



Universidad  
Carlos III de Madrid

## TECNOLOGÍAS *DEL* SECTOR FINANCIERO

### Práctica 6: In-memory DataGrid

Andoni Alcelay

MÁSTER UNIVERSITARIO EN TECNOLOGÍAS DEL SECTOR FINANCIERO:  
FINTECH

## Introducción

En este documento se explicarán los resultados obtenidos en la práctica 6 realizada con la clase Hazelcast, la cual permite meter datos in-grid. La práctica está dividida en tres apartados, la primera, la creación de 3 instancias donde comparten una misma para enviar y recibir el mismo dato, la segunda, con 3 tipos de datos diferentes y además con concurrencia, y la tercera con un cliente que se conecta al servidor.

## Práctica 1: Creación de un cluster Hazelcast

Para la primera practica se crea un cluster con una sola clase que escribe tres veces el mismo dato. En la captura de pantalla puede observarse cómo hay 3 miembros en nuestro cluster, uno por cada ejecución del programa realizado, esto significa que la conexión ha sido satisfactoria.

```
nov 22, 2018 4:37:48 PM com.hazelcast.internal.cluster.ClusterService
INFORMACIÓN: [192.168.1.112]:5703 [dev] [3.7.2]

Members [3] {
  Member [192.168.1.112]:5701 - 00282faa-baab-436d-a8b2-6fc8c793ee94
  Member [192.168.1.112]:5702 - 71b5290c-4c9f-4b3d-9c42-acc3933920dc
  Member [192.168.1.112]:5703 - 9aec8417-8f7a-47a9-9368-b0838e3c27f6 this
}
```

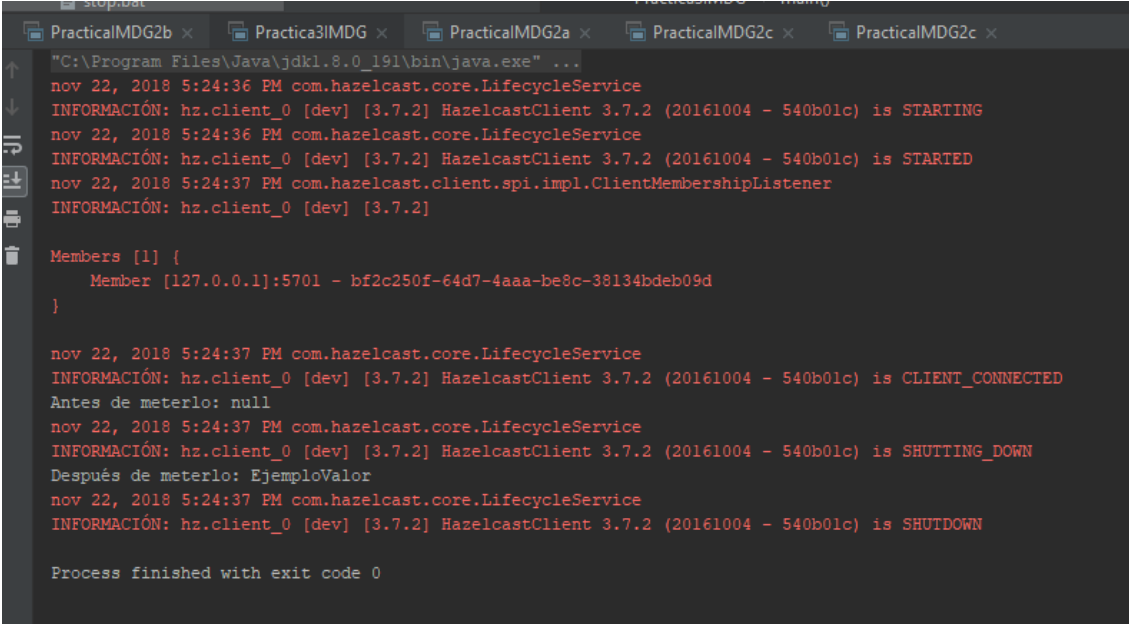
## Práctica 2: Uso básico de cache distribuida

En la segunda parte, disponemos de tres clases diferentes que se conectarán a un mismo shocket. Cada clase insertará un valor en el cache que comparten entre los tres, y además esperará a que todos los hilos escriban los datos, que en este caso son 3, y al final pinta lo que ha escrito cada conexión, tal y como se ve en el pantallazo.

```
Run: PracticalMDG2b x PracticalMDG2c x PracticalMDG2a x PracticalMDG2c x PracticalMDG2c x
}
nov 22, 2018 4:51:50 PM com.hazelcast.nio.tcp.SocketAcceptorThread
INFORMACIÓN: [192.168.1.112]:5702 [dev] [3.7.2] Accepting socket connection from /192.168.1.112:56084
nov 22, 2018 4:51:50 PM com.hazelcast.nio.tcp.InitConnectionTask
INFORMACIÓN: [192.168.1.112]:5702 [dev] [3.7.2] Connecting to /192.168.1.112:5701, timeout: 0, bind-any: true
nov 22, 2018 4:51:50 PM com.hazelcast.nio.tcp.TcpIpConnectionManager
INFORMACIÓN: [192.168.1.112]:5702 [dev] [3.7.2] Established socket connection between /192.168.1.112:56085 and /192.1
nov 22, 2018 4:51:50 PM com.hazelcast.nio.tcp.TcpIpConnectionManager
INFORMACIÓN: [192.168.1.112]:5702 [dev] [3.7.2] Established socket connection between /192.168.1.112:5702 and /192.16
nov 22, 2018 4:51:52 PM com.hazelcast.core.LifecycleService
INFORMACIÓN: [192.168.1.112]:5702 [dev] [3.7.2] [192.168.1.112]:5702 is STARTED
Person(name='Andoni', zipCode=28051, streetName='Pago Kalea', fullAddress='Pago Kalea 27, Durango')
Person(name='Ines', zipCode=28052, streetName='Calle Calle', fullAddress='Calle Calle 1, Pamplona')
Person(name='Ander', zipCode=28053, streetName='Gran via', fullAddress='Gran via 1, Bilbao')
```

## Práctica 3: Utilizando nuestro propio servidor

Por último, se ha realizado un cliente de conexión al servidor lanzado por las clases propias de Hazelcast. Se ha lanzado el servidor, y una vez lanzado el cliente se han hecho dos pruebas. La primera, conseguir el valor sin haber insertado nada, y la segunda, conseguirlo una vez se ha insertado en el servidor. El resultado ha sido el siguiente:



```
"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
nov 22, 2018 5:24:36 PM com.hazelcast.core.LifecycleService
INFORMACIÓN: hz.client_0 [dev] [3.7.2] HazelcastClient 3.7.2 (20161004 - 540b01c) is STARTING
nov 22, 2018 5:24:36 PM com.hazelcast.core.LifecycleService
INFORMACIÓN: hz.client_0 [dev] [3.7.2] HazelcastClient 3.7.2 (20161004 - 540b01c) is STARTED
nov 22, 2018 5:24:37 PM com.hazelcast.client.spi.impl.ClientMembershipListener
INFORMACIÓN: hz.client_0 [dev] [3.7.2]

Members [1] {
  Member [127.0.0.1]:5701 - bf2c250f-64d7-4aaa-be8c-38134bdeb09d
}

nov 22, 2018 5:24:37 PM com.hazelcast.core.LifecycleService
INFORMACIÓN: hz.client_0 [dev] [3.7.2] HazelcastClient 3.7.2 (20161004 - 540b01c) is CLIENT_CONNECTED
Antes de meterlo: null
nov 22, 2018 5:24:37 PM com.hazelcast.core.LifecycleService
INFORMACIÓN: hz.client_0 [dev] [3.7.2] HazelcastClient 3.7.2 (20161004 - 540b01c) is SHUTTING_DOWN
Después de meterlo: EjemploValor
nov 22, 2018 5:24:37 PM com.hazelcast.core.LifecycleService
INFORMACIÓN: hz.client_0 [dev] [3.7.2] HazelcastClient 3.7.2 (20161004 - 540b01c) is SHUTDOWN

Process finished with exit code 0
```