|  |
| --- |
| Máster en Big Data |
| Tecnologías de Almacenamiento |
| 1. Hands-On: Desarrollo Apache Spark |

Índice

[1. Introducción 3](#_Toc96244195)

[2. Entorno 3](#_Toc96244196)

[3. Primer contacto 3](#_Toc96244197)

[4. Carga e Inspección de datos 4](#_Toc96244198)

# Introducción

El objetivo de este Hands-On es el de familiarizarse con el framework de Spark para el análisis y procesamiento de datos

# Entorno

Para la realización de los ejercicios se va a utilizar *spark-shell* en scala ya que nos proporciona un entorno muy dinámico para la introducción de funciones y nos permite recibir una respuesta inmediata.

Para ello, utilizaremos la máquina virtual desplegada en Hands-On anteriores llamada Developer\_Hadoop y ejecutaremos el Spark Shell ubicado en */home/training/spark-1.3.1/bin*

El dataset que utilizaremos se llama auctiondata.csv y está ubicado en */home/training/training\_materials/developer/data/auction.csv*

# Primer contacto

Lo primero que vamos a hacer en este Hands-On es la realización del ejemplo que aparece en las transparencias del tema *3-Apache Spark* a partir de la página 64. Sigue los pasos que se describen a continuación:

3.1. Cargar el primer RDD con el archivo auctiondata.csv (el path ha de ser el correcto y tiene que hacer referencia a la localización del mismo en el file system local)

3.2. ¿Cuantos elementos contiene el dataset cargado?

3.3. Aplicar una transformación de filtrado para los resultados que contengan la palabra “xbox” (p. 73)

3.3. Aplica una acción para contar el número de resultados que aparecen

# Carga e Inspección de datos

Primero se recomienda mapear las variables:

val auctionid = 0

val bid = 1

val bidtime = 2

val bidder = 3

val bidderrate = 4

val openbid = 5

val price = 6

val itemtype = 7

val daystolive = 8

Contesta a los apartados siguientes enganchando la función que contesta a la pregunta y el resultado obtenido si es que existe

4.1. Carga el archivo *auctiondata.csv* haciendo un Split sobre el mismo con el separador “,”. El RDD debe llamarse *auctionRDD (*Pista: La función recibirá un condición del tipo: \_.split(","))

4.2. Que transformaciones y/o acciones se deben utilizar en cada uno de los casos?

* 1. ¿Como ver el primer elemento del inputRdd?
  2. ¿Como ver los 5 primeros elementos del RDD?
  3. ¿Cuál es el número total de pujas?
  4. ¿Cuál es el número total de elementos diferentes que se han subastado? (guardar resultado en una variable llamada totitems)
  5. ¿Cuál es el número total de tipos de elementos que se han subastado? (guardar resultado en una variable llamada totitemtype)
  6. ¿Cuál es el número total de ofertas por cada tipo de artículo? (guardar resultado en una variable llamada bids\_itemtype)

A partir de aquí, se recomienda utilizar: import java.lang.Math

* 1. En todos los artículos subastados, ¿Cuál es el número máximo de pujas?
  2. En todos los artículos subastados, ¿Cuál es el número mínimo de pujas?
  3. ¿Cuál es el número medio de pujas?