Método del Trapecio

Motivación: Un nuevo método.

Motivación

- Ecuaciones diferenciales ordinarias.
- ¿Existe solución y es única?
- Métodos de discretización.
- Método de Euler.

./Images/interpret-pvi.p

Figura: Representación del campo vectorial asociado a la ecuación logística $y'(t) = cy(t)(1 - y(t))$.

Motivación: Método de Euler

$$\begin{cases} w_0 = y_0 \\ h_i = t_{i+1} - t_i \\ w_{i+1} = w_i + h_i f(t_i, w_i) \end{cases} \tag{1}$$

Método de Euler.

Motivación: Necesidad de tratar y analizar los datos

Índice

¿Qué es TPCx-HS?

TPC™

Transaction Processing Performance Council Express Hadoop System

Benchmarking Hadoop

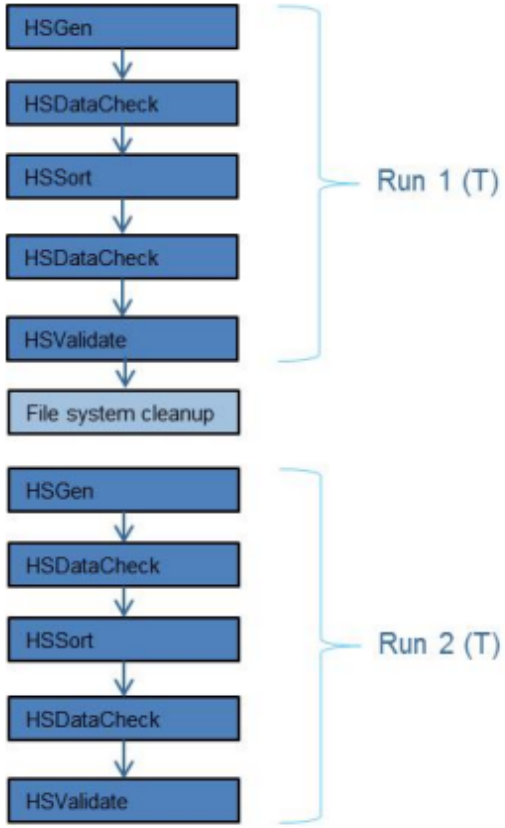
Carga de trabajo de TPCx-HS

- HSGen: generación de datos con un factor de escala.
- HSDataCheck: comprobación de los datos.
- HSSort: Implementación en Hadoop de TeraSort.
- HSValidate: comprobación de la salida.

Funcionamiento de TPCx-HS

Dos ejecuciones de cinco fases cada una.

- Fase 1: Generación de los datos.
  3-ways replication
- Fase 2: Verificación de la validez de los datos.
- Fase 3: Ordenación de los datos.
  3-ways replication
- Fase 4: Verificación de la validez de los datos.
- Fase 5: Validación de la salida



Rendimiento

Medida del rendimiento.

$$HSph@SF = \frac{SF}{T/3600}$$

Medida del rendimiento-precio.

$$\$/HSph@SF = \frac{P}{HSph@SF}$$

Parámetros:

- SF: factor de escala escogido.
- T: tiempo total de las dos ejecuciones.
- P: costo del sistema bajo estudio.

¿Por qué surge Map Reduce?

Problemas de la programación distribuida

- Implementaciones a bajo nivel: OpenMP, MPI ...
- Reimplementación de paralelismos equivalentes
- Implementación de la tolerancia a fallos

Map Reduce

- Nuevo paradigma de programación distribuida
- Simple, eficiente y de alto nivel
- Basado en las funciones map y reduce
- Gestiona los datos y la tolerancia a fallos

Ejemplo en un lenguaje funcional

$$f(x) = x^2$$

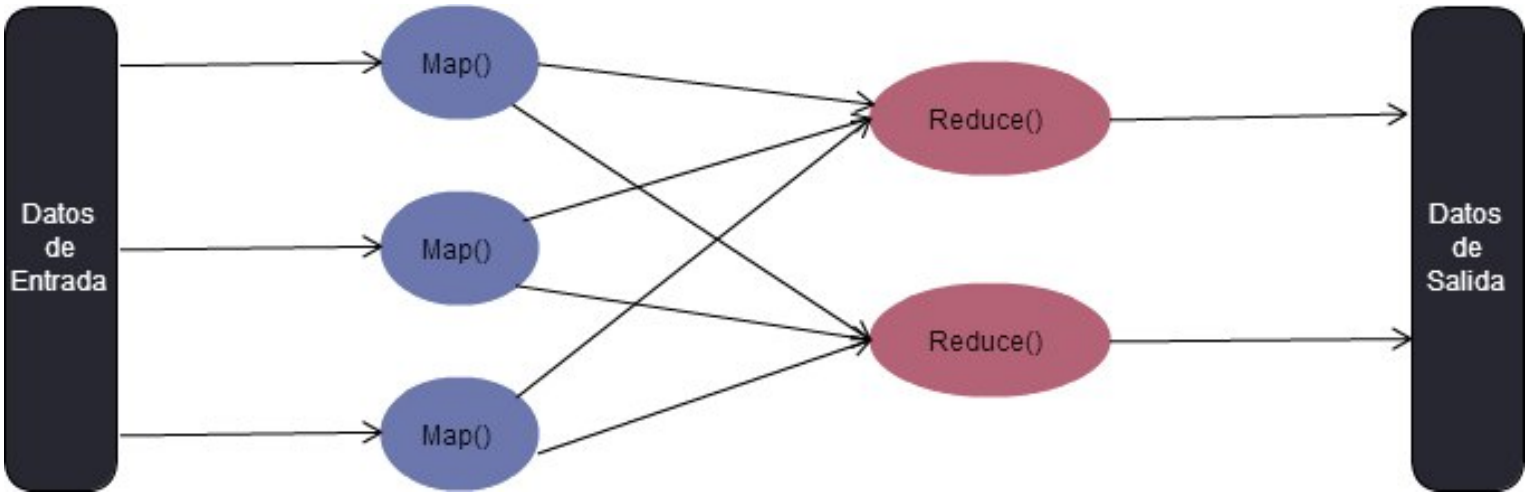
$$reduce(+, map(f, [1, 2, 3])) = reduce(+, [1, 4, 9]) = 14$$

¿Cómo funciona map reduce?

Algoritmo: Obtención del número de ocurrencias de cada una de las palabras de un texto.

```
key: Nombre del documento
value: Texto del documento
function MAP(String key, String value)
    for each word w in value do
        EmitIntermediate(w, "1" )
    end for
end function
```

```
key: Palabra
values: Conjunto con las ocurrencias de la palabra
function REDUCE(String key, Iterator values)
    result = 0
    for each value v in values do
        result += Int(v);
    end for
    return key, String(result);
end function
```

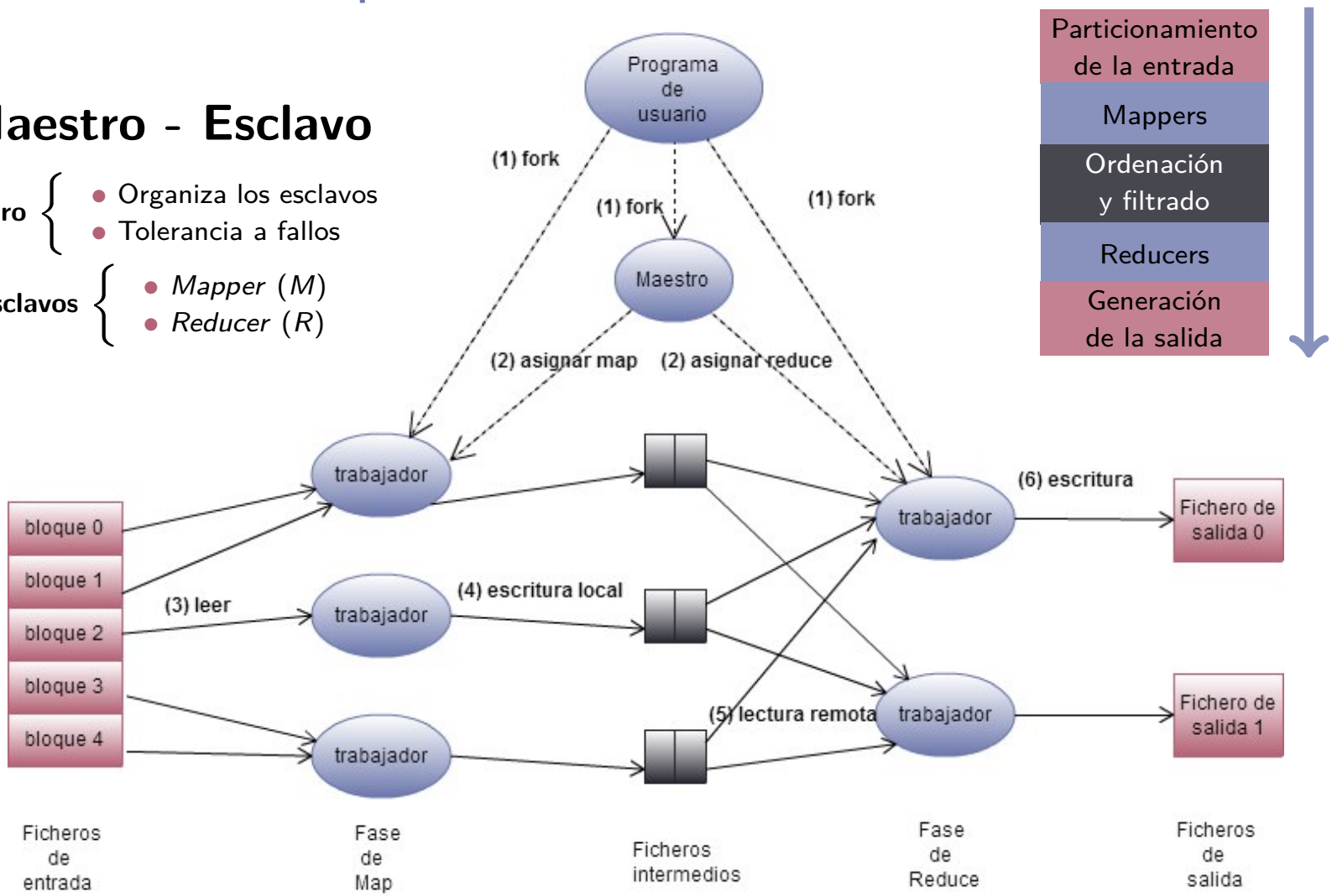


¿Cómo funciona map reduce?

Maestro - Esclavo

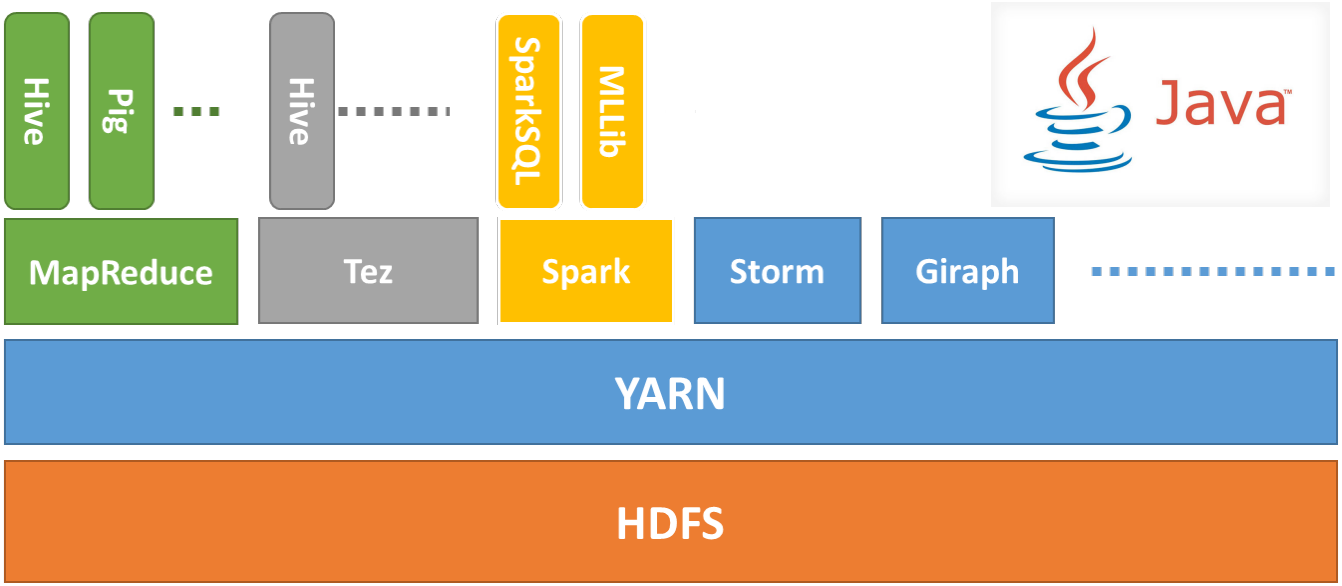
Maestro
- Organiza los esclavos
- Tolerancia a fallos

Esclavos
- *Mapper* (M)
- *Reducer* (R)

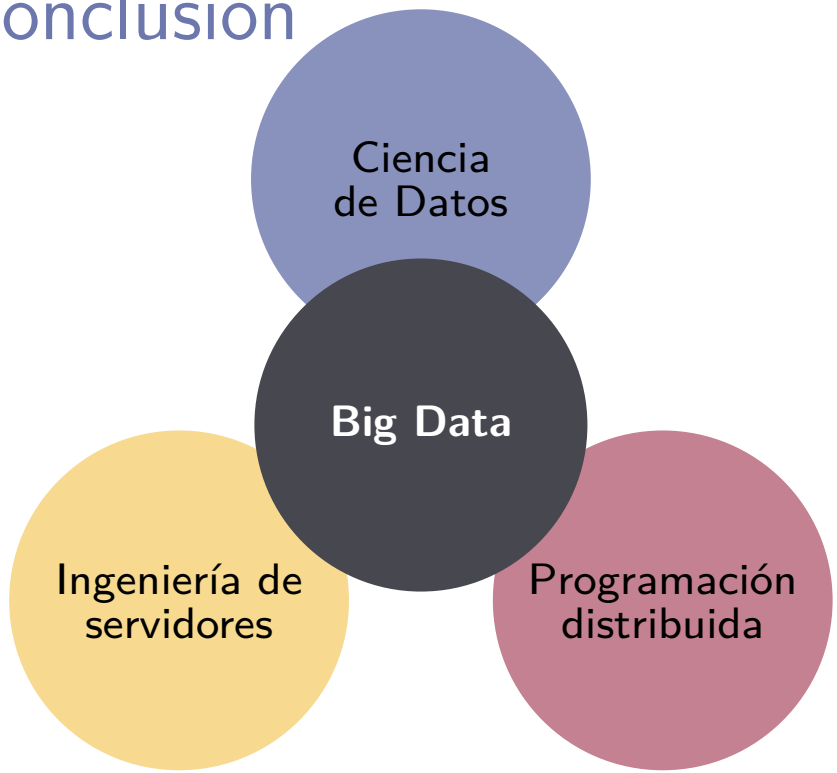


Open-source software for reliable, scalable, distributed computing

- HDFS: Sistema de archivos distribuido basado en Google File System (GFS).
- YARN: Gestión de tareas, recursos y nodos.
- SPARK: Map Reduce + procesamiento iterativo y en memoria.



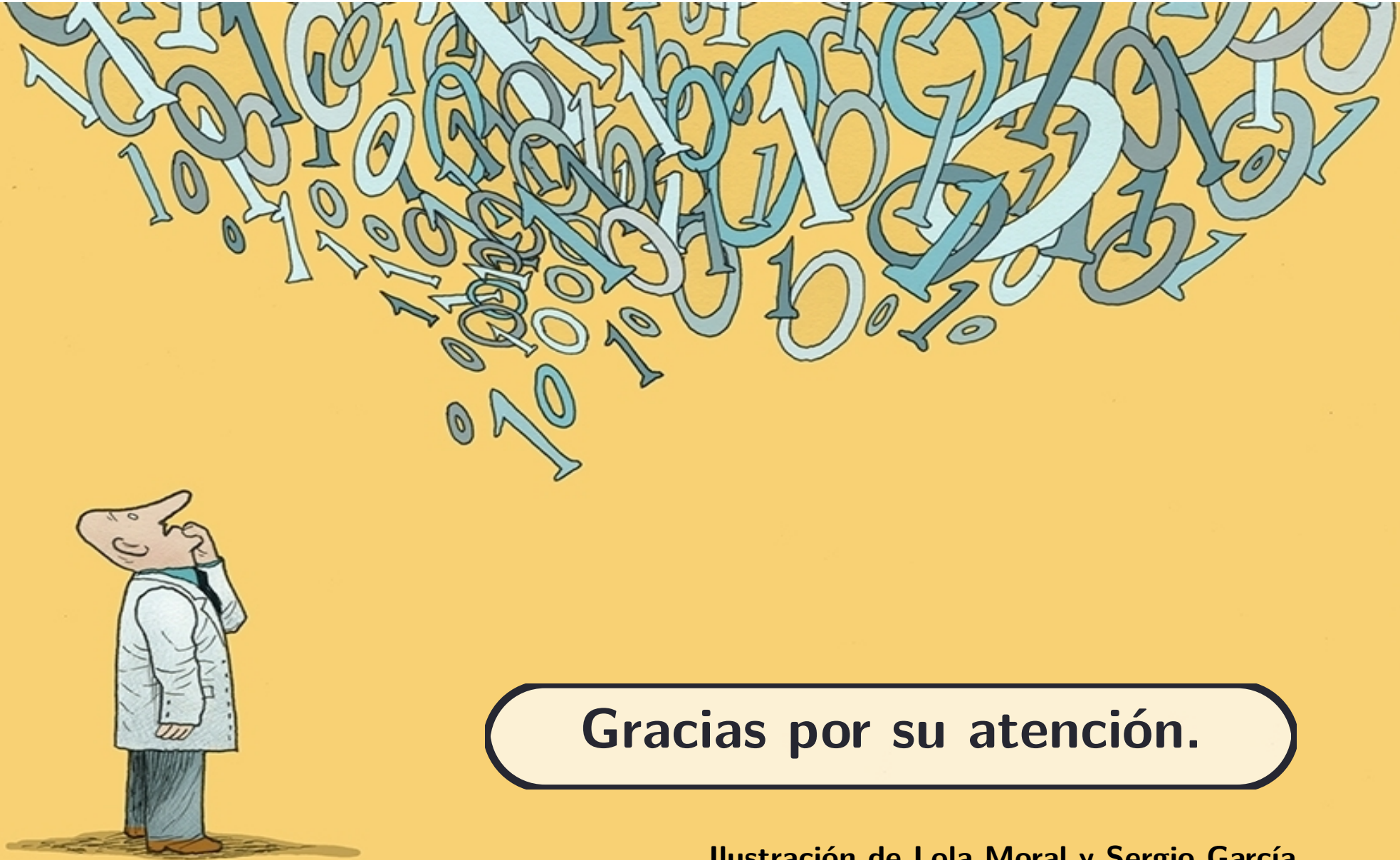
Conclusión



- Nuevas tecnologías: Spark, Flink...
- Desarrollo y diseño de algoritmos
- Benchmarks para las nuevas tecnologías

"Vivimos en la era de la información. El progreso y la innovación no se ve obstaculizado por la capacidad de recopilar datos sino por la capacidad de gestionar, analizar, sintetizar y descubrir el conocimiento subyacente en dichos datos. Este es el reto de las tecnologías de Big Data."

Francisco Herrera Triguero, Prof. Universidad de Granada



Gracias por su atención.

Ilustración de Lola Moral y Sergio García