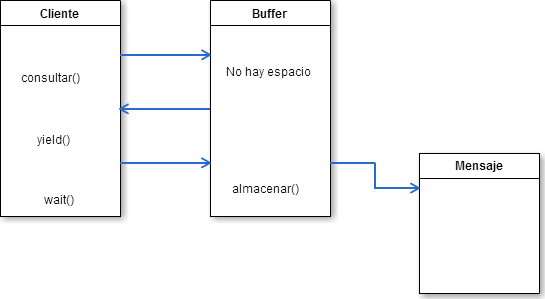
Caso 1 infraestructura computacional

**Interacción Cliente Buffer**

Ya que el cliente no debe comunicarse directamente con el servidor el Buffer hace de intermediario.

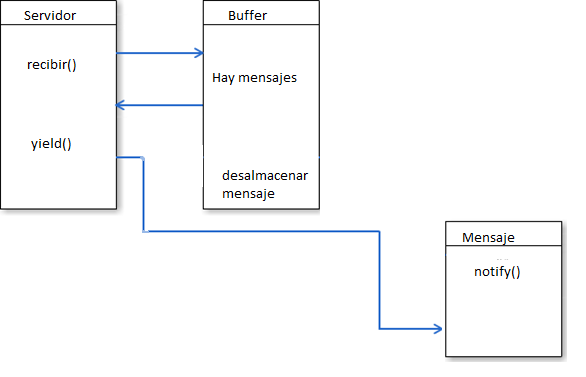
Primero al iniciar el Thread cliente este trata de hacer sus consultas. El método hacer consulta no es sincronizado puesto que el número de consultas es de cada Thread (Cliente). Cada vez que el cliente trata de hacer una consulta el buffer le dice si pudo procesarla o no. Si el buffer no la puede procesar el cliente sede el procesador en espera activa. Si el Buffer si puede procesarla se hace espera pasiva (Hasta que el servidor recoja el mensaje, lo cual a su vez despierta al cliente). El procedimiento anterior se repite hasta que el cliente no tenga más consultas. La reducción del número de consultas de un cliente no se sincroniza puesto que este parámetro solo lo modifica el mismo cliente. Cuando no le quedan más consultas el cliente le avisa al Buffer que se retira. Este aviso si es sincronizado puesto que el número de clientes es del Buffer y puede llegar a ser modificado por varios clientes al tiempo.

La interacción del Cliente con el Buffer y el mensaje



**Interacción Servidor Buffer**

El servidor luego, sincrónicamente retira el mensaje y despierta al cliente, si no hay mensajes el servidor pasivamente espera a que haya para poder retirarlos, una vez ya no hayan mensajes ni clientes el servidor para.



Para saber el resultado se hizo el método aumentar estático y sincronizado. Esto ya que varios servidores modifican la misma variable. Además cuando ya no quedan consultas (no hay más clientes que atender y el buffer esta vacío) se solicita el aumentar (0). Este retorna el valor modificado. Con esto se logra esta última instrucción sea la última de la cola en el aumento de la variable de interés. Así se tiene la variable con su último estado.