



Instituto Politécnico Nacional
Escuela Superior de Cómputo ★★★

UNIDAD DE APRENDIZAJE

“DESARROLLO DE SISTEMAS DISTRIBUIDOS”

**“TAREA 1. SISTEMA DISTRIBUIDO QUE VERIFICA SI UN
NÚMERO ES PRIMO”**



ALUMNO:

MORENO HURTADO ALEJANDRO

GRUPO 4CV13

PROFESOR:

PINEDA GUERRERO CARLOS

Desarrollo de la practica

Por medio de 4 servidores multi-thread verificar si un numero ingresado por una petición de navegador es primo o no primo.

La petición será procesada por el servidor B enviando el número y los rangos a evaluar por medio de hilos para los servidores A. El servidor A una vez encontrando un numero que divide al numero ingresado tendrá que enviar al servidor B la palabra *DIVIDE* enviando al navegador el resultado de *no es primo* , en caso contrario enviara *es primo*.

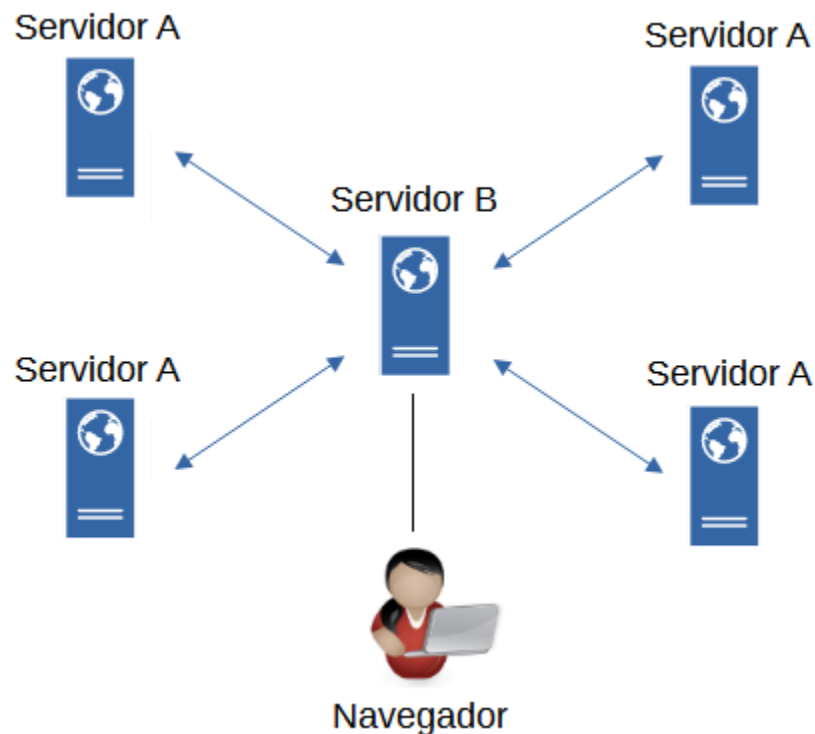
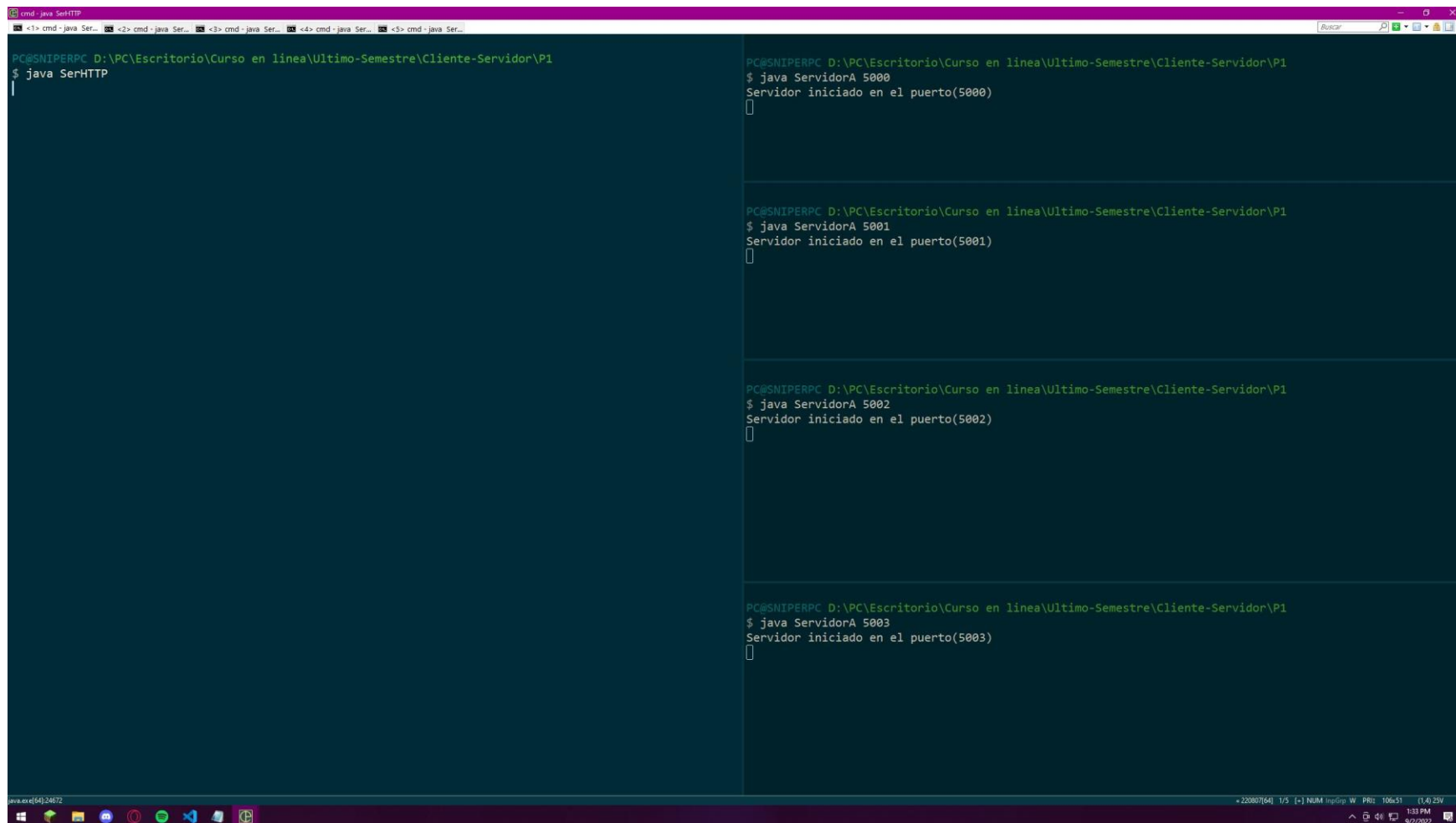


Ilustración 1: Diagrama de referencia

A continuación se mostrara el funcionamiento de la practica corriendo los servidores A en distintos puestos , el navegador enviando la solicitud al servidor B y la respuesta en navegador del servidor B.



```
PC@SNIPERPC D:\PC\Escritorio\Curso en línea\Ultimo-Semestre\Cliente-Servidor\P1
$ java SerHTTP

PC@SNIPERPC D:\PC\Escritorio\Curso en línea\Ultimo-Semestre\Cliente-Servidor\P1
$ java ServidorA 5000
Servidor iniciado en el puerto(5000)

PC@SNIPERPC D:\PC\Escritorio\Curso en línea\Ultimo-Semestre\Cliente-Servidor\P1
$ java ServidorA 5001
Servidor iniciado en el puerto(5001)

PC@SNIPERPC D:\PC\Escritorio\Curso en línea\Ultimo-Semestre\Cliente-Servidor\P1
$ java ServidorA 5002
Servidor iniciado en el puerto(5002)

PC@SNIPERPC D:\PC\Escritorio\Curso en línea\Ultimo-Semestre\Cliente-Servidor\P1
$ java ServidorA 5003
Servidor iniciado en el puerto(5003)
```

Ilustración 2: ejecución de todos los servidores

Empezamos con la ejecución de todos los servidores a la espera de los datos enviados por el navegador , en este caso los servidores A estan en los puertos 5000,5001,5002,5003 y el servidor B esta a la escucha en el puerto 8081.

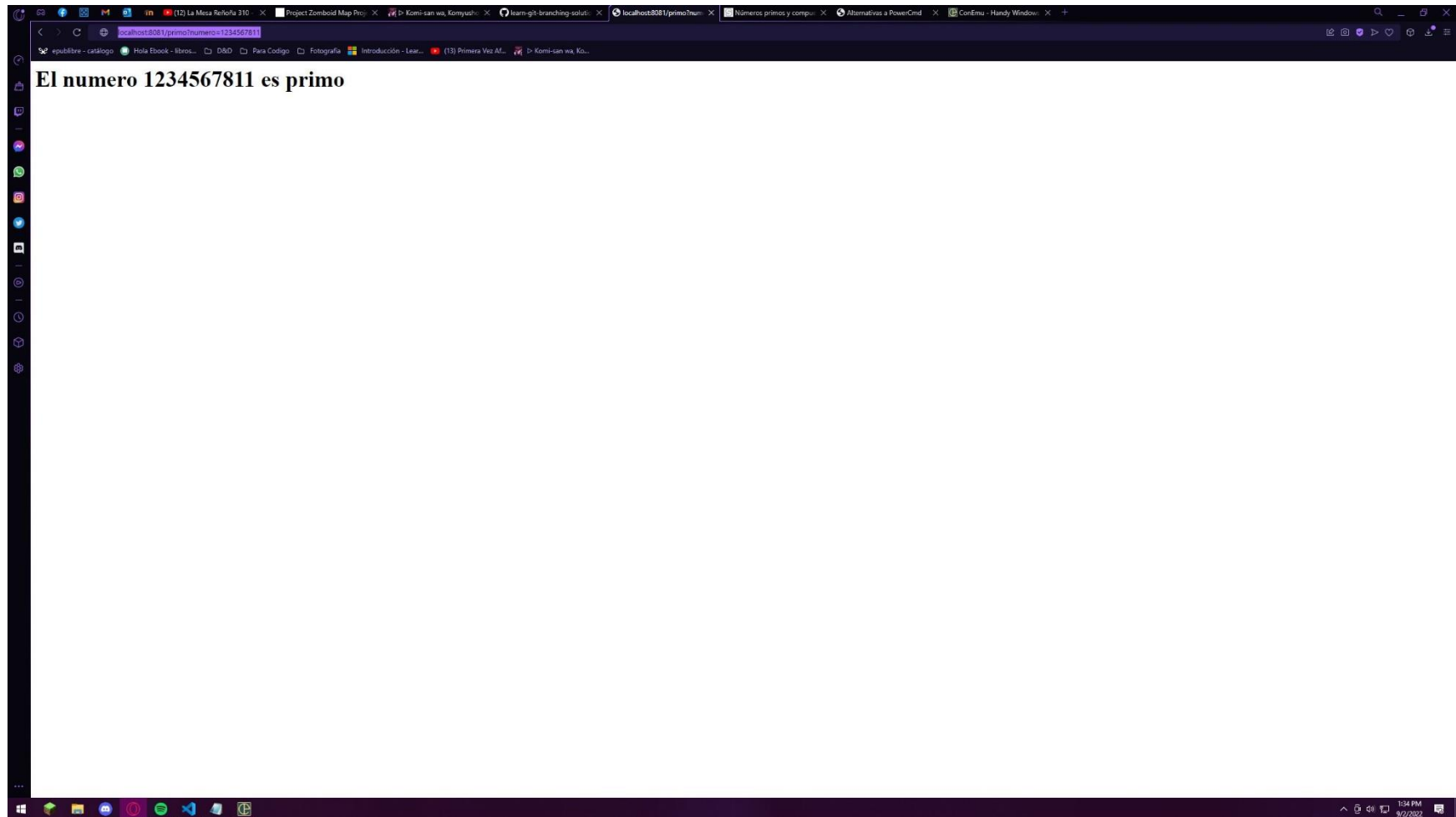
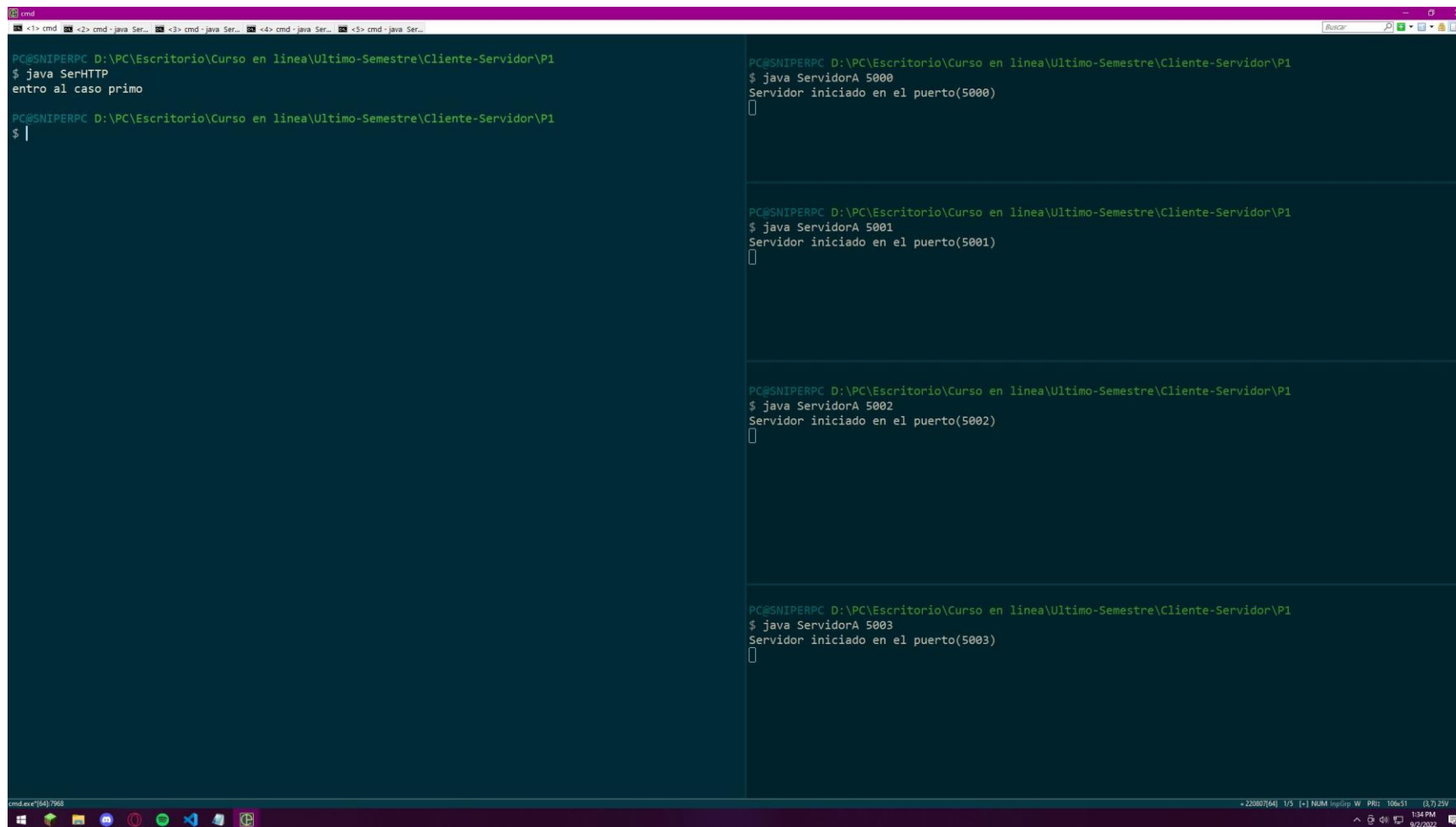


Ilustración 3: Envío de datos por el navegador y respuesta del servidor B

Envío al servidor B <http://localhost:8081/primo?numero=1234567811> el cual procesa el numero 1234567811 para poder verificar si el numero es o no un numero primo a través de los servidores A los cuales regresan al navegador la respuesta **El numero 1234567811 es primo** después de terminar las comprobaciones en los servidores A.



```
PC@SNIPERPC D:\PC\Escritorio\Curso en línea\Ultimo-Semestre\Cliente-Servidor\P1
$ java SerHTTP
entro al caso primo

PC@SNIPERPC D:\PC\Escritorio\Curso en línea\Ultimo-Semestre\Cliente-Servidor\P1
$ |

PC@SNIPERPC D:\PC\Escritorio\Curso en línea\Ultimo-Semestre\Cliente-Servidor\P1
$ java ServidorA 5000
Servidor iniciado en el puerto(5000)

PC@SNIPERPC D:\PC\Escritorio\Curso en línea\Ultimo-Semestre\Cliente-Servidor\P1
$ java ServidorA 5001
Servidor iniciado en el puerto(5001)

PC@SNIPERPC D:\PC\Escritorio\Curso en línea\Ultimo-Semestre\Cliente-Servidor\P1
$ java ServidorA 5002
Servidor iniciado en el puerto(5002)

PC@SNIPERPC D:\PC\Escritorio\Curso en línea\Ultimo-Semestre\Cliente-Servidor\P1
$ java ServidorA 5003
Servidor iniciado en el puerto(5003)
```

Ilustración 4 : Servidores después de la comprobación

Los servidores A quedan nuevamente en escucha para la nueva petición mientras en el servidor B podemos ver que entro en el caso que un numero es primo y se detiene para su próxima ejecución. Ejecutamos nuevamente el servidor B e ingresamos el siguiente numero.

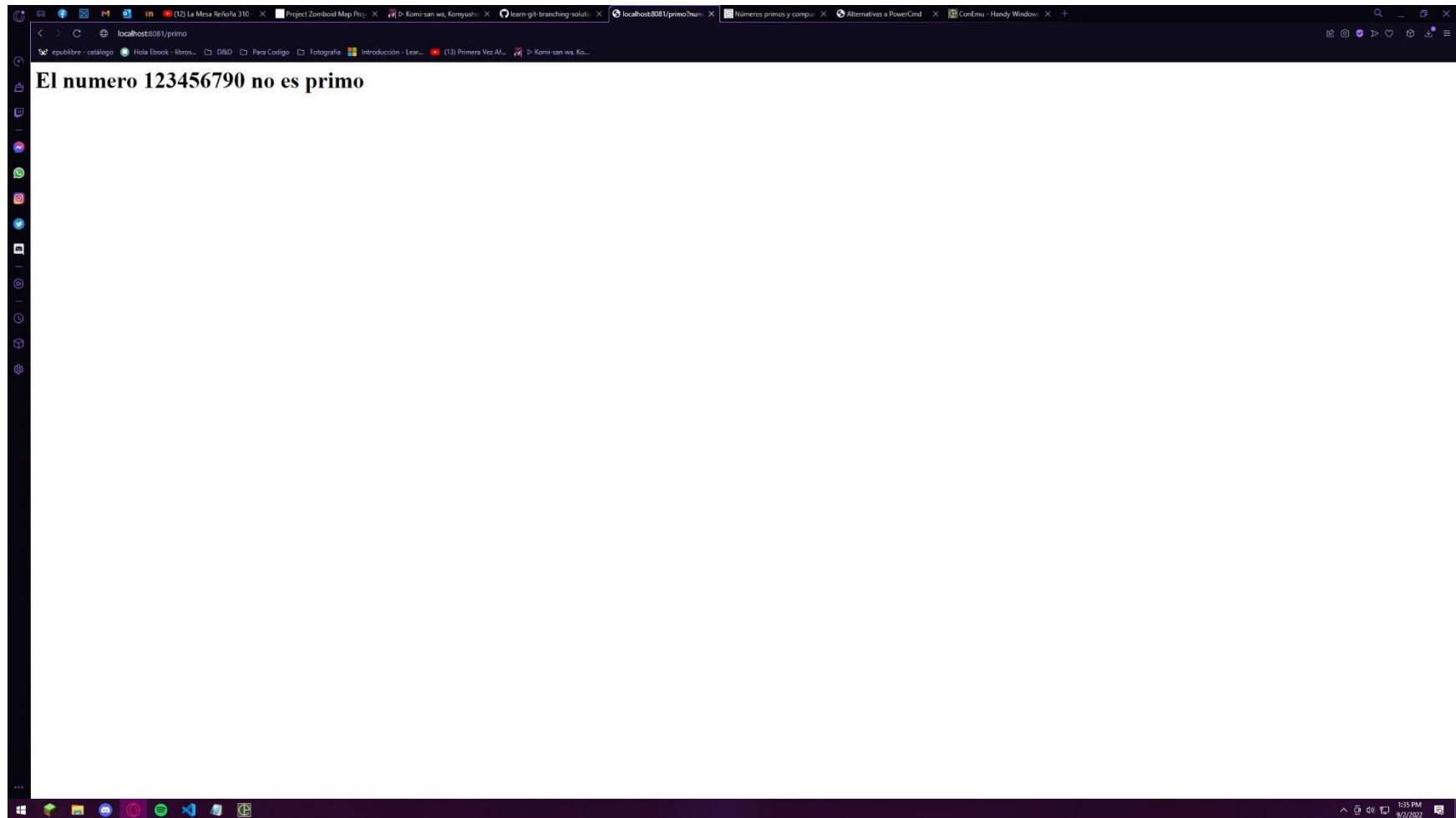


Ilustración 5 : Comprobación de un número no primo

Envío al servidor B <http://localhost:8081/primo?numero=123456790> el cual procesa el numero 123456790 para poder verificar si el número es o no un numero primo a través de los servidores A los cuales regresan al navegador la respuesta **El numero 123456790 no es primo** después de terminar las comprobaciones en los servidores A.

```
PC@SNIPERPC D:\PC\Escritorio\Curso en línea\Ultimo-Semestre\Cliente-Servidor\P1
$ java SerHTTP
entro al caso primo

PC@SNIPERPC D:\PC\Escritorio\Curso en línea\Ultimo-Semestre\Cliente-Servidor\P1
$ java SerHTTP

PC@SNIPERPC D:\PC\Escritorio\Curso en línea\Ultimo-Semestre\Cliente-Servidor\P1
$ |

PC@SNIPERPC D:\PC\Escritorio\Curso en línea\Ultimo-Semestre\Cliente-Servidor\P1
$ java ServidorA 5000
Servidor iniciado en el puerto(5000)
[]

PC@SNIPERPC D:\PC\Escritorio\Curso en línea\Ultimo-Semestre\Cliente-Servidor\P1
$ java ServidorA 5001
Servidor iniciado en el puerto(5001)
[]

PC@SNIPERPC D:\PC\Escritorio\Curso en línea\Ultimo-Semestre\Cliente-Servidor\P1
$ java ServidorA 5002
Servidor iniciado en el puerto(5002)
[]

PC@SNIPERPC D:\PC\Escritorio\Curso en línea\Ultimo-Semestre\Cliente-Servidor\P1
$ java ServidorA 5003
Servidor iniciado en el puerto(5003)
[]
```

Ilustración 6 : Servidores después de la segunda comprobación

Una vez enviada la respuesta al navegador el servidor B se detiene y los servidores A entran en espera de nuevos datos por parte del servidor B.

Conclusión

Al trabajar con hilos y procesamiento de manera distribuida debemos tener muy en mente la sincronización de las variables para asegurar que se este escribiendo y leyendo los datos que corresponden. Al igual que una barrera para los hilos de procesamiento y asegurar que todos empiecen el procesamiento al mismo tiempo y terminen al mismo tiempo dependiendo el caso.

Otra cosa de tener en cuenta en esta practica es la manera de implementar banderas y asegurar que la respuesta de los servidores entren en los casos que corresponden.