

## Reporte parte 1

### saturación del servidor

Como primera parte del ejercicio decidimos poner a prueba el servidor del proyecto *hello-word* haciendo varias solicitudes desde 3 equipos diferentes.

#### Diseño de la prueba:

Equipos usados:

Equipo	Función
Hgrid7	Servidor
Hgrid1	Cliente
Hgrid2	Cliente
Hgrid3	Cliente

La prueba se llevo acabo de la siguiente manera; se tenían pensadas 3 entradas para solicitarle al servidor en la función de Fibonacci, las cuales eran 40, 50 y 80. La idea es que por cada entrada se ejecutara el cliente y le realizara la solicitud al servidor, después se guardaba las salidas por cada petición en el archivo finalLog.txt donde podríamos evidenciar las salidas obtenidas y la eventual respuesta del servidor.

Todo lo anterior apoyado por medio del script de bash *run\_clients.sh* el se encarga de todo lo anterior y también de enviar los archivos jar del cliente con cualquier modificación realizada.

#### Resultado de la prueba y conclusiones:

Después de ejecutar la prueba observamos lo siguiente en el archivo finalLog.txt y en la consola del servidor.

Johan Stiven Ricardo - A00370269  
Camilo Campaz Jiménez - A00370258

```
reports > finalLog.txt
1   Es el turno de hgrid1.icesi.edu.co
2   hgrid1.icesi.edu.co esta probando con : 50 el server respondió: 12586269025
3
4   com.zeroc.Ice.ConnectionTimeoutException
5   com.zeroc.Ice.ConnectionTimeoutException
6
7   Es el turno de hgrid2.icesi.edu.co
8   hgrid2.icesi.edu.co esta probando con : 40 el server respondió: 102334155
9
10  com.zeroc.Ice.ConnectionTimeoutException
11  com.zeroc.Ice.ConnectionTimeoutException
12
```

```
[swarch@hgrid7 callback-server-deploy]$ java -jar server.jar
Cliente: hgrid2.icesi.edu.co numero: 1,1,2,3,5,8,13,21,34,55,89,144,233,377,610,987,1597,2584,4181,6765,10946,17711,28657,46368,75025,121393,196418,
317811,514229,832040,1346269,2178309,3524578,5702887,9227465,14930352,24157817,39088169,63245986,102334155,
Cliente: hgrid1.icesi.edu.co numero: 1,1,2,3,5,8,13,21,34,55,89,144,233,377,610,987,1597,2584,4181,6765,10946,17711,28657,46368,75025,121393,196418,
317811,514229,832040,1346269,2178309,3524578,5702887,9227465,14930352,24157817,39088169,63245986,102334155,165580141,267914296,433494437,701408733,1
134903170,1836311903,2971215073,4807526976,7778742049,12586269025,
```

Es por todo lo anterior que, después de ver los resultados obtenidos, llegamos a la conclusión de que al momento de recibir múltiples solicitudes con números muy grandes se lanza la excepción *com.zeroc.Ice.ConnectionTimeoutException*, pues el servidor se encuentra ocupado con un calculo complejo, lo cual hace que los otros procesos, provenientes de los otros clientes, se queden esperando la respuesta a la solicitud y eventualmente mueran de inanición lanzando la excepción y demostrándonos que no existe concurrencia en esta primera implementación del proyecto