

APACHE NETBEANS

Contornos de Desenvolvimento

Desenvolvimento de Aplicacíons Multiplataforma. Curso 2024/2025

Álvarez Rodríguez, Fabián

Pena Piñeiro, Aitor

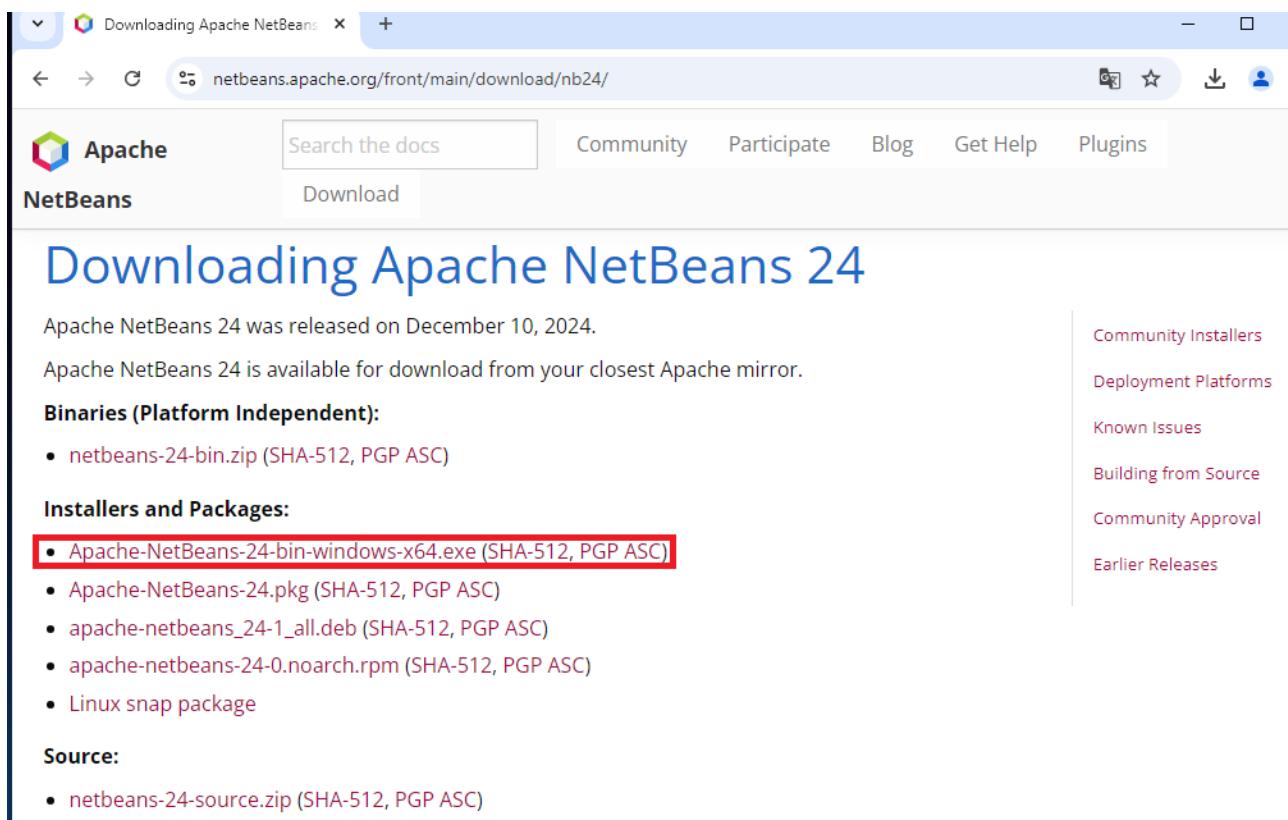
Resumen	3
Guia de Instalación	3
Interfaz	6
Opciones de personalización del IDE.	12
Creación del Proyecto	13
Configuración del Proyecto	20
Plugins	25
GIT	27
Asistente de IA	30

Resumen

NetBeans fundada por Sun Microsystems es un IDE (Integrated Development Environment) o un entorno de desarrollo integrado libre, es decir es un programa para ejecutar código, principalmente hecho para java, es gratuito, es de código abierto, NetBeans funciona a través de módulos, un módulo es un archivo Java que contiene clases Java con código escrito para que se ejecute con las API (Application Programming Interface).

Se puede descargar desde la pagina oficial de Apache NetBeans y para los ordenadores de la clase usaremos el NetBeans 24.

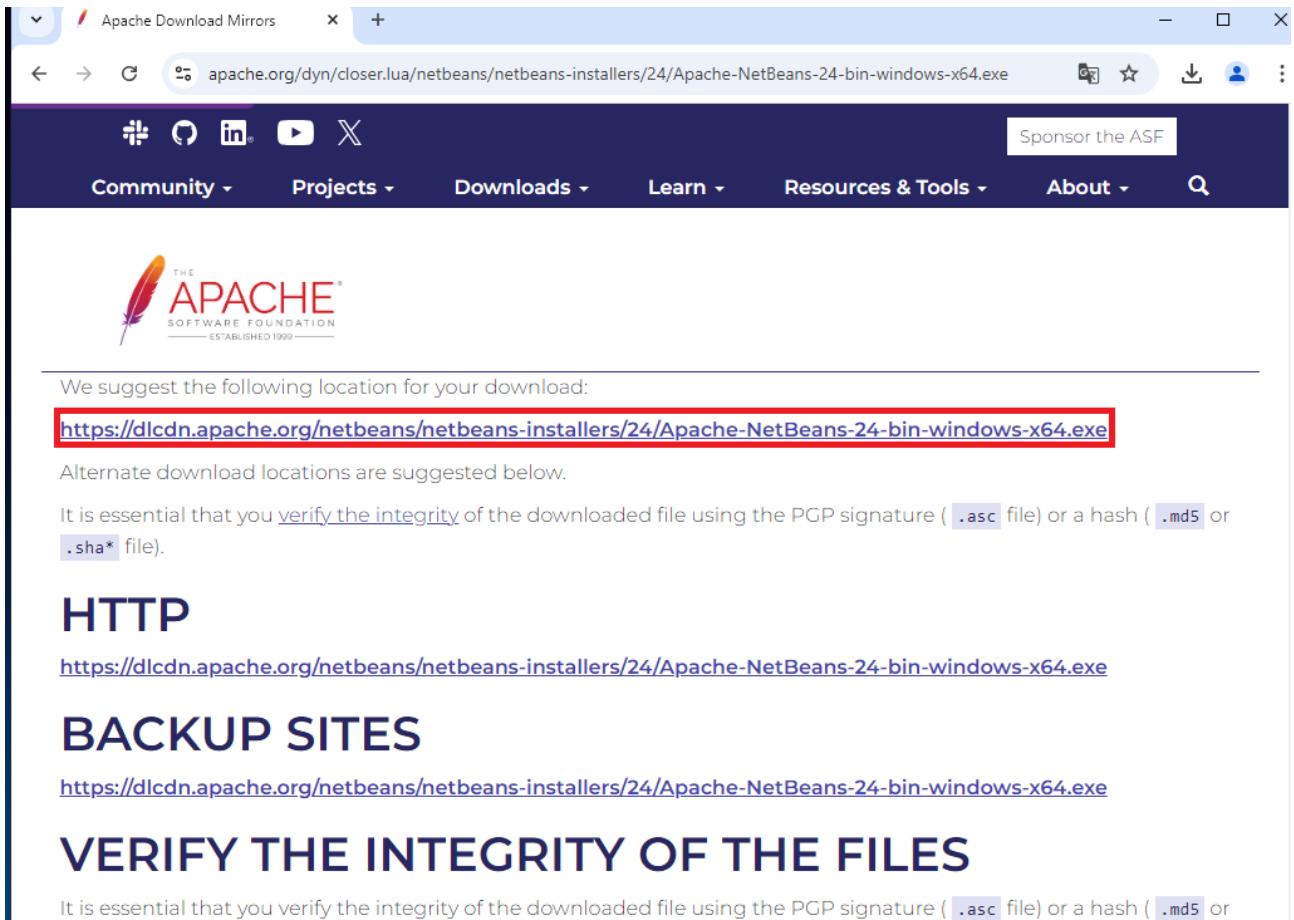
Guia de Instalación



The screenshot shows a web browser window with the title "Downloading Apache NetBeans" and the URL "netbeans.apache.org/front/main/download/nb24/". The page features the Apache NetBeans logo and navigation links for "Search the docs", "Community", "Participate", "Blog", "Get Help", and "Plugins". A prominent blue header reads "Downloading Apache NetBeans 24". Below it, a message states "Apache NetBeans 24 was released on December 10, 2024." and "Apache NetBeans 24 is available for download from your closest Apache mirror." A section titled "Binaries (Platform Independent)" lists a single item: "netbeans-24-bin.zip (SHA-512, PGP ASC)". Another section, "Installers and Packages:", lists several options, with the first one, "Apache-NetBeans-24-bin-windows-x64.exe (SHA-512, PGP ASC)", highlighted by a red border. To the right, a sidebar contains links for "Community Installers", "Deployment Platforms", "Known Issues", "Building from Source", "Community Approval", and "Earlier Releases". At the bottom, a "Source:" section lists "netbeans-24-source.zip (SHA-512, PGP ASC)".

Primero para utilizar el NetBeans tendremos que tener instalado el JDK el cual podremos descargar en Java Downloads en la página de Oracle y descargaremos la versión de JDK 21.

El entorno de desarrollo Apache NetBeans podemos descargarlo desde la página oficial de Apache. Buscamos en el navegador “Apache NetBeans”, y le damos a Download luego a Apache NetBeans 24, que es la versión actual, y seleccionamos esta opción de descarga “Apache-NetBeans-24-bin-windows-x64.exe (SHA-512, PGP ASC)”.



The screenshot shows a web browser window with the Apache Download Mirrors page for Apache NetBeans. The URL in the address bar is <https://apache.org/dyn/closer.lua/netbeans/installers/24/Apache-NetBeans-24-bin-windows-x64.exe>. The page features the Apache Software Foundation logo and navigation links for Community, Projects, Downloads, Learn, Resources & Tools, About, and a search bar. A red box highlights the download link <https://dlcdn.apache.org/netbeans/netbeans-installers/24/Apache-NetBeans-24-bin-windows-x64.exe>.

We suggest the following location for your download:

<https://dlcdn.apache.org/netbeans/netbeans-installers/24/Apache-NetBeans-24-bin-windows-x64.exe>

Alternate download locations are suggested below.

It is essential that you verify the integrity of the downloaded file using the PGP signature ([.asc](#) file) or a hash ([.md5](#) or [.sha*](#) file).

HTTP

<https://dlcdn.apache.org/netbeans/netbeans-installers/24/Apache-NetBeans-24-bin-windows-x64.exe>

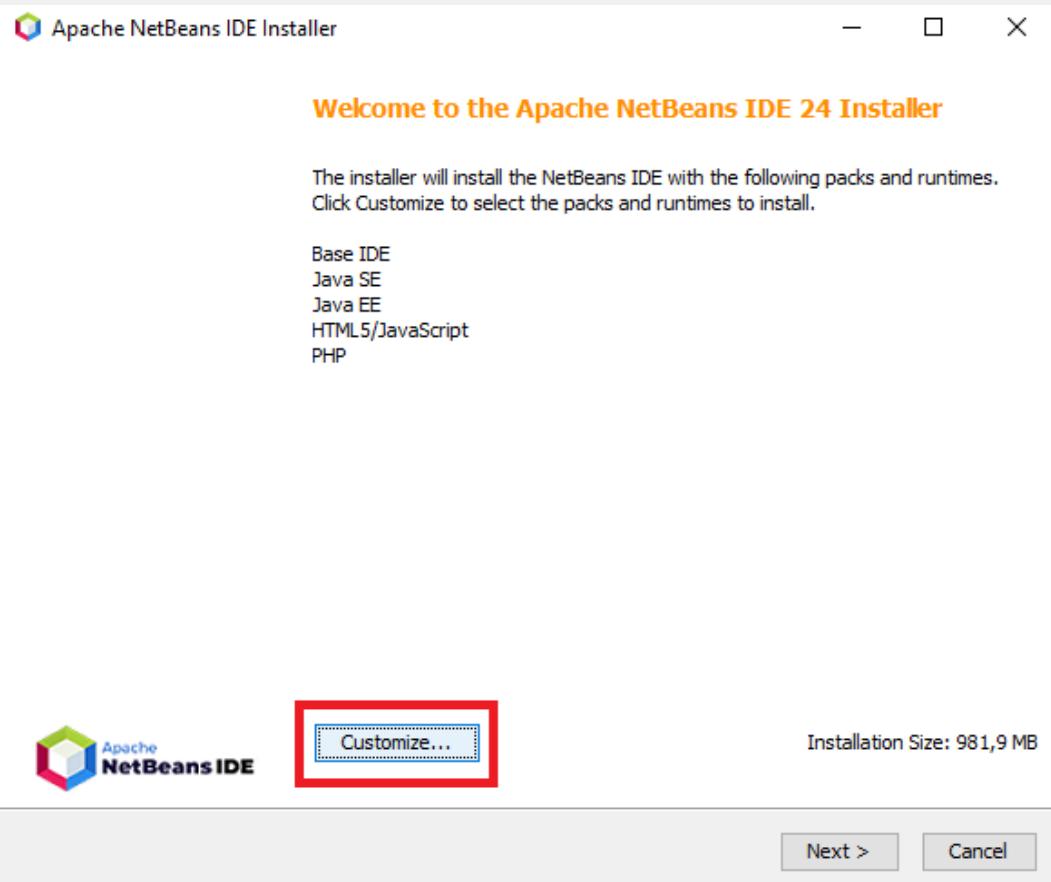
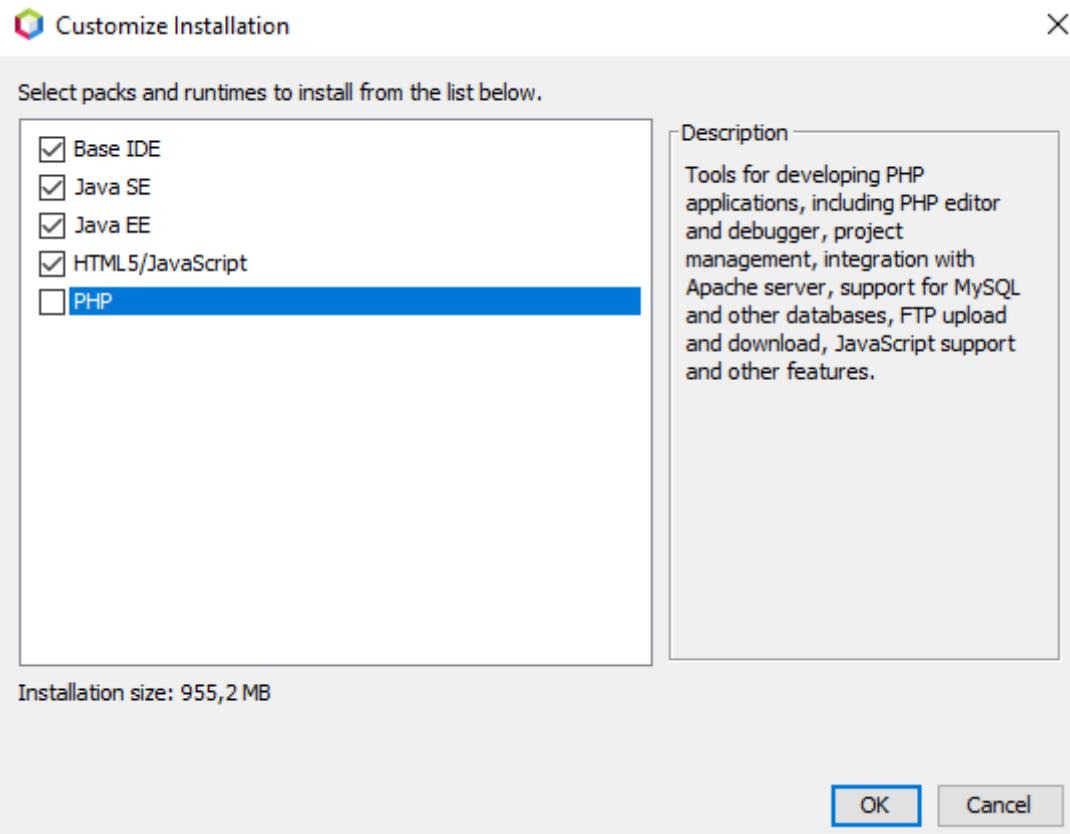
BACKUP SITES

<https://dlcdn.apache.org/netbeans/netbeans-installers/24/Apache-NetBeans-24-bin-windows-x64.exe>

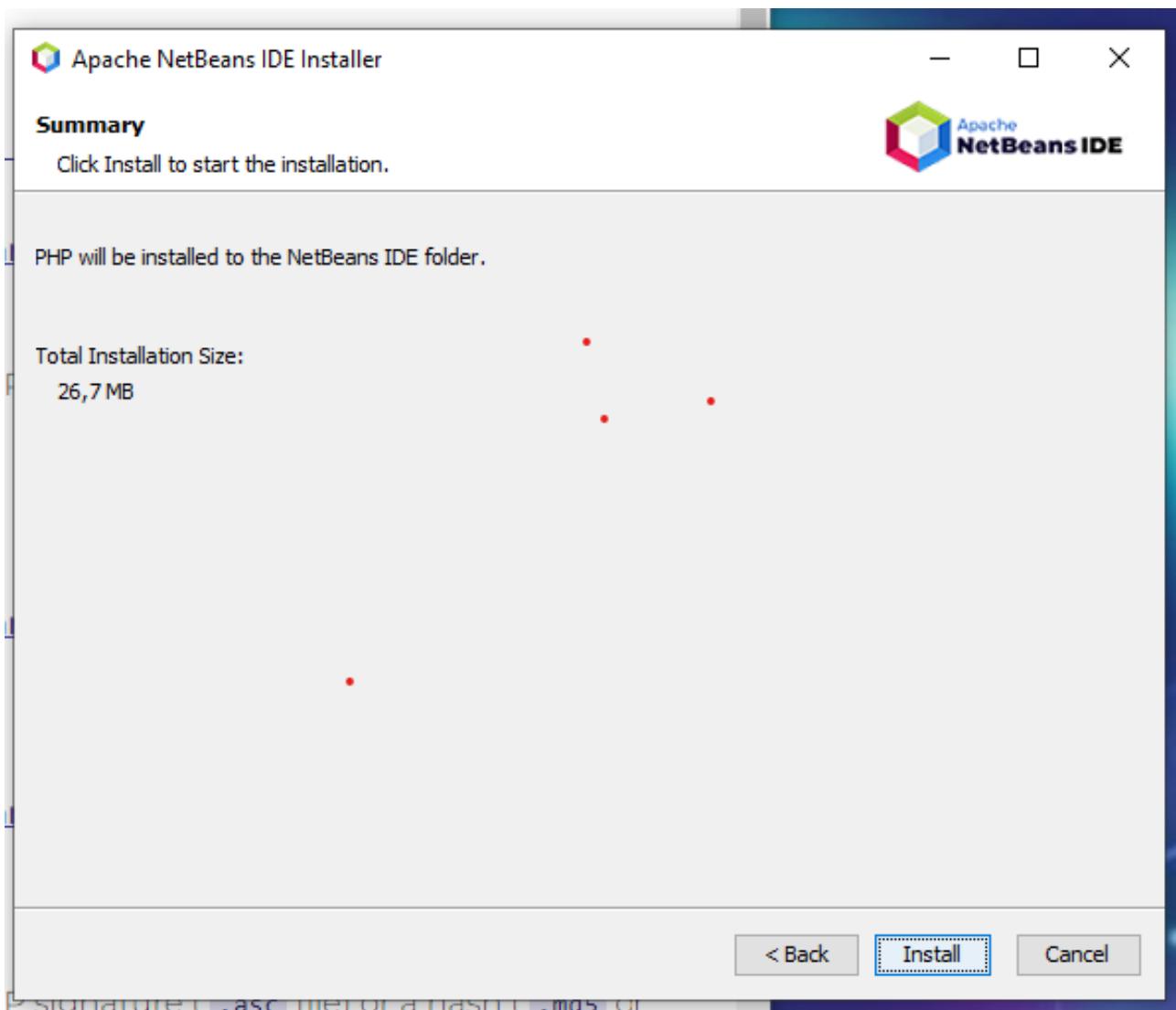
VERIFY THE INTEGRITY OF THE FILES

It is essential that you verify the integrity of the downloaded file using the PGP signature ([.asc](#) file) or a hash ([.md5](#) or [.sha*](#) file).

Nos redirigirá a otra pagina con el enlace para el instalador.

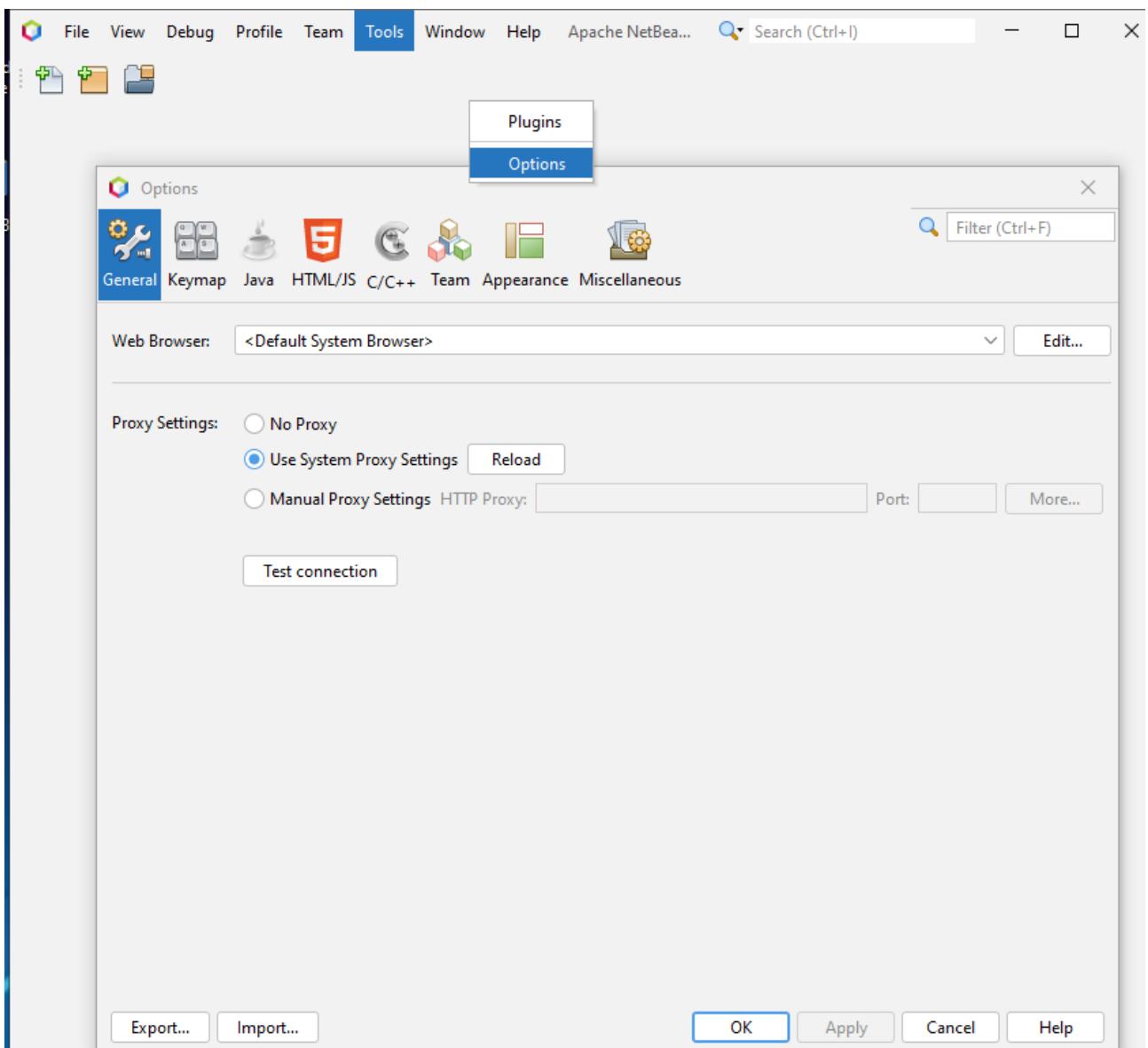


Y por último seleccionamos instalar.

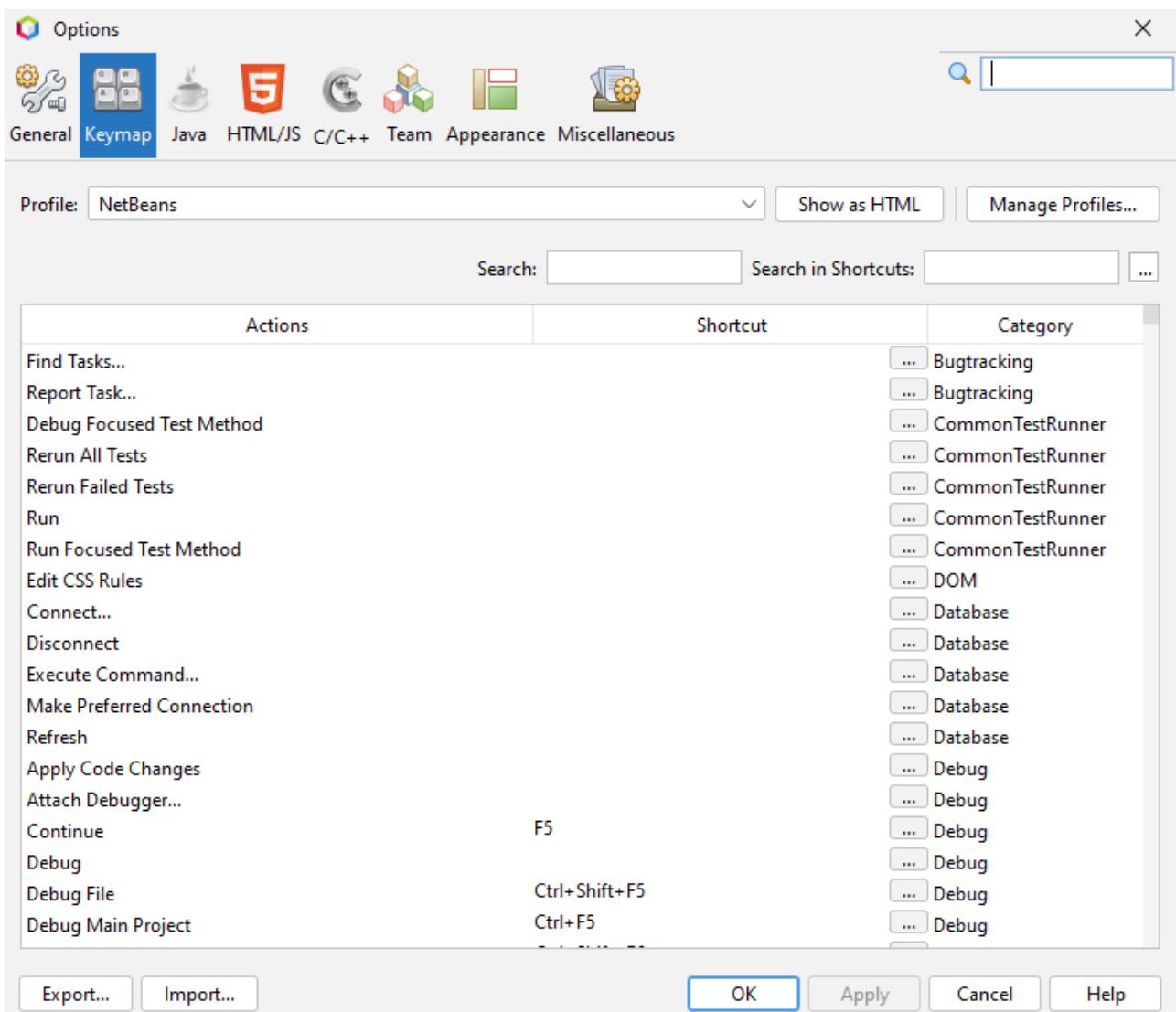


Interfaz

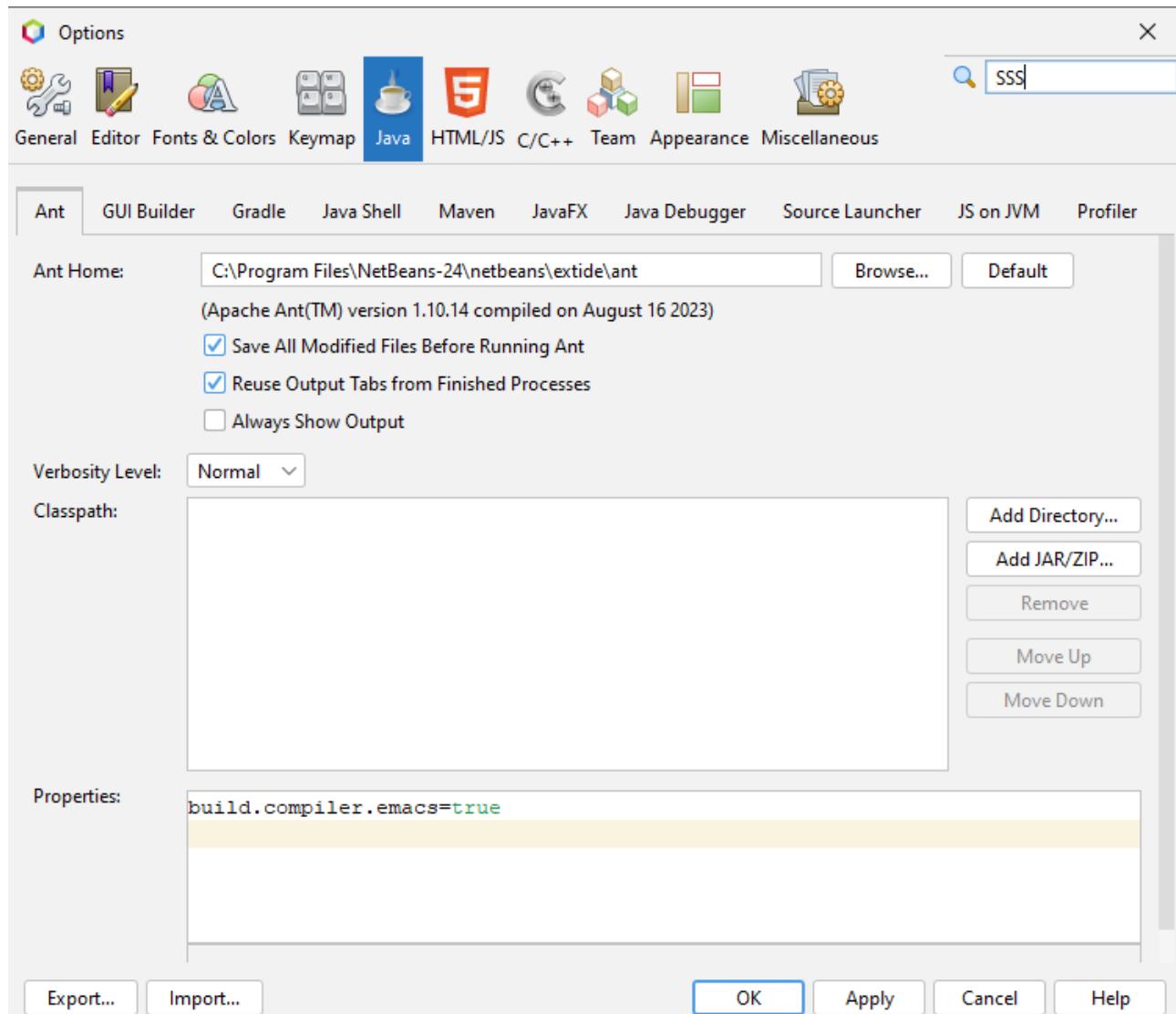
Para ir a la configuración de la interfaz iremos a “Tools” luego a “Options” y se nos abrirá el menú de la imagen. Primero observamos los ajustes generales en ella podemos cambiar el buscador que usa la aplicación y los ajustes del Proxy.



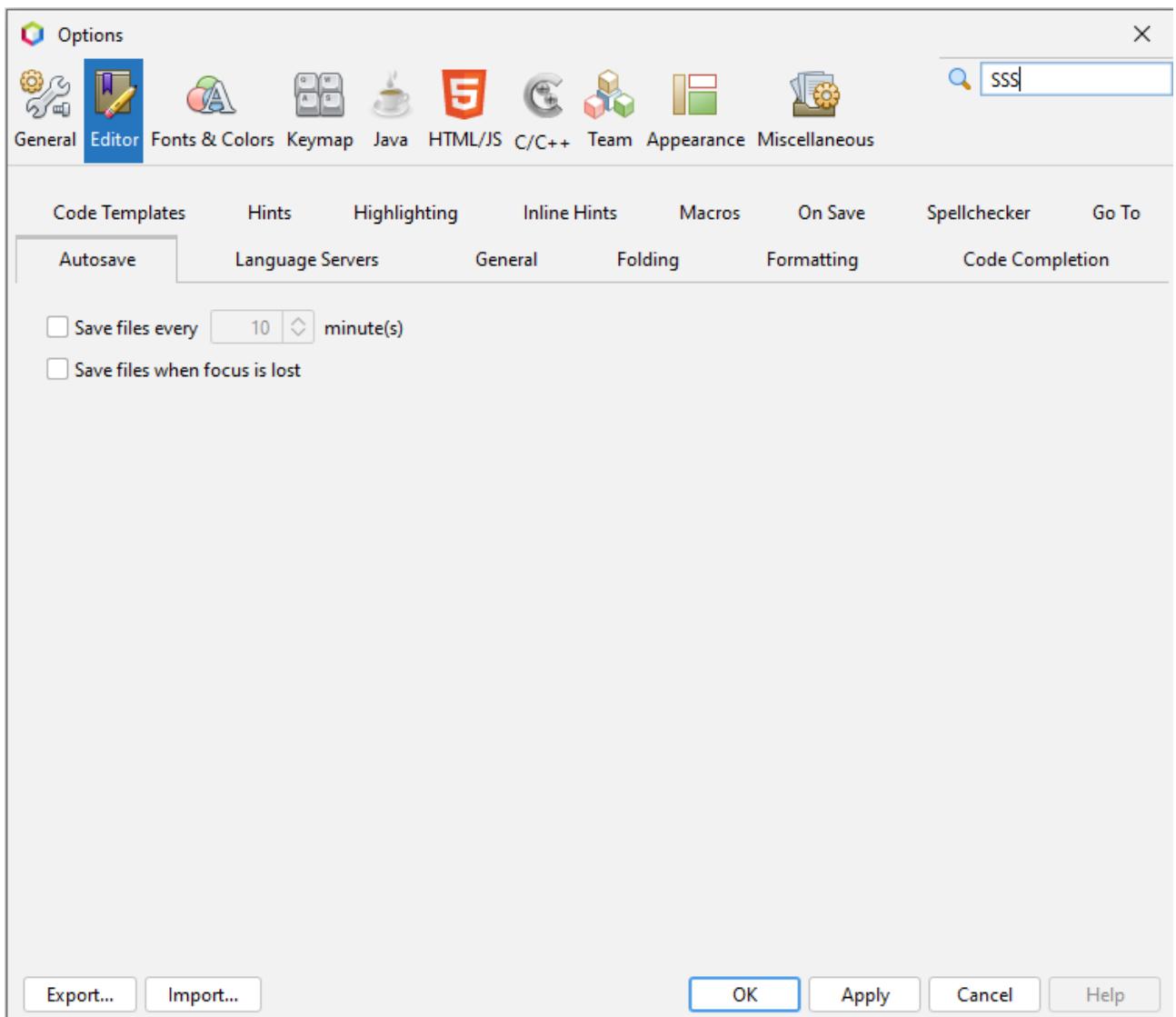
La segunda opción son los ajustes de personalización del teclado como los atajos del teclado.



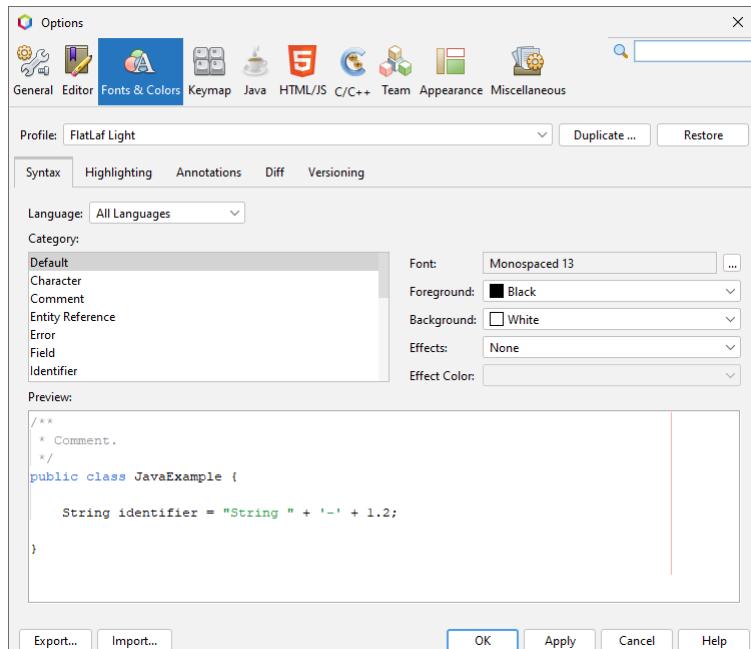
En la opción de Java podemos cambiar opciones del compilador.



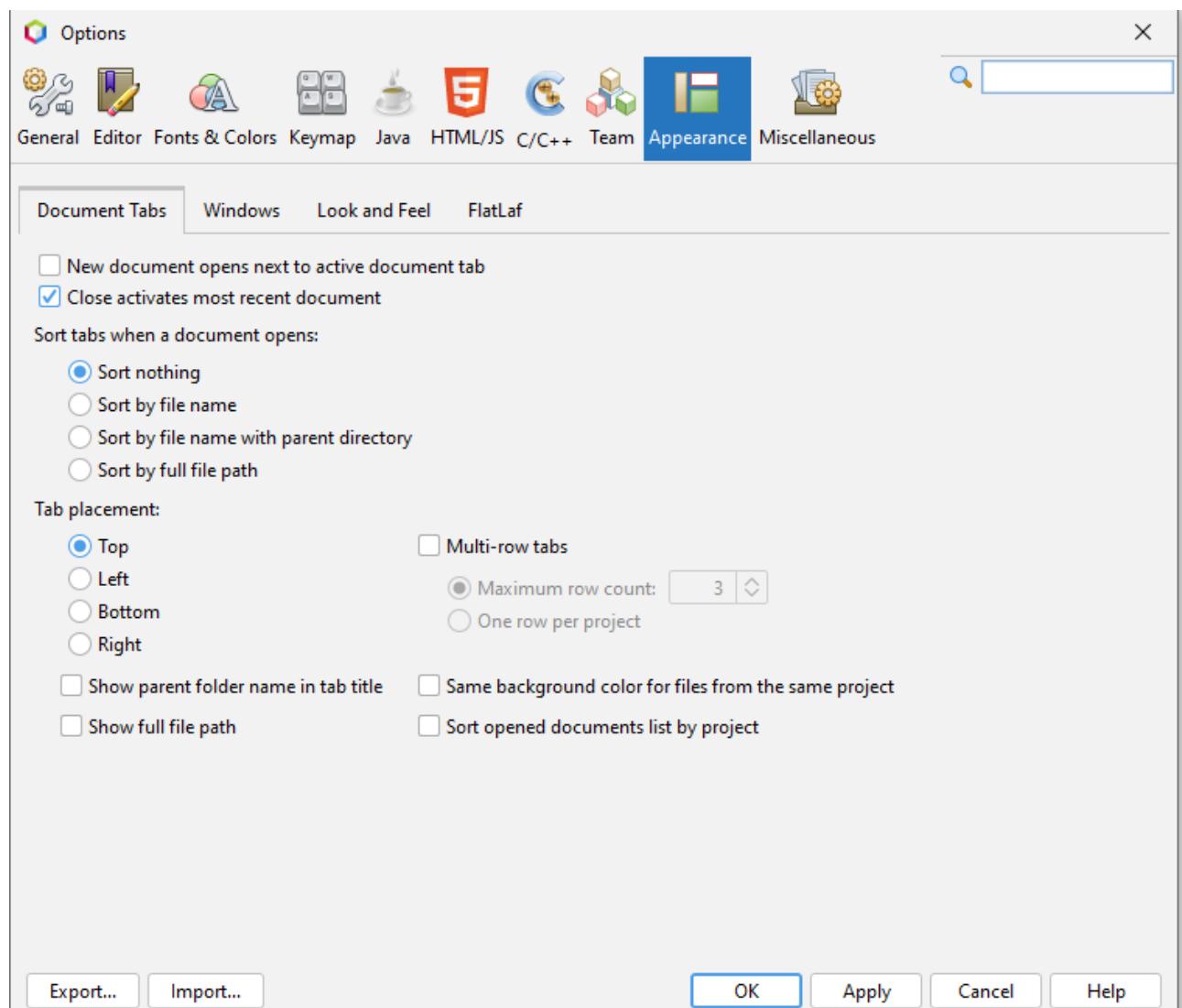
En la opción de editor tenemos varias opciones. Las más interesantes son el autoguardado por tiempo y los macros que sirven para pulsar un botón y que se ejecute lo que tenemos programado que puede ser texto o opciones como poner 10 breakpoints en X líneas y DeBugear.



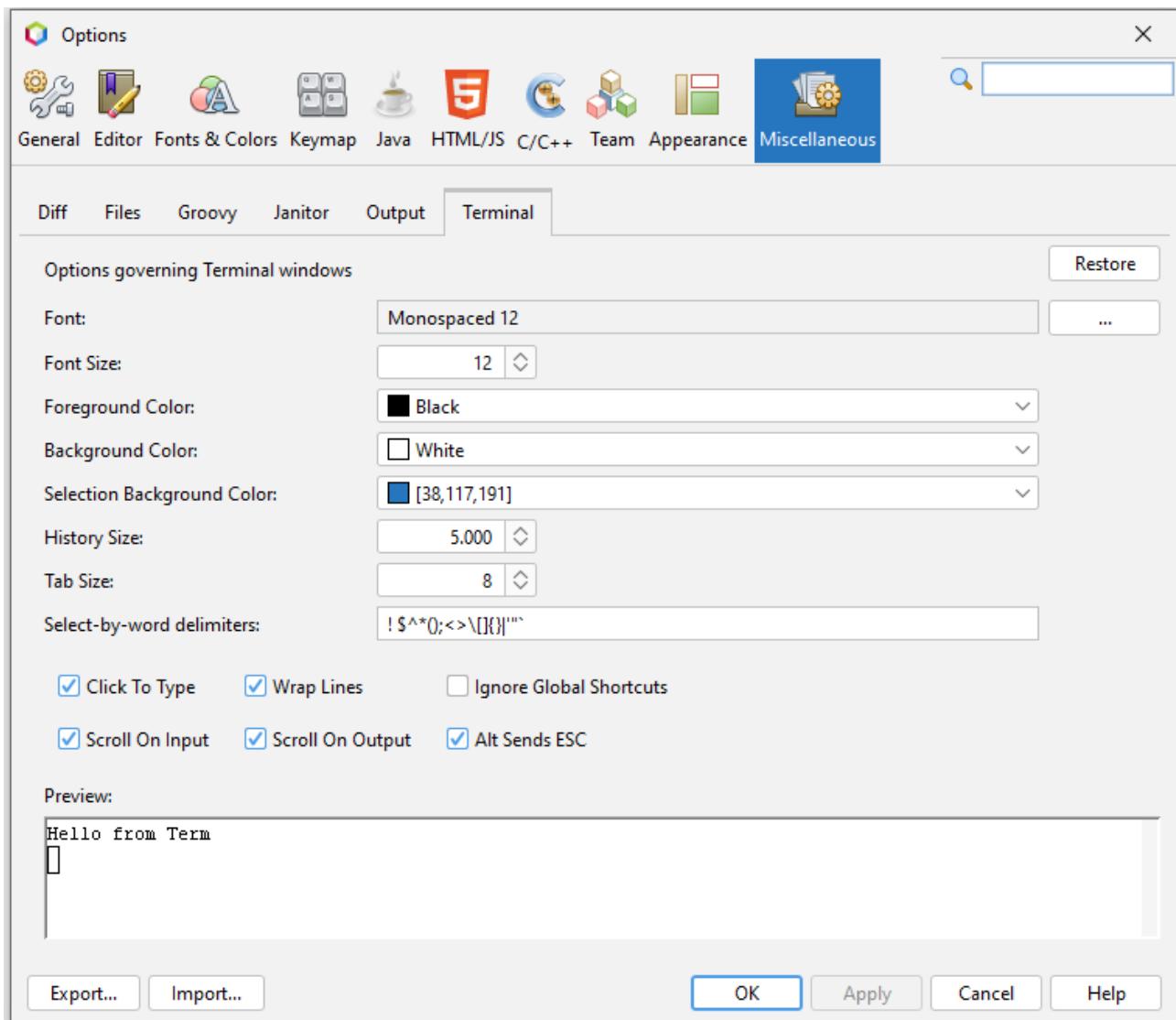
En la opción de Fonts & Colors puedes cambiar el tipo de letra el tamaño y todo lo relacionado con el texto.



En Appearance se puede cambiar la tabulación cuantos espacios pone, la apariencia de la interfaz como el fondo y demás opciones.

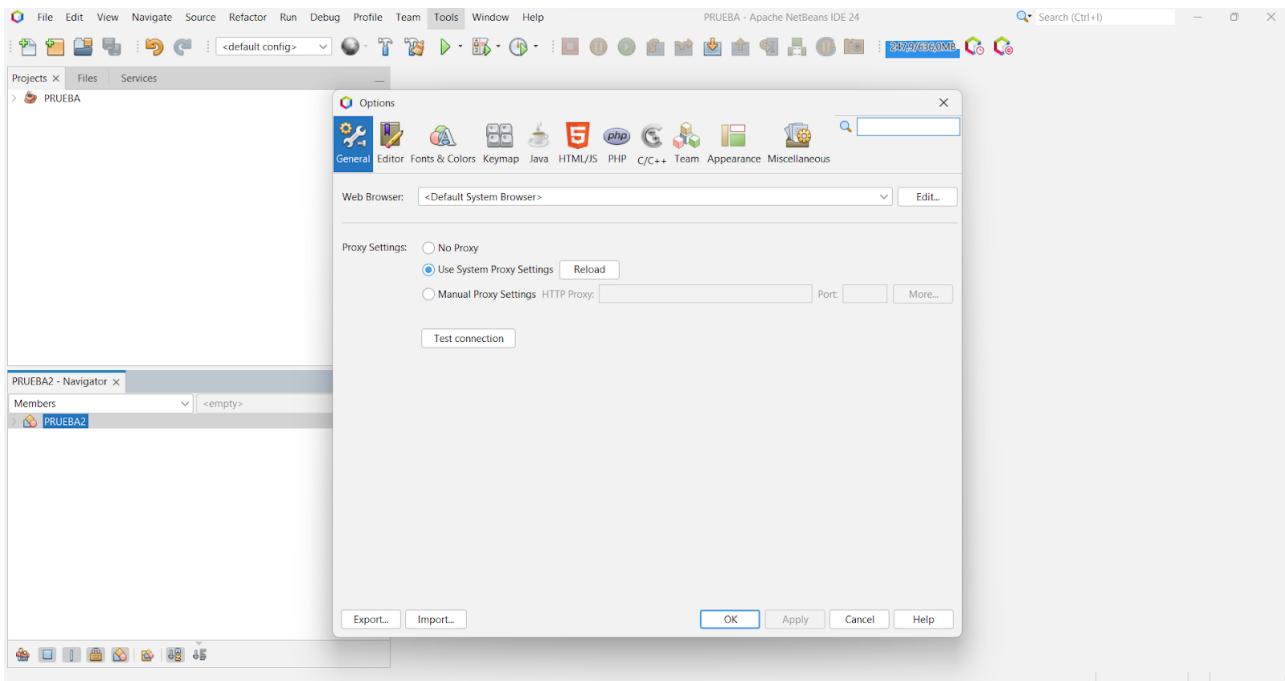


En Miscellaneous puedes cambiar opciones de la terminal como el tamaño de la tetra y el tipo de letra.



Opciones de personalización del IDE.

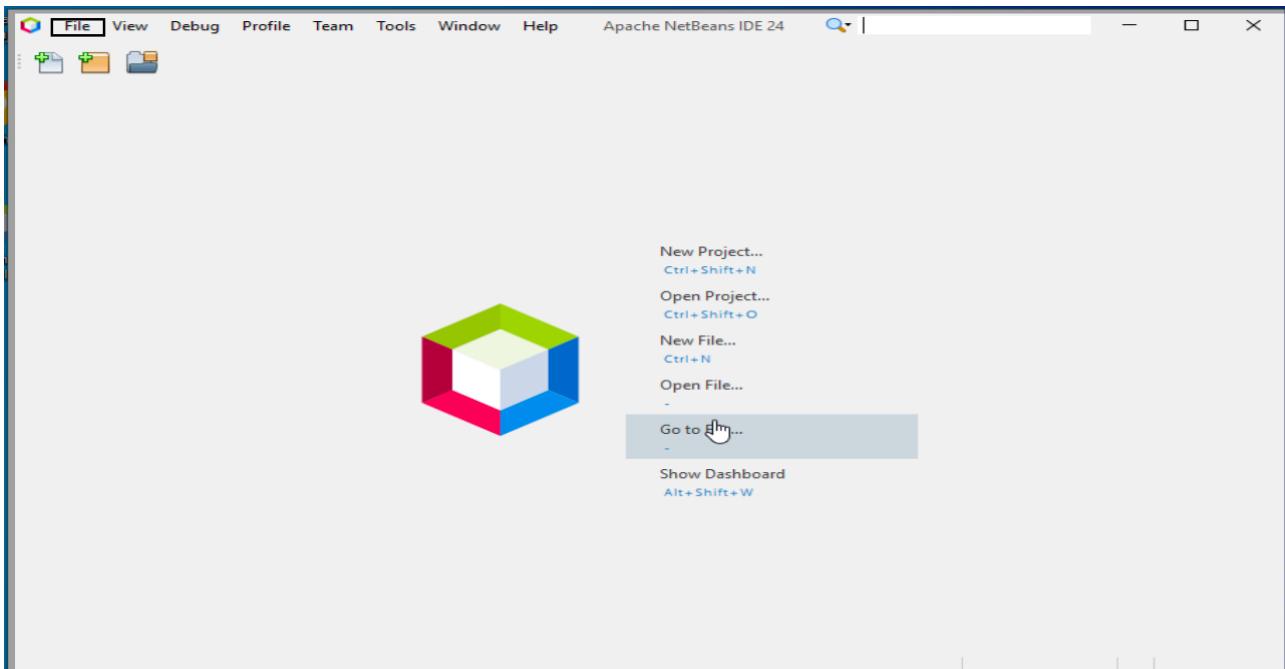
Para exportar a configuración del IDE iremos a Tools después a Options y nos abrirá este menú en el cual vemos al final las opciones de Importar y exportar. Al cliclar en exportar se nos abrirá un menú en el cual seleccionaremos que configuración del IDE queremos hacer como atajos de teclado. Al darle a OK se nos creara un archivo ZIP el cual lleva toda la configuración, lo guardaremos y lo copiaremos a la nueva maquina en la que usemos el IDE y dandole a importar y seleccionando el archivo ZIP ya tendremos la configuración de nuestra antigua maquina.



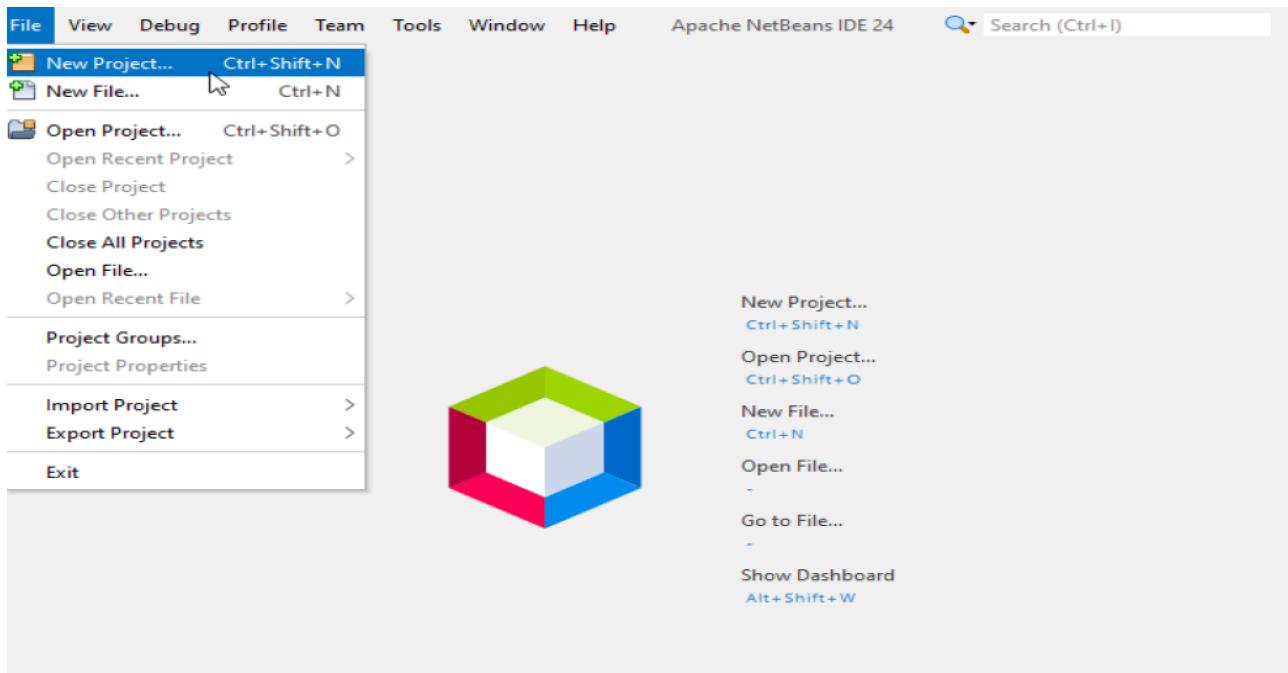
Creación del Proyecto

Para la creación de un nuevo proyecto en el entorno de desarrollo Netbeans en primero lugar es necesario abrir la aplicación y una vez dentro hemos de realizar los siguientes pasos:

1^a. Lo primero que haremos será clicar en la opción Archivo/File.



2º. Una vez hemos realizado el primer paso, se abrirá una pestaña y deberemos elegir la opción de Nuevo Proyecto/New Project.



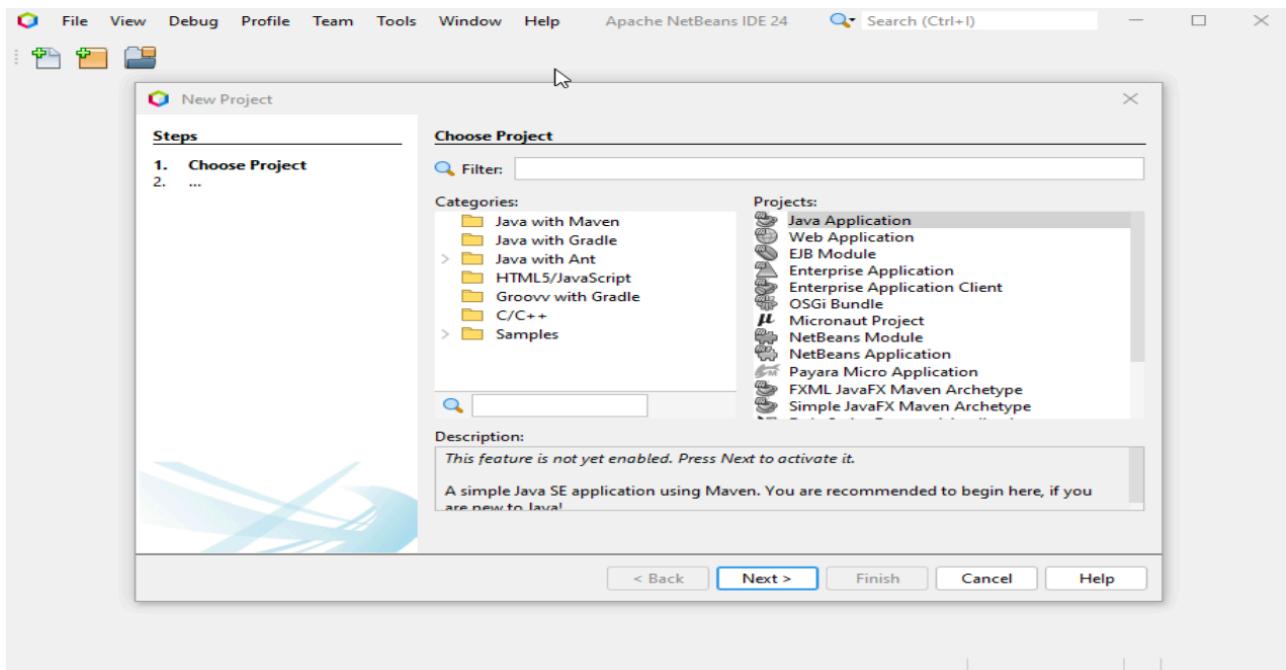
3º. Una vez dentro, aparecerá un cuadro donde deberemos elegir las opciones en función al proyecto que queremos llevar a cabo. En este caso, queremos llevar a cabo un proyecto de una aplicación Java por lo tanto deberemos elegir entre las siguientes categorías que nos ofrece Netbeans.

En primer lugar, tenemos Java with Maven/Java con Maven. Esta es una herramienta de gestión de proyectos que tiene una configuración y estructura de proyecto estándar, es decir no tiene muchas opciones de personalización. No obstante, es una herramienta que dispone de muchos plugins y amplio soporte.

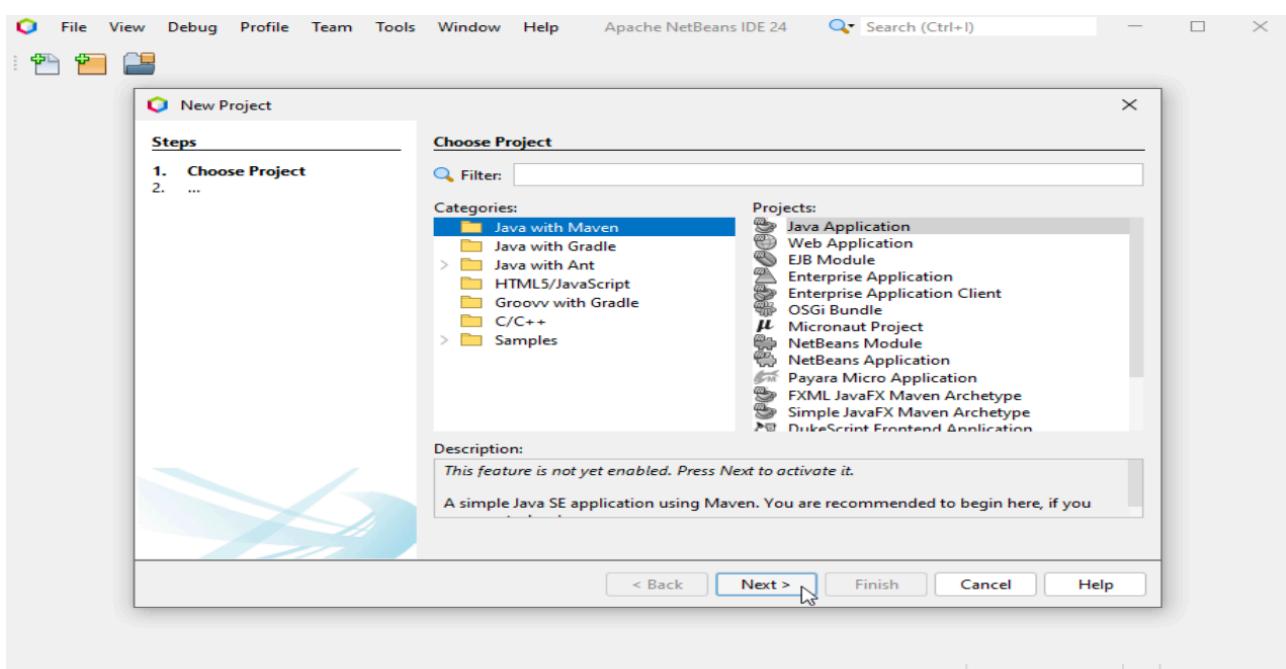
En segundo lugar, encontramos que podemos trabajar con Java with Ant/Java con Ant. Es una herramienta Open-Source que se utiliza en la creación y compilación de programas Java. Es una herramienta que al contrario que Maven posee una flexibilidad total en su configuración. Es para proyectos antiguos o muy simples.

En tercer lugar, podemos observar que existe otra opción que es Java with Gradle/Java con Gradle. Es una herramienta que permite la compilación de código abierto y se centra en la flexibilidad y el rendimiento. Es una herramienta muy estable que completa las tareas de una forma rápida y precisa.

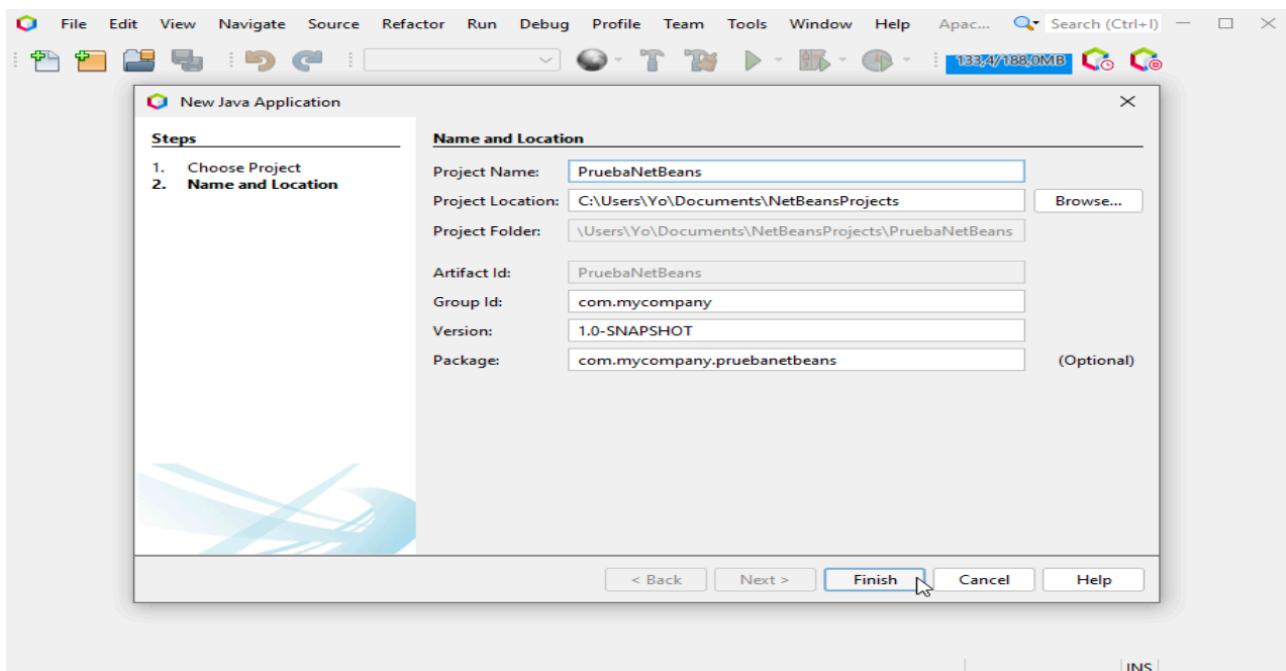
Java with Ant/Java con Ant podría ser una buena opción de cara a empezar a utilizar NetBeans porque nos permitiría empezar a trabajar de una forma rápida y sin demasiados pasos previos. No obstante, en este caso elegiremos Java with Maven/Java con Maven porque más adelante explicaremos como se integra con Git y para dicho fin es mejor utilizar Maven ya que Ant no tiene integración directa con el citado control de versiones.



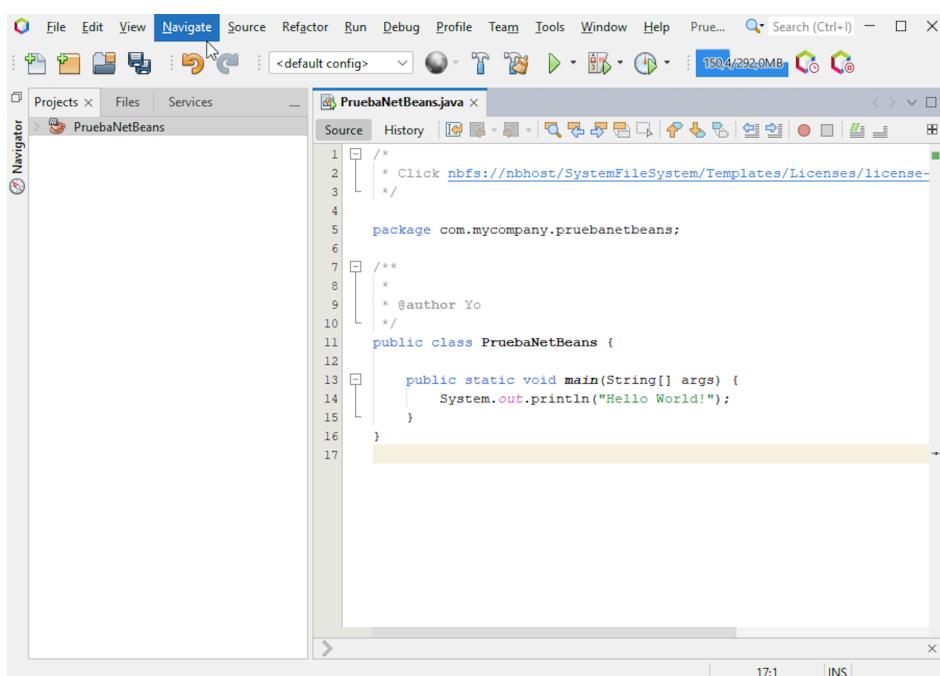
A continuación, marcamos Java with Maven/ Java con Maven y seguimos haciendo click en Next.



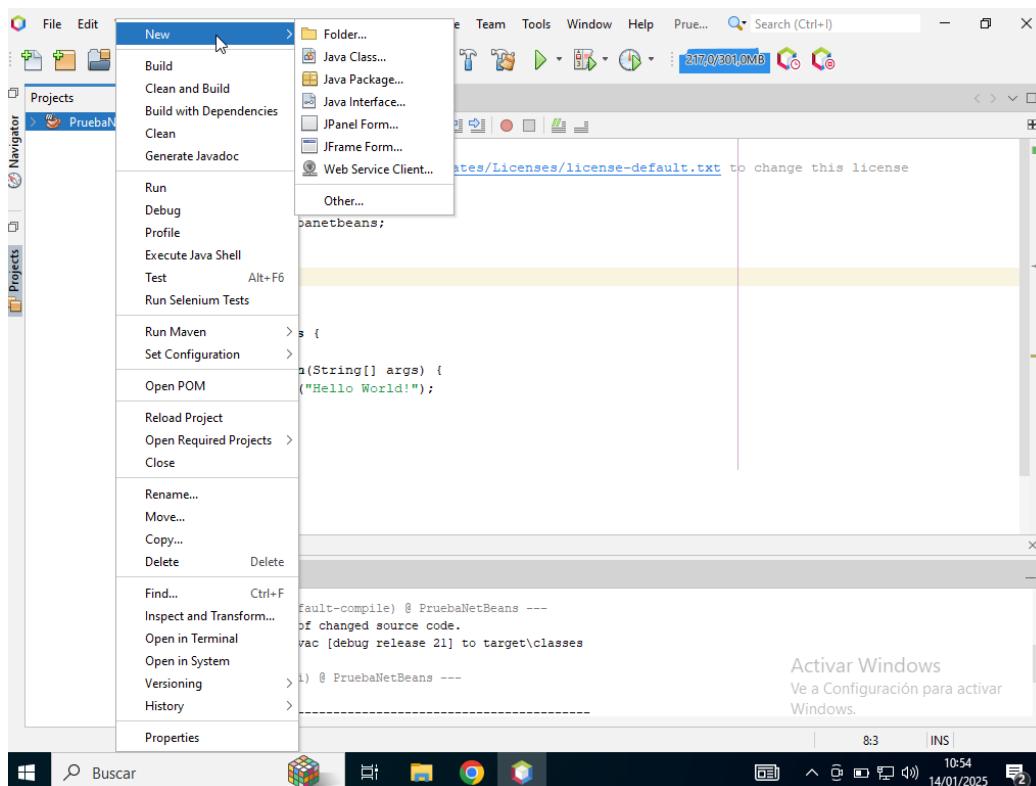
4^a Esperamos a que se configure y se active Java SE y una vez activado encontraremos la siguiente pantalla:



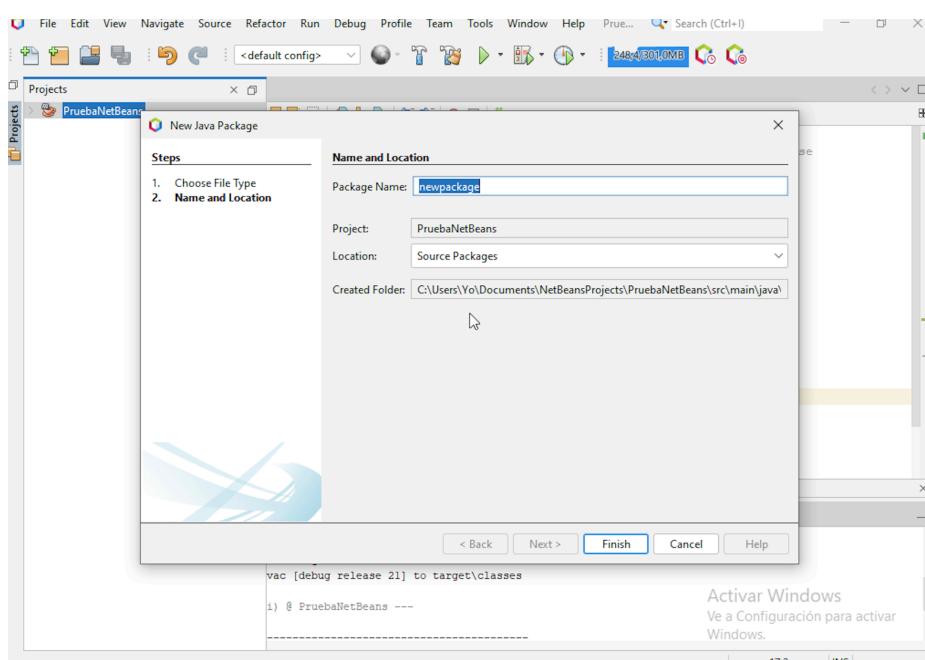
En este caso deberemos elegir un nombre para nuestro proyecto y una dirección como una carpeta o carpetas donde se alojaran los archivos que pertenezcan al mismo. Ignoramos el resto de opciones porque para el objetivo actual no es necesario configurarlo. Esas opciones están enfocadas a crear un proyecto único y que no entre en conflicto con otros proyectos en marcha o que se han realizado dentro de la empresa u organización. Una vez realizado este paso, ya tendremos nuestro proyecto con una estructura creada y se abrirá el entorno de desarrollo.



5. Ahora continuaremos creando el paquete y la clase java que queramos dentro del proyecto.
Para ello pulsamos con el botón derecho encima del proyecto que hemos creado con anterioridad:

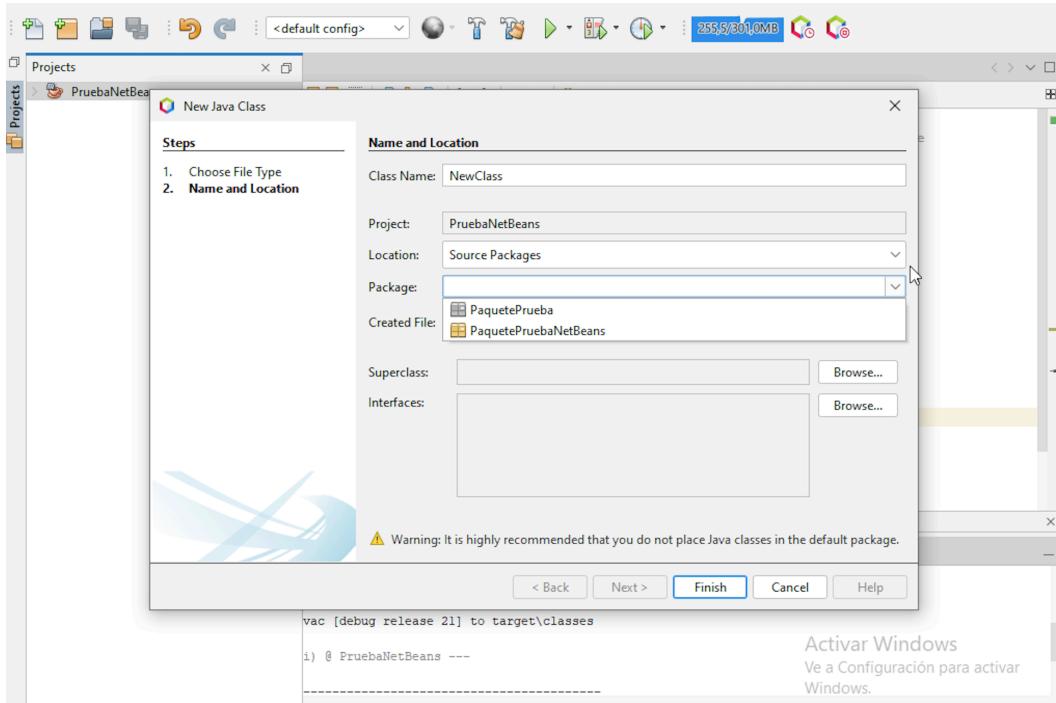


Si clicamos en New > Java Package crearemos un nuevo paquete dentro de nuestro proyecto y si clicamos en New > Java Class crearemos una nueva clase Java. En esta captura mostraremos como es el proceso para un nuevo paquete:

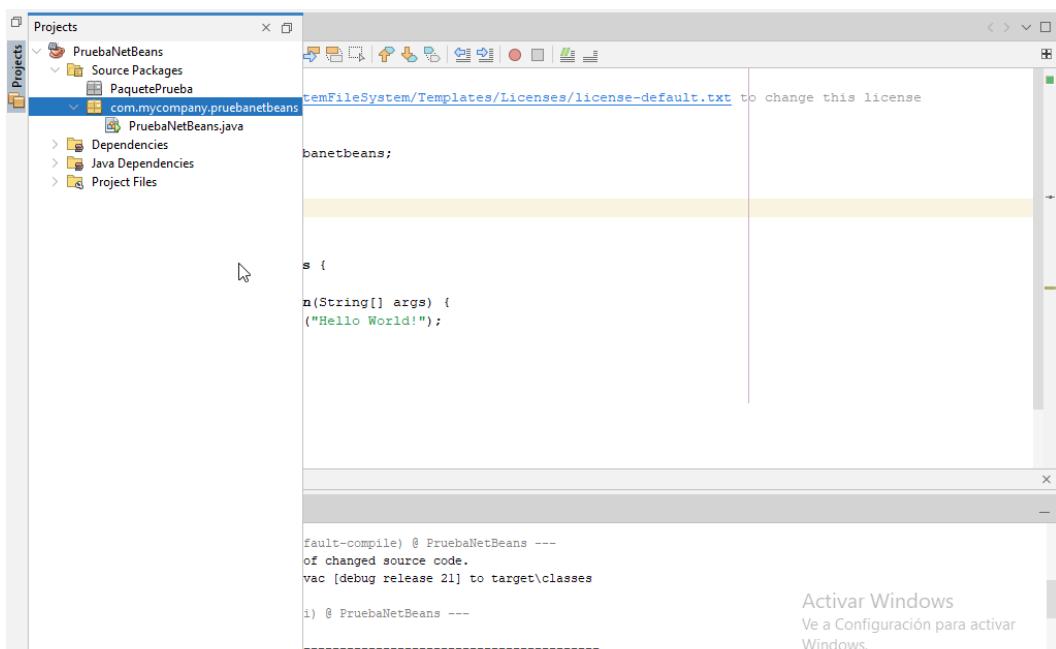


En Package Name escribiríamos el nombre elegido, así como donde nos gustaría que estuviese alojado y después finalizaríamos con Finish.

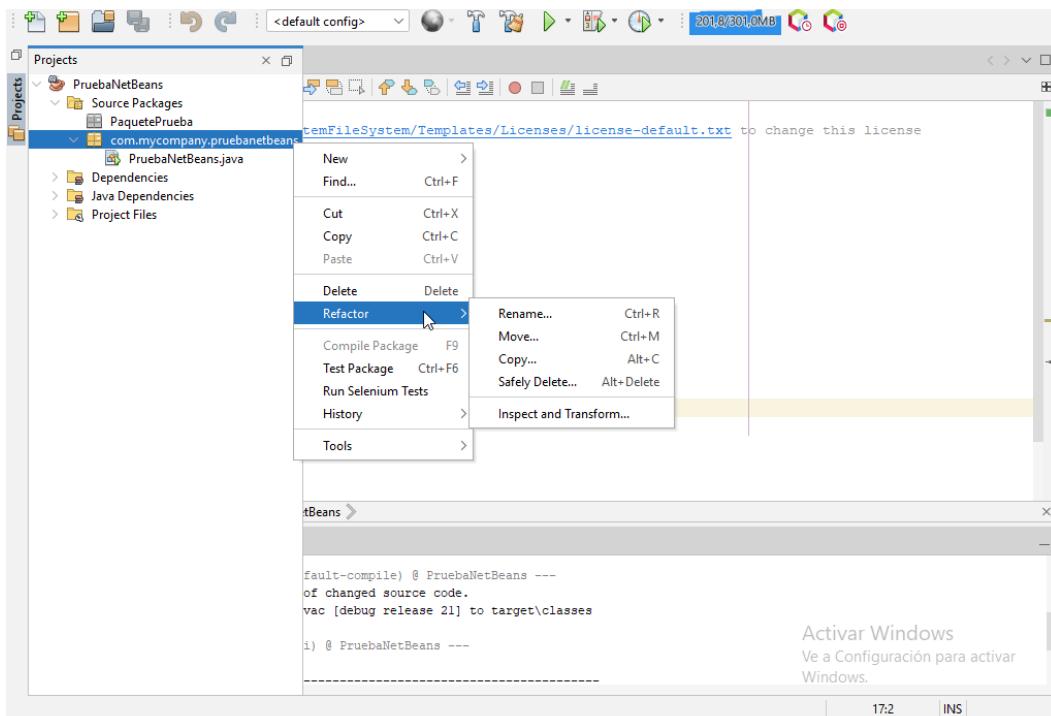
Para crear una clase Java el proceso sería el siguiente:



Si abrimos nuestro proyecto podremos observar que ya teníamos un paquete creado por defecto anteriormente con el nombre “com.mycompany.pruebanetbeans” con una Java Class en su interior llamada “PruebaNetBeans.java”.



6. Podríamos cambiar el nombre del paquete y de nuestra clase Java. Para ello, tenemos que refactorizar nuestro proyecto. En primer lugar, utilizamos el botón derecho sobre el paquete anterior y elegimos la opción Refactor.



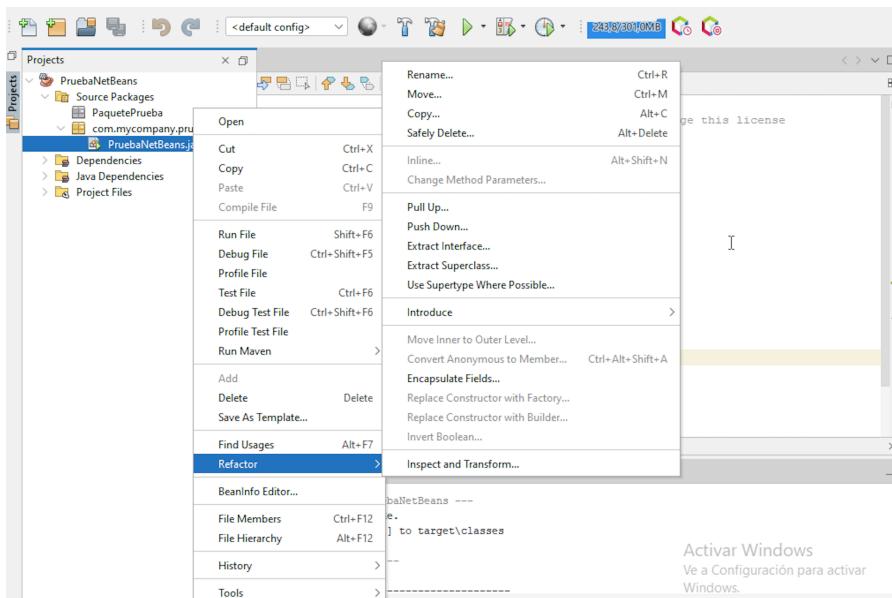
Tras este paso, nos encontraremos esta pestaña con diferentes opciones:

Rename (Renombrar): Esto nos permite cambiar el nombre de una clase, método, variable, paquete o archivo.

Move (Mover): Permite mover una clase o un paquete a otra ubicación dentro del paquete.

Copy (Copiar): Permite duplicar una clase, un paquete o un archivo dentro del proyecto.

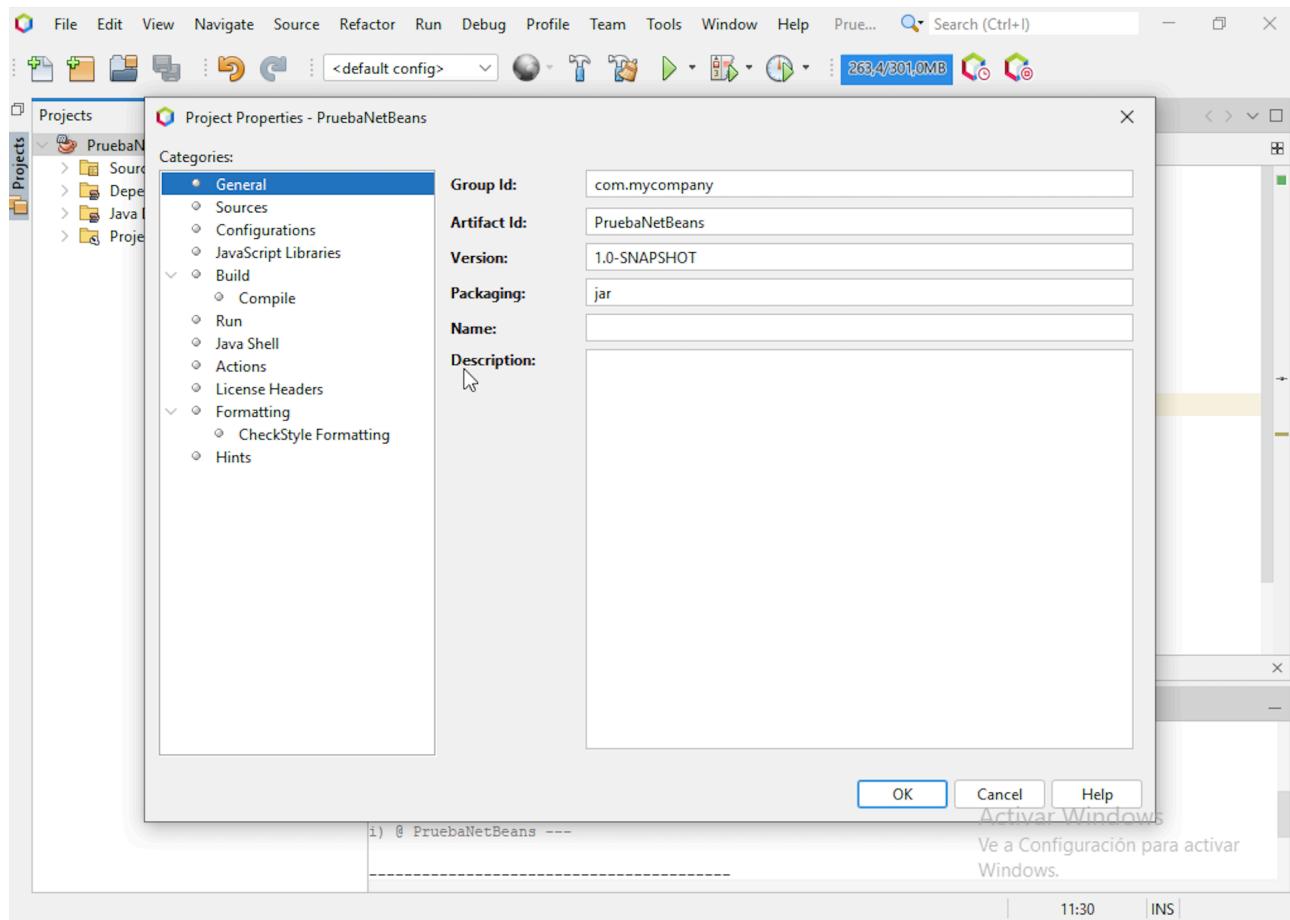
Safety Delete (Borrar): Permite borrar de forma segura un archivo, clase o paquete.



También podríamos refactorizar nuestra clase Java.

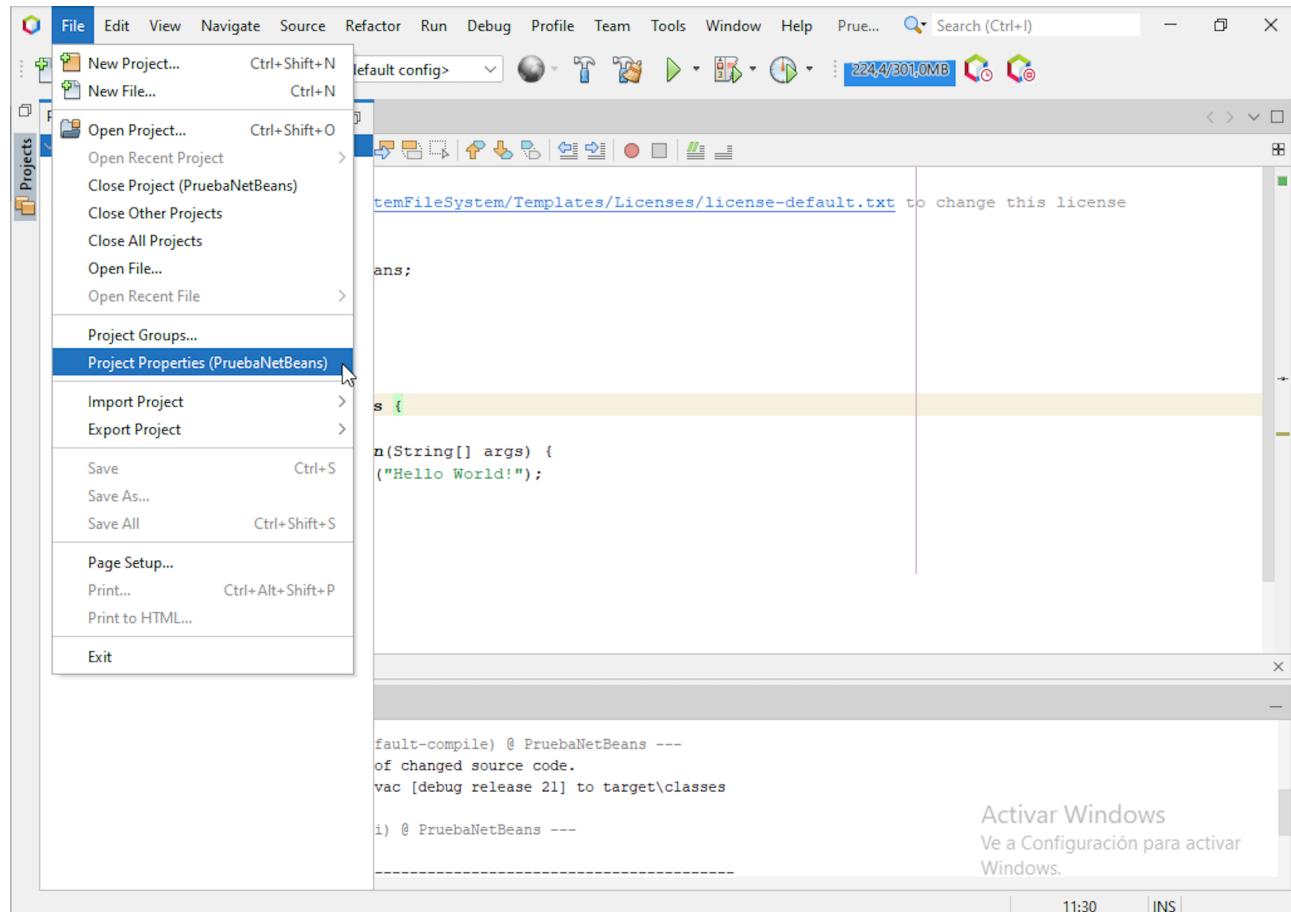
Una vez realizados estos pasos ya tendríamos estructurado nuestro proyecto con las paquetes y las clases deseadas.

Configuración del Proyecto



Para poder configurar nuestro proyecto creado en primer lugar es necesario abrirlo. Para ello iremos a File>Open Project (también podemos usar ctrl + shift + O) y seleccionaremos nuestro proyecto en la carpeta o dirección en la que está alojado.

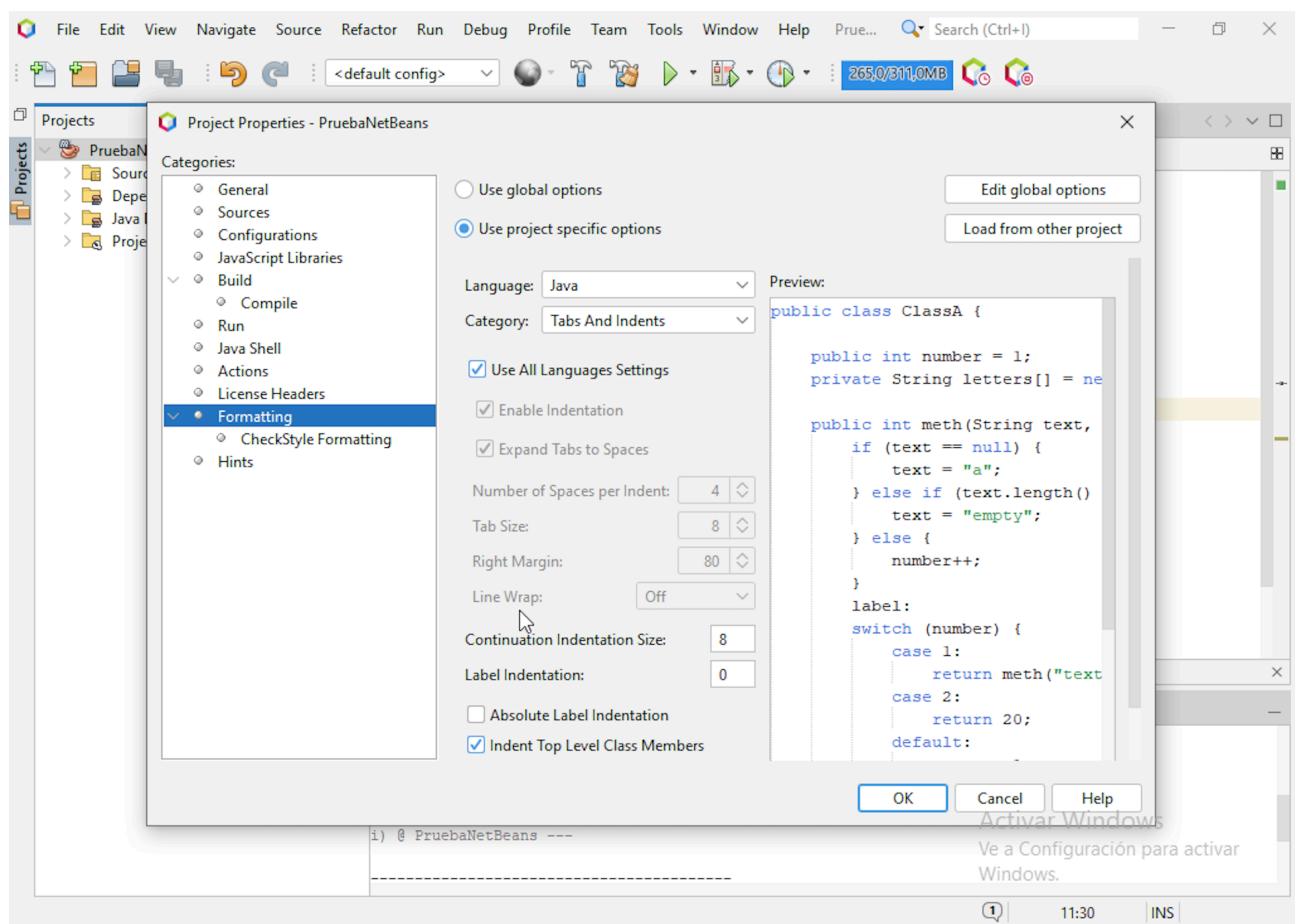
Dentro de File encontraremos la opción de Project Properties o propiedades del proyecto que nos permitirá modificar numerosos aspectos de nuestro proyecto como podemos observar en la captura anterior.

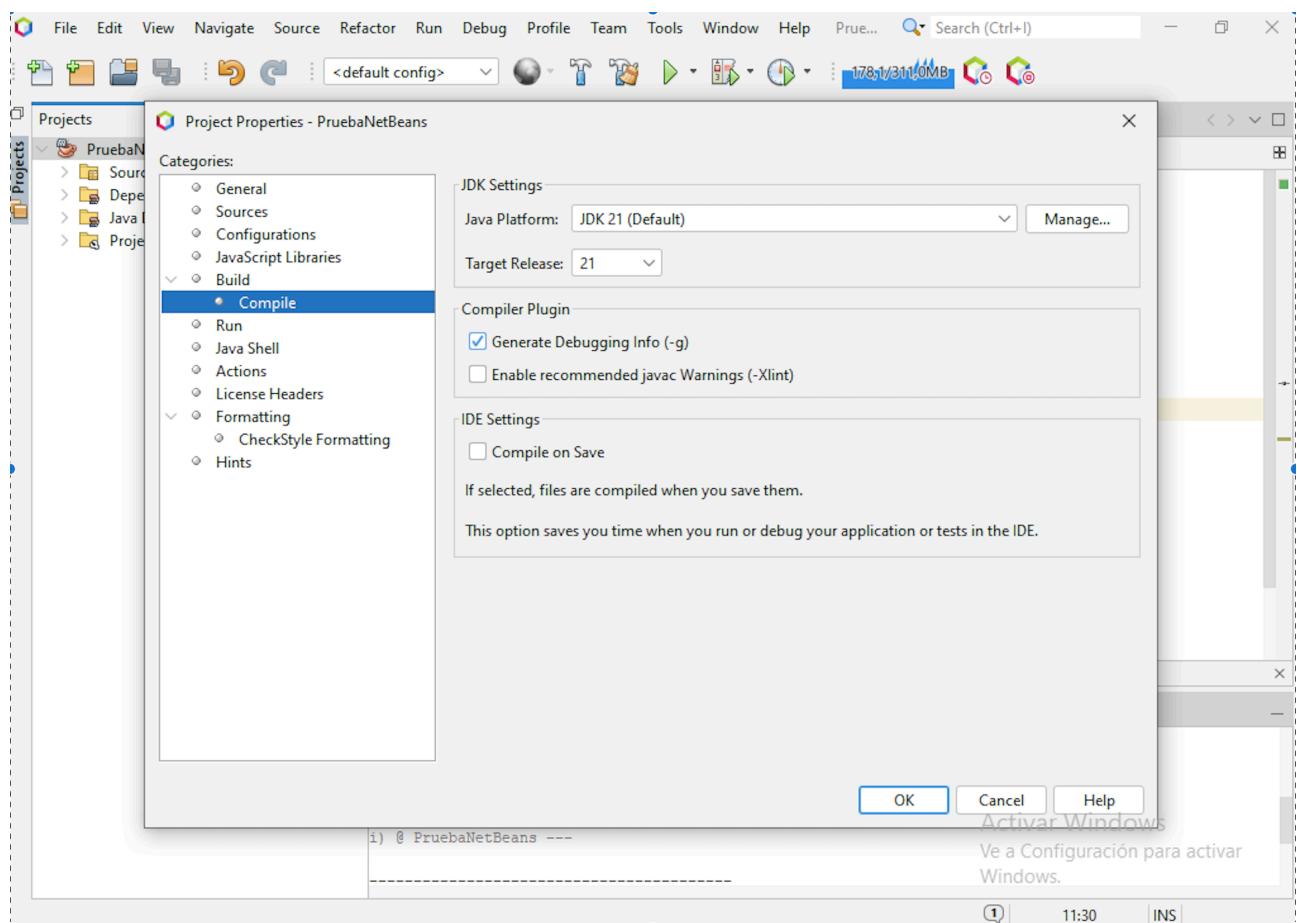


En la sección de Compile podemos seleccionar la versión del JDK. En este caso, nosotros hemos seleccionado la versión JDK 21.

En la categoría Formatting, encontraremos todas las opciones para dar formato a nuestro código. Podríamos formatear el indentado y las tabulaciones en función a las convenciones de cada lenguaje y en función al formato que se establezca en el proyecto.

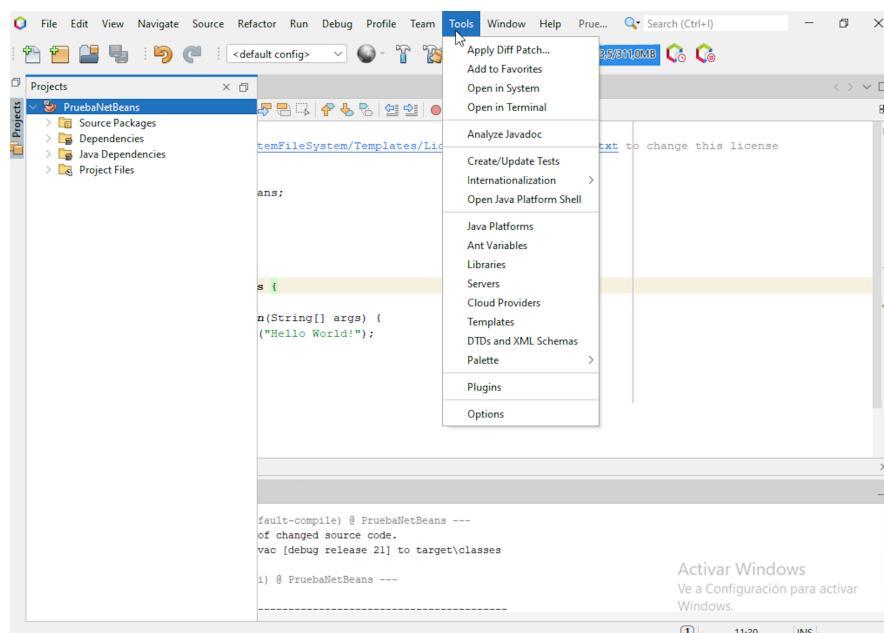
En resumen, configura el formato del código fuente para mantener la consistencia en la escritura del código. Permite definir reglas de indentación y el uso de los tabuladores.



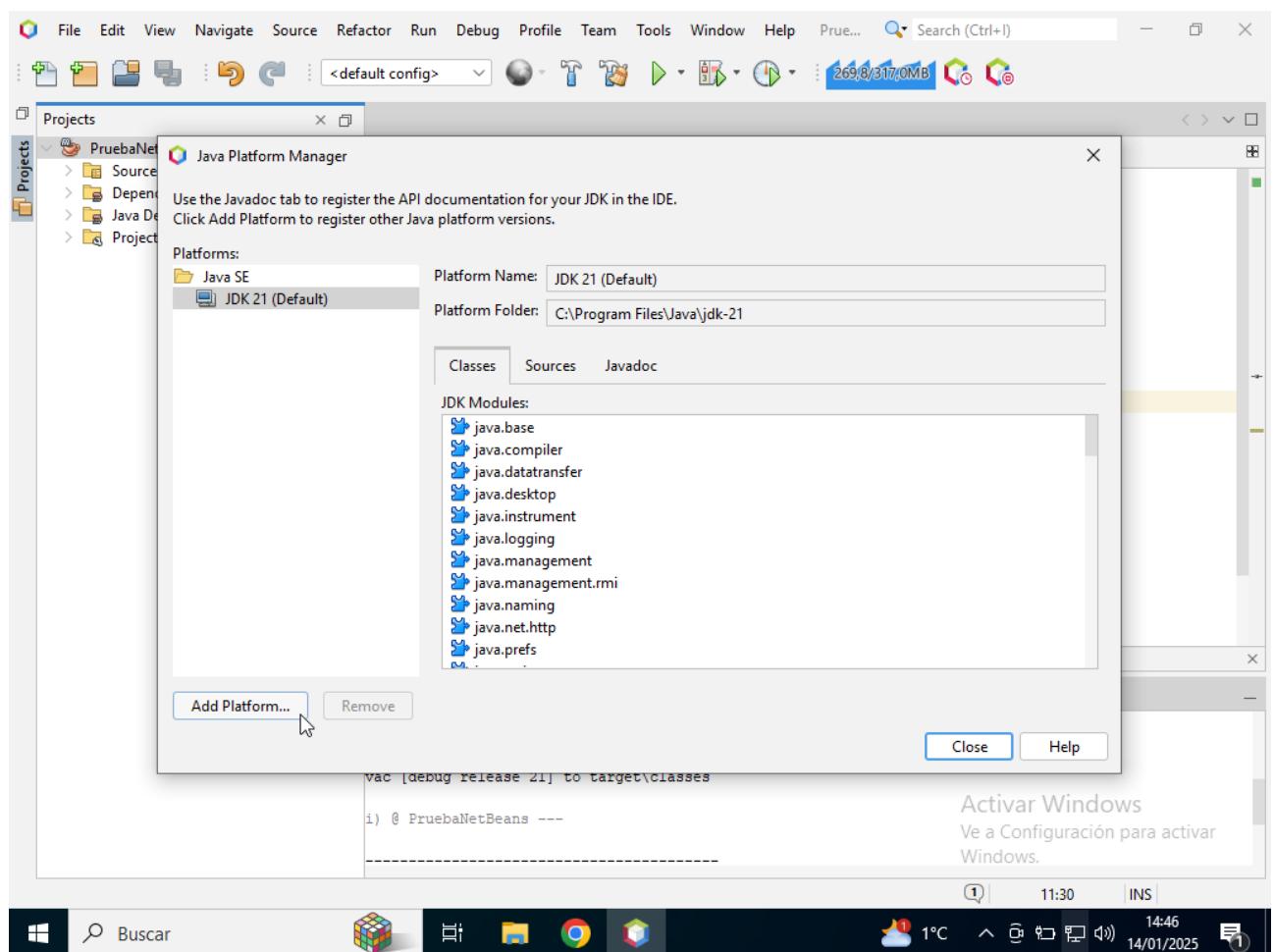


Por último, la sección de sources nos permite definir las carpetas que contienen el código fuente y los archivos de prueba.

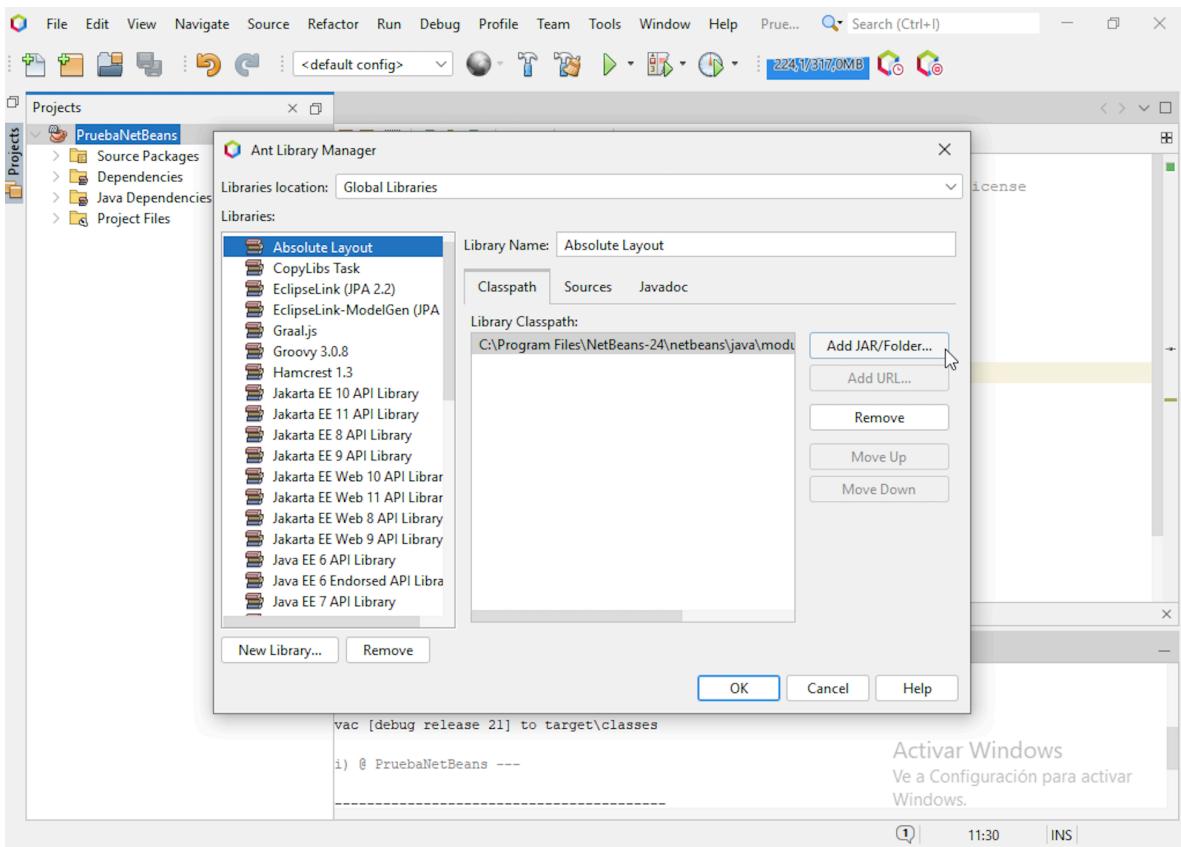
Estos serían los puntos más importantes a configurar en la parte de Project Properties. No obstante, si vamos a la pestaña de Tools (herramientas) encontraremos más opciones de cara a seguir configurando nuestro proyecto.



En esta parte encontramos Java Platforms que nos permite añadir nuevas versiones del JDK y establecer una versión de como la predeterminada.

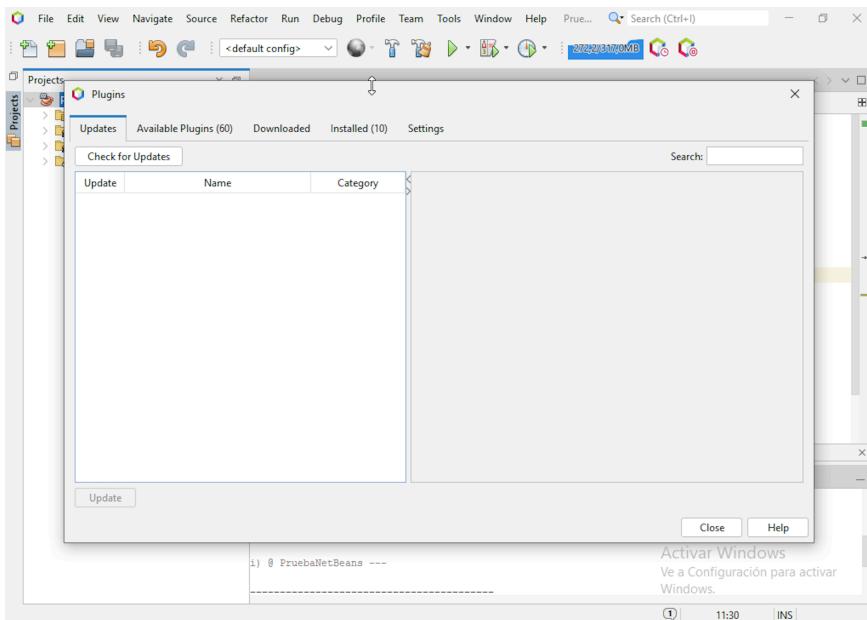


Otra opción importante sería la de Libraries (Librerías). Nos permite gestionar las librerías que se pueden añadir al proyecto. Se pueden crear y añadir nuevas bibliotecas mediante un archivo ZIP o JAR.



Plugins

Los plugins son aplicaciones informáticas que nos permiten extender las funciones de otra aplicación. El entorno de desarrollo NetBeans nos permite instalar estas extensiones para mejorar sus funciones.



Para instalar los plugins o extensiones tendremos que volver a la pestaña de Tools y esta vez seleccionar Plugins.

En esta pestaña encontraremos:

Updates. Nos mostrará todas aquellas actualizaciones de las extensiones que tenemos instaladas.

Available Plugins. Nos enseñará una lista de los plugins que están disponibles para instalar.

Downloaded. Será una lista de todos aquellos plugins que hemos descargado.

Installed. Aquí se ubicarán las extensiones que hemos instalado en NetBeans.

Para la instalación de plugins lo que haremos será ir a la pestaña de available plugins y en la barra con el nombre search buscaremos la extensión que queremos instalar. En este caso, hemos hecho una selección de los plugins que pueden resultar de mayor utilidad para programar.

1. **External Code Formatters.** Este plugin permite la integración de formateadores de Código Externos (Formateador de Java y JS de Eclipse, formateador de Java de Google, formateador de Java de Spring , etc) para NetBeans. Este plugin permite elegir un formateado predeterminado para cada lenguaje soportado y sobreescribirlo por proyecto en la configuración de NetBeans.
2. **SonarLint4NetBeans.** Es una herramienta de análisis de código que ayuda a los desarrolladores a identificar y corregir errores, vulnerabilidades, y problemas de calidad en su código de manera proactiva. Proporciona comentarios en tiempo real mientras se escribe el código.
3. **Rainbow Braces.** Es una herramienta que permite dar color a los paréntesis, llaves y corchetes para identificar mejor los bloques de código y evitar errores.
4. **GitHub Issue Support.** Es un plugin que proporciona una integración entre tu entorno de desarrollo y GitHub para gestionar los problemas y tareas asociadas al proyecto.
5. **Jeddict AI.**

GIT

Apache NetBeans proporciona el control de versiones integrando directamente GIT en el IDE. Esto nos permite trabajar con el código y el control de versiones directamente desde la aplicación.

Iniciar un repositorio de GIT.

Antes de nada hemos de asegurarnos de tener Git instalado en nuestro equipo.

Para crear un repositorio de GIT con uno de nuestros proyectos debemos seguir los siguientes pasos:

1. En la ventana de proyectos, seleccionamos aquel que todavía no tiene un repositorio Git.
Clicamos con el botón derecho sobre el proyecto deseado.
2. Elegimos Versioning > Initialize Git Repository (también podemos hacerlo desde el menú principal en Team>Git>Initialize).
3. Una vez realizado este paso, es necesario establecer la ubicación del repositorio para guardar las versiones y le damos a OK.

Una vez realizado este paso veremos que nuestros archivos han sido añadidos al repositorio y se marcará con un Added (A). En caso de que todavía no aparezca clicaremos con el botón derecho en el archivo y seleccionaremos ADD.

Clonar Repositorio

En primer lugar, para copiar un repositorio de Git ya existente necesitamos clonarlo con la opción Team > Git > Clone. Por ejemplo, para clonar un repositorio desde GitHub podríamos realizar los siguientes pasos:

- 1º Especificar la ruta de la ubicación de nuestro repositorio de GitHub, es necesario completar nuestro usuario y contraseña. Una vez completado este paso clicamos en Next.

2º En la sección Remote Branches (ramas remotas), especificamos la rama que queremos descargar (Fetch) en nuestro repositorio local.

3º Directorio de destino (Destination Directory). Aquí especificaremos la ruta donde queremos que se clone nuestro repositorio.

- En el campo Directorio Principal (Parent Directory), especificaremos la ruta del directorio donde se clonará el repositorio.
- En el campo Clone Name, escribiremos el nombre de la carpeta local donde se clonará el proyecto.
- En el campo Checkout Branch (rama para revisar), seleccionamos la rama que se revisará en el árbol de trabajo.
- El campo Remote Name, especifica el nombre que representa el repositorio original que se está clonando (el valor predeterminado es origin).

Es aconsejable dejar seleccionada la opción Escanear Proyectos de NetBeans después de Clonar para activar el escaneo tras finalizar la clonación. Hacemos clic en Finish. Al clonar, se crea la carpeta .git dentro del directorio seleccionado.

Ahora que ya tenemos nuestro repositorio remoto, podemos comenzar a trabajar con él.

Repositorios remotos

En un ambiente real de trabajo necesitaremos trabajar con otros desarrolladores y compartir nuestro trabajo, es por ello que encontramos las opciones de Commit, Fetch, Pull, Push.

Fetching en Git es el proceso de obtener los cambios del repositorio remoto sin modificar las ramas locales. Esto te permite ver las actualizaciones antes de integrarlas. Para realizar un Fetch en NetBeans, iremos a **Team > Remote > Fetch**.

En el asistente de NetBeans, en la página de Repositorio Remoto, selecciona entre un repositorio configurado previamente o especifica una nueva ubicación de repositorio Git, incluyendo nombre, login, y configuración de proxy si es necesario. Luego, elige las ramas de las que deseas obtener actualizaciones en la página de Ramas Remotas. Al finalizar, se crea una copia local de la rama remota seleccionada. Estas actualizaciones pueden ser fusionadas en una rama local posteriormente.

Hacer pulling es obtener actualizaciones desde un repositorio remoto y fusionarlas en nuestra rama local actual. Es decir, sincroniza nuestro repositorio local con los cambios realizados en remoto. Para realizar un pull en NetBeans, selecciona **Team > Remote > Pull**. Eligimos un repositorio configurado o especificamos una nueva ubicación de repositorio Git. Luego, seleccionamos las ramas desde las cuales deseas extraer cambios. Al finalizar, nuestro repositorio local se sincronizará con el repositorio remoto.

Hasta este momento estaríamos actualizando nuestro repositorio local con los cambios realizados en remoto. Ahora veremos como realizar el camino inverso. Como guardar los cambios realizados a nivel local y actualizar el repositorio remoto. Para ello realizaremos un Commit y un Push.

Hacer un commit sería hacer un guardado local de los cambios realizados en los archivos del proyecto. Para ello, clicamos con el botón derecho en el proyecto del que queremos guardar los cambios y en el menú podremos seleccionar **Git > Commit**.

El cuadro de diálogo de **Commit** contiene los siguientes componentes:

- **Área de mensaje de commit:** destinada a describir los cambios realizados.
- **Listas desplegables de Autor y Committer:** para diferenciar entre quien hizo los cambios y quien realizó el commit si es necesario.
- Sección de archivos para hacer commit muestra:
 - Archivos modificados.
 - Archivos eliminados en el árbol de trabajo (localmente).
 - Archivos nuevos (que aún no existen en el repositorio Git).
 - Archivos renombrados.

Ahora que hemos guardado los cambios, vamos a enviarlos a nuestro repositorio en remoto haciendo pushing. Para realizar un push elegimos **Team > Remote > Push**. Se abrirá el asistente **Push to Remote Repository**.

En la página **Remote Repository**, seleccionamos el repositorio configurado o la opción para especificar la ubicación de un repositorio remoto nuevo, incluyendo nombre, usuario y contraseña si es necesario, y haz clic en **Next**.

En la página **Select Local Branches**, elegimos las ramas a las que queremos hacer push y hacemos clic en **Next**.

En la página de **Update Local References**, seleccionamos las ramas a actualizar en el directorio de Remotes de tu repositorio local y haz clic en **Finish**.

Asistente de IA

Actualmente no existe ninguna extensión que permita integrar GitHub Copilot en Apache NetBeans. No existe una forma nativa de hacerlo. Aunque resulta tedioso y poco productivo se podría hacer de forma indirecta. Es posible usar otro entorno de desarrollo que tenga integrado GitHub Copilot y después trasladar el código a nuestro proyecto de NetBeans.

Aunque no ofrece GitHub Copilot, NetBeans nos ofrece otras opciones para utilizar Inteligencia Artificial a la hora de programar nuestro código.

Por una lado, tenemos **Jeddict AI Assistant**. Es una herramienta que nos permite tener una variedad de características impulsadas por inteligencia artificial que mejoran la eficiencia y calidad del código. Podemos descargarla desde el propio IDE como un plugin.

Permite la generación de documentación: Automatiza la creación de documentación para clases, métodos o interfaces, asegurando una documentación completa y bien estructurada basada en el contexto del código. Así como corrección gramatical de los documentos.

Optimiza la lógica, estructura o eficiencia de los métodos.

Sugerencias de nombres de variables: Propone nombres de variables más significativos y apropiados según el contexto para mantener mejor el código.

Mejora de calidad de texto: Ofrece alternativas más concisas y efectivas para comentarios, documentación o mensajes de registro, mejorando la calidad y legibilidad del contenido textual en el código.

Generación de casos de prueba: Crea automáticamente casos de prueba para clases o métodos basándose en indicaciones contextuales proporcionadas por el asistente de IA, facilitando la escritura de pruebas y asegurando que el código esté bien testado.

Por otro lado, tenemos el complemento **NetBeansChatGPT**. Este es un complemento (o add-on, es una especie de herramienta que extiende las funcionalidades de forma más general que un plugin) que permite integrar todas las funcionalidades.

Para instalar este complemento, lo que tenemos que hacer es descargar la versión más reciente desde el repositorio oficial de GitHub. Es un archivo .nbm.

Para instalar el complemento vamos a **Tools > Plugins**. Cuando esté dentro de esta pestaña, nos iremos a **Downloaded > Add Plugins**. Y ahí seleccionamos la carpeta previamente descargada.

Una vez instalado, solo nos quedaría reiniciar para hacer efectivos los cambios y ya podremos acceder a las funcionalidades añadidas.