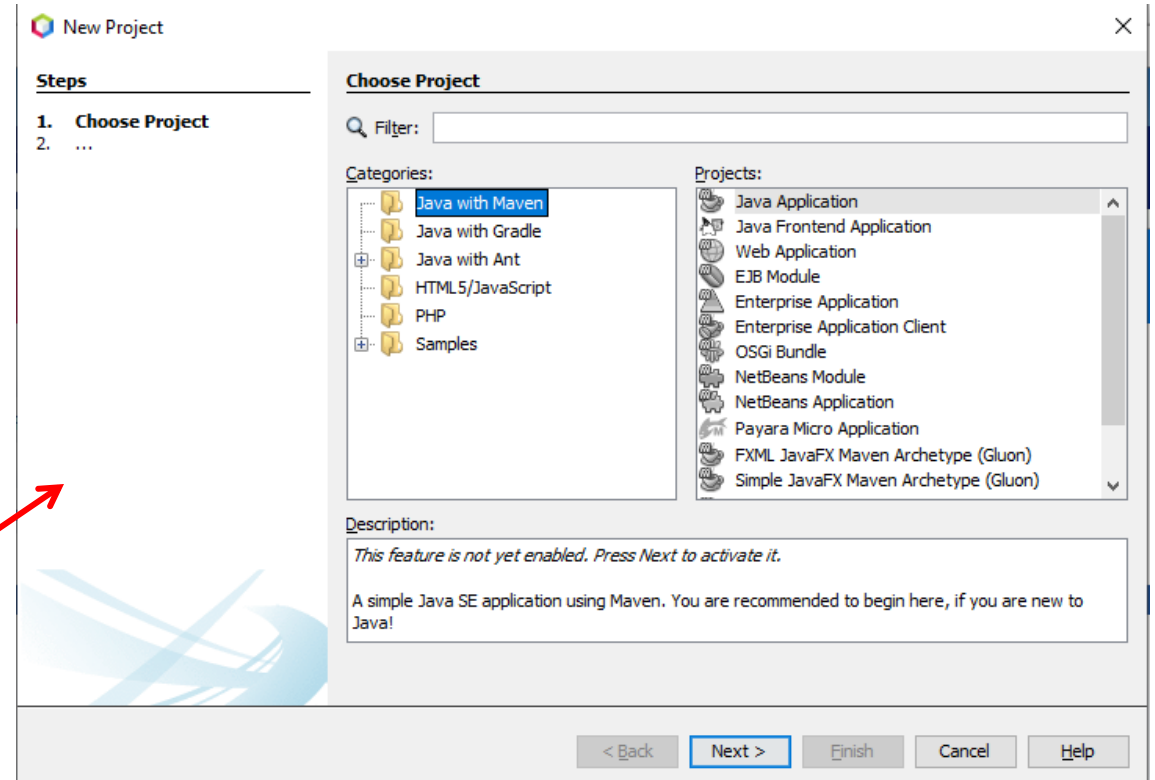
- ▣ En la consola de comandos, un poco más compleja que en el caso del JDK.
- ▣ Darle al archivo permiso de ejecución y ejecutarlo.
- ▣ Instalación de NetBeans en Ubuntu

Instalación de IDE NetBeans

Tras instalar la aplicación se nos mostrará la siguiente ventana

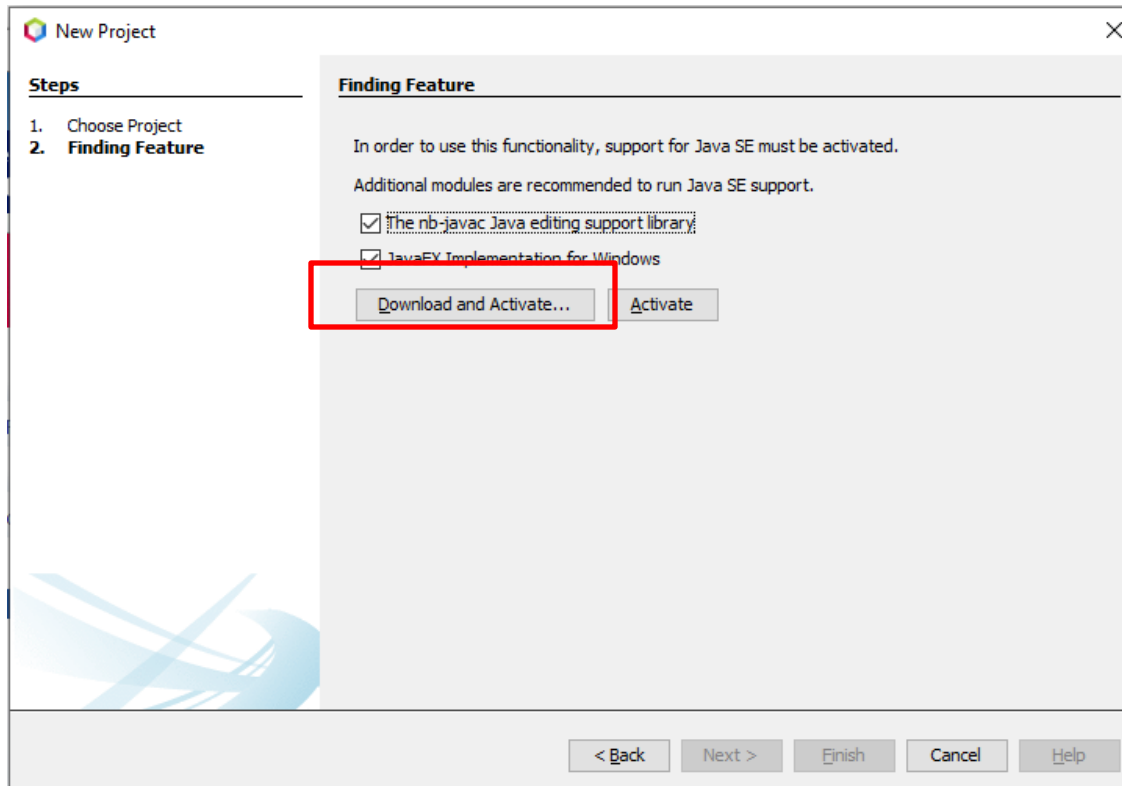


Seleccionaremos el menú File opción New Project



Instalación de IDE NetBeans

Seleccionaremos “Java with Maven” en Categorías y en Projects “Java Application” y pulsaremos Next



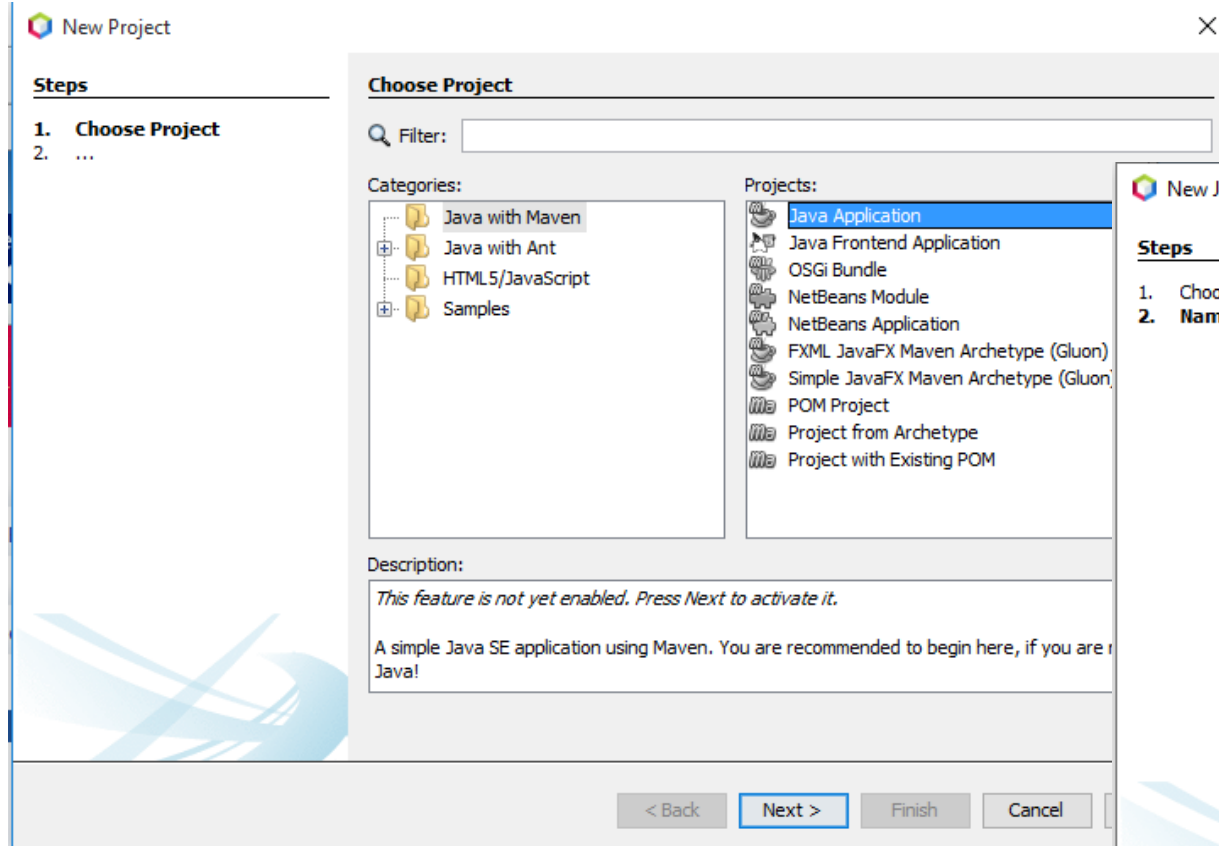
En esta ventana seleccionaremos uno o ambas opciones y pulsaremos “Download and activate” (para activar los componentes).

JavaFX es un Framework (conjunto de librerías), es el sucesor de Swing, para la programación de aplicaciones Web.

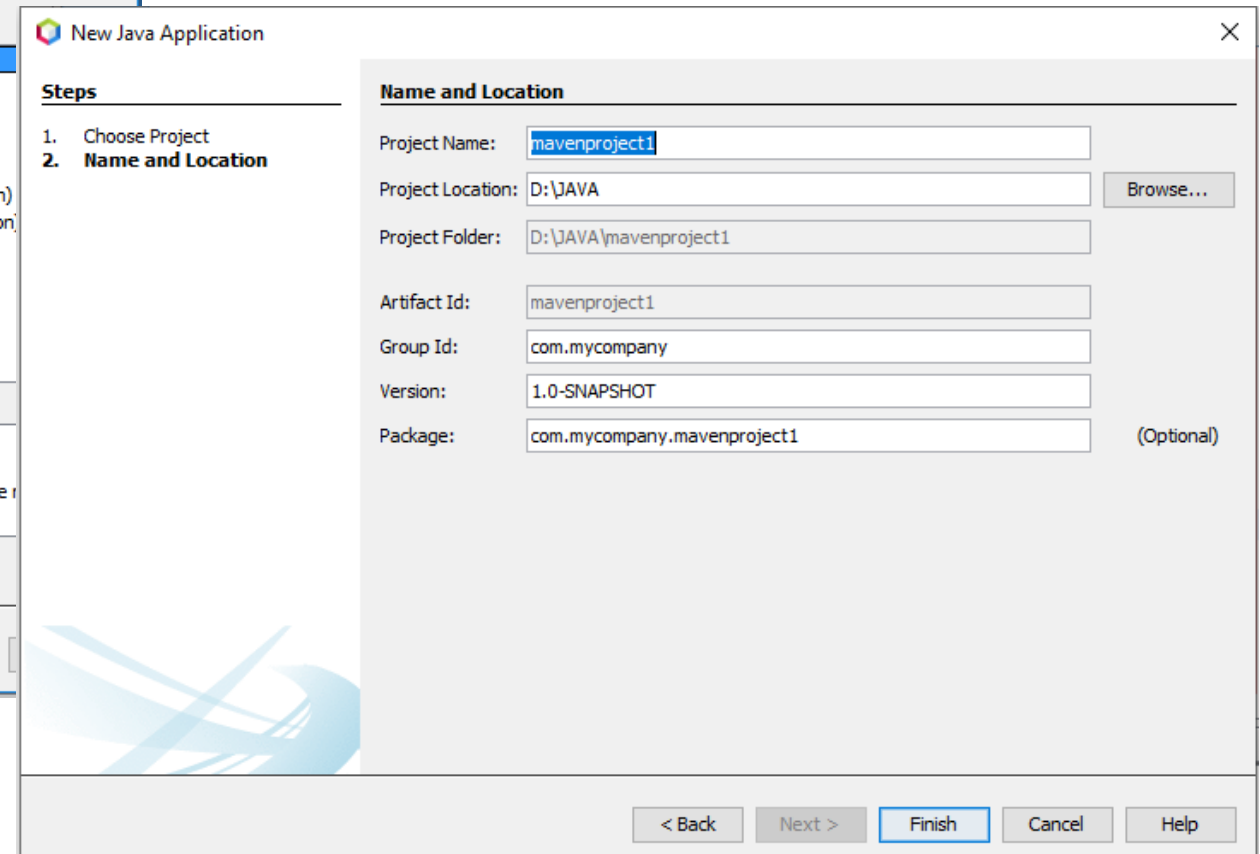
Si hemos marcado la librería JavaFX nos indicara que aceptemos los términos e instalemos, aunque no es necesario instalar este componente.

7. Uso básico de entornos de desarrollo.

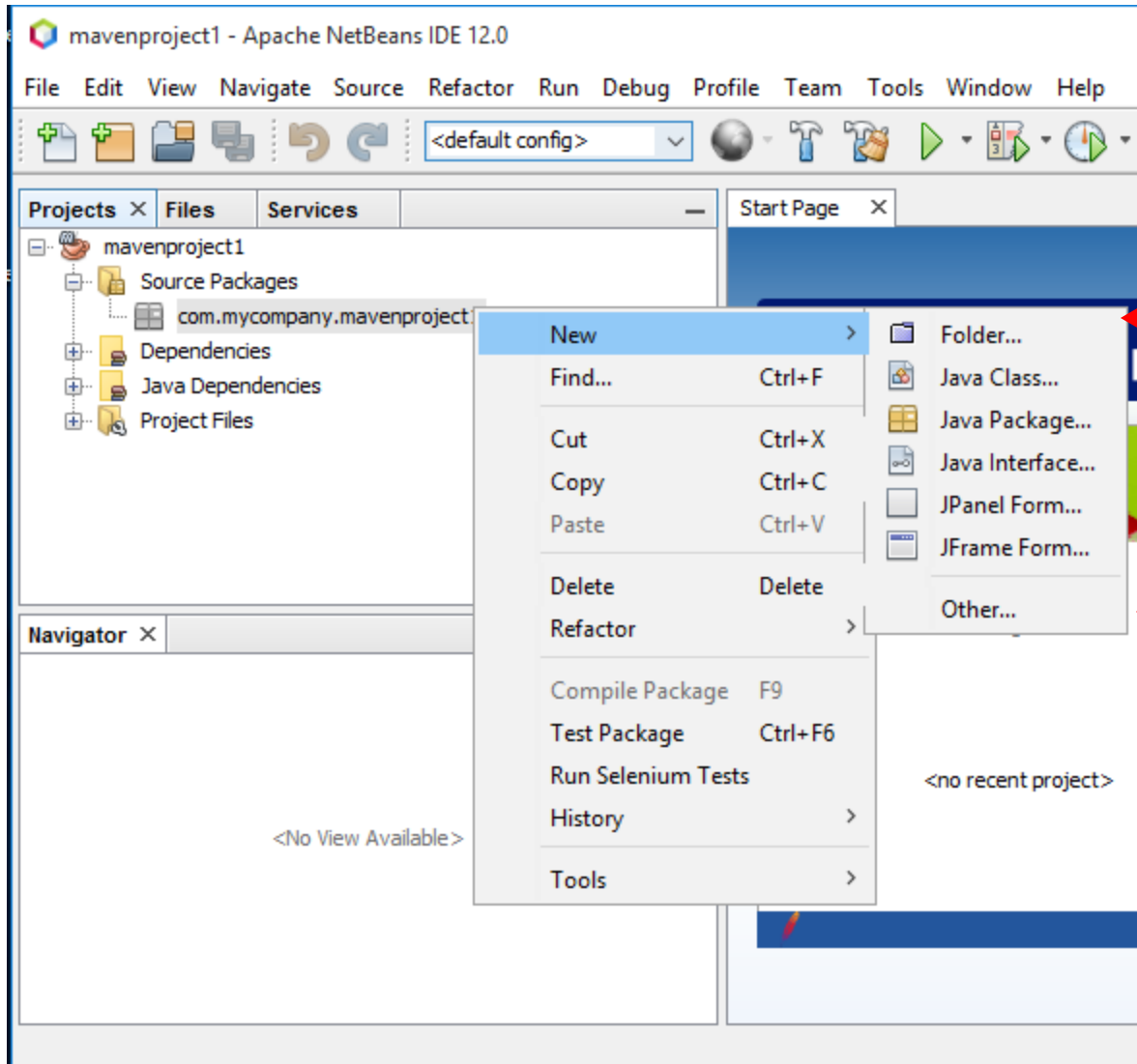
Uso básico de entornos de desarrollo



Creación de Proyectos en NetBeans



Uso básico de entornos de desarrollo



Crear elementos dentro del Proyecto

Crear una carpeta
Crear una Clase
Crear un Paquete

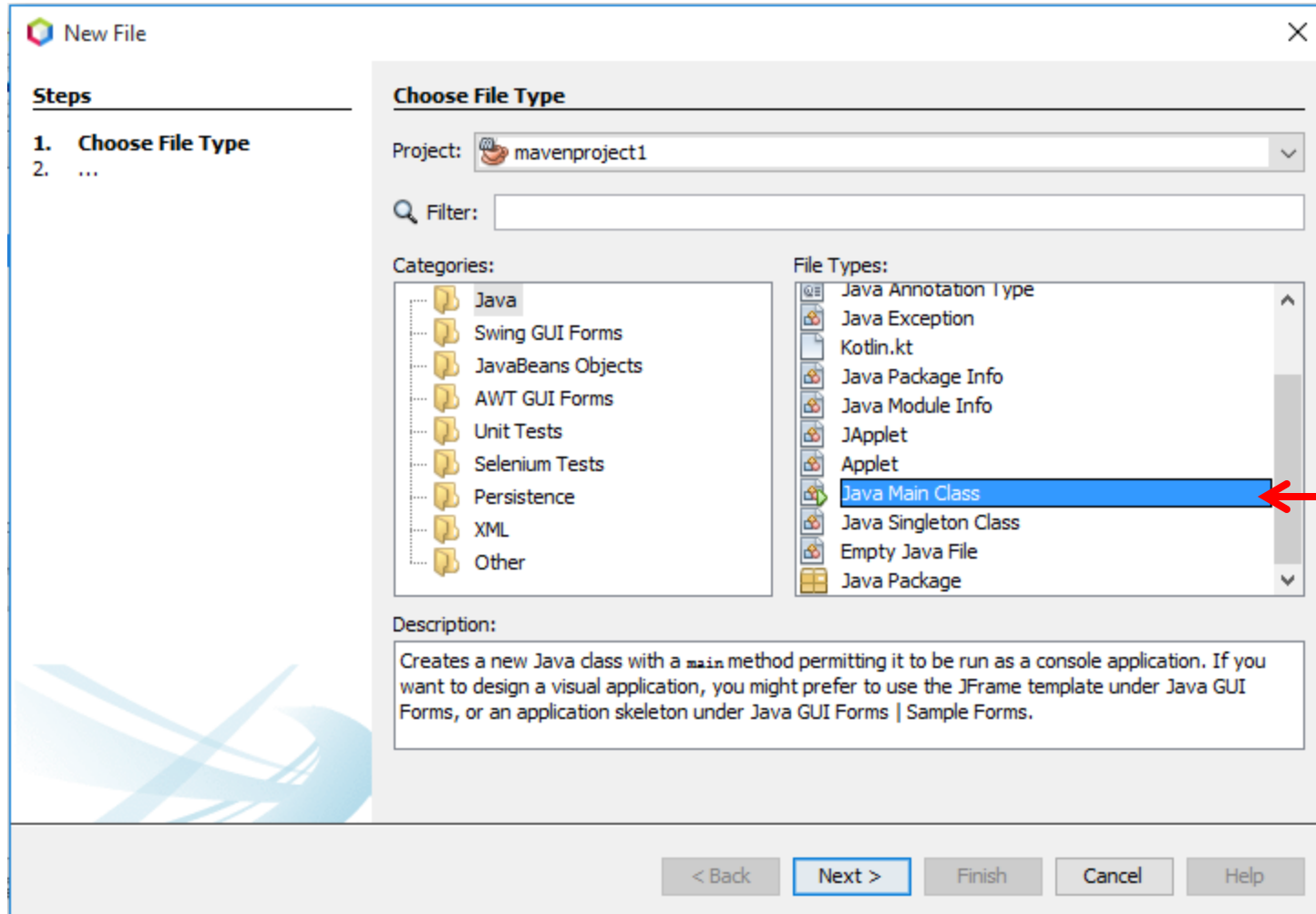
...
Crear un Panel
Crear una Ventana

} Aplicaciones basadas
en ventanas

Otros..



Uso básico de entornos de desarrollo

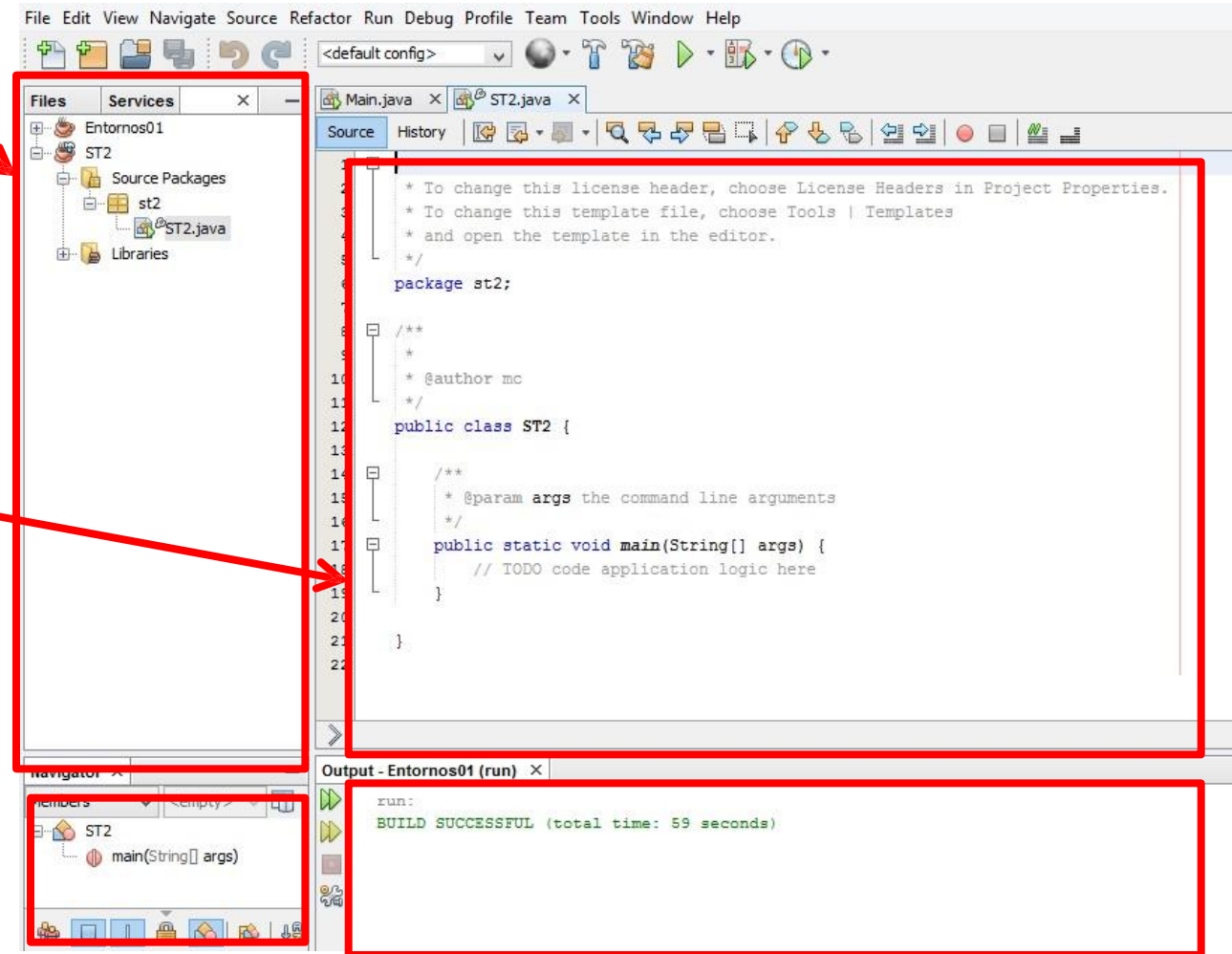


Creamos una Clase ejecutable

```
1  /**
2   * To change this license header, choose License Headers in P
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package com.mycompany.mavenproject1;
7
8   /**
9    *
10   * @author admin
11   */
12   public class NewMain {
13
14       /**
15        * @param args the command line arguments
16        */
17       public static void main(String[] args) {
18           // TODO code application logic here
19       }
20   }
```

Uso básico de entornos de desarrollo

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



Uso básico de entornos de desarrollo. Generación de ejecutables.

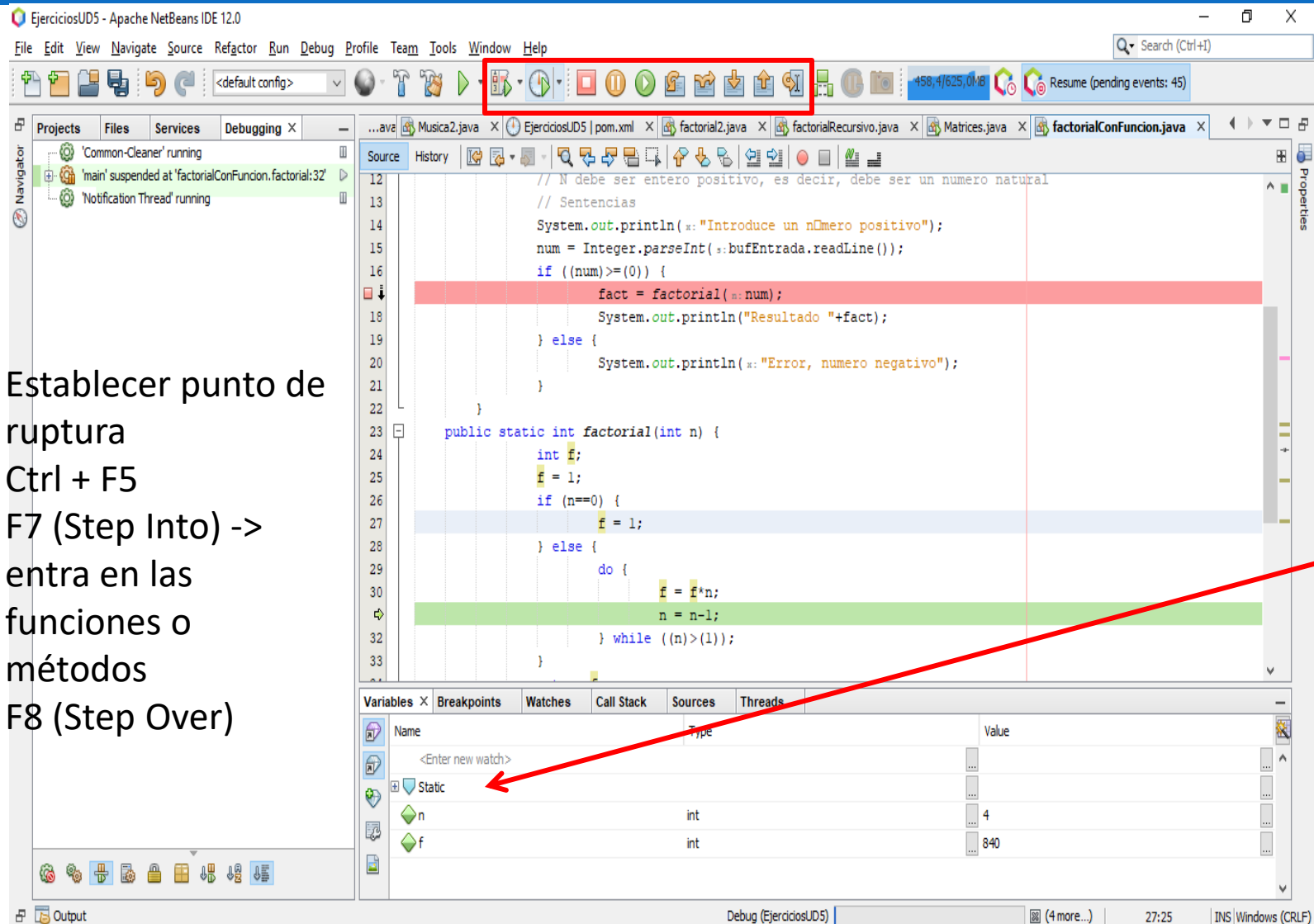
- Siguiendo pasos son:
 - ▣ Compilación
 - ▣ Depuración
 - ▣ Ejecución

Uso básico de entornos de desarrollo. Edición de programas

- Tarea. Crear programa “hola, mundo”
 - Crear un proyecto.
 - Editar código: crear la clase “Hola Mundo”
 - Compilar y ejecutar.

Uso básico de entornos de desarrollo. Depuración de programas

1. Establecer punto de ruptura
2. Ctrl + F5
3. F7 (Step Into) -> entra en las funciones o métodos
4. F8 (Step Over)



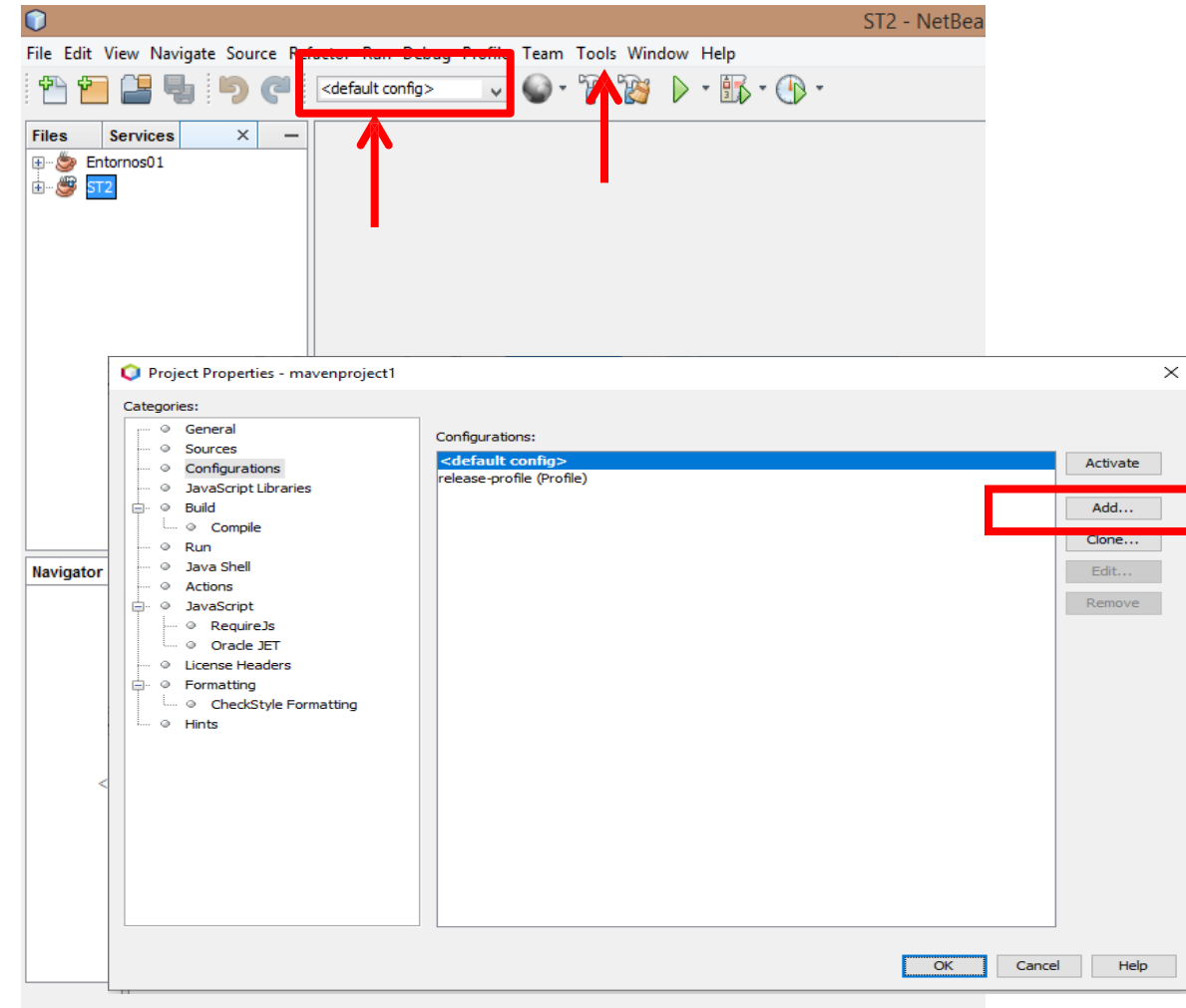
Podremos ver el valor de las variables

8. Configuración y personalización de entornos de desarrollo

Configuración y personalización de entornos de desarrollo

- Es posible personalizar la configuración de NetBeans para cada proyecto o para todos los proyectos.
 - Seleccionar el desplegable con el nombre de “configuración”.
 - Botón Dcho sobre el proyecto->Propiedades
- En cuanto a la configuración del entorno:
 - Menú Tools → Options

[Video de Fco. Hernandez Pérez \(ED. A Distancia\). Configuración NetBeans.](#)

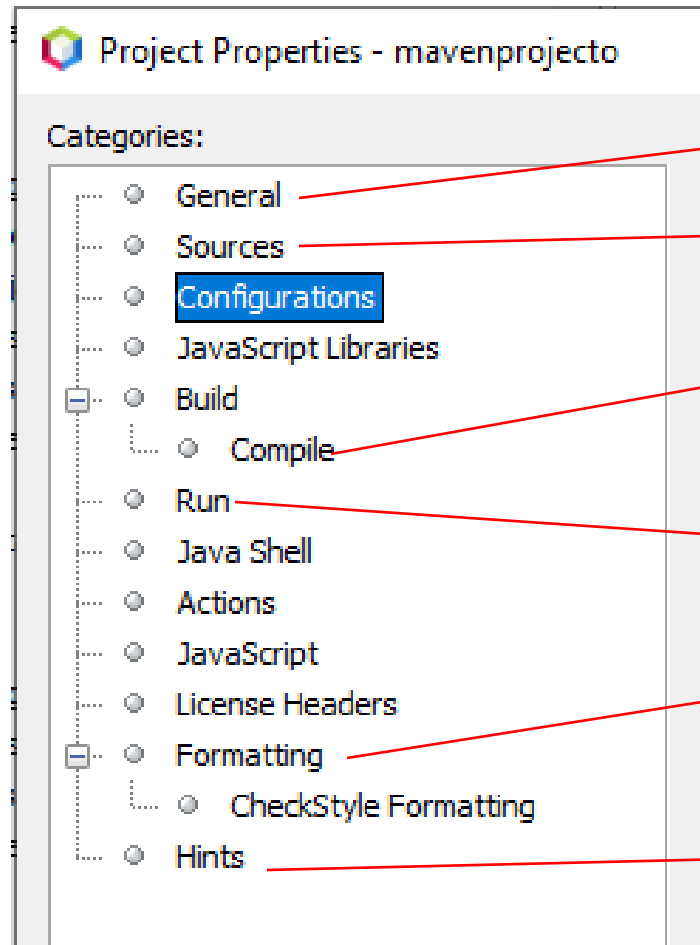


Configuración y personalización de entornos de desarrollo

□ **Parámetros configurables del entorno**

- Carpeta o carpetas donde se alojarán todos los archivos de los proyectos, paquetes fuente y test de pruebas (Source)
- Administración de la plataforma del entorno de desarrollo. (Compile)
- Opciones de la compilación de los programas: compilar al grabar, generar información de depuración. (Compile)
- Opciones de empaquetado de la aplicación: nombre del archivo empaquetado (con extensión .jar) y momento del empaquetado. (Actions)
- Opciones de generación de documentación asociada al proyecto. (Actions)
- Descripción de los proyectos, para una mejor localización de los mismos. (General)
- 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. (Formatting)
- Opciones de combinación de teclas en teclado. (Formatting)
- Plantillas de Código (abreviaciones para bloques de código)

Configuración de Netbeans



☐ Opciones de la aplicación.

☐ Fuentes.

☐ Bibliotecas y Generación de código.

☐ Ejecución de código.

☐ Formato del código en el editor de textos.

☐ Otros

Configuración de Netbeans

□ **Código fuente:**

- ▣ 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.

□ **Ejecución de 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 genera automáticamente por NetBeans al inicio de la sesión.

□ 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.

□ Plantillas de código

- ▣ sout para `System.out.println(..)`
- ▣ main para `public static main (String [] args) {}`
- ▣ ...

Configuración de Netbeans

- Tarea 07.- Configurar el entorno
 - Abrir un proyecto.
 - Configuración del entorno:
 - Autocompletado de código.
 - Plantillas de código.
 - Refactorización
 - Configuración Editor de texto
 - Abrir Ventanas Cerradas
 - Compilar al guardar.

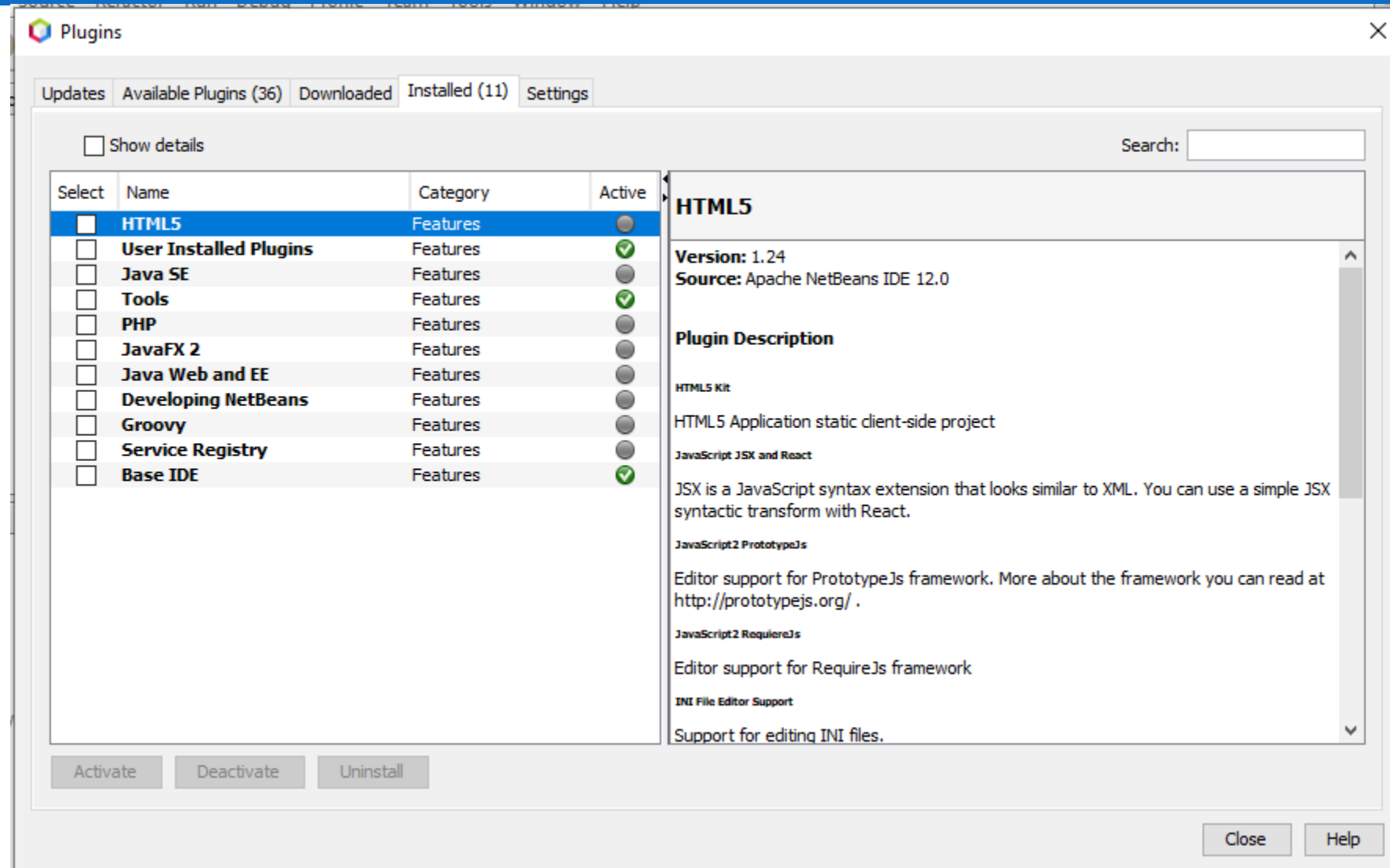
8. Gestión de módulos

Gestión de módulos

- NetBeans puede ser extendido a través de plugins o módulos.
- Para añadir un nuevo módulo tenemos varias opciones:
 - Módulos que NetBeans instala por defecto (imagen1)
 - Módulos que descargamos desde un sitio web permitido y añadimos al IDE (imagen 2):
 - Formato .nbm
 - Se denomina **adición off-line**: <http://plugins.netbeans.org/>
 - Módulos que instalamos de **forma on-line** en el entorno (imagen3)
 - Requiere tener instalado el plugin Portal Update Center
 - Instala complementos desde nuestro mismo IDE, sin descargarlos previamente.
 - Módulos desarrollados por nosotros mismos.

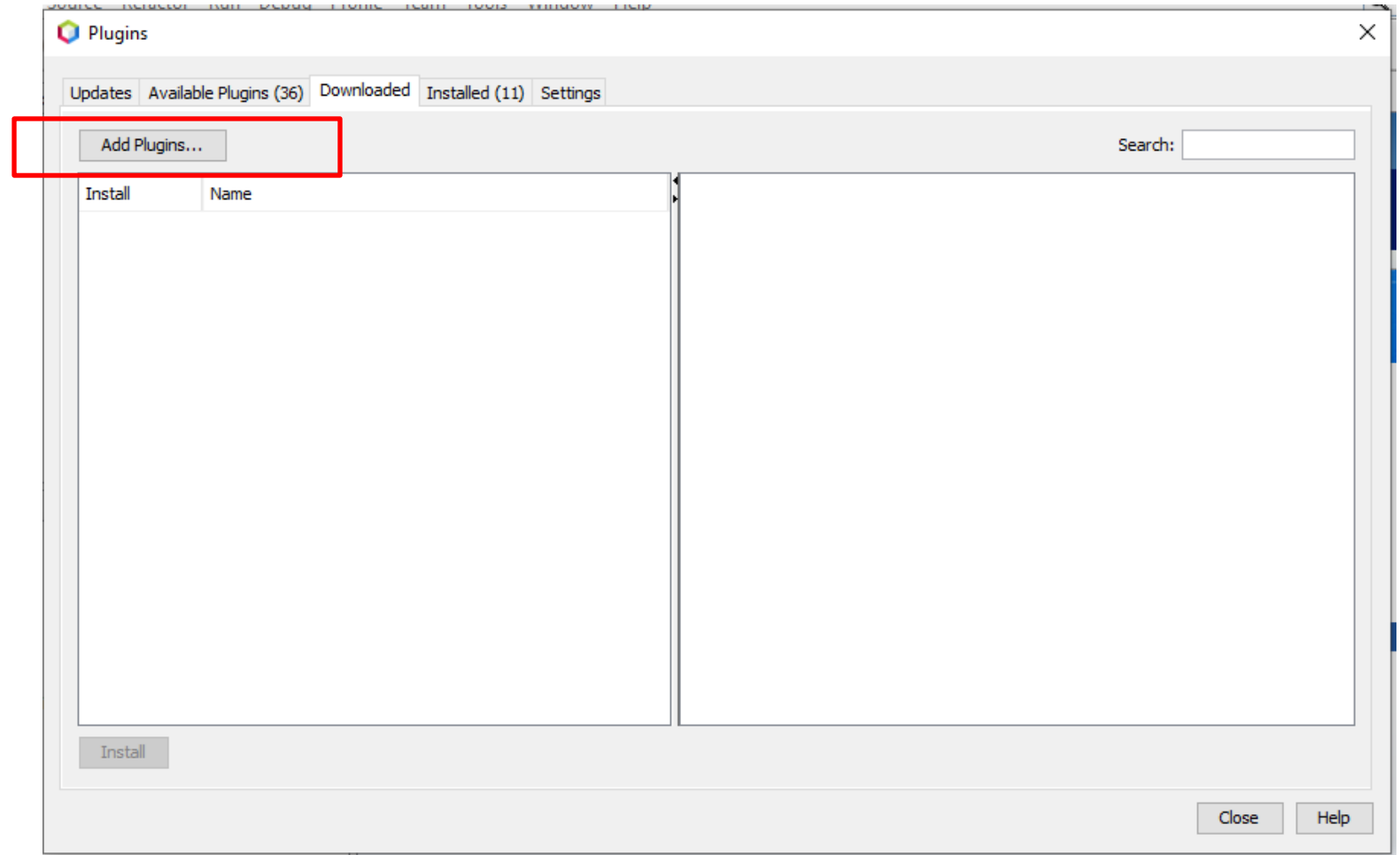
Gestión de módulos

□ Imagen 1



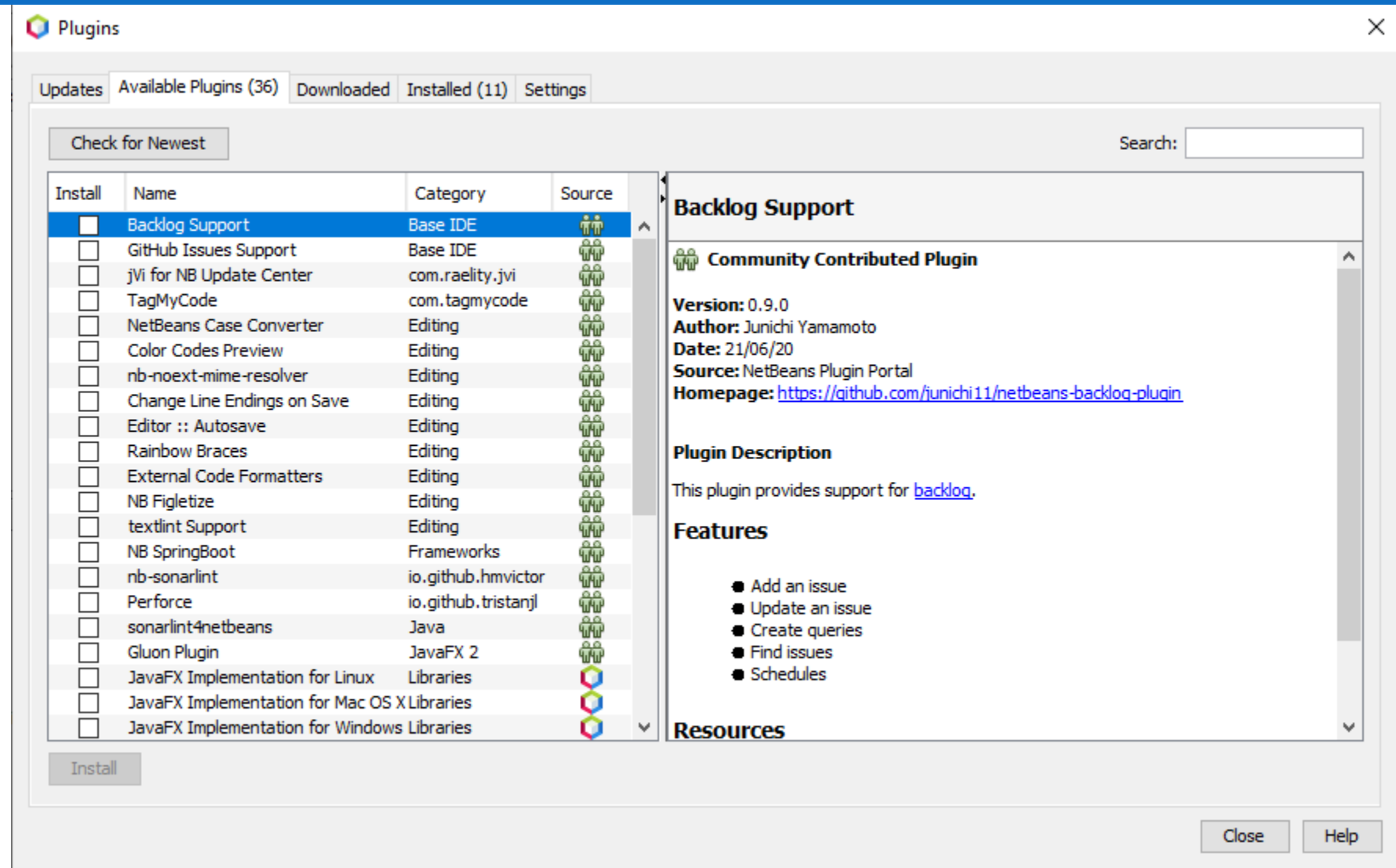
Gestión de módulos

□ Imagen 2



Gestión de módulos

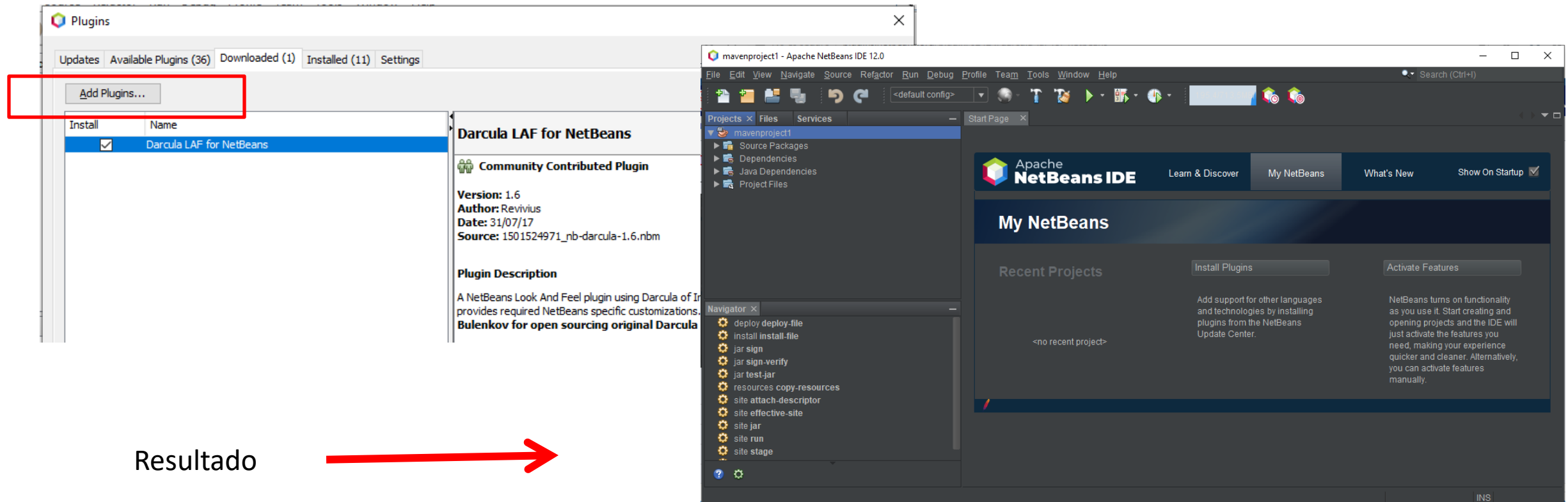
□ Imagen 3



Gestión de módulos

□ Ejemplo 1: instalación offline -> Plugin Darcula

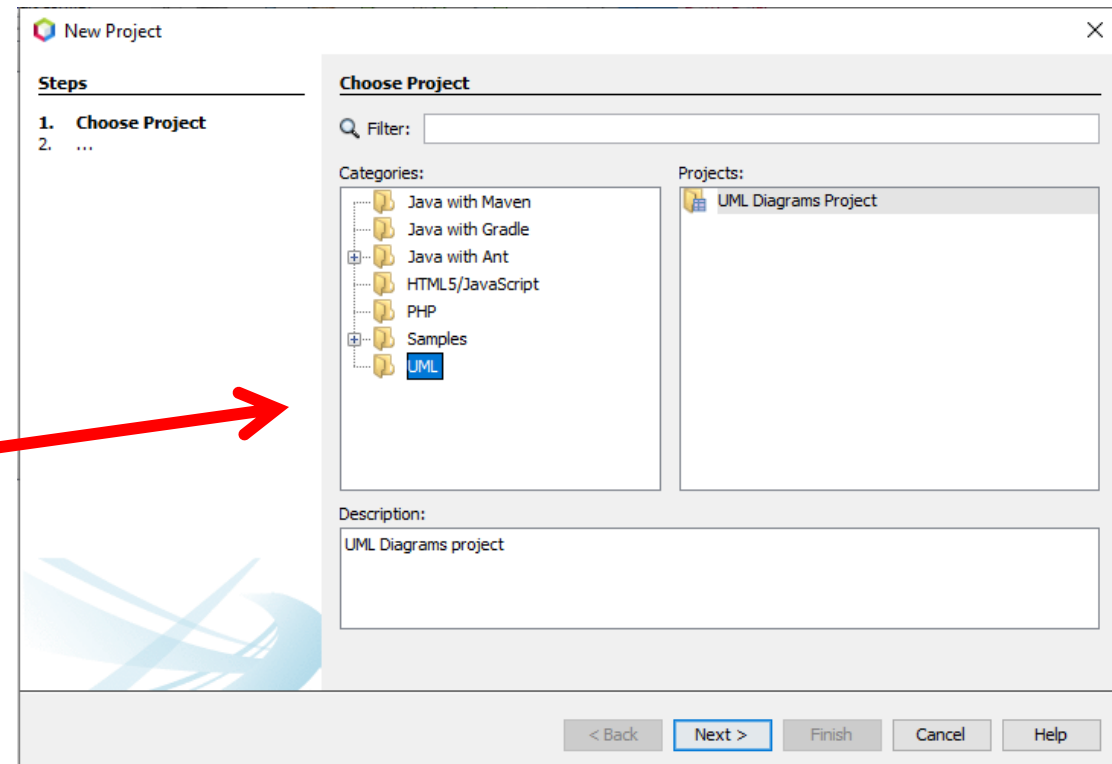
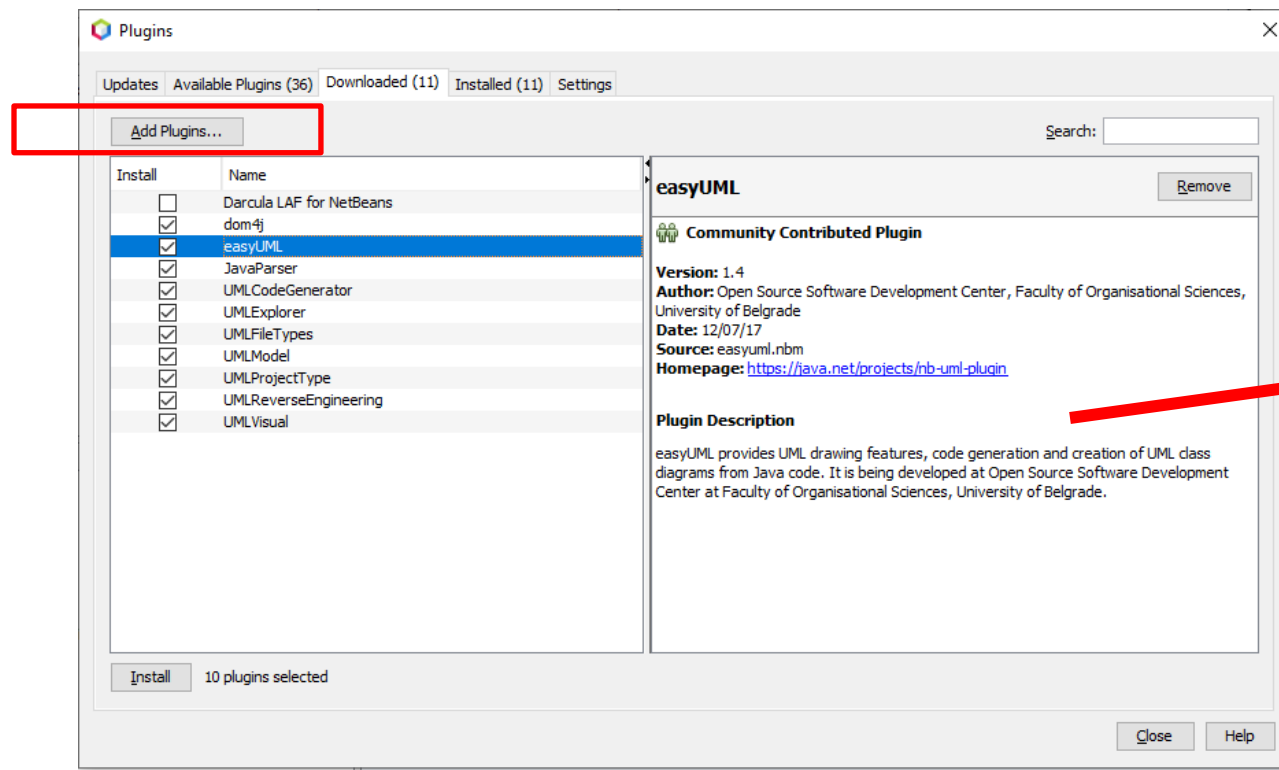
- Descargamos el plugin desde un sitio oficial como <http://plugins.netbeans.org/> -> Fichero .nbm
- Menú Tools/Plugins -> Pestaña Downloaded



Gestión de módulos

□ Ejemplo 2: instalación offline -> Plugin EasyUML

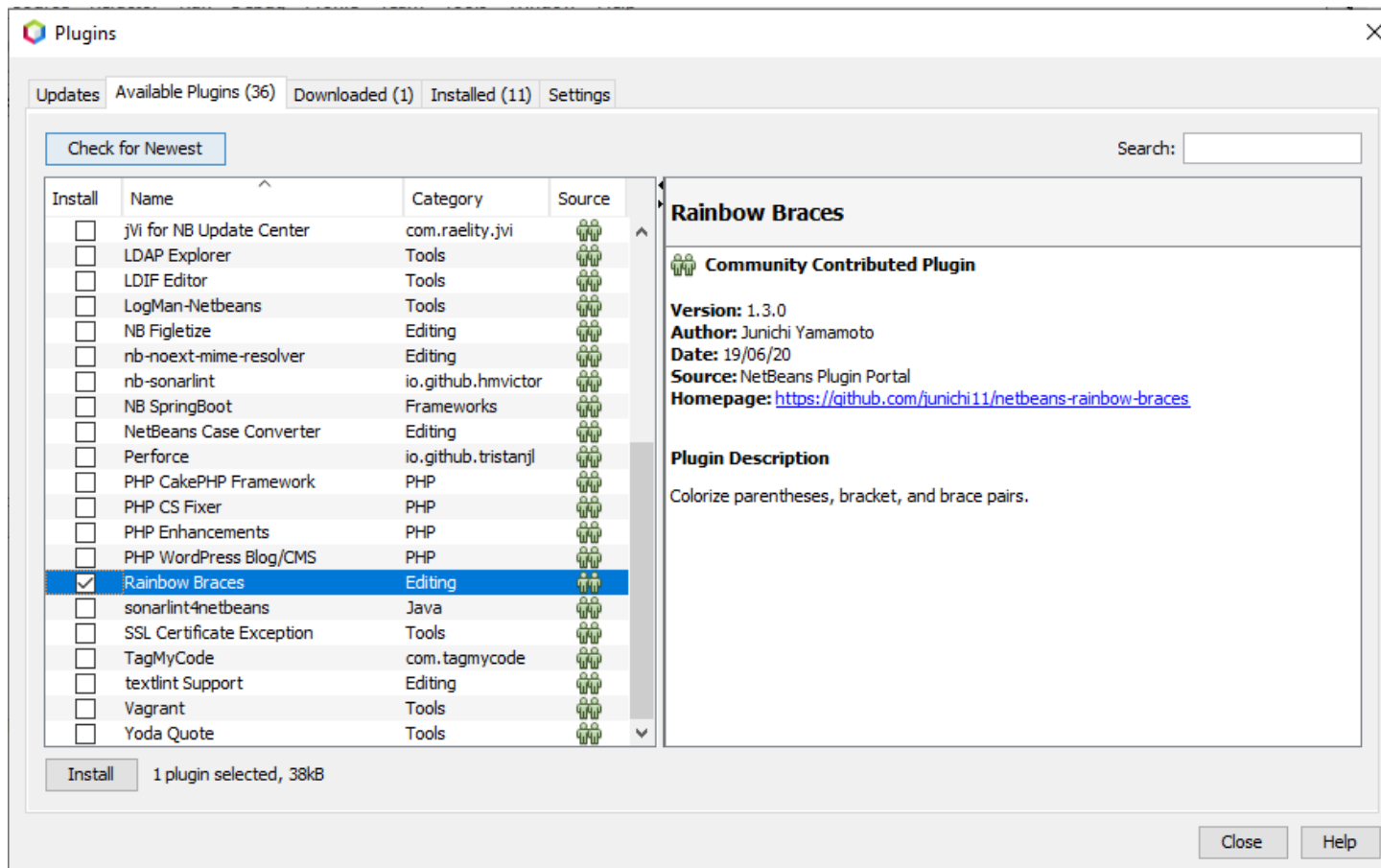
- Descargamos el plugin desde un sitio oficial como <http://plugins.netbeans.org/> -> Fichero .zip y lo descomprimos
- Menú Tools/Plugins -> Pestaña Downloaded



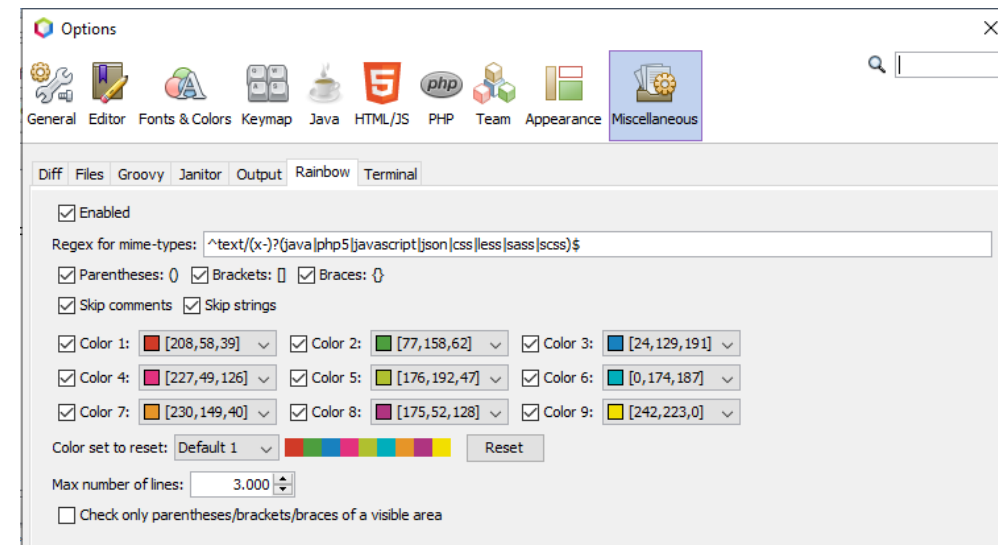
Gestión de módulos

□ Ejemplo 3: instalación online-> Rainbow Braces

➤ Menú Tools/Plugins -> Pestaña Available Plugins



A partir de su instalación observamos determinados elementos se colorean de manera anidada. Podemos configurar este Plugin desde: Tools/options /Miscellaneous



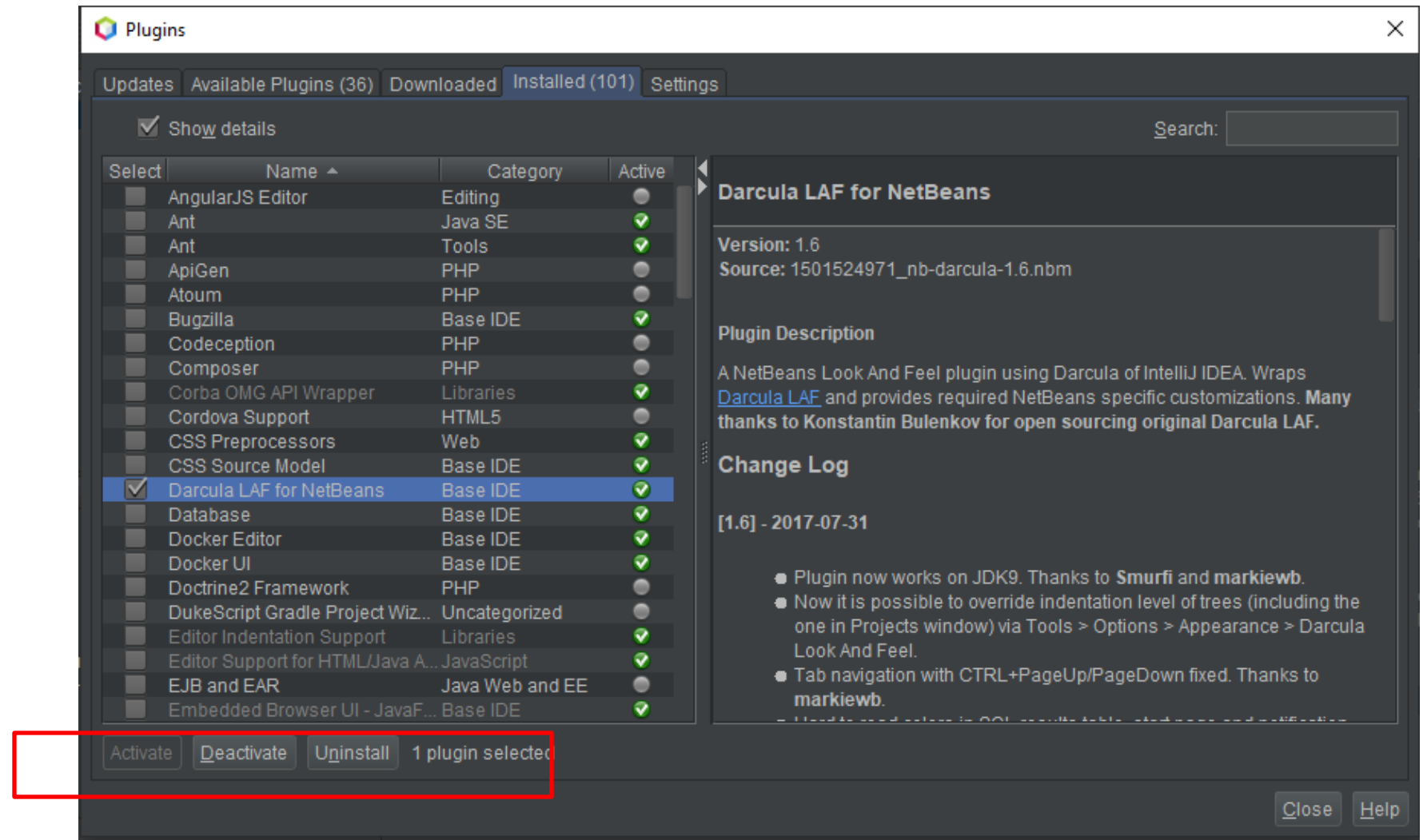
Gestión de módulos

- 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.

[Vídeo de Fco. Hernández \(ED. A Distancia\). Instalar y desinstalar Plugins en NetBeans.](#)

Gestión de módulos

Ejemplo de desinstalación de un Plugin:



Gestión de módulos

- **Categorías y funcionalidades de módulos y plugins:**
 - ▣ **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.

Gestión de módulos

- **Tarea 08:** investiga sobre el tema de plugins para NetBeans indicando dos de ellos que te parezcan interesantes o curiosos. A ser posible uno cuya instalación se realice de forma offline y otro online y explica la utilidad de ambos, aportando ejemplos de uso y capturas de pantalla, tanto de su instalación como de su utilización dentro de la herramienta.
- Esta práctica se entregará el 11 de diciembre a las 15h y se valorará como una nota de examen.
- Se valorará:
 - la originalidad (se penalizará la no existencia de este factor)
 - La utilidad del plugin
 - La documentación aportada y ejemplos de uso.

10.- 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.
- ❑ Una vez instalado, nos permitirá realizar continuas revisiones del entorno y actualizaciones de todos los plugins.

Actualización y mantenimiento de entornos de desarrollo.

