Josue Jimenez De Lucio

**Java y mas**

***INTRODUCCIÓN***

Una de las formas más sencillas de comprender la concurrencia es imaginar a una persona la cual trabaja en múltiples tareas al mismo tiempo, y que rápidamente cambia de una tarea a otra. Por ejemplo, imaginemos a una persona la cual se encuentra programando, realizando cálculos en Excel y contestando correos electrónicos, todo esto al mismo tiempo. Dedica un par de segundos a cada tarea, y rápidamente, con un ágil cmd + shift cambia de tarea.

Concluimos que la persona trabaja de forma concurrente. Las tareas que realiza no necesariamente deben seguir un orden, quizás, después de contestar un correo regresa con los cálculos en Excel, le dedica un par de segundo, regresa a responder otro correo y finaliza con la codificación del programa, u, otro escenario pudiera ser que después finalizar ciertos cálculos, la persona continúa codificando un par de segundos para después responder un par de correos y regresar con los cálculos.

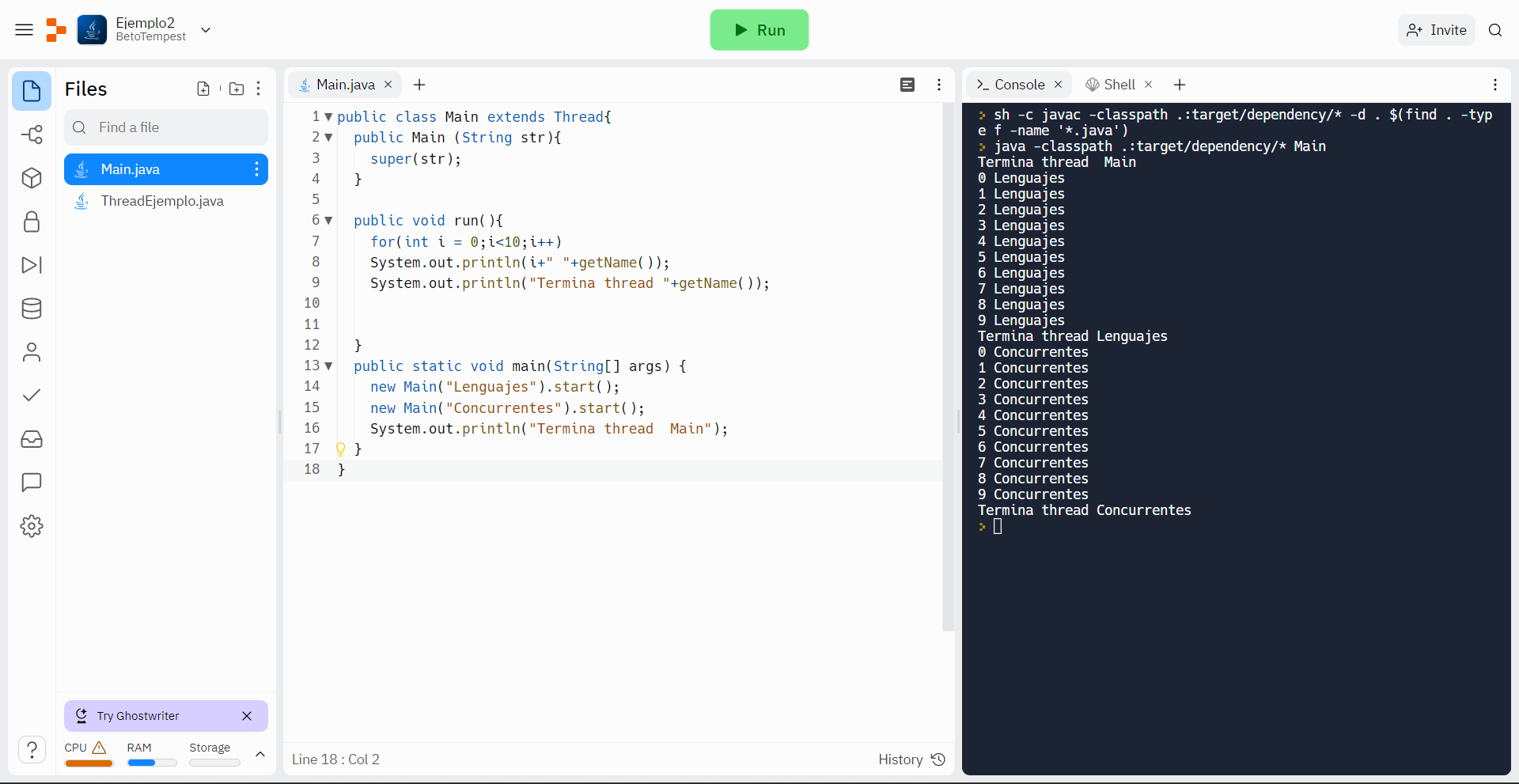
***DESARROLLO DE LA PRÁCTICA***

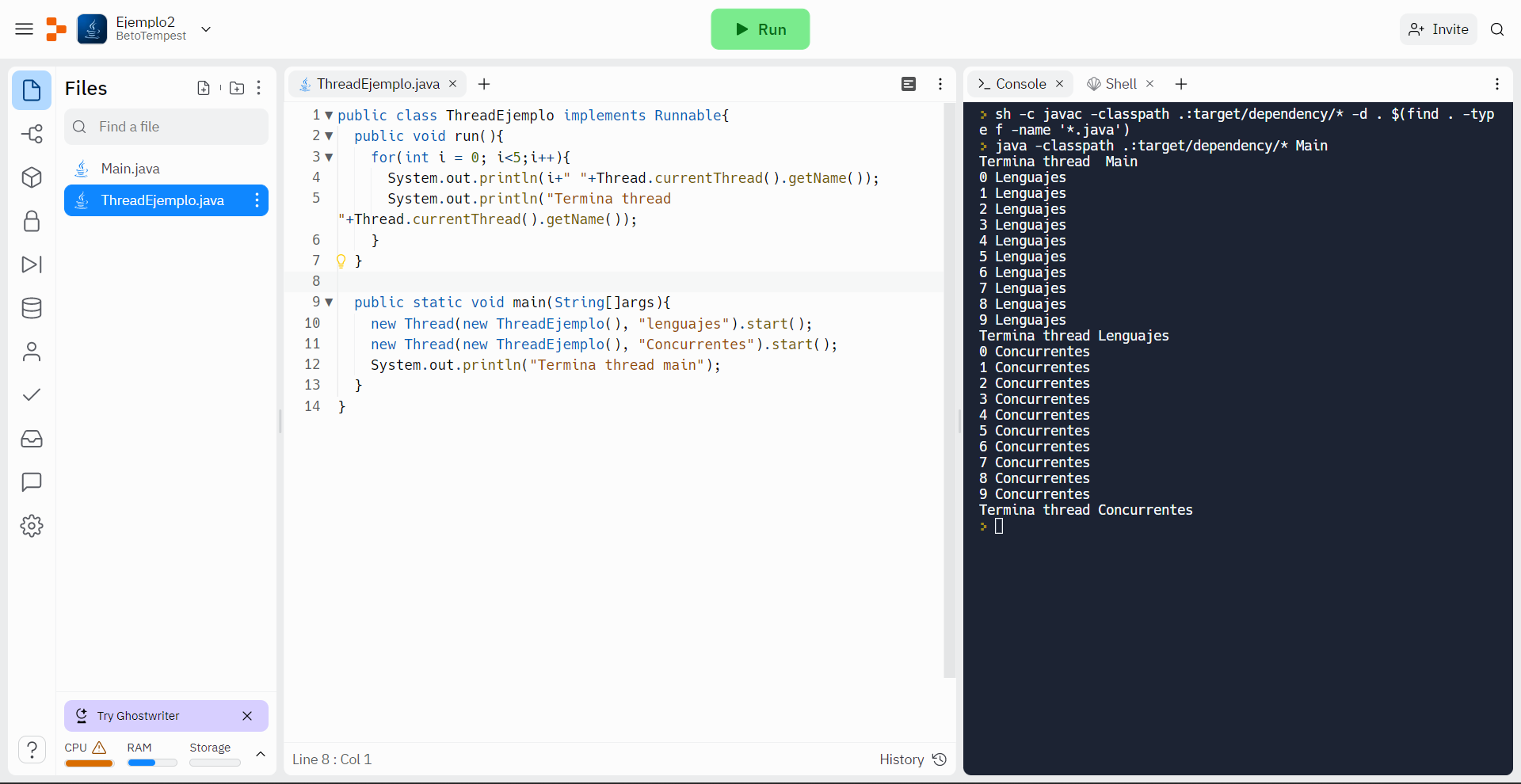
La Máquina Virtual Java (JVM) es un sistema multihilo. Es decir, es capaz de ejecutar varios hilos de ejecución simultáneamente. La JVM gestiona todos los detalles, asignación de tiempos de ejecución, prioridades, etc., de forma similar a cómo gestiona un Sistema Operativo múltiples procesos.

La diferencia básica entre un proceso de Sistema Operativo y un Thread Java es que los hilos corren dentro de la JVM, que es un proceso del Sistema Operativo y por tanto comparten todos los recursos, incluida la memoria y las variables y objetos allí definidos. A este tipo de procesos donde se comparte los recursos se les llama procesos ligeros (lightweight process).

Java da soporte al concepto de Thread desde el propio lenguaje, con algunas clases e interfaces definidas en el paquete java.lang y con métodos específicos para la manipulación

de Threads en la clase Object. Desde el punto de vista de las aplicaciones los hilos son útiles porque permiten que el flujo del programa sea divido en dos o más partes, cada una ocupándose de alguna tarea de forma independiente.





***DISCUSIÓN***

Además, Java cuenta con muchos frameworks, o código utilizable prefabricado que se suele emplear como plantillas para aplicaciones de escritorio y móviles. Los desarrolladores utilizan estos frameworks en lugar de crear código personalizado desde cero.

Por lo que aprender java es realmente importante para nuestro futuro profesional.

**CUESTIONARIO**

1***.- ¿Cuáles son las ventajas de utilizar la programación concurrente?*** En sistemas monoprocesador permite optimizar el uso de los recursos. Facilita la programación de tiempo real, ya que se pueden concebir como procesos cuya ejecución se planifica de acuerdo con la urgencia.

2***.- ¿Cuáles son las desventajas de utilizar la programación concurrente***? Este tipo de programación aparece también que su uso puede llegar a producir interbloqueos, que impiden la disponibilidad del sistema y requieren un gran esfuerzo para dar solución a los inconvenientes que se presenten como consecuencia de este.

3***.- ¿Son iguales la programación paralela y la concurrente? Explica ¿por qué***? No porque un programa es concurrente si puede soportar dos o más acciones en progreso. Un programa es paralelo si puede soportar dos o más acciones ejecutándose simultáneamente.

4***.- ¿Cuáles son las diferencias entre la programación secuencial y la programación***

***concurrente?*** La programación concurrente no es más que la forma en la cual podemos resolver ciertas problemáticas de forma concurrente, es decir, ejecutando múltiples tareas a la misma vez y no de forma secuencial.