

ENTORNO DE DESARROLLO: INTELLIJ IDEA

Sergio Martinez Martinez - 1º DAM

Instalación Java SDK	3
Descarga Java SDK	3
Instalación Java SDK	4
Configuración variable de entorno PATH	6
Instalación IntelliJ	7
Descargar IntelliJ IDEA	7
Instalación IntelliJ IDEA	7
Configuración IntelliJ	10
Primeros pasos en IntelliJ	13
Creación de proyecto	13
Ejecutar clases en IntelliJ	15
Ajustes y preferencias de IntelliJ	18
Instalación y uso de plugins (IDE Features Trainer)	24

Instalación Java SDK

Descarga Java SDK

A diferencia de otros entornos de desarrollo u otros lenguajes, IntelliJ no viene con los archivos necesarios para desarrollar aplicaciones Java. Como se indica en la página web del producto:

<https://www.jetbrains.com/help/idea/sdk.html>

El primer paso será descargar, instalar y configurar correctamente estos archivos. Para ello acudiremos a la página oficial de Java:

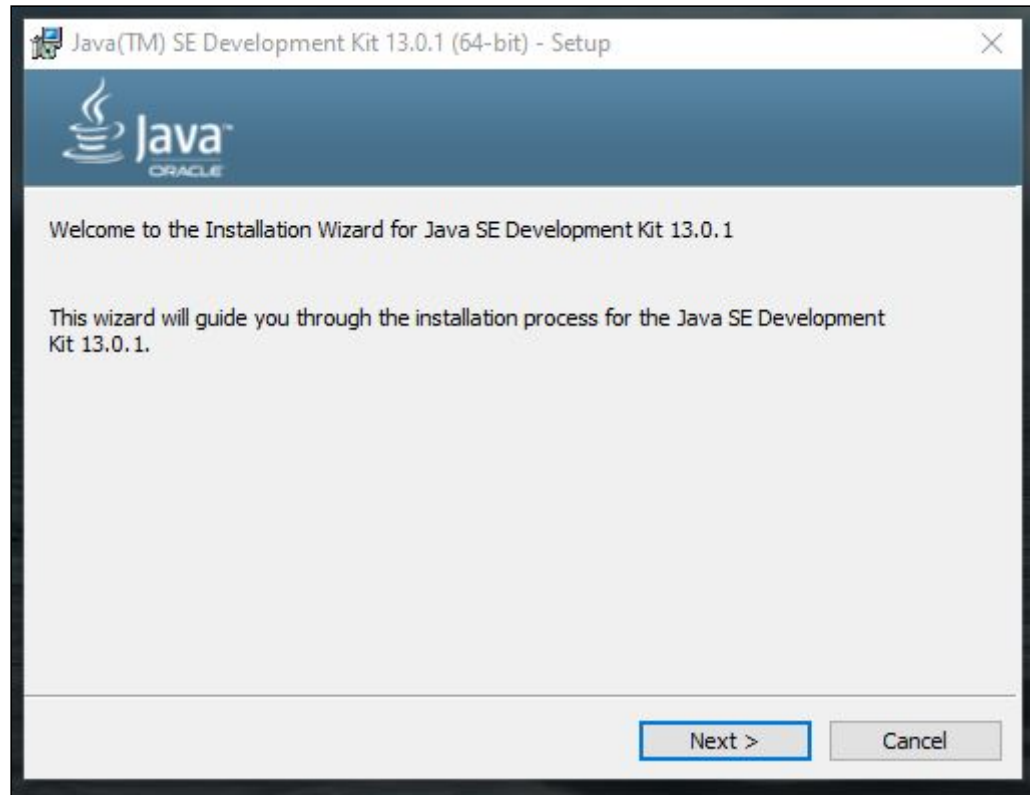
<https://www.oracle.com/technetwork/java/javase/downloads/index.html>

En la página nos aparecerá un botón para iniciar la descarga, el cual nos descargará el archivo necesario para la instalación.

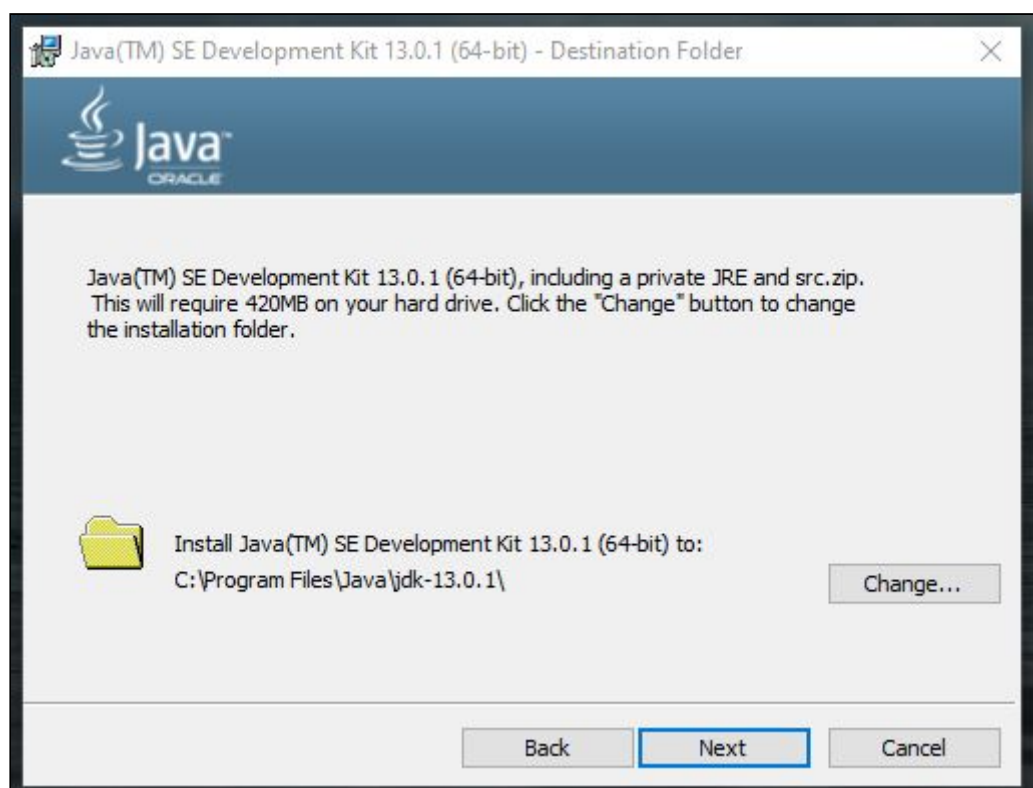


Instalación Java SDK

Ejecutamos el archivo `jdk-13.0.1_windows-x64_bin.exe` y nos aparecerá la siguiente ventana:



Haremos click en Next y nos aparecerá la siguiente ventana:



En esta ventana nos aparece el directorio de instalación de los archivos SDK el cual no es recomendable cambiar, pero es posible. Seleccionaremos Next y terminaremos la instalación.



Configuración variable de entorno PATH

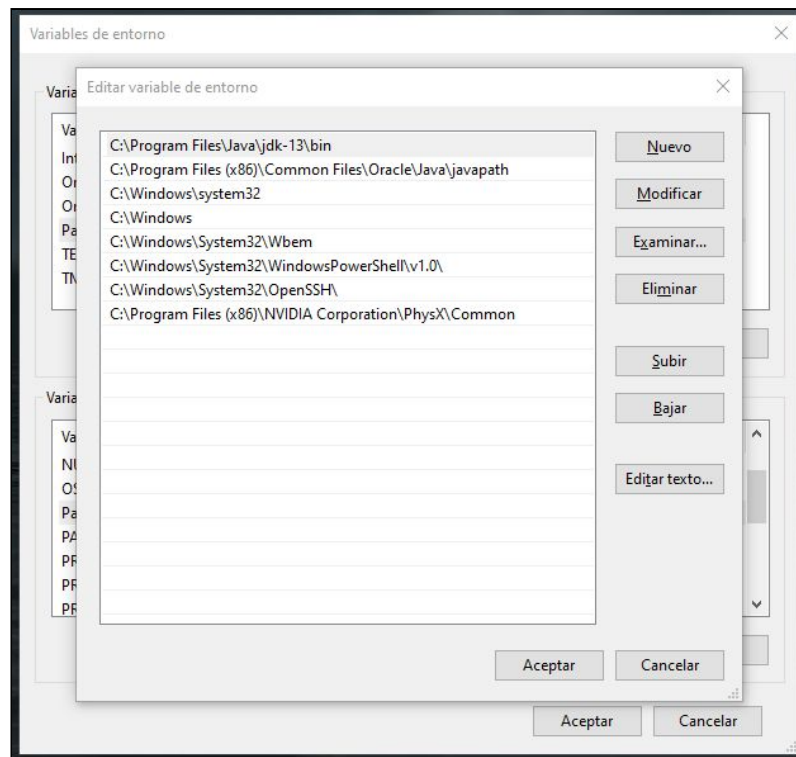
Necesitaremos configurar la variable de entorno PATH para que los entornos de desarrollo encuentren los archivos SDK de Java.

1. Abrir el Panel de control > Sistema y seguridad
2. Click en Sistema > Configuración avanzada del sistema
3. Click en Variables de entorno y haremos doble click en la variable llamada: "Path"
4. Añadiremos la siguiente entrada:

```
C:\Program Files\Java\jdk-13\bin
```

En caso de haber seleccionado otro directorio en la instalación de los archivos SDK deberemos añadir esa dirección.

Nos tiene que aparecer algo parecido a esto, haremos click en Aceptar:



Instalación IntelliJ

Descargar IntelliJ IDEA

Acudiremos a la página oficial del producto:

<https://www.jetbrains.com/es-es/idea/>

Donde tendremos una introducción y presentación de las funcionalidades del mismo. Además de las descargas del mismo, tanto en la parte superior como en la inferior.

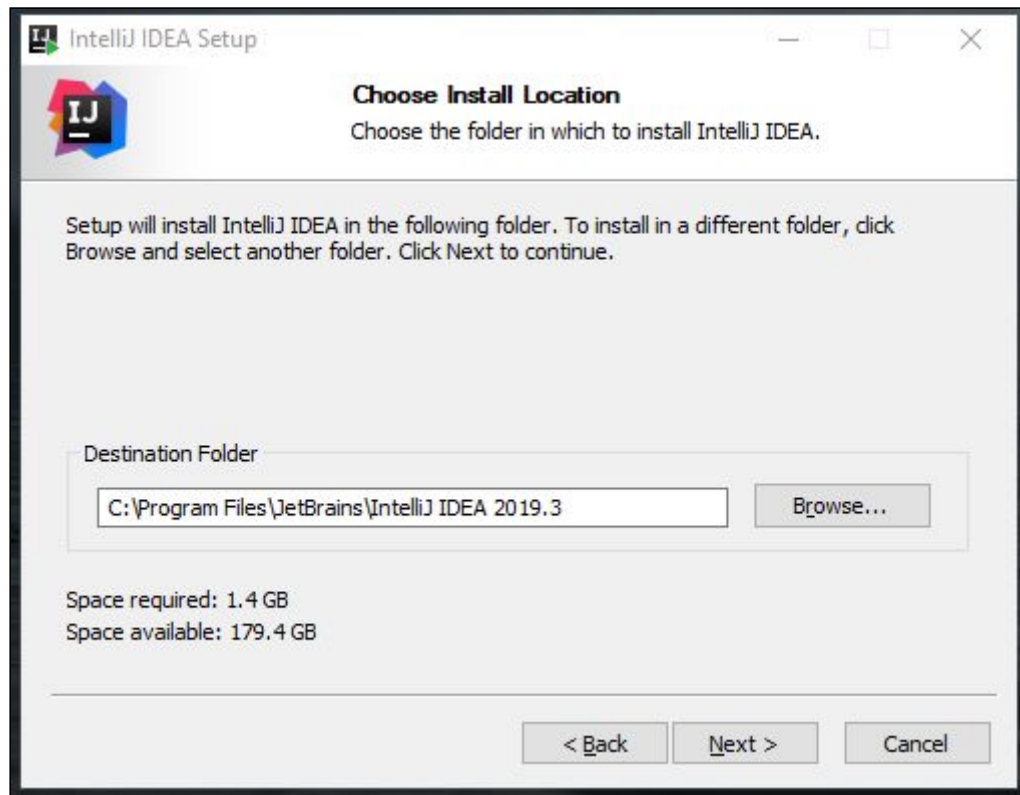
Descargamos la versión Comercial o Ultimate, que dispone de un periodo de prueba de 30 días y es la más completa.

Instalación IntelliJ IDEA

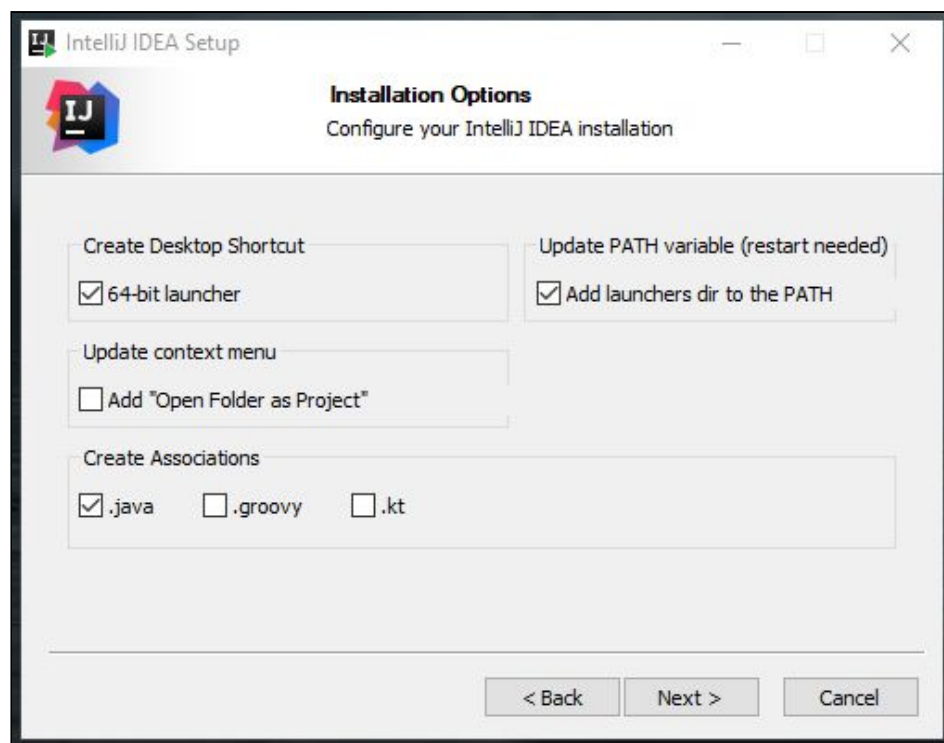
Ejecutaremos el archivo idealU-2019.3.exe y haremos click en Next:



En esta ventana nos aparecerá el directorio de instalación del programa, lo podemos cambiar si queremos:

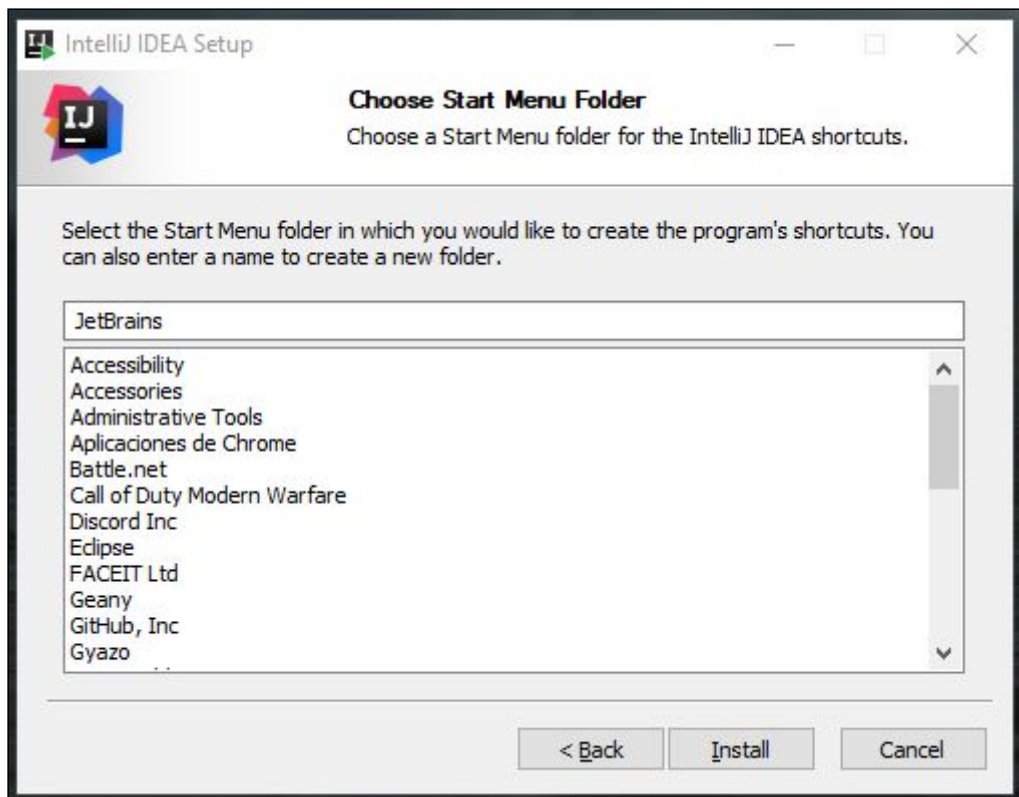


En la siguiente ventana seleccionaremos las siguientes opciones:



Con esta configuración se nos creará un acceso directo en el escritorio del programa, añadirá la variable Path del programa y nos abrirá de forma predeterminada los archivos .java con el programa.

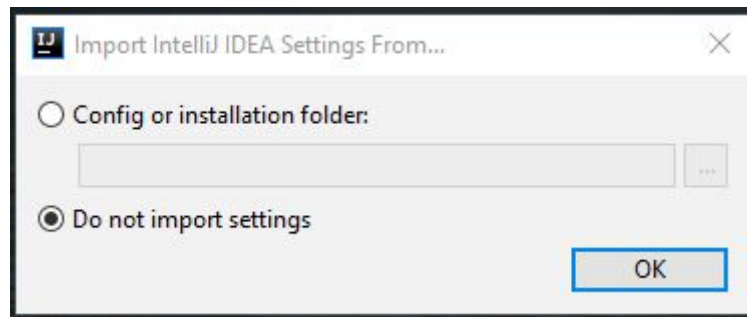
Haremos click en Next y le daremos a Install:



El proceso de instalación llevará unos minutos y tras él será necesario reiniciar el ordenador, el programa nos dará la opción de reiniciar automáticamente o manualmente.

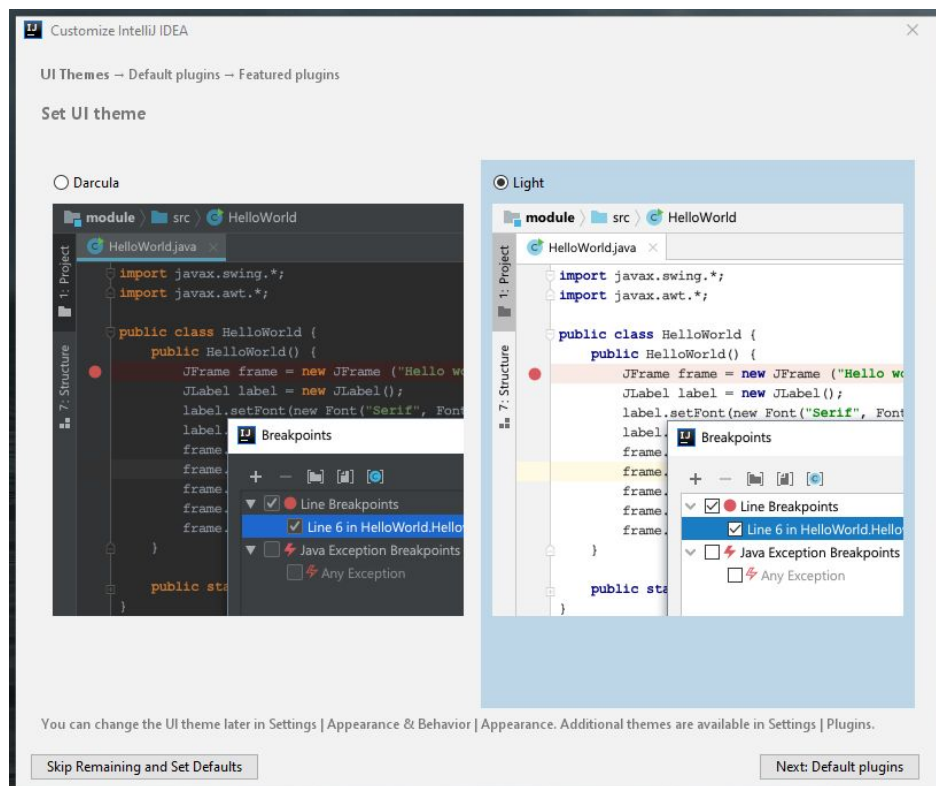
Configuración IntelliJ

Ejecutaremos el acceso directo que nos ha creado el proceso de instalación en el Escritorio, llamado: IntelliJ IDEA 2019.3 x64.

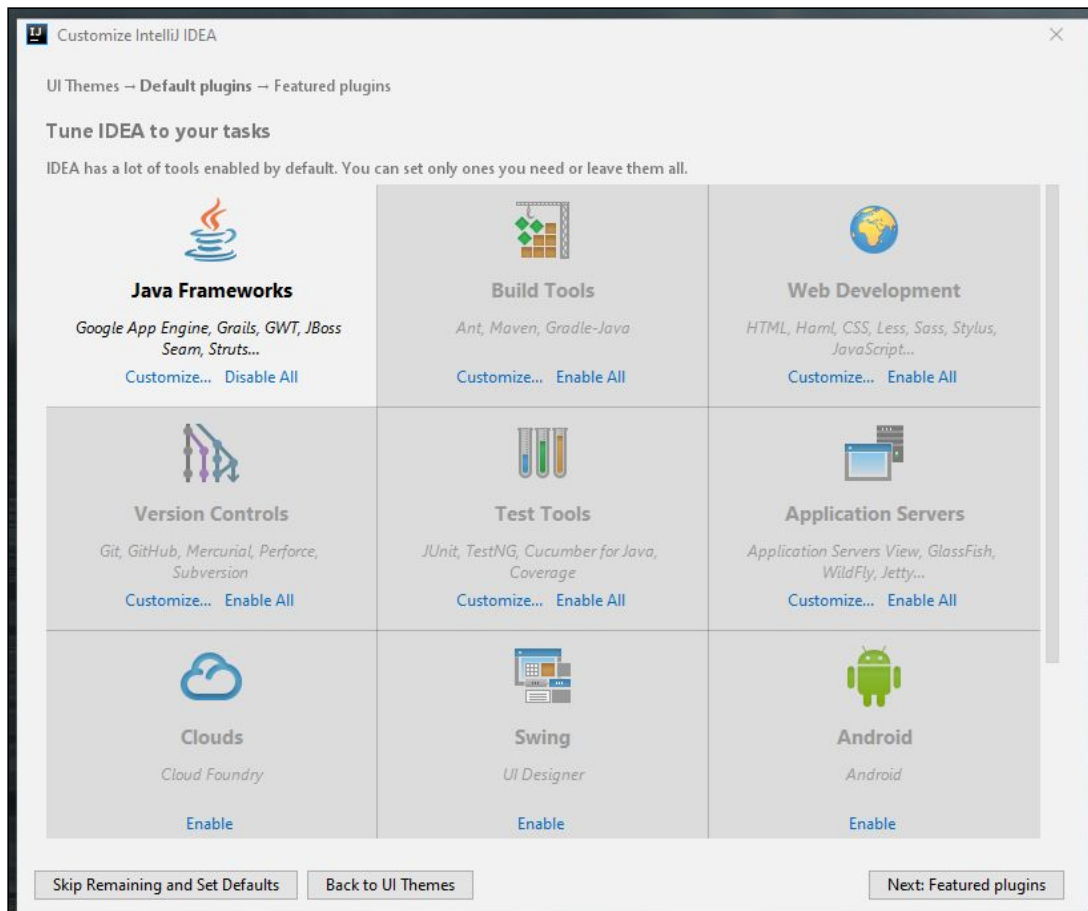


La primera ventana que nos aparecerá nos pedirá un archivo de configuración, seleccionaremos la opción “Do not import settings” ya que es la primera vez que usamos el programa.

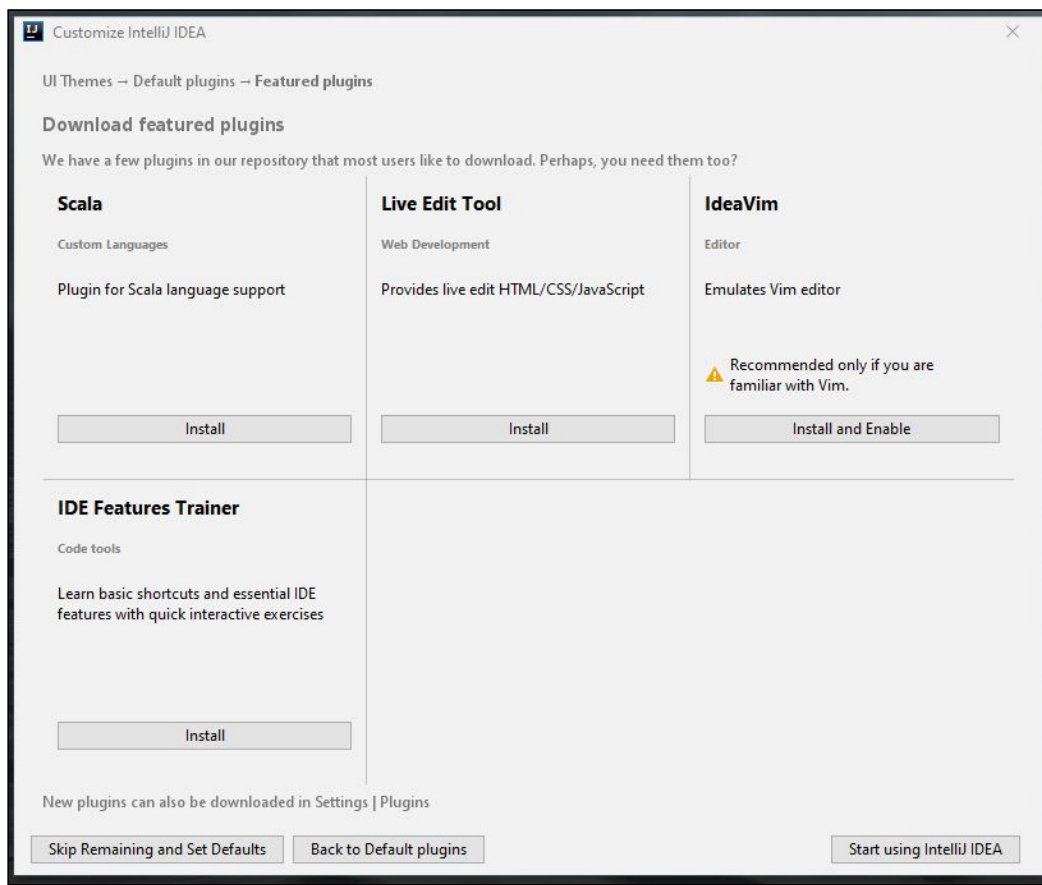
Seguidamente, tendremos la opción de seleccionar el tema del programa entre Oscuro y Claro, esta opción es totalmente estética y de gusto personal.



Ahora nos dara la opcion de instalar los plugins más básicos, como solo vamos a desarrollar aplicaciones Java, deshabilitamos el resto. Estos plugins se pueden volver a instalar en cualquier momento.



Y lo mismo haremos con la siguiente ventana pero directamente clickaremos la opción de “Start using IntelliJ IDEA”.



Primeros pasos en IntelliJ

Creación de proyecto

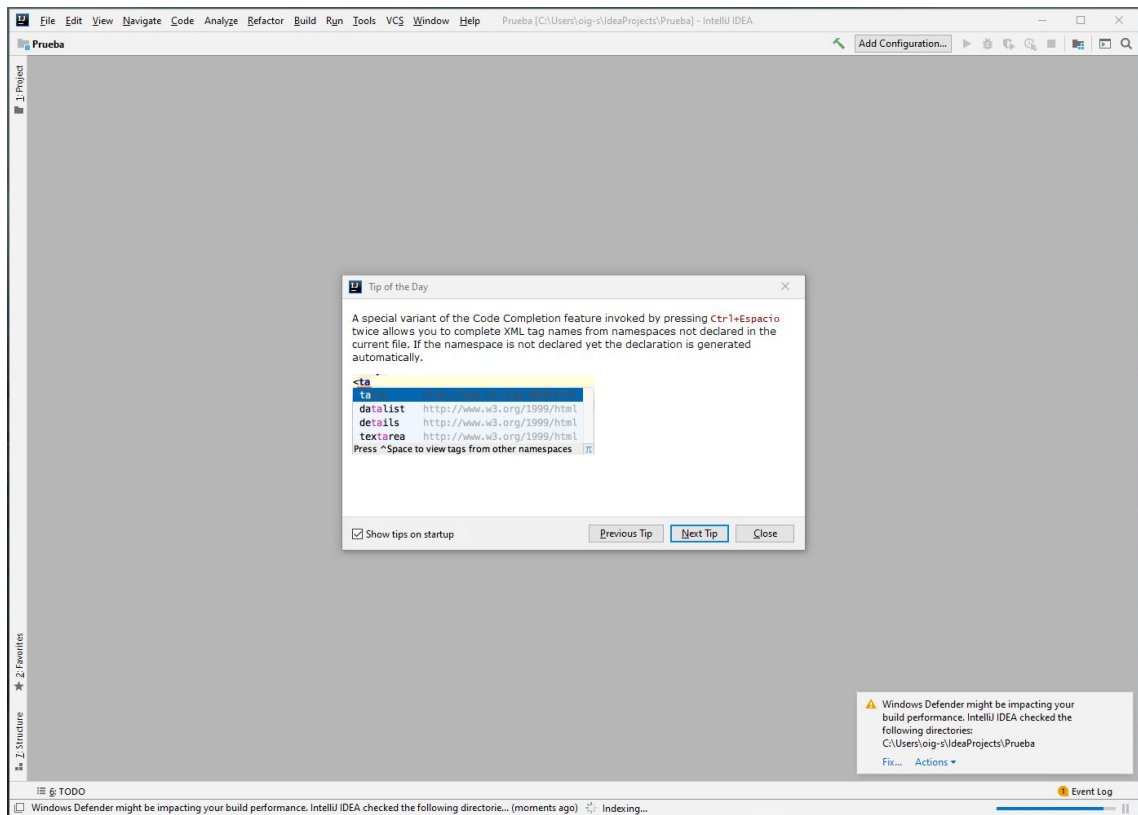
Por fin tendremos instalado y configurado IntelliJ, ahora nos aparecerá la ventana para crear, importar o abrir un proyecto, como no tenemos ninguno todavía clickaremos en “Create New Project”.

Haremos click en “Next” para crear un proyecto de Java directamente, de nuevo clickaremos “Next” para saltar el paso de la plantilla y nos pedirá que le pongamos un nombre a nuestro proyecto, pondremos “Prueba” por ejemplo.

Justo debajo seleccionaremos el directorio donde queremos que se nos guarde el proyecto, lo podemos cambiar o dejarlo como viene por defecto. En mi caso lo guardaré en:

```
D:\IntelliJ Projects\Prueba
```

Ya tendremos abierta la ventana principal del programa y nuestro primer proyecto creado.



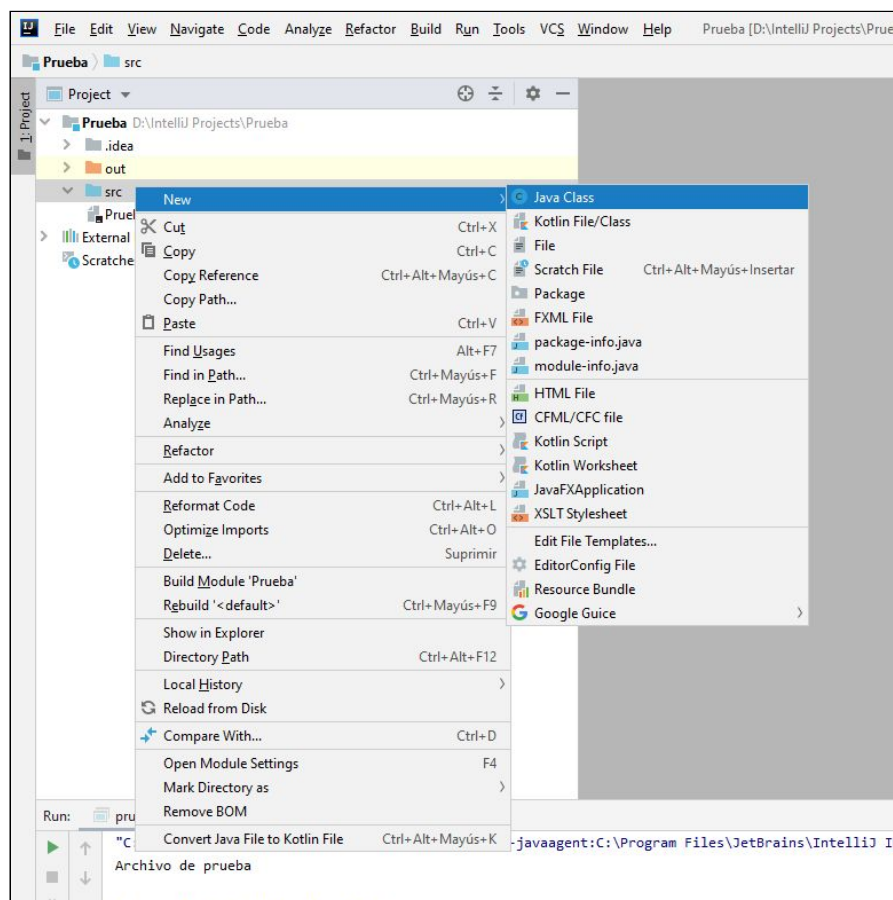
La cual nos mostrará consejos sobre su uso, además, en la esquina inferior izquierda nos aparece un aviso de sobre el antivirus. El cual nos dice que nuestro antivirus está afectando al rendimiento del programa, clickaremos en “Fix...” y después en “Configure Automatically” para que IntelliJ se autoexcluya de los escaneos del antivirus para no verse afectado por ellos.

Ejecutar clases en IntelliJ

Este programa está diseñado para trabajar en grandes proyectos que se dividen en paquetes. Por lo que compilar y ejecutar un archivo .java no sera tan facil como en Geany.

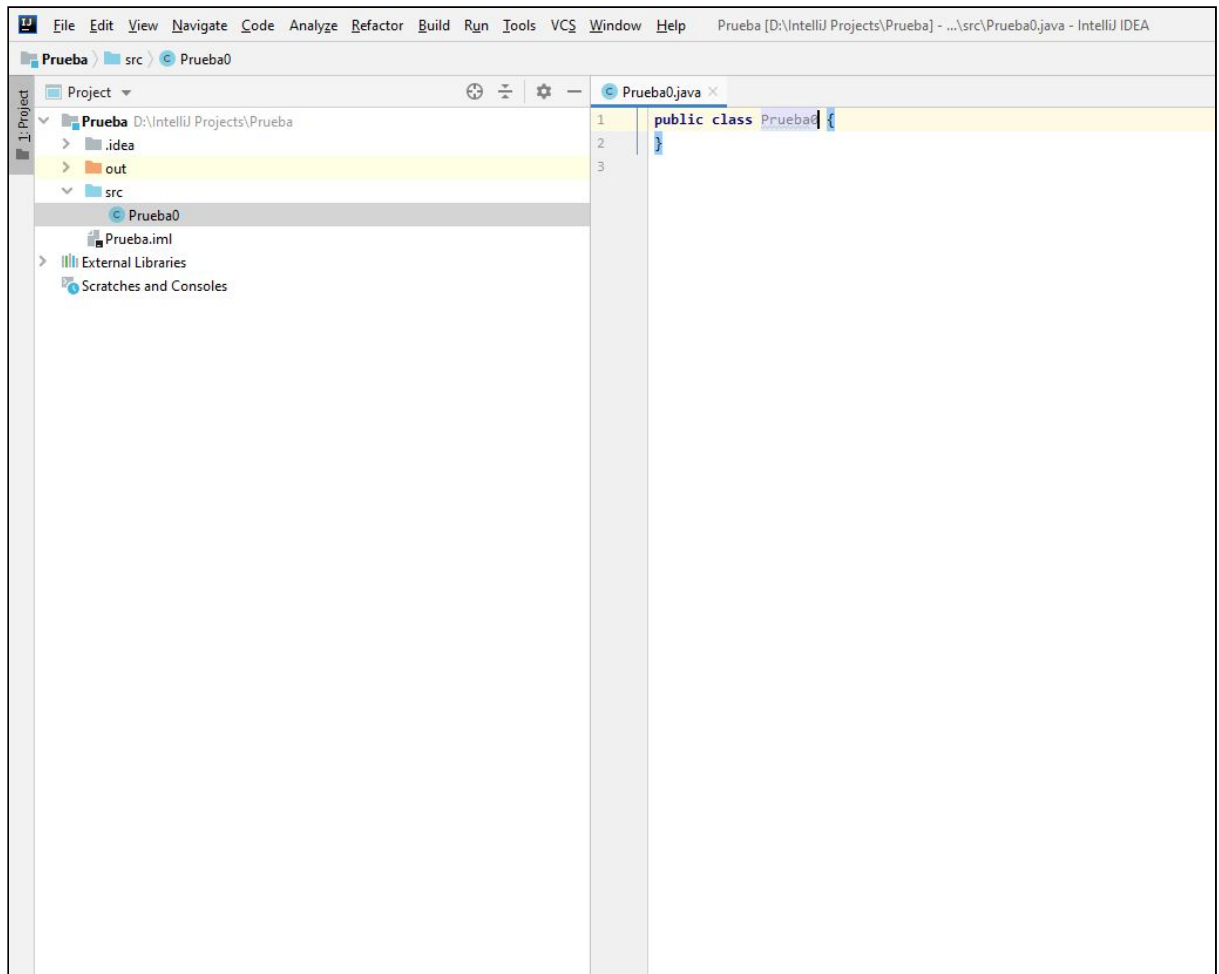
Tendremos dos opciones, crear nosotros el archivo .java desde cero dentro del proyecto o copiar archivos existentes a la carpeta del proyecto “src” desde el Explorador de Windows.

Para la primera opción deberemos seleccionar la carpeta “src” con click derecho que se encuentra en el menú de la izquierda llamado “1. Project”. Seleccionaremos New > Java Class.



Pondremos un nombre a nuestro archivo. En mi caso, le pondré “Prueba0”.

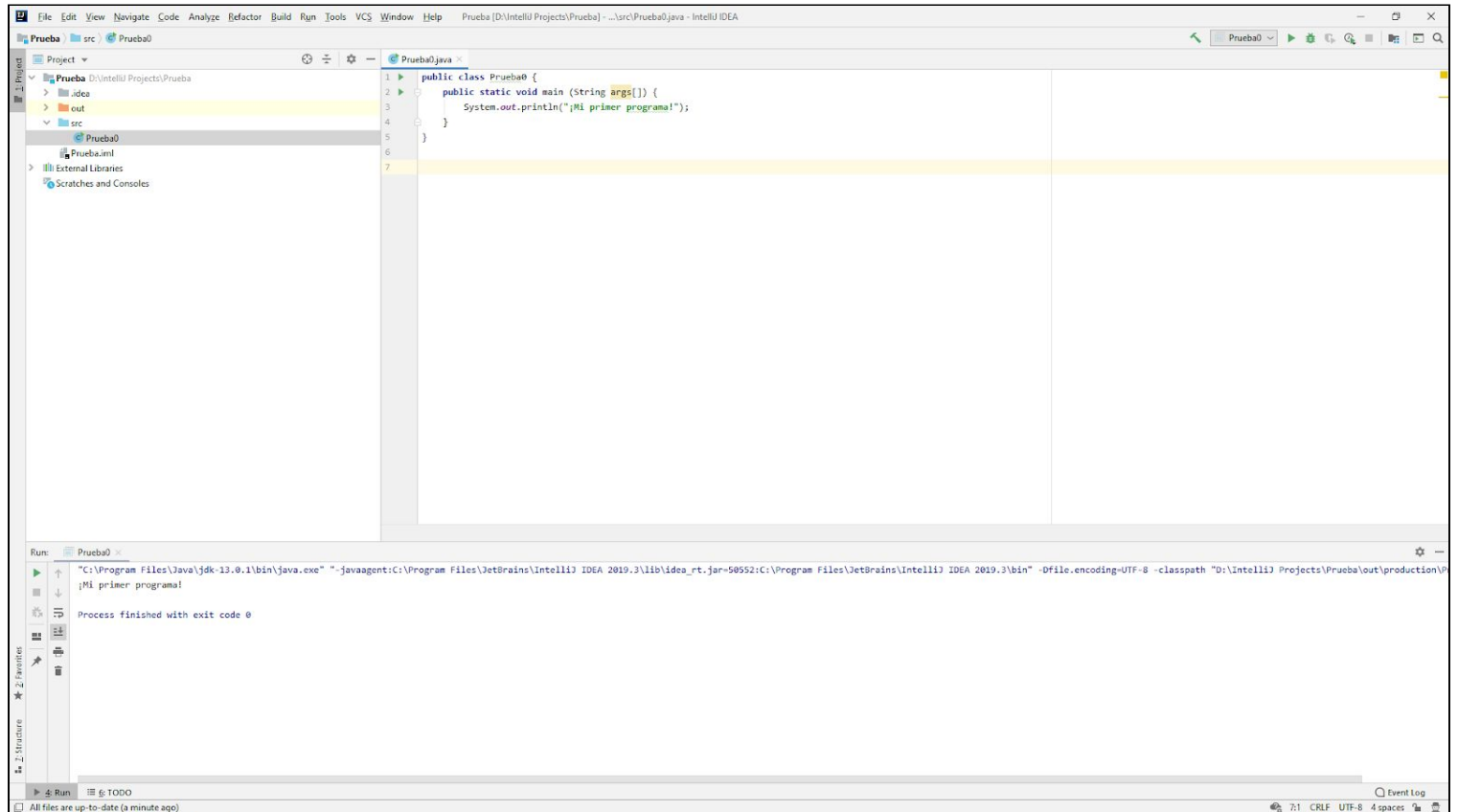
El resultado tiene que ser como este:



Ahora copiaremos y pegaremos este código:

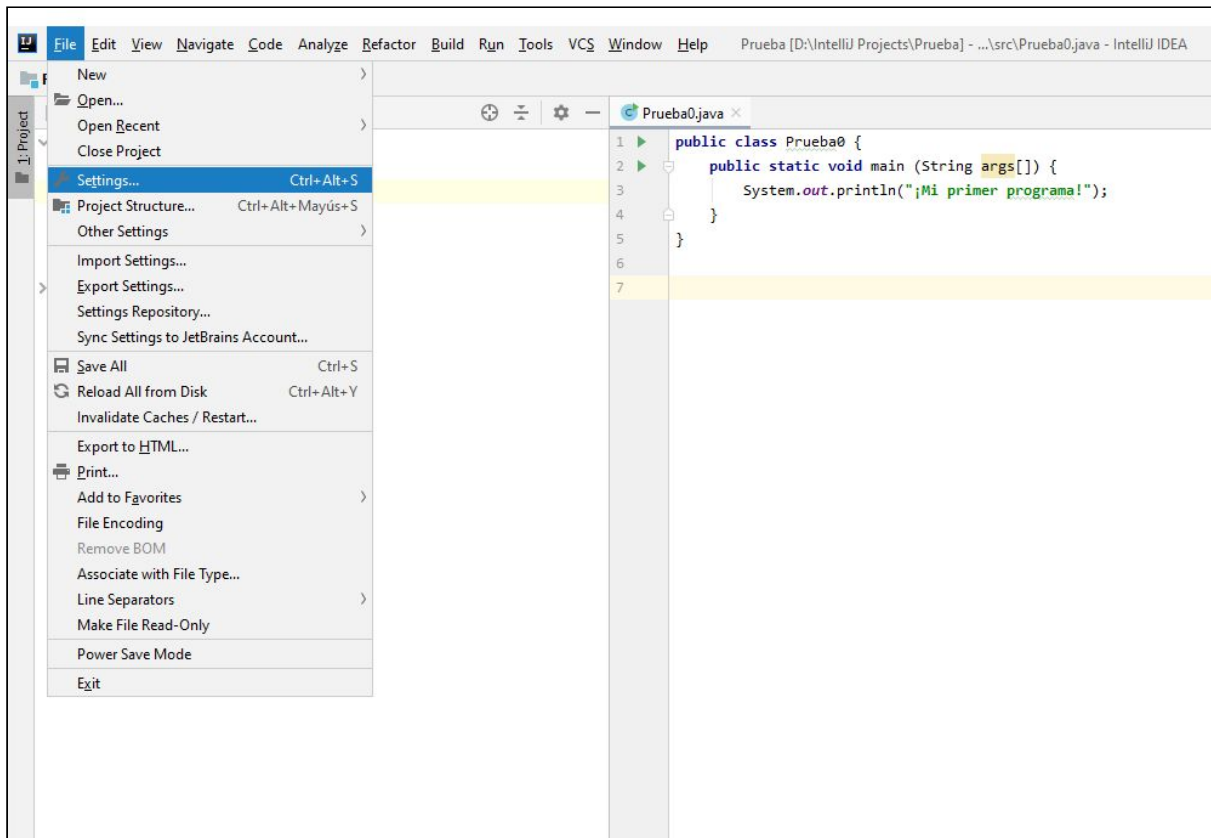
```
public class Prueba0 {  
    public static void main (String args[]) {  
        System.out.println("¡Mi primer programa!");  
    }  
}
```


Tendremos varias opciones para ejecutar el programa: Con la combinación Mayús + F10, dando botón derecho sobre cualquier punto del programa y seleccionando: “Run ‘Prueba0.main()’”, también tendremos un botón en la parte superior derecha de la pantalla. En cualquier caso, nos saldrá en la parte inferior del programa una nueva ventana con el resultado de la ejecución.

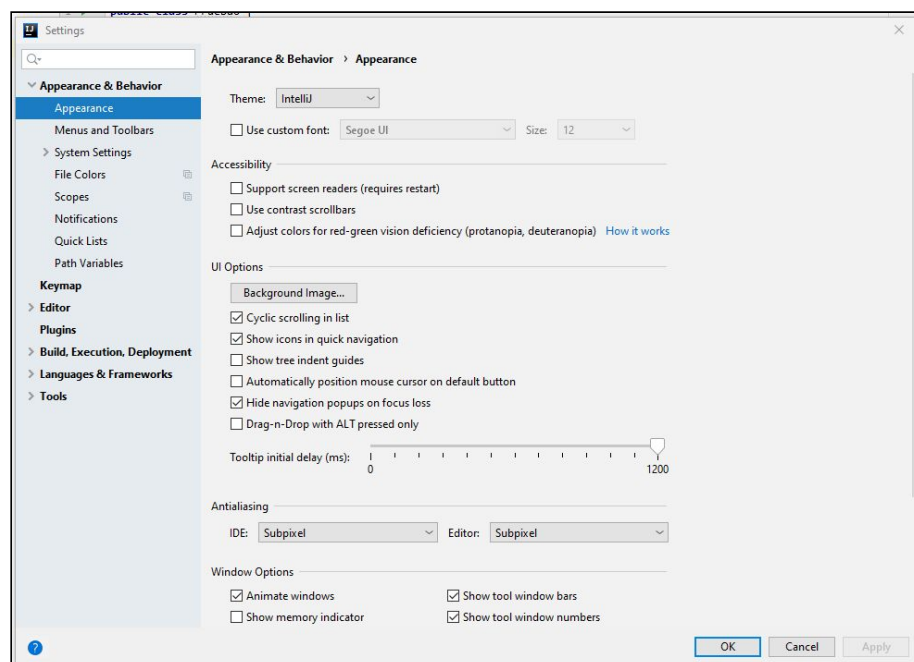


Ajustes y preferencias de IntelliJ

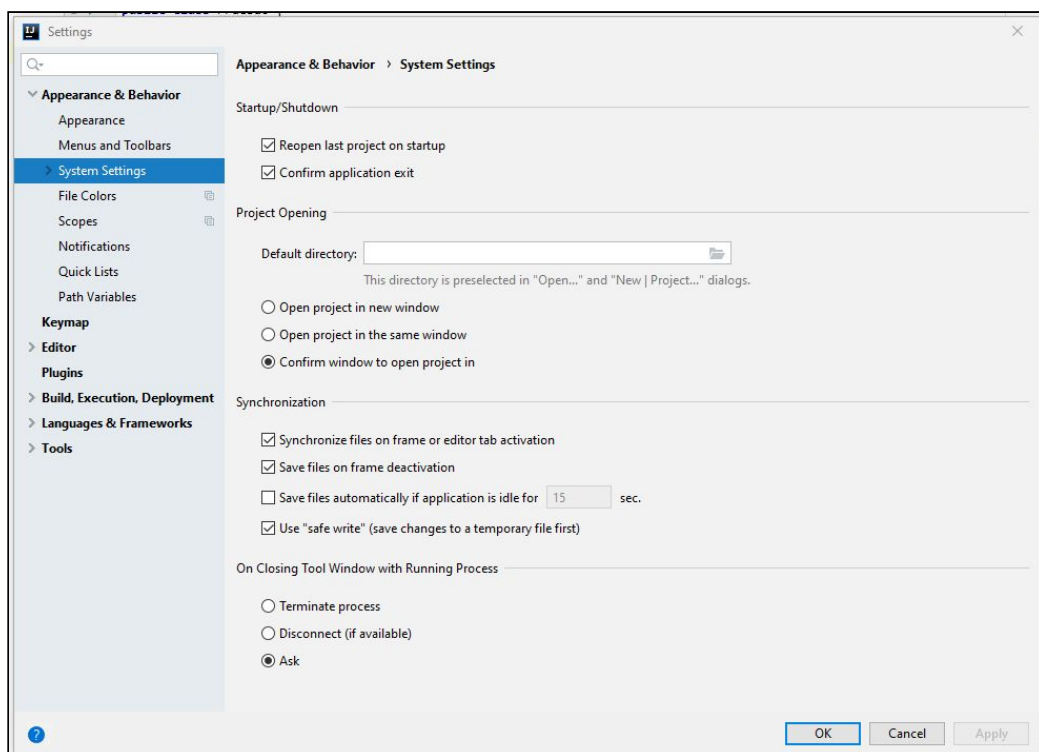
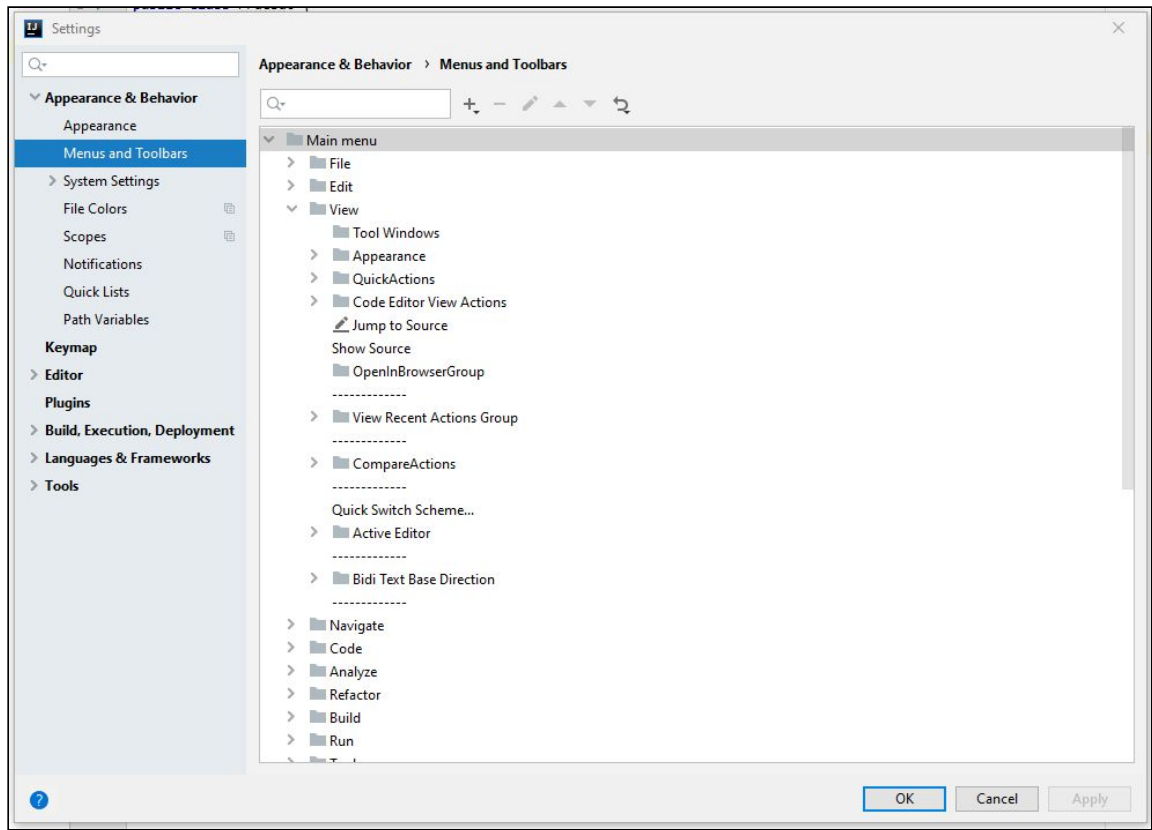
Entrando en el menú “File” encontraremos las opciones más generales y entre ellas se encuentra la opción “Settings”. También podemos acceder a ella mediante Ctrl+Alt+S.



La primera seccion sera la de Apariencia donde podremos personalizar en tema, la fuente, opciones de daltonismo etc.



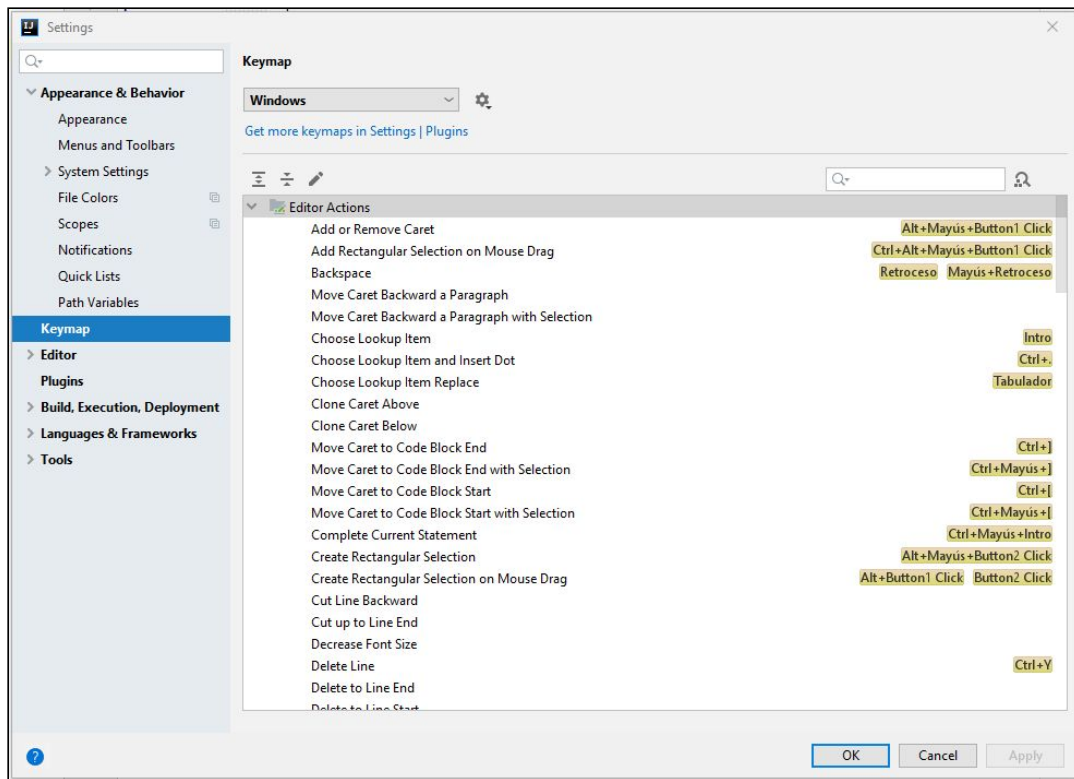
En la siguiente sección encontraremos la personalización de menús y barras de herramientas.



Seguido de los ajustes de sistema donde podremos ajustar como se comporta IntelliJ respecto a los proyectos (directorio, como abrir, guardados automáticos...)

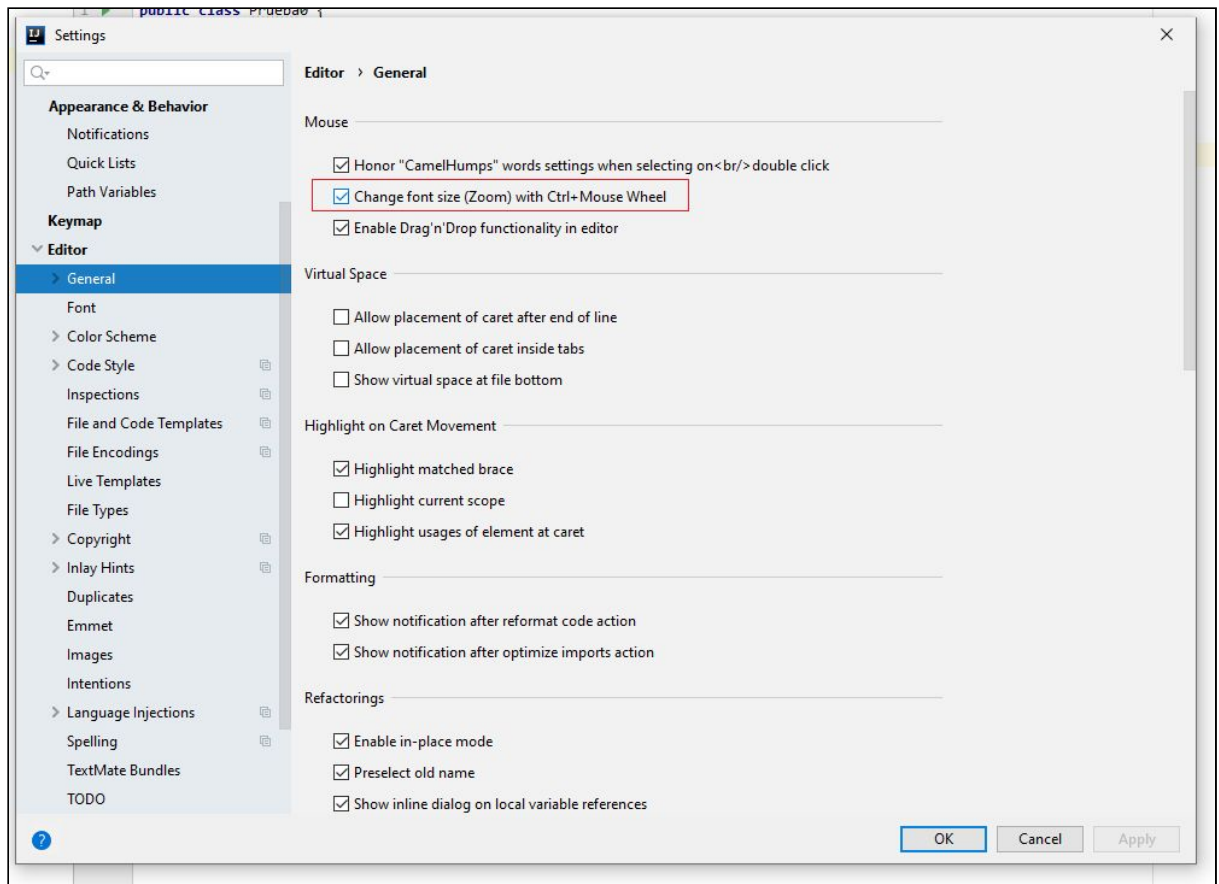
Estas son todas las opciones de apariencia y comportamiento más útiles.

La siguiente sección cubre las funciones de las teclas y las combinaciones de teclas.



Podremos modificar este apartado por si estamos acostumbrados a otros entornos de desarrollo con diferentes combinaciones, pero en general, son bastante similares.

En la siguiente sección tendremos las opciones del editor, dentro de la sección “General”. Entre todas estas opciones hay una muy útil que es la de hacer zoom presionando la tecla Ctrl + Rueda del ratón.



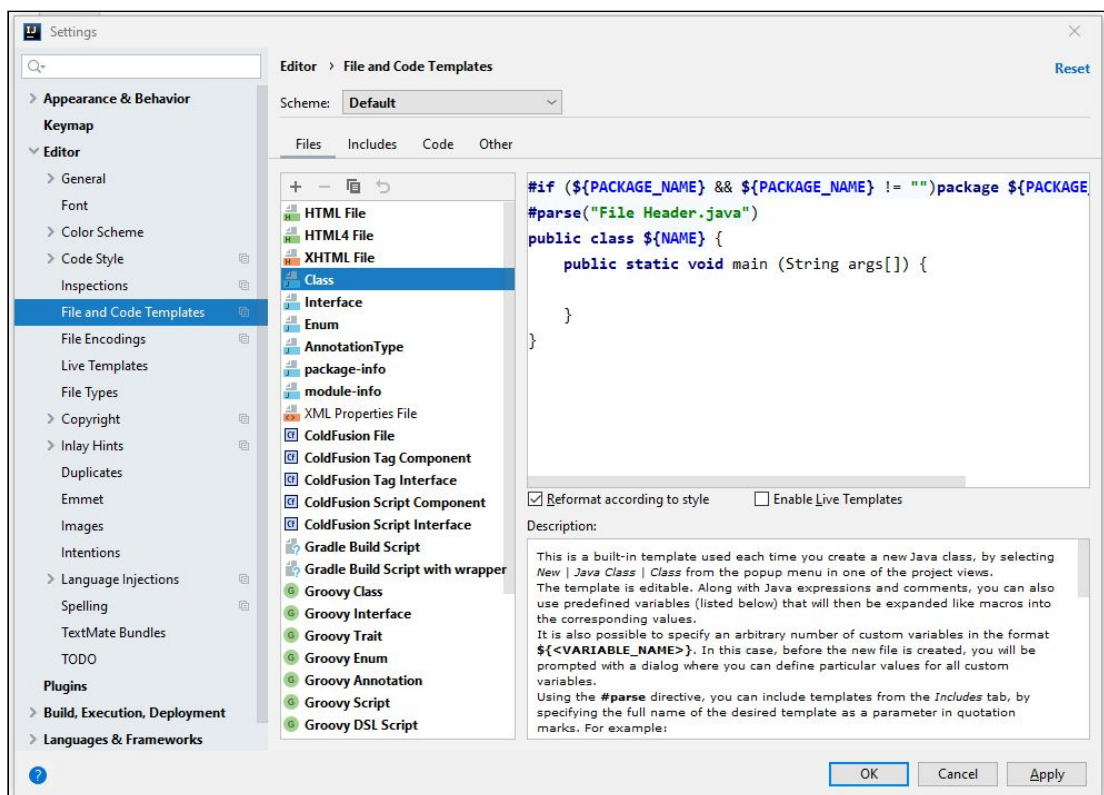
También encontraremos la opción de personalizar la fuente del editor en “Font”, los colores de los diferentes comandos, la inspección del código... Y dentro de estas opciones encontraremos la sección “File and Code Templates” donde podremos personalizar que aparece cuando creamos un archivo nuevo.

En IntelliJ al crear un archivo .java, no nos aparecerá automáticamente la línea del método “main”, aquí podremos cambiarlo.

Nos dirigiremos a “Class” y pegaremos este código:

```
#if (${PACKAGE_NAME} && ${PACKAGE_NAME} != "")package  
${PACKAGE_NAME};#end  
#parse("File Header.java")  
public class ${NAME} {  
    public static void main (String args[]) {  
  
    }  
}
```

Le daremos a aplicar, tiene que quedar tal que así:



Desde ahora, cuando creamos un archivo .java, nos aparecerá el método main automáticamente.

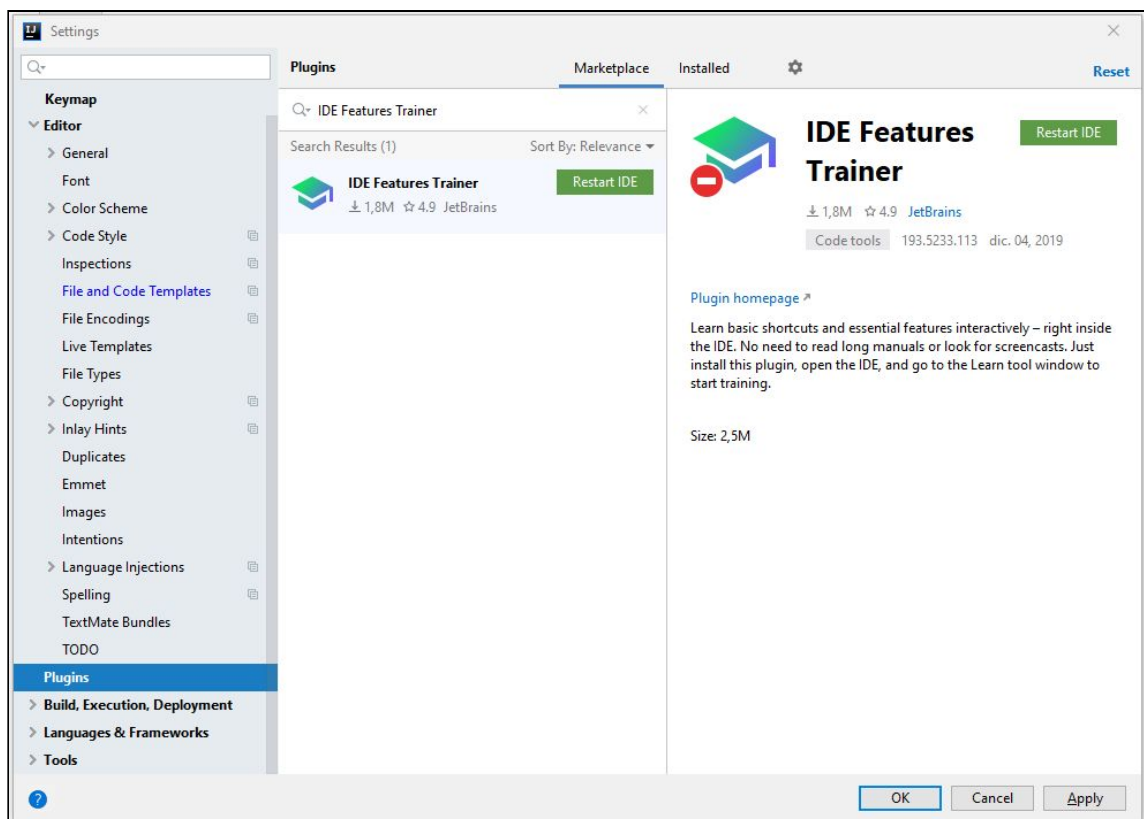
El resto de secciones no serán relevantes, salvo “Plugins”. Donde encontraremos múltiples plugins para IntelliJ (compatibilidades con otros lenguajes, temas, nuevas funcionalidades...). Estos plugins pueden ser propios de JetBrains o de los usuarios.

Instalación y uso de plugins (IDE Features Trainer)

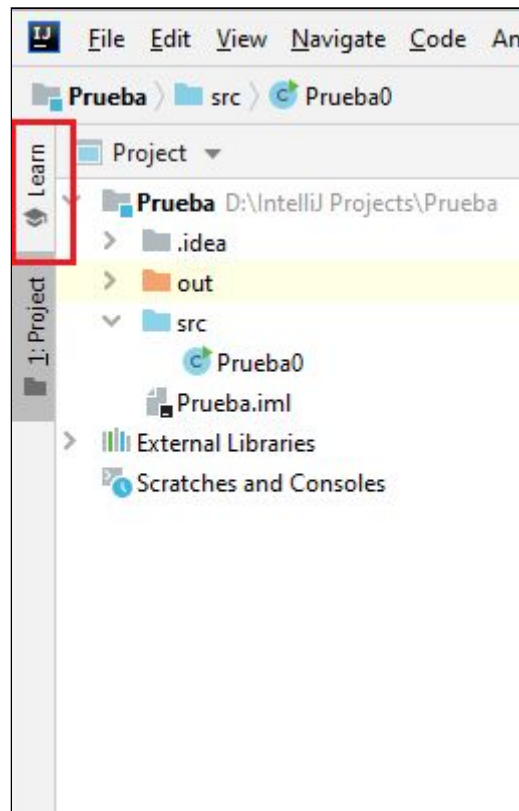
Entre ellos encontraremos “IDE Features Trainer”. Este plugin nos ofrece varios tutoriales sobre las funcionalidades más útiles que vienen incorporadas en el programa (selección múltiple, búsquedas, auto completar, autocorrección, mover bloques enteros de código...)

Para instalarlo, debemos buscarlo en la barra de búsqueda, escribiremos “IDE Features Trainer” y le daremos a “Install”.

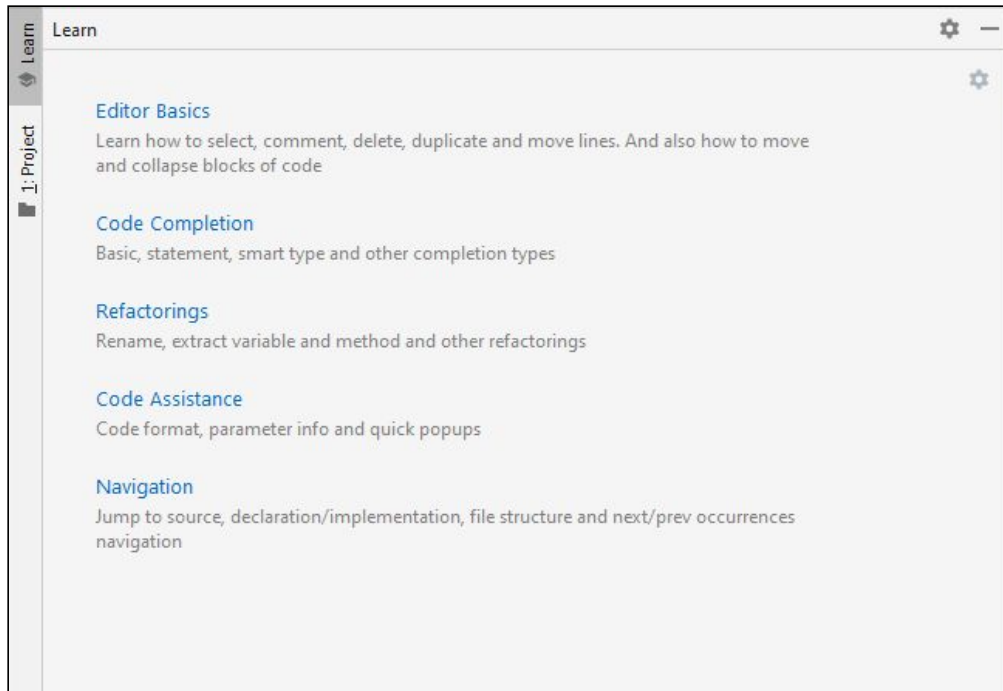
Una vez instalado el programa deberá reiniciarse para poder usar el plugin. Podemos hacer click en “Restart IDE” o hacerlo manualmente.



Una vez reiniciado el programa nos aparecera un nuevo menu “Learn” en la parte izquierda del programa.



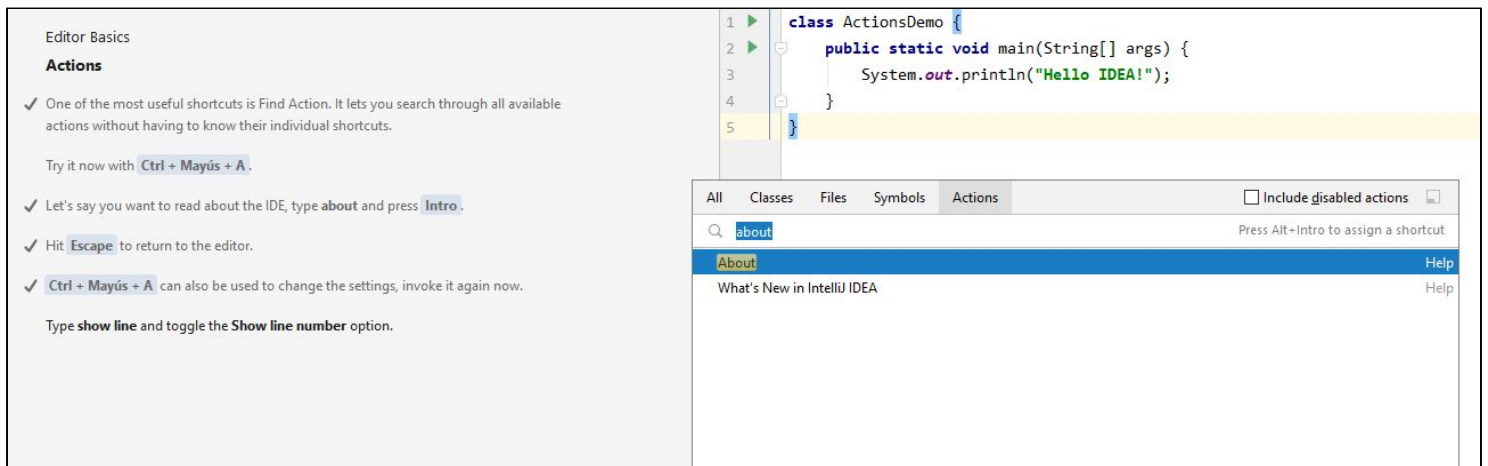
Donde encontraremos un menú con las diferentes secciones del tutorial: Funciones básicas del editor, auto completar código, refactorizar, asistencia de código y navegación.



Haremos click en “Editor Basics” para comenzar el primer nivel.

El programa nos mostrara los pasos que debemos seguir en la parte izquierda, cuando completamos uno mostrará el siguiente.

En el primer tutorial nos enseña a abrir la barra de búsqueda donde encontraremos acciones, opciones y ajustes, personalización... Sin necesidad de navegar por todos los menús.



Recomiendo completar el tutorial al 100% para aprender funciones útiles que nos ayudarán a programar más rápido.

Por suerte para ti, ya has finalizado este documento de 28 páginas y ya estas listo para desarrollar Java en IntelliJ. Aunque de momento no estemos preparados para usarlo al 100%, espero que esto te sirva para cuando de verdad necesitemos un entorno de desarrollo potente y resolutivo como este.

