Enunciado.

Ejercicio 1

De igual manera a lo visto en el tema, ahora te proponemos un ejercicio del tipo productor-consumidor que mediante un hilo productor almacene datos (15 caracteres) en un búfer compartido, de donde los debe recoger un hilo consumidor (consume 15 caracteres). La capacidad del búfer ahora es de 6 caracteres, de manera que el consumidor podrá estar cogiendo caracteres del búfer siempre que éste no esté vacío. El productor sólo podrá poner caracteres en el búfer, cuando esté vacío o haya espacio.

Te mostramos una posible salida del programa que debes realiza

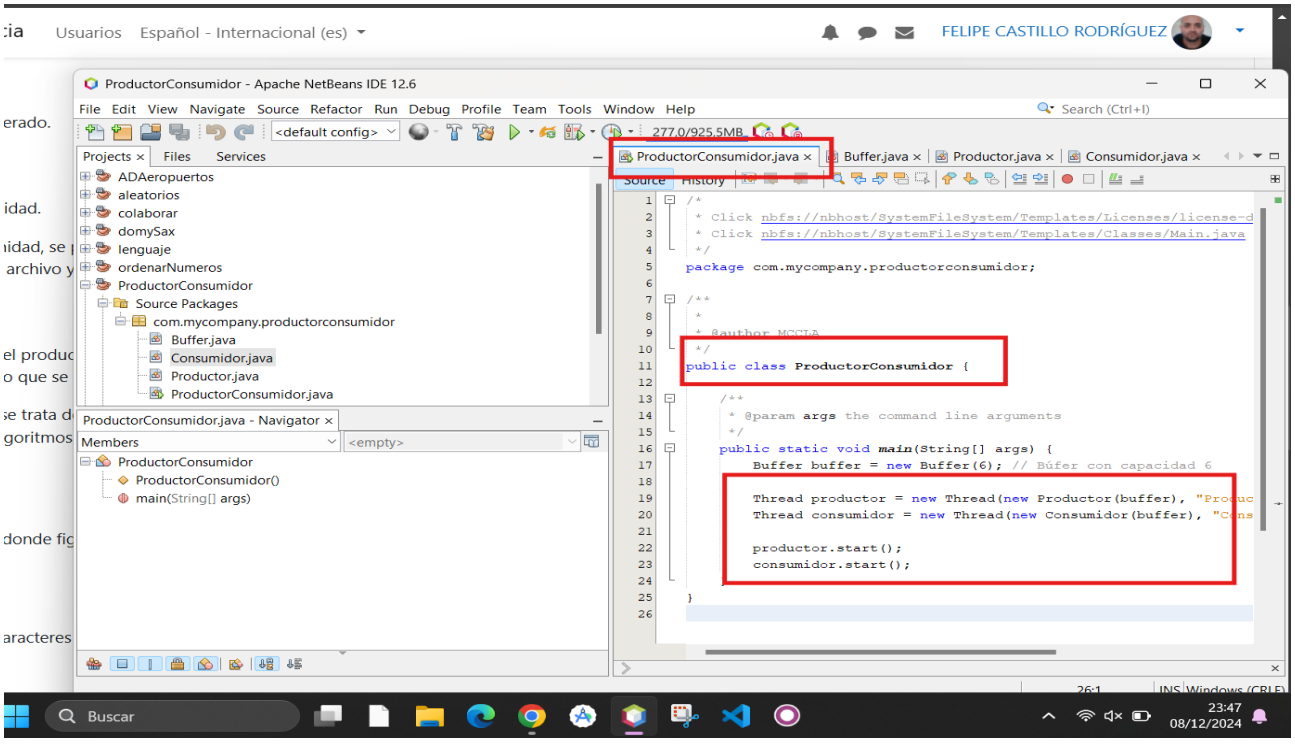
Ejercicio 2

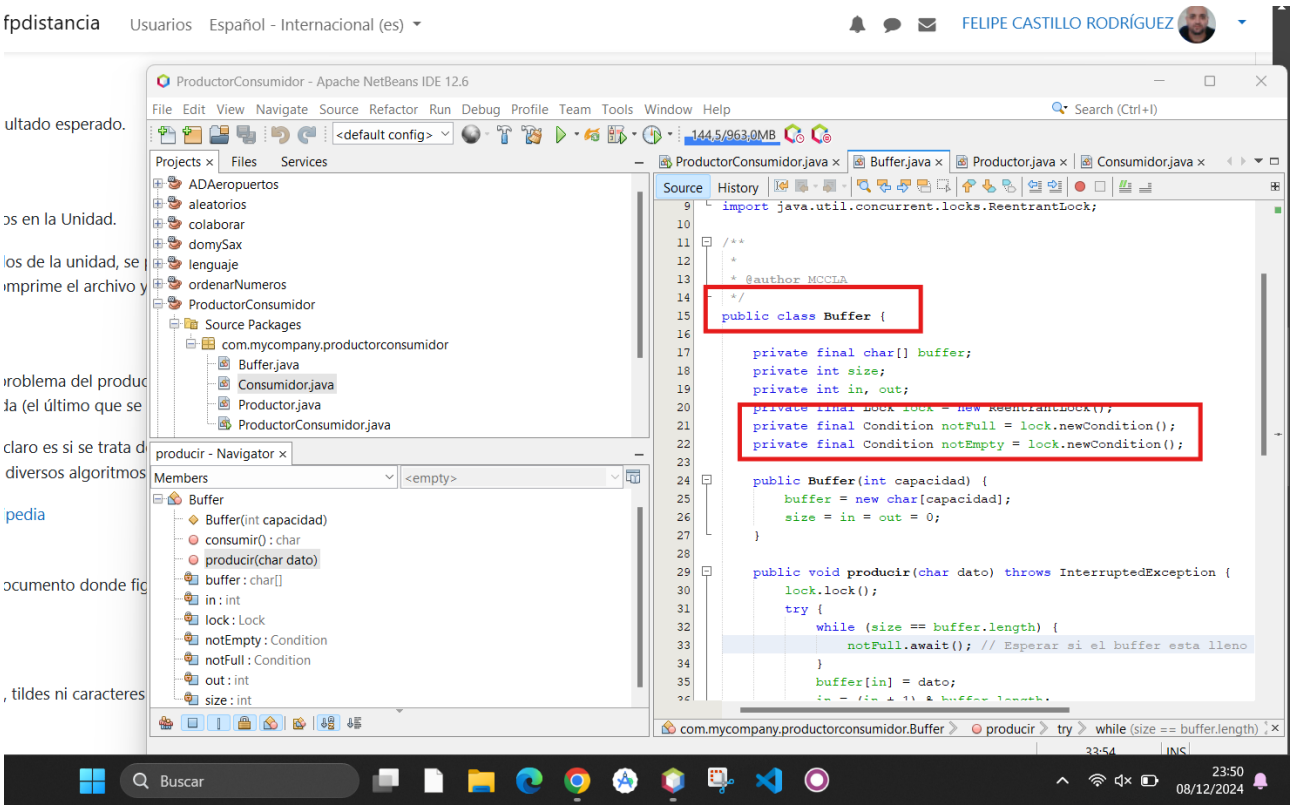
De igual manera a lo visto en el tema, ahora te proponemos que resuelvas el clásico problema denominado "La cena de los filósofos" utilizando la clase `Semaphore` del paquete `java.util.concurrent`.

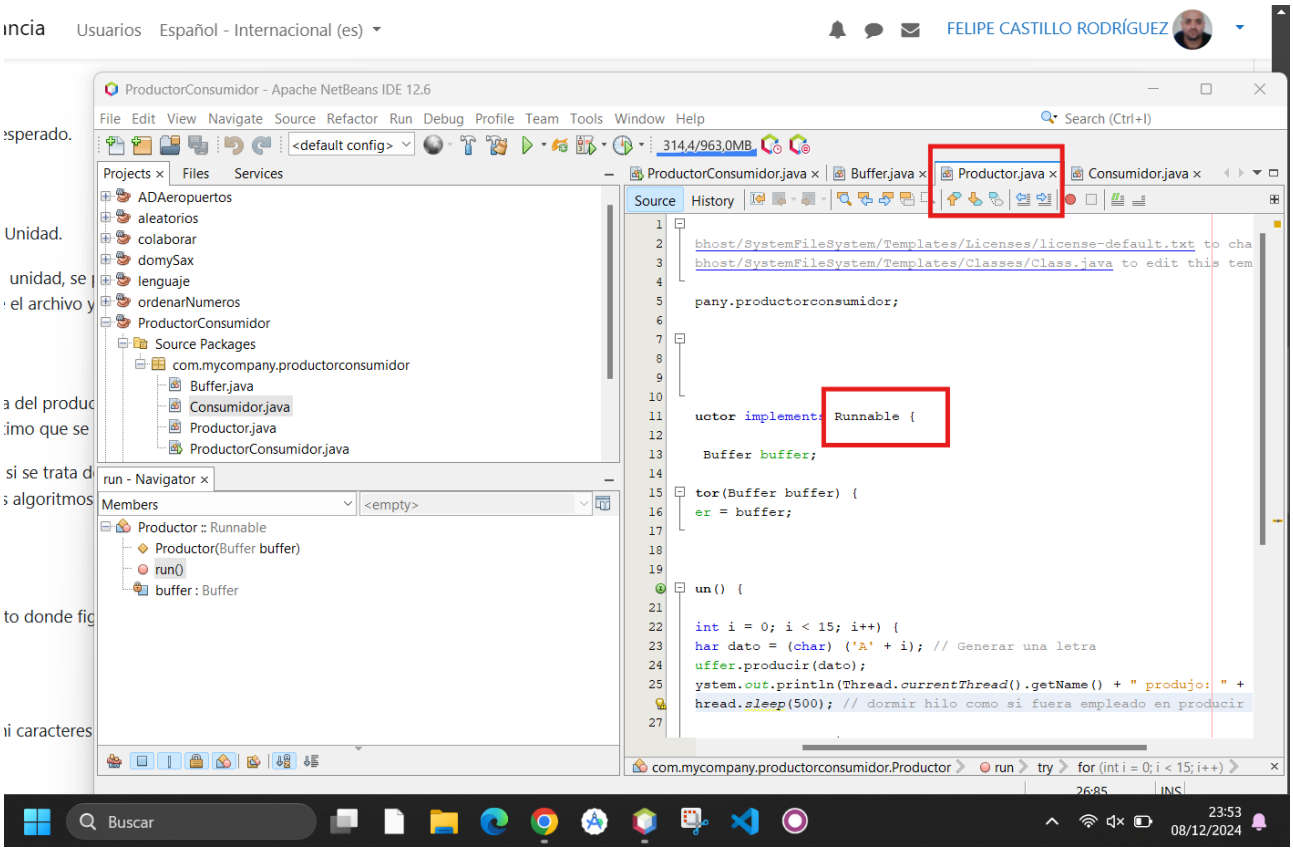
El problema es el siguiente: Cinco filósofos se sientan alrededor de una mesa y pasan su vida comiendo y pensando. Cada filósofo tiene un plato de arroz chino y un palillo a la izquierda de su plato. Cuando un filósofo quiere comer arroz, cogerá los dos palillos de cada lado del plato y comerá. El problema es el siguiente: establecer un ritual (algoritmo) que permita comer a los filósofos. El algoritmo debe satisfacer la exclusión mutua (dos filósofos no pueden emplear el mismo palillo a la vez), además de evitar el interbloqueo y la inanición.

EJERCICIO 1

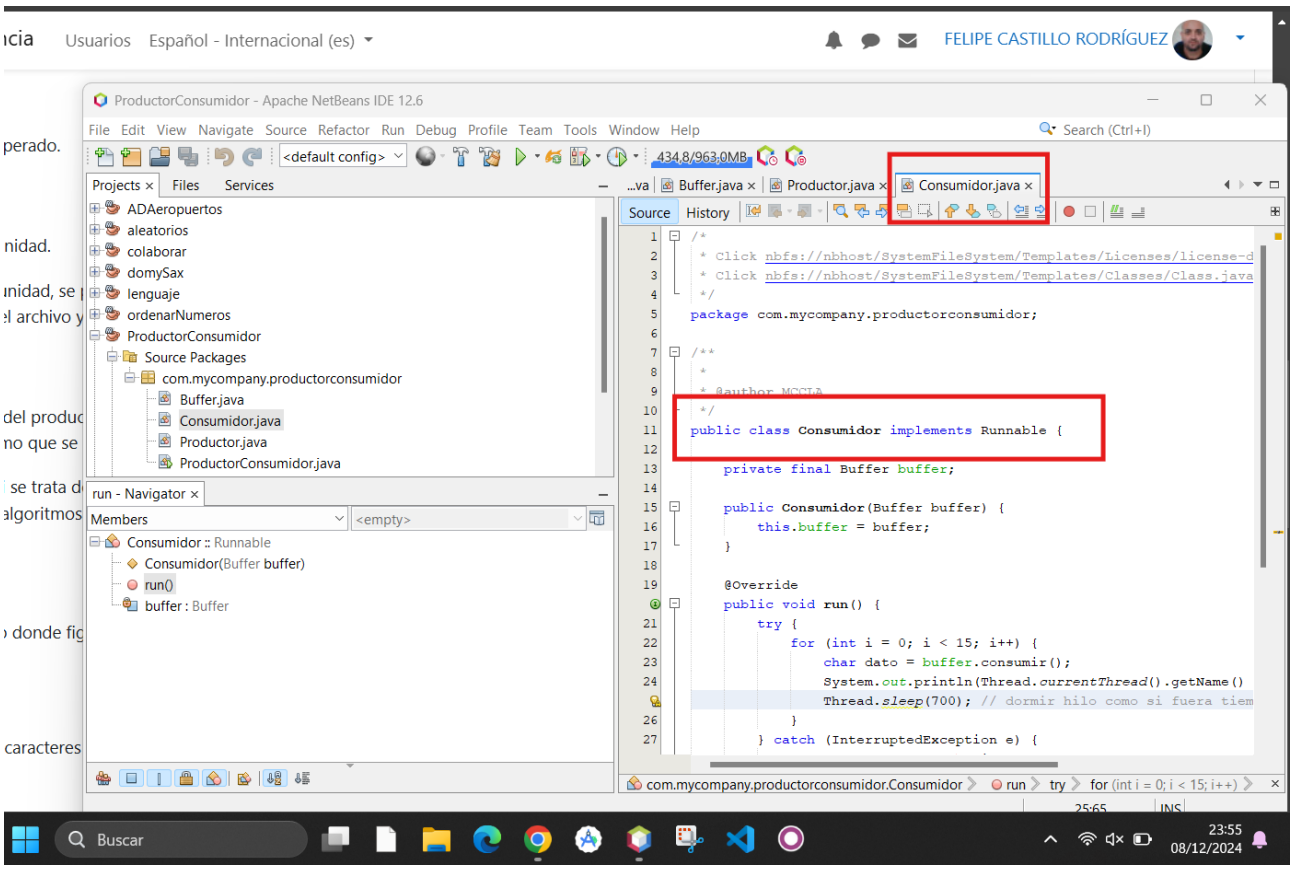
Hice 4 clases para tenerlo todo bien organizado , una principal , una clase Buffer, una productor y una consumidor, a continuación las imágenes y la ultima la ejecución del ejercicio con éxito.



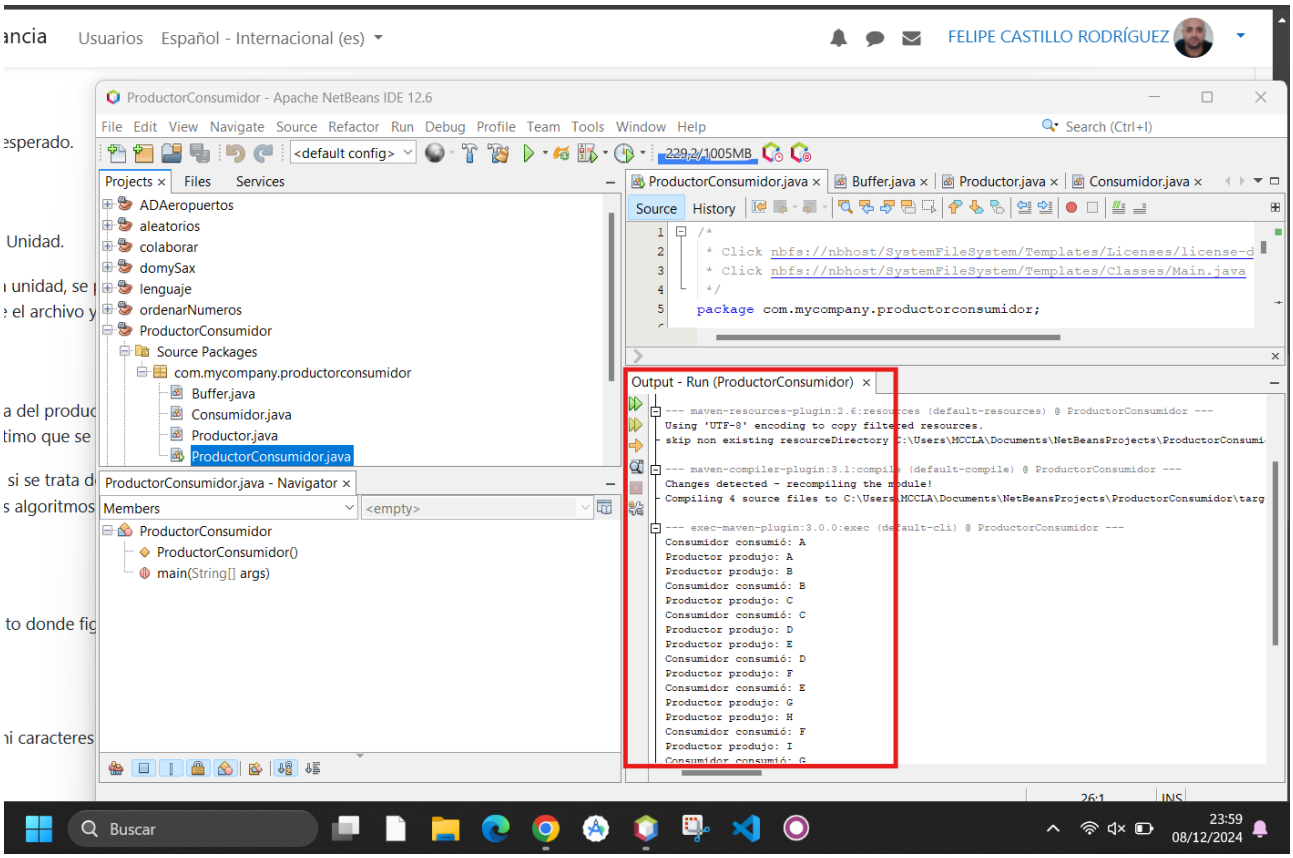




Tanto las clases productor como consumidor las he implementado con Runnable porque con herencia si hereda de otra clase no existe la herencia múltiple.

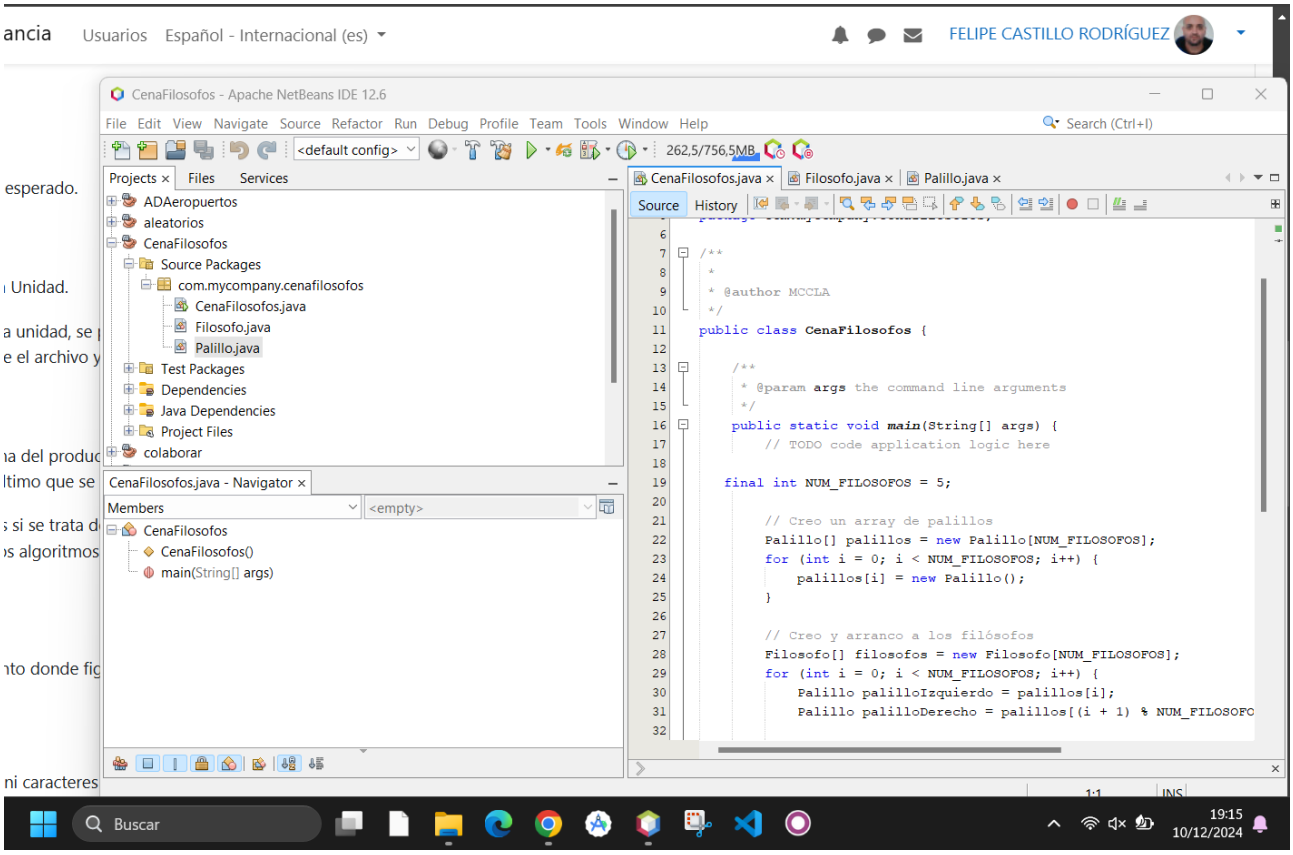


Aquí finalmente el resultado del ejercicio.

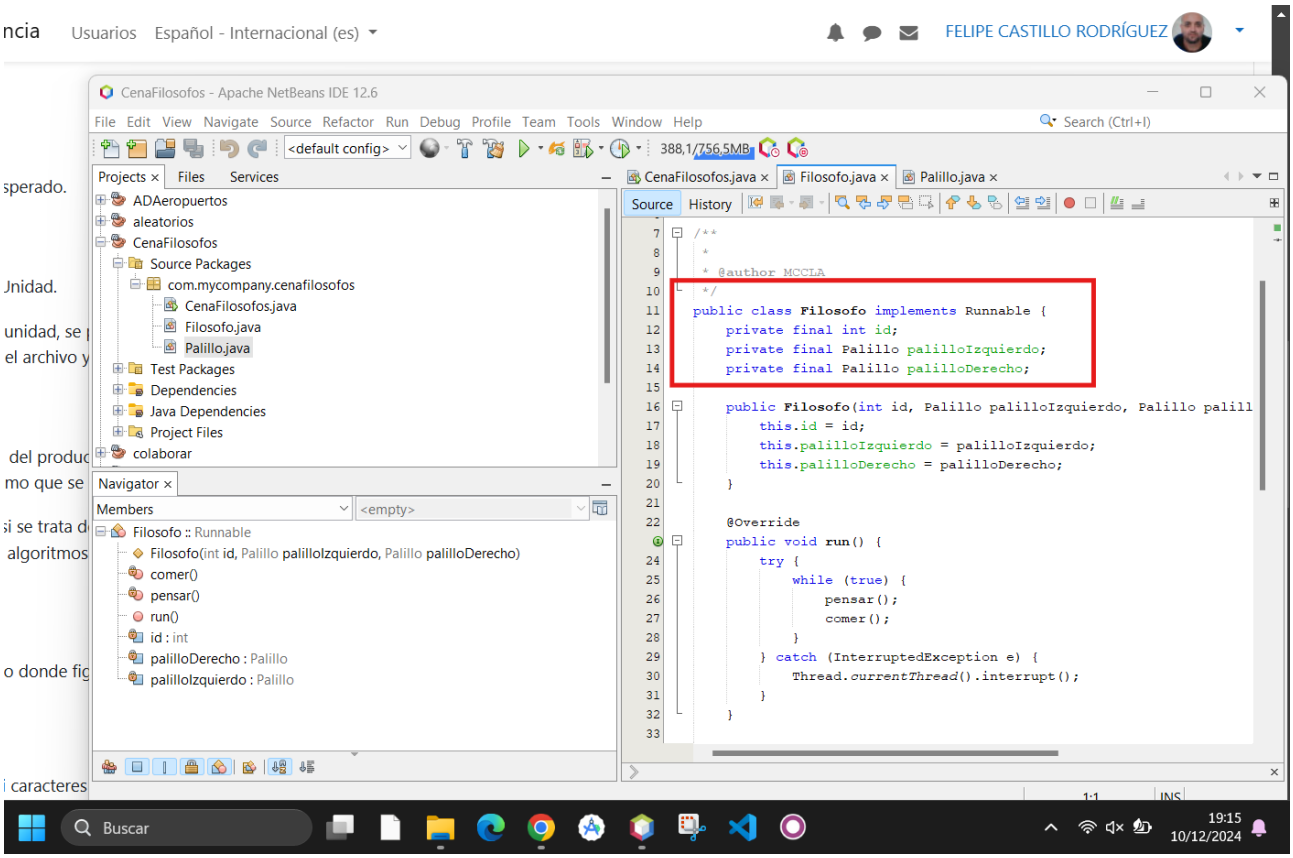


EJERCICIO 2

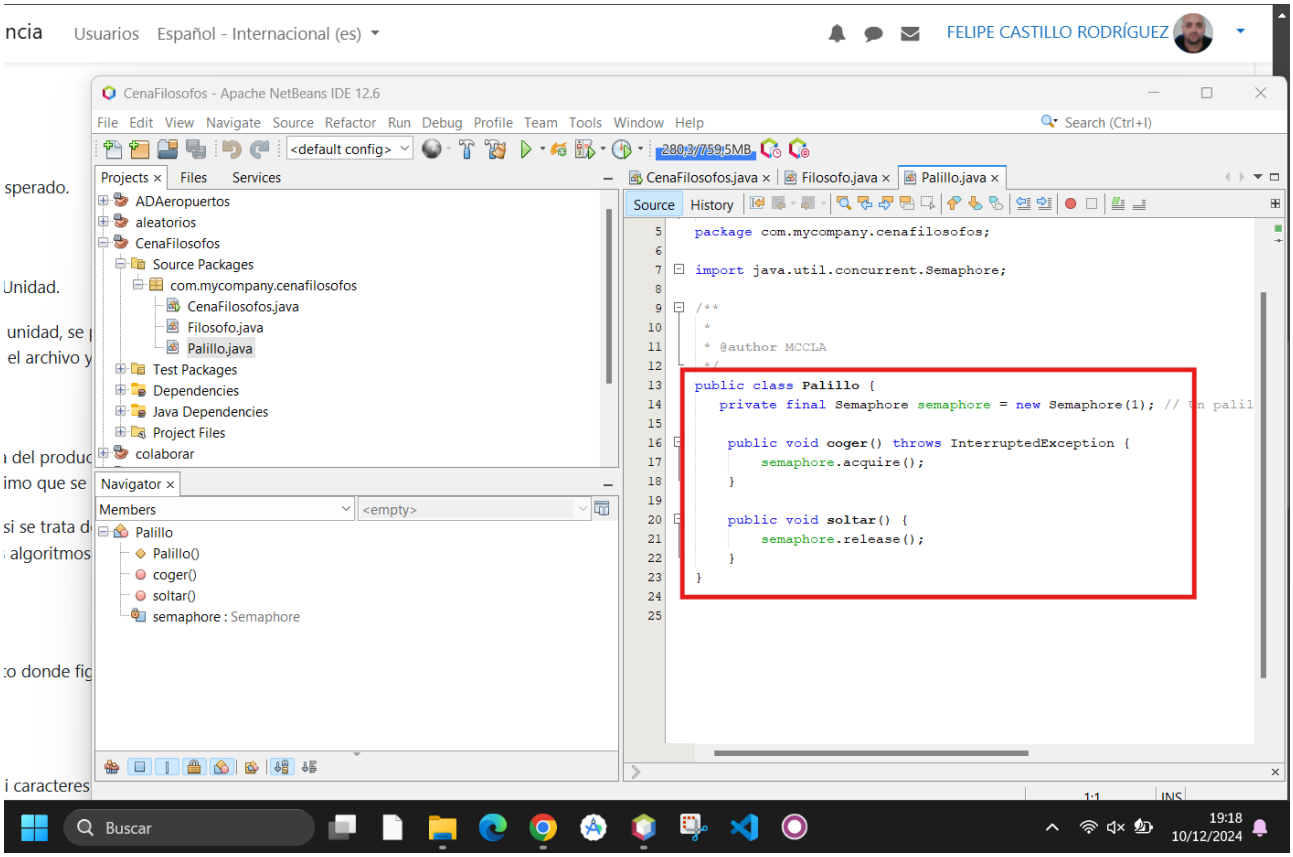
En este ejercicio una clase main con la que cree los objetos , asigne palillos a cada filosofo e inicio los hilos, también cree 2 arrays uno de filósofos y otro de palillos



clase filosofo , represento al filosofo que alterna entre pensar y comer dolo come si consigue los dos palillos e intento evitar el interbloqueo



clase palillo, representa el recurso compartido palillo, tiene un bloqueo que evita la exclusión mutua con dos métodos uno para coger el palillo si no esta ocupado y otro para soltarlo.



CONCLUSIÓN: lo mas difícil de esta practica para mi fue la lógica … tarde en entender como debía ir el ejercicio , también aprovecho para decir que el recurso compartido de el enunciado del ejercicio no funciona. Gracias por todo