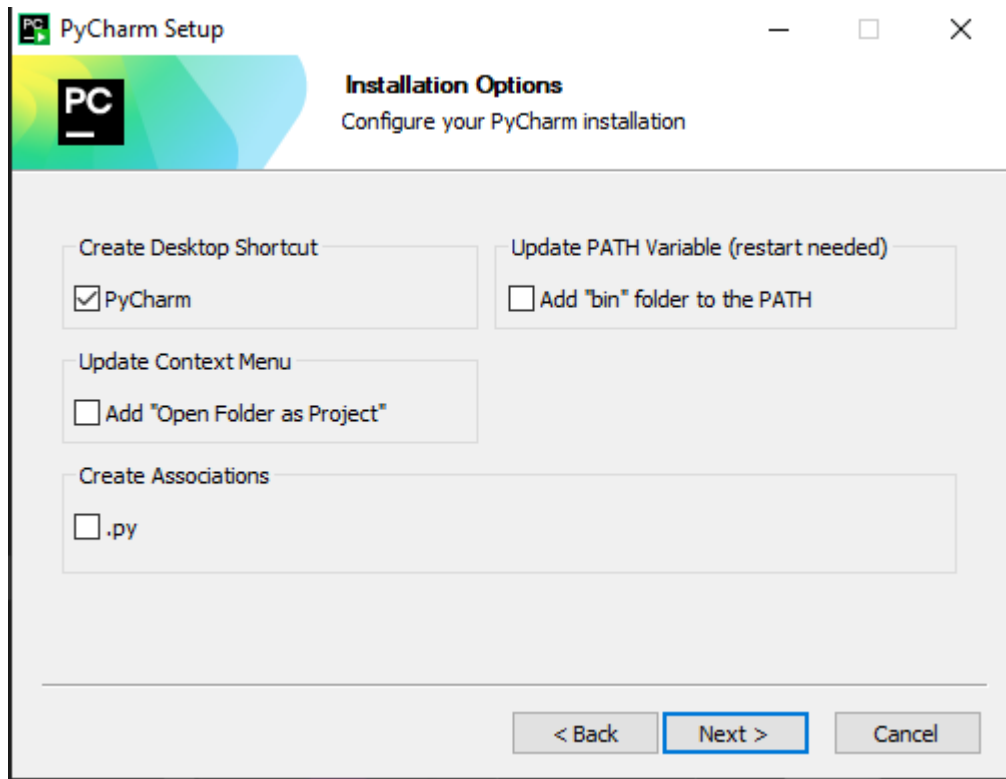


2.1. Instalación de entornos de desarrollo, propietarios y libres (CE 2.a)

1. ¿Qué diferencias encontraste en el proceso de instalación entre el IDE propietario y el libre?



Al instalar PyCharm me he dado cuenta de que es un IDE más completo y que toma más tiempo durante su instalación.

Por el contrario, Visual Studio Code o VCode es muchísimo más ligero y no requiere de configuraciones

2. ¿Qué ventajas identificaste en cada uno de los entornos durante la instalación?

En la instalación de Visual Code era muchísimo más rápida que en la instalación de un IDE propietario como PyCharm que era mucho más lenta, pero permitía mucha más configuración en su instalador.

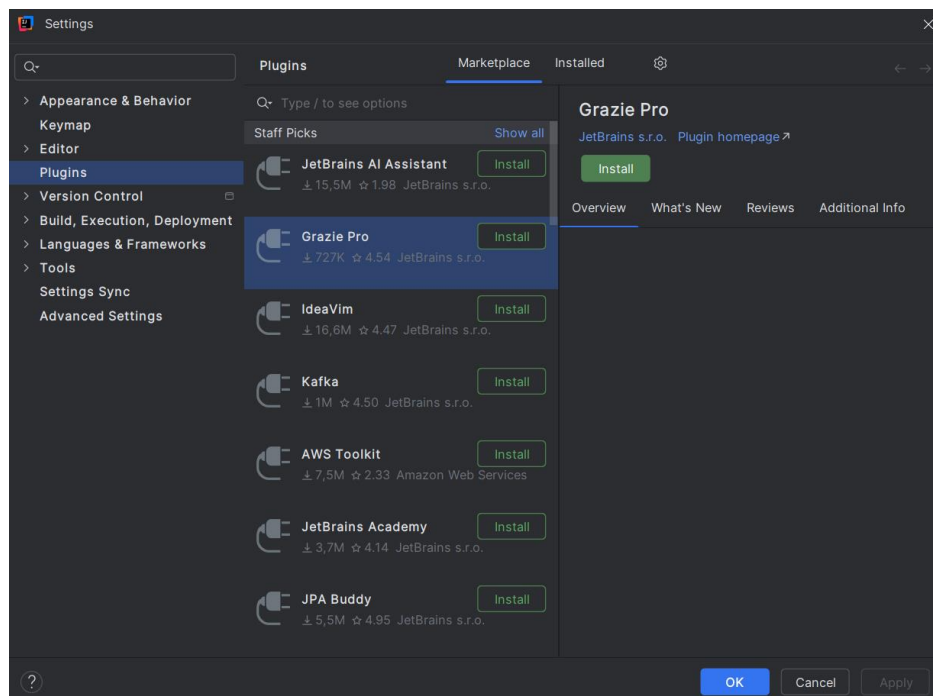
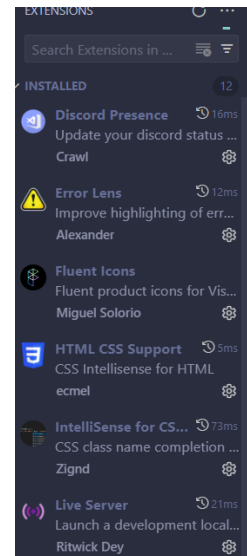
2.2. Gestión de módulos y extensiones en el entorno de desarrollo (CE 2.b)

1. ¿Cómo fue el proceso de instalación de extensiones o módulos en cada IDE?

Las extensiones en VCode son muy fáciles de instalar, ya que el propio IDE te trae un repo con todas las extensiones publicadas, que no son pocas, mientras que otros IDE como IntelliJ o Pycharm tienen un repositorio más pequeño pero centrado en el lenguaje que maneja cada IDE

2. ¿Qué beneficios proporcionan las extensiones o plugins que instalaste para el desarrollo de tus proyectos?

En VCode me permiten detectar errores de sintaxis en el código fácilmente y accesibilidad para muchos otros lenguajes, en IntelliJ, los plugins que he visto mayormente son compatibilidades del lenguaje de ese IDE con otros o algunas IA para el código

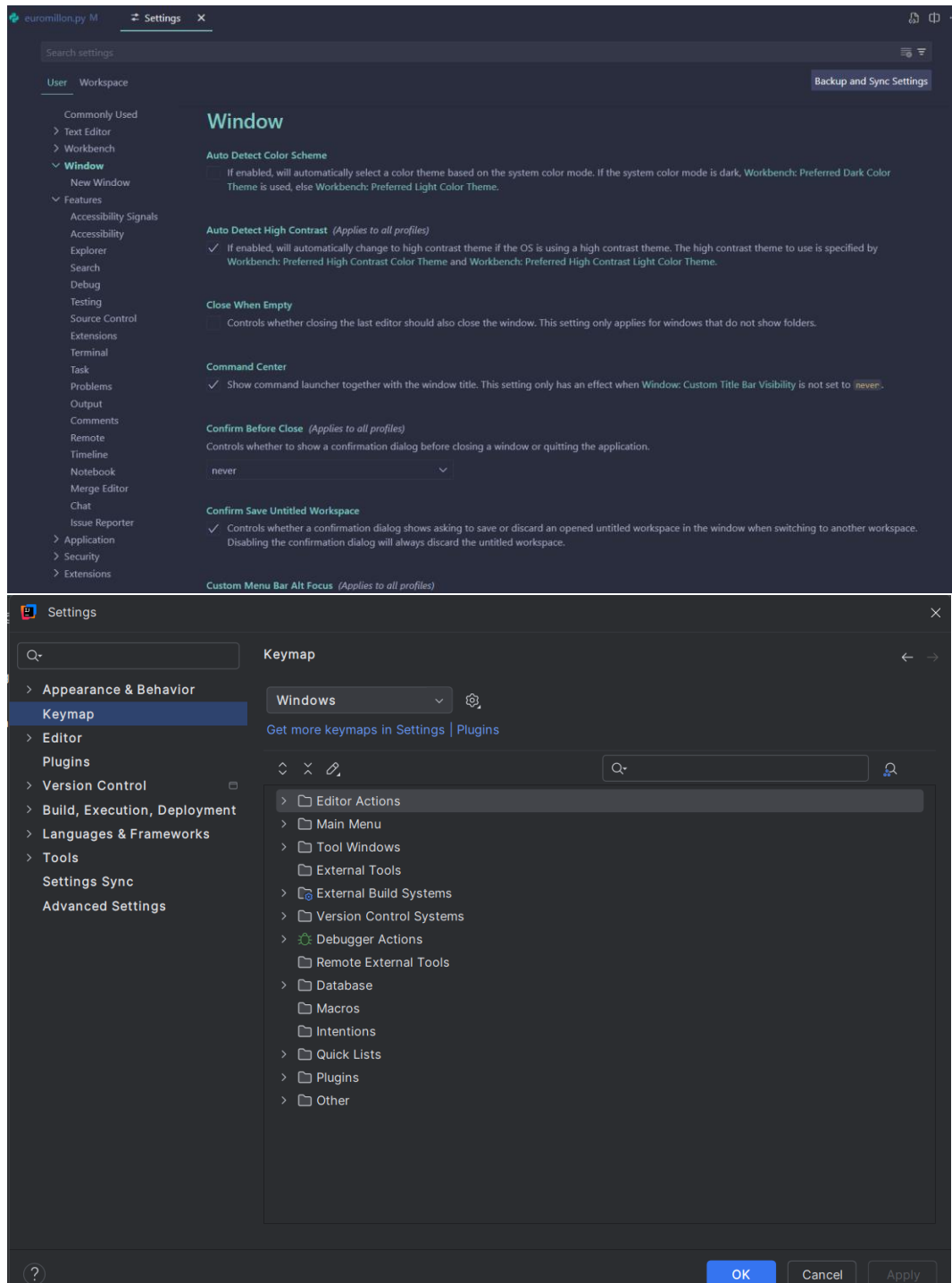


2.3 Personalización y automatización del entorno (CE 2.c)

1. ¿Qué aspectos del entorno personalizaste y cómo mejoró tu experiencia de desarrollo?

La verdad que, en tema de personalización, el IDE propietario, en este caso IntelliJ es mucho más completo que Visual Studio Code, ya que permite personalizar todo tipo de configuraciones dentro del propio entorno.

No toque ninguna configuración, ya que me gusta trabajar con las herramientas por defecto.



- ¿Cómo configuraste la automatización de tareas y en qué te benefició durante el trabajo?

En VCode se podrían usar extensiones como Prettier o Task Runner para automatizar tareas como el formateo del código, análisis del código o realizar pruebas.

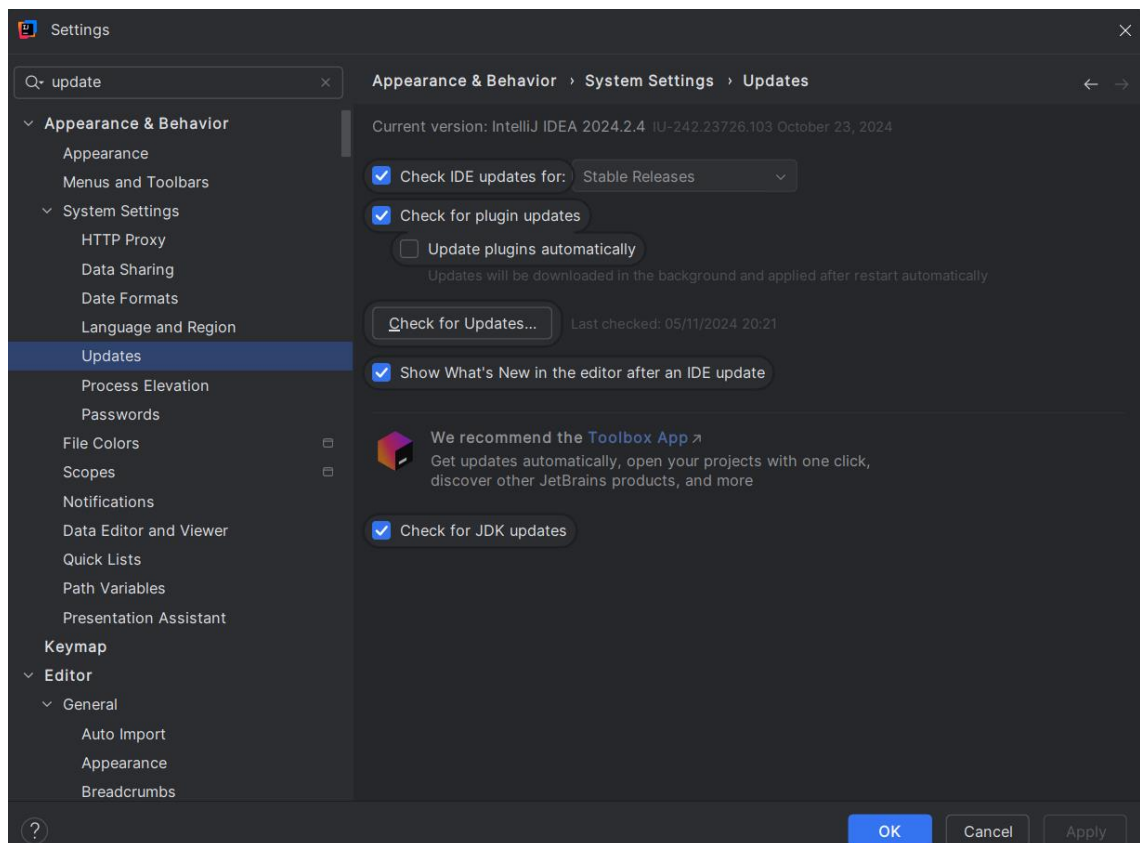
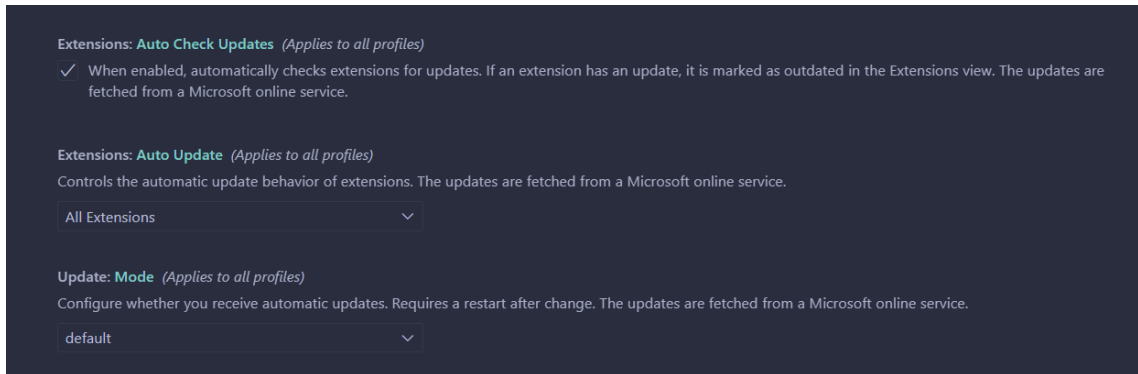
Mientras que, en Pycharm tenemos que configurar su debugger para ejecutar scripts que automaticen las tareas

2.4 Configuración del sistema de actualización del entorno de desarrollo (CE 2.d)

1. ¿Cómo configuraste las actualizaciones automáticas en cada IDE?

En Visual Studio Code solo busqué updates y marqué todo encendido para que se me actualizaran tanto las extensiones como el IDE solo.

En IntelliJ, hay un apartado específico para personalizar las actualizaciones.



2. ¿Por qué es importante mantener el IDE actualizado en proyectos de desarrollo?

Para poder recibir las últimas novedades de cada entorno que podrían tener nuevas herramientas para facilitar la escritura de código o mejoras de rendimiento, o calidad de vida.

2.5 Generación de ejecutables a partir de código fuente en distintos lenguajes en un mismo IDE (CE 2.e)

1. ¿Cuál fue el proceso para ejecutar el mismo programa en diferentes lenguajes dentro del mismo IDE?

La verdad que a la hora de ejecutar un script en Python y en Java, noté la fluidez de la ejecución de Java, que fue un poco más veloz a la hora de ejecutar un programa

```

prueba.py > ...
1 for i in range(10, -1, -1):
2     print(i)
3     print("¡Despegue!")
4

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```

6
5
4
3
2
1
0
¡Despegue!
PS C:\Users\coman\Documents\DAM\pruebas> & C:/Users/coman/AppData/Local/Programs/Python/Python312/python.exe c:/Users/coman/Documents/DAM/pruebas/prueba.py
10
9
8
7
6
5
4
3
2
1
0
¡Despegue!
PS C:\Users\coman\Documents\DAM\pruebas>

```

```

prueba.java > ...
1 public class prueba {
2     public static void main(String[] args) {
3         for (int i = 10; i >= 0; i--) {
4             System.out.println(i);
5         }
6         System.out.println(x:"¡Despegue!");
7     }
8 }
9

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```

PS C:\Users\coman\Documents\DAM\pruebas> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\coman\AppData\Roaming\Code\User\workspaceStorage\5761641b246c018ac647375f7ec564b7\redhat.java\jdt_ws\pruebas_b91554ec\bin' 'prueba'
10
9
8
7
6
5
4
3
2
1
0
¡Despegue!
PS C:\Users\coman\Documents\DAM\pruebas>

```

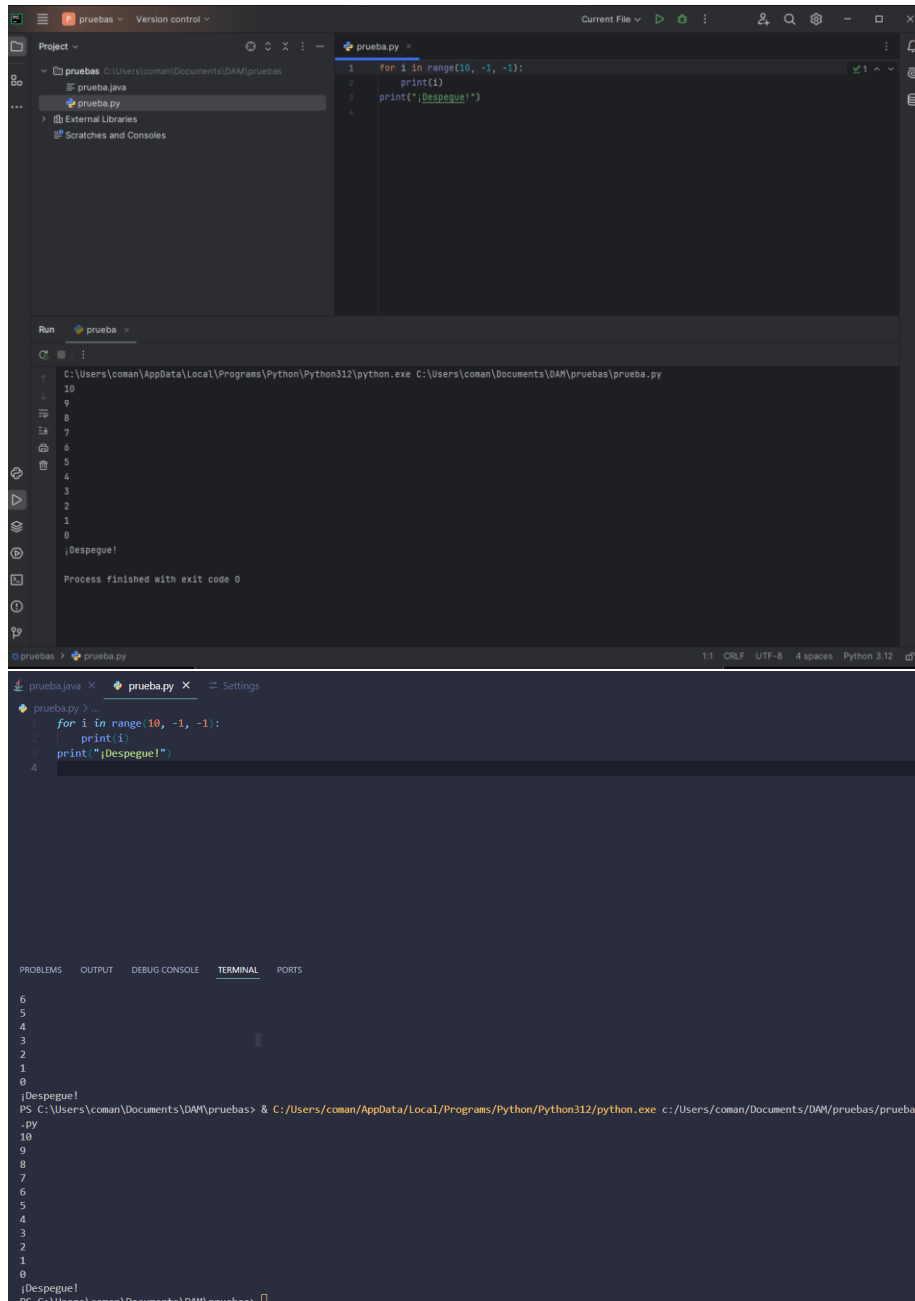
- ¿Qué diferencias encontraste en la generación del ejecutable entre los dos lenguajes?

A la hora de ejecutar Python solamente hice uso de un intérprete para poder ejecutarlo, pero cuando tuve que ejecutar el script de Java, tuve que instalar más extensiones y el JDK, comúnmente conocido como la JVM en la que trabaja Java.

2.6 Generación de ejecutables con diferentes IDEs a partir del mismo código fuente (CE 2.f)

1. ¿Qué diferencias encontraste al ejecutar el mismo código fuente en diferentes IDEs?

A la hora de ejecutar un script pequeño tanto en Pycharm como en Visual Studio Code, en visual studio lo ejecuté directamente, pero al ejecutarlo en Pycharm me pidió configurar el intérprete.



2. ¿Cuál de los IDEs te pareció más cómodo o eficiente para ejecutar el código Python o el lenguaje que hayas elegido? ¿Por qué?

Me pareció más cómodo Visual Studio Code porque ya lo tengo acomodado con todas las extensiones que uso normalmente y porque estoy más acostumbrado a usarlo