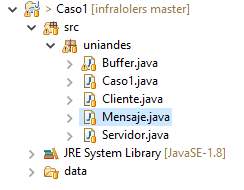
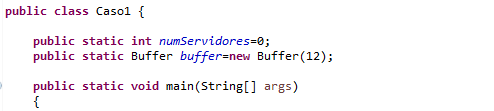
**Diseño del programa**

**Contexto de las clases:**

Para el desarrollo del caso 1 se pensó principalmente en el uso de 4 clases para modelar el problema propuesto. La primera clase es una a la que llamamos caso1, la cual se encarga de ejecutar un método *main* y crear unos objetos estáticos que usara para simular al buffer y al numero de servidores que se va a usar. Luego, están la clase Servidor y la clase Cliente las cuales están pensadas para simular a estos actores; ambas extienden de la clase *thread* y será usados para procesar los procesos del problema. En último lugar, están las clases Mensaje que su función será la de contener el mensaje y guardar el contenido de este, y la clase buffer que será la que va a coordinar la comunicación de mensajes concurrentemente entre el cliente y el servidor. En la clase Buffer estarán los métodos de colocar y responder, de modo que sean usados por los clientes y los servidores, para manejar la transmisión de mensajes.

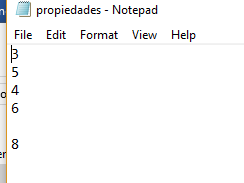


Estructura de la clase Caso1:

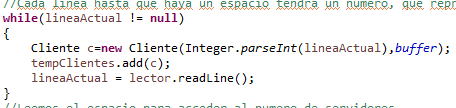


**Funcionamiento:**

El programa comienza cuando se ejecuta el método *main* de la clase Caso 1, en el cual se hace la lectura de un archivo de texto plano donde se encuentran números que representan los mensajes.



Luego con estos mensajes se van creando a los clientes para que se ejecute el programa, el número representa cuantas consultas va a hacer el cliente.



Se deja un espacio en blanco antes del último número en el archivo de texto para que represente el numero de servidores a crear. Luego de que se lee el ultimo numero del archivo de propiedades se establece el numero de servidores a crear. Con toda la información ya recolectada se crean los clientes y servidores para simular el problema y se inicializan.

