INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

Materia: *Aplicaciones para Comunicaciones en Red*

Grupo: *3CM17*

Nombre del Trabajo:

Reporte de la Práctica 6. Sockets no bloqueantes

Alumnos:

Peña Atanasio Alberto

Martínez Coronel Brayan Yosafat

Profesor: *Moreno Cervantes Axel*

Introducción

Buffers no bloqueantes

Un buffer tiene mejor desempeño que un flujo (stream), ya que tiene un tamaño finito (capacidad), así como un estado (interno) que permite llevar el registro y control de cuantos datos se han puesto o leído de él.

Sockets bloqueantes

Las entradas y salidas son por naturaleza bloqueantes (no permiten realizar nada más hasta que terminen). En el caso de los sockets si no hay nada que procesar la instrucción se queda dormida hasta que ocurra un evento que permita terminar la operación.

Si realizamos operaciones de entrada (read, recv, recvfrom, etc.) sobre el socket y no hay datos disponibles es proceso entrara al estado de dormido hasta que haya datos para leer

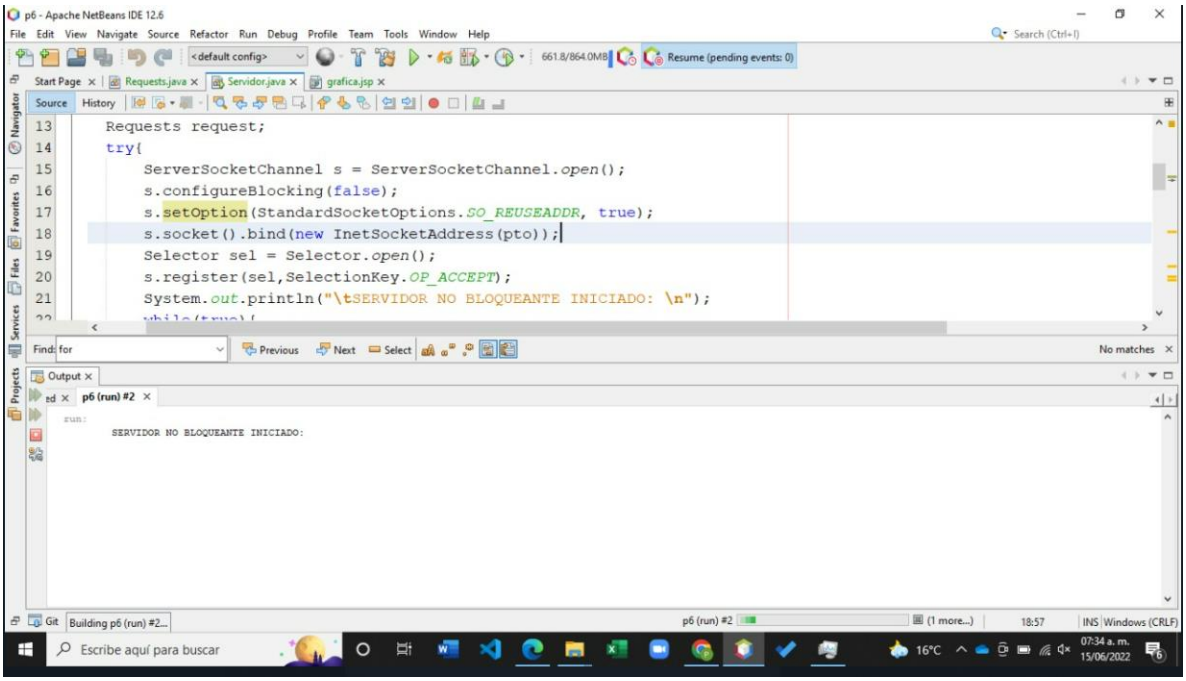
Si realizamos operaciones de salida (write, send, sendto, etc.) sobre el socket, el kernel copia los datos del buffer de la aplicación en el buffer de envío de datos, si no hay espacio en este último el proceso se bloqueara hasta tener suficiente espacio.

En algunas ocasiones es preferible que no exista el bloqueo mencionado ya quepermite realizar otras tareas si no hay datos que manejar. Hay dos maneras básicas de manejo:

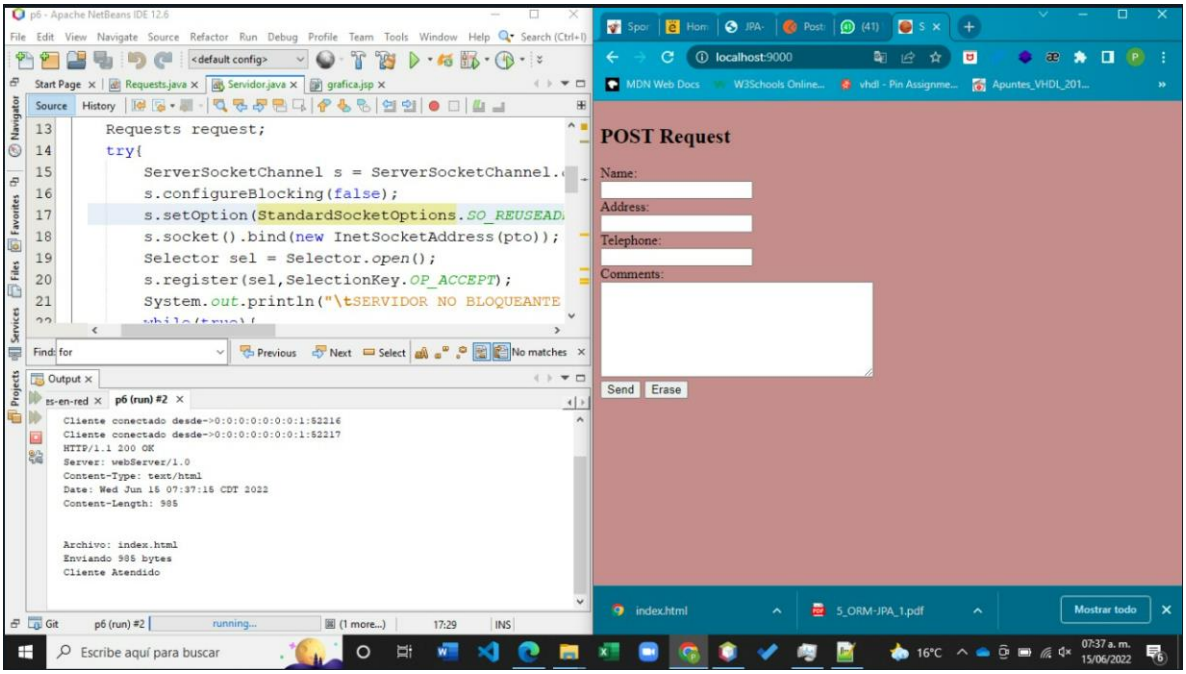
- Polling: Consiste en una operación de consulta constante, eso lo vuelve síncrono, ya que solo se procesa en un momento determinado.
- Asíncrono: En este caso, hay que esperar a que ocurra un evento de entrada o salida y actuar en consecuencia.

Desarrollo

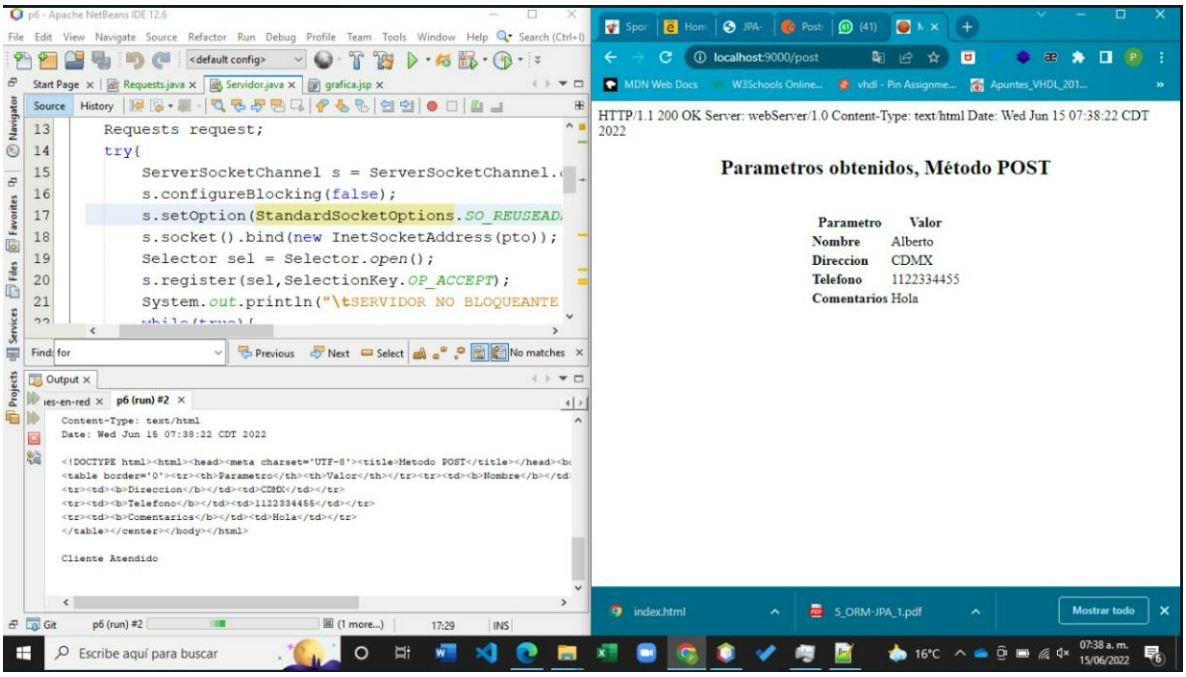
Se tomaron las siguientes capturas durante el desarrollo de la aplicación, asi como su ejecución.



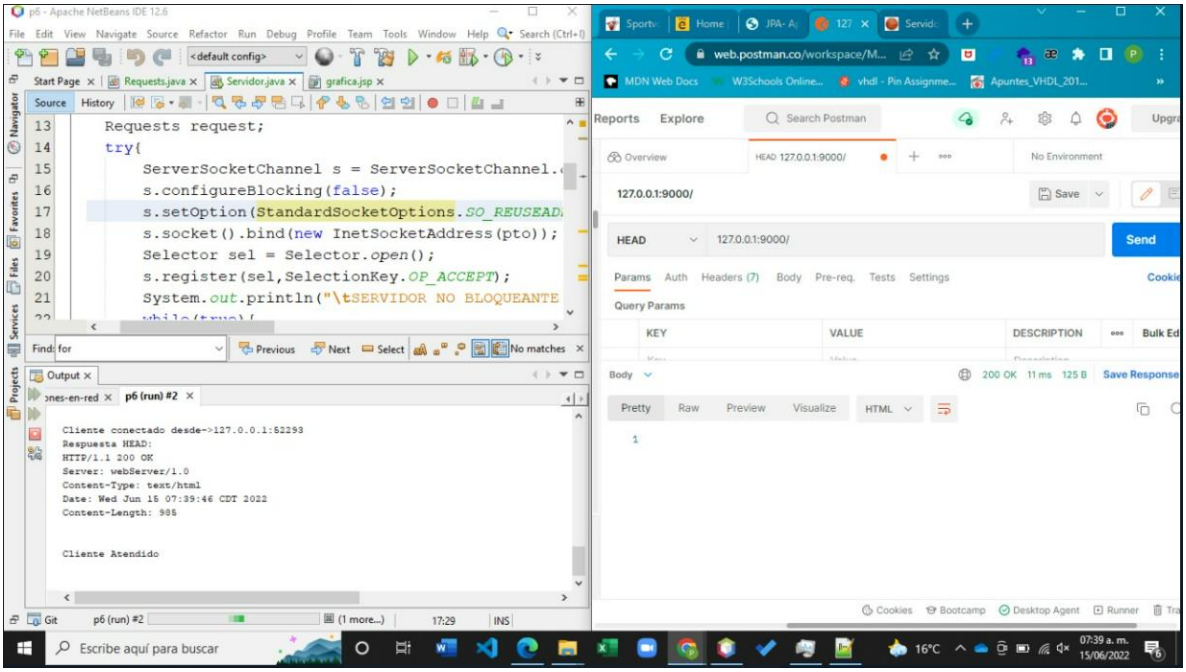
Comenzamos a ejecutar el servidor no bloqueante



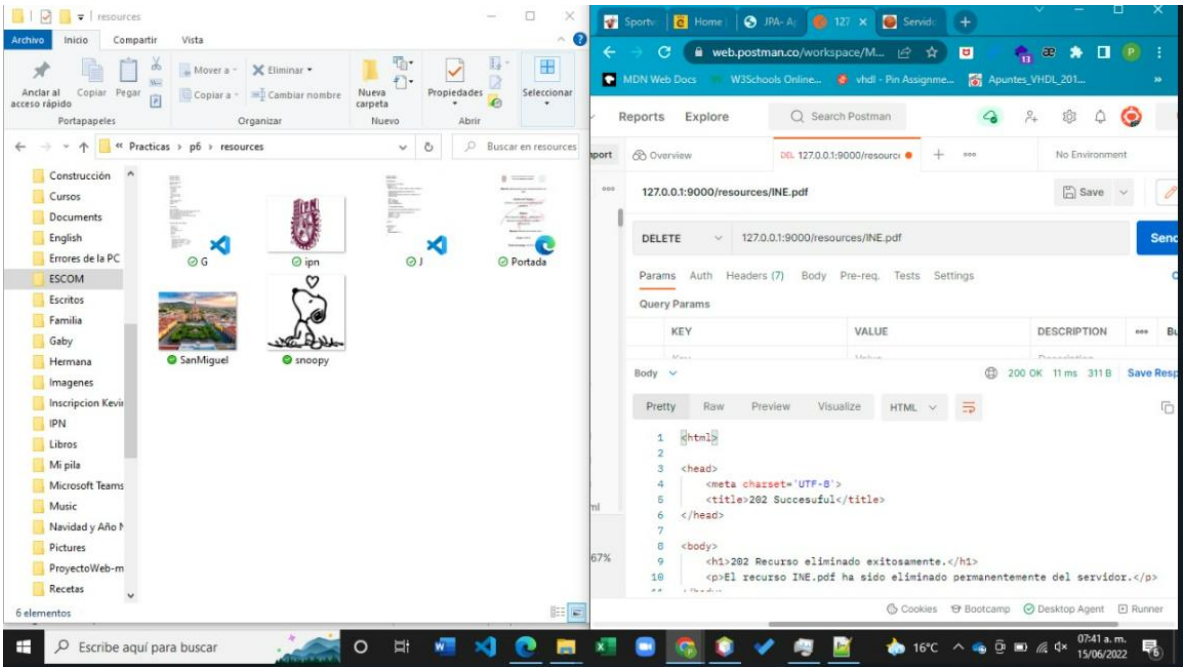
Comenzamos a correr un cliente



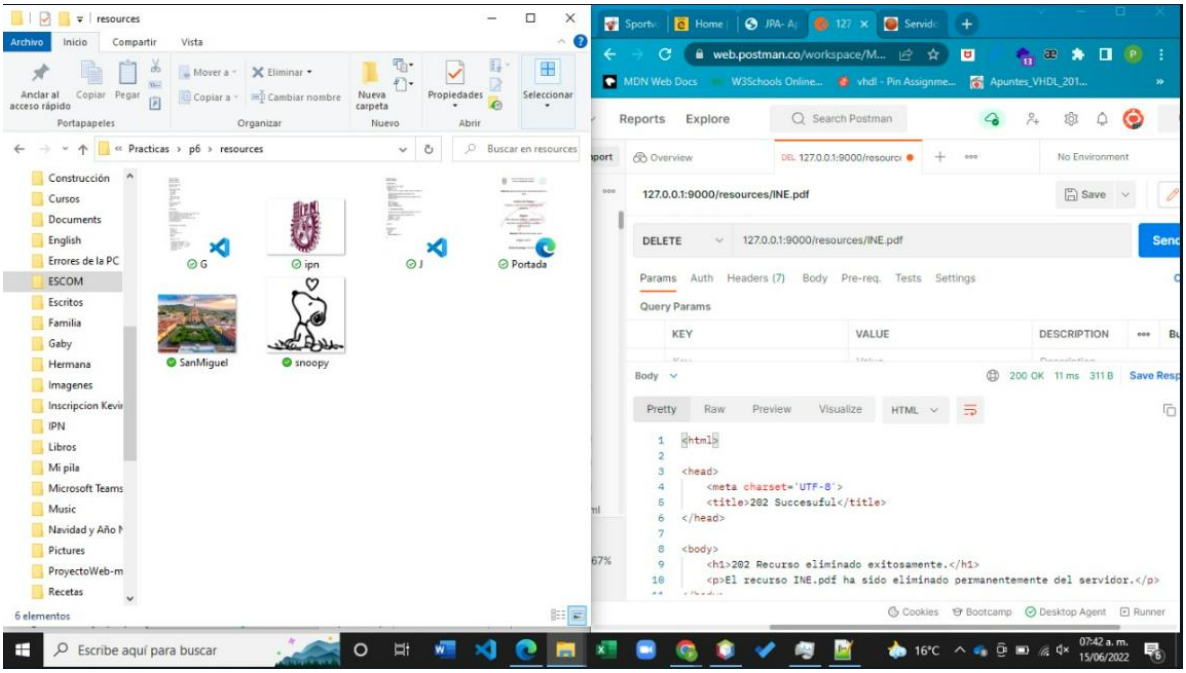
Enviamos el formulario



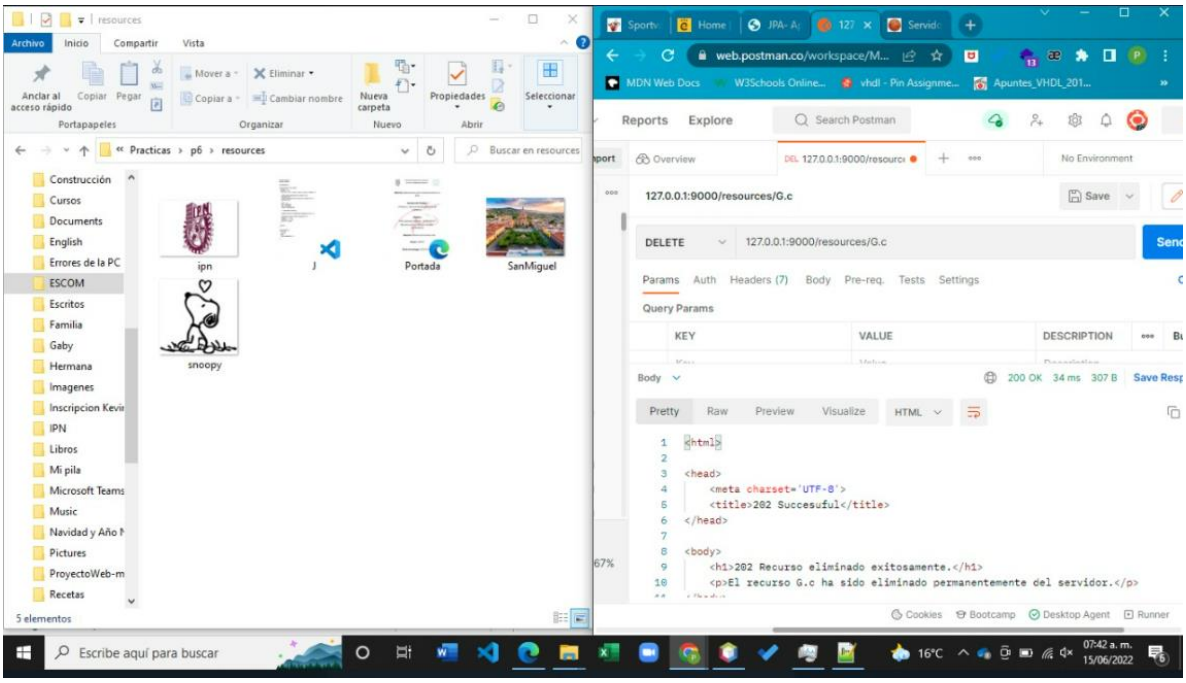
Abrimos PostMan



Primera petición DELETE



Segunda petición DELETE



Tercera petición DELETE

Conclusiones

Pensamos que el uso de los sockets no bloqueantes es de gran ayuda ya que, en caso de requerir realizar diversas actividades con los datos que se comparten entre los sockets, es más fácil realizarlo de esta forma ya que no existe un bloqueo que detenga el flujo de la comunicación permitiendo realizar operaciones de lectura y escritura de datos más constantes, aunque se debe ser cuidadoso con el tratado de esta para no tener problemas con la información enviada.