



Universidad
Carlos III de Madrid

TECNOLOGÍAS *DEL* SECTOR FINANCIERO

Práctica 8: Protocolos de mensajería

Andoni Alcelay

MÁSTER UNIVERSITARIO EN TECNOLOGÍAS DEL SECTOR FINANCIERO: FINTECH

1. Introducción

En esta práctica que serializan y deserializan objetos java a ficheros mediante, Json, XML, protos y kryo. La práctica se divide en 5 partes en la cual se usa cada uno de estos métodos y en la última se lanzan todos de golpe en bucle para lanzar una comparación de tiempo entre ellos.

2. Práctica 1: Jaxb

El objetivo de esta primera parte de la práctica es anotar los modelos con JAXB para que sea capaz de leer el xml de ejemplo y cumpla con el esquema XSD llamado Reference.xsd.

Tras poner las anotaciones correspondientes, el resultado ha sido el siguiente.

```
C:\devsoft\java\jdk\1.8.0_51\win64\bin\java.exe ...
18:40:10.731 [main] DEBUG com.cnebrera.uc3.tech.lesson9.jaxb.JaxbTest - [Practica 1] Size of referenceData instrument list 5
18:40:10.738 [main] DEBUG com.cnebrera.uc3.tech.lesson9.jaxb.JaxbTest - [Practica 1] Algorithm identifierTWAP
18:40:10.738 [main] DEBUG com.cnebrera.uc3.tech.lesson9.jaxb.JaxbTest - [Practica 1] Algorithm marketIdl
Process finished with exit code 0
```

3. Práctica 2: Json

Esta práctica ha sido similar a la anterior, solo que en este caso lo que se lee es un fichero Json. Las anotaciones son diferentes a las de XML, pero no tienen conflictos entre ellas, por lo que no ha dado problemas, una vez lanzado el ejecutable el resultado ha sido el siguiente:

```
JsonTest
C:\devsoft\java\jdk\1.8.0_51\win64\bin\java.exe ...
18:46:13.390 [main] DEBUG com.cnebrera.uc3.tech.lesson9.json.JsonTest - [Practica 1] Size of referenceData instrument list 5
18:46:13.393 [main] DEBUG com.cnebrera.uc3.tech.lesson9.json.JsonTest - [Practica 1] Algorithm identifierTWAP
18:46:13.393 [main] DEBUG com.cnebrera.uc3.tech.lesson9.json.JsonTest - [Practica 1] Algorithm marketIdl
18:46:13.447 [main] DEBUG com.cnebrera.uc3.tech.lesson9.json.JsonTest - [Practica 2] Json Serializer [true]
Process finished with exit code 0
```

4. Práctica 3: Proto

El objetivo de este ejercicio es aprender a como realizar un fichero .proto y a como compilarlo así como rellenar una clase de protocol buffer en base a nuestro modelo utilizando el patrón builder que provee protocol buffers en su compilación y generación de clases.

Para ello se ha creado un fichero .proto que simula las clases del modelo. Una vez creadas, se tratan igual que las anteriores, dándoles así el valor que se quiera, solo que ya serializado. El resultado tras lanzar el ejecutable es el siguiente.

```
ProtoTest x
C:\devsoft\java\jdk\1.8.0_51\win64\bin\java.exe ...
18:48:26.984 [main] DEBUG com.cnebrera.uc3.tech.lesson9.proto.ProtoTest - [Practica 3] Proto Serializer [true]

Process finished with exit code 0
```

5. Práctica 5: Kryo

Esta práctica ha sido únicamente de explicación, por lo que no se explicará.

6. Práctica 6: Comparación de rendimiento

En esta práctica se han unificado todos los métodos para comparar su rendimiento. Como dimos en lecciones anteriores, hace falta calentar la máquina virtual de java para que esta optimice la ejecución del código. Una vez lanzado todo 100 millones de veces, los resultados de las medias de rendimiento en nanosegundos han sido estos.

```
-----Serialize-----
Jaxb mean = 3346
Json mean = 919
Proto mean = 158
Kryo mean = 526
-----Deserialize-----
Jaxb mean = 6240
Json mean = 1264
Proto mean = 372
Kryo mean = 569
-----Serialize and Deserialize-----
Jaxb mean = 10224
Json mean = 2436
Proto mean = 472
Kryo mean = 1105
```