



# Abordando el Big Data con Apache Spark

**Gervasio Varela**

Responsable de Innovación y Producto en Redegal

@gervarela

[gervasio.varela@redegal.com](mailto:gervasio.varela@redegal.com)

## ÍNDICE

- **Redegal**
- **Qué es el big data**
- **Qué es Apache Spark**
- **Tour rápido de Spark**
- **Q&A**





Nuestra experiencia es crear la tuya



**Somos** una agencia de ingeniería digital especializada en generación de **negocio online**. Expertos en **desarrollo ecommerce** y **marketing online**, especialistas en **desarrollo de apps**, excelentes **consultores de negocio**, técnicos en mantenimiento de **sistemas...**

Cada uno de nuestros casos es una experiencia para nosotros.

Con presencia en **España, México y Colombia**, somos la única agencia digital que engloba todos los servicios que tu proyecto necesita.

# NUESTROS SERVICIOS



INTEGRACIÓN



WEB



MOBILE



ECOMMERCE



MARKETING



ANALÍTICA



INTERNACIONALIZACIÓN




**+12**

**Años de experiencia en  
ingeniería y marketing online**



**+100**

Cientes de todos  
los sectores



**+80**

**Profesionales  
especializados**





Nuestras oficinas se encuentran en **Ourense, A Coruña, Madrid, Barcelona, Ciudad de México y Bogotá**, con próximas aperturas y expansión internacional en Latinoamérica.

The image features three trophies, likely silver or chrome, arranged in a row from left to right. The trophies are cup-shaped with ornate handles. The background is a solid, deep blue color. The trophies are slightly out of focus, with the one in the foreground being sharper than the others. The text "NUESTROS RECONOCIMIENTOS" is centered over the trophies in a white, bold, sans-serif font.

# **NUESTROS RECONOCIMIENTOS**





**Ardán de Gacela**, gracias a la tasa de crecimiento elevada y constante (por encima del 25%) en cifras de ingresos, durante tres años consecutivos



**Ardán Potencial Competitivo**, evaluado mediante factores relacionados con el cambio estratégico, negocio, sistemas, relaciones y finanzas



**Ardán Empresa Global**, por los procesos de internacionalización, teniendo en cuenta el número de clientes extranjeros, países y personal



**Ardán Igual en Género**, debido a la igualdad en funciones, retribuciones y actuaciones





**Ardán de Gacela**, gracias a la tasa de crecimiento elevada y constante (por encima del 25%) en cifras de ingresos, durante tres años consecutivos



**Ardán Empresa Innovadora**, evaluando altos niveles en el desempeño innovador medido a través de indicadores como el esfuerzo en I+D interna, capacitación en I+D+i o en tecnología en procesos



**Ardán Empresa Global**, por los procesos de internacionalización, teniendo en cuenta el número de clientes extranjeros, países y personal



# eawardsMÉXICO



Premiados por **eShow México** como la **Mejor Agencia Digital de Captación de Tráfico** en el año 2016



Premiados por **eShow México** como la **Mejor Agencia Digital de creación y diseño de páginas web** en el año 2017



Premiados por **eShow México** como la **Mejor Agencia Social Media** en el año 2018



# Big Data



*“extremely large data sets that may be analysed computationally to reveal patterns, trends, and associations, especially relating to human behaviour and interactions”*

*John Mashey*

# Qué es el Big Data

- *extremely large data set*
  - **Conjuntos de datos que sobrepasan las capacidades de máquinas individuales.**
- *analysed computationally*
  - **Computación paralela.**
- *reveal patterns, trends, and associations*
  - **Algoritmos inteligentes, ML no supervisado, forecasting, etc.**
- *human behaviour and interactions* + machine behaviour and anomalies
  - **Análisis de textos, análisis de emociones, análisis de imagen/vídeo.**

# Qué es el Big Data

- **Generar y capturar datos es fácil y barato**
  - Webs y Apps móviles
    - Interacción de ingentes volúmenes de usuarios, logs
  - Dispositivos conectados, IoT y maquinaria
    - Comportamiento, sensores, etc.
- **Almacenarlos, procesarlos y sacarles valor es complejo y costoso**
  - Clusters de almacenamiento
  - Procesamiento paralelo
  - Personal cualificado

## Qué es el Big Data

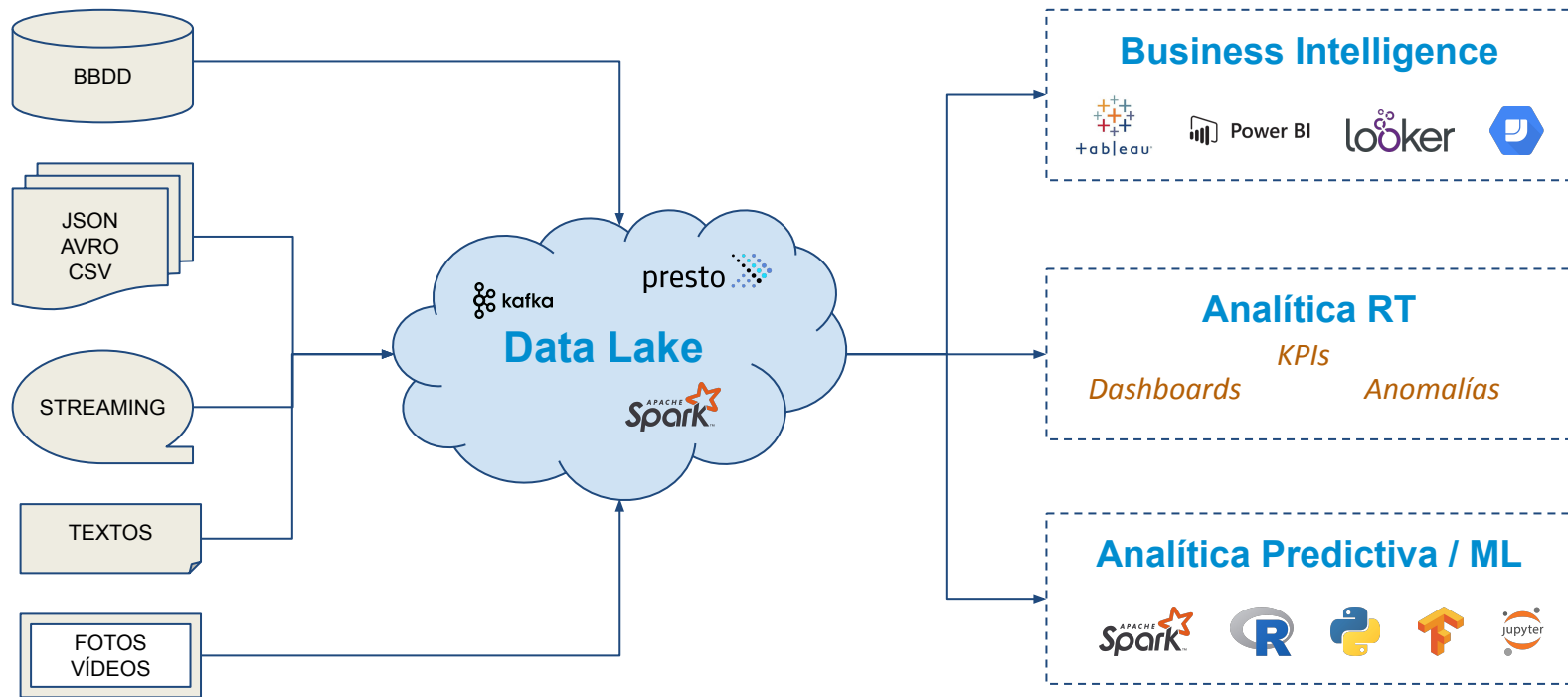
- El término se acuñó en los 90
- En la última década:
  - Los servicios en la nube lo han hecho asequible
  - Explosión de herramientas
  - Know-how y personal cualificado
- Empieza a llegar más allá de las grandes organizaciones



## Business Intelligence vs Big Data

- Conjunto de datos estructurados.
  - Esquema fijo y definido para un/os propósito/s particulare/s.
  - Objetivos de analítica definidos y cerrados.
- Multitud de conjuntos de datos con formatos y orígenes heterogéneos
    - Estructurados
    - Semi estructurados
    - No estructurados
  - Datos masivos en tiempo real
  - Analítica predictiva, machine learning e IA
  - Dinamismo y agilidad

# Business Intelligence ++



# Apache Spark

## Apache Spark

*Motor de ejecución y conjunto de librerías para el procesamiento paralelo de datos en clusters*



## Enfocado hacia el big data

- **Plataforma unificada para programar aplicaciones big data**
  - Soporte para un amplio rango de tareas típicas de big data y data analytics
  - **Carga y transformación de datos**
  - **Consulta de datos (SQL)**
  - **Streaming**
  - **Analítica predictiva y Machine Learning**

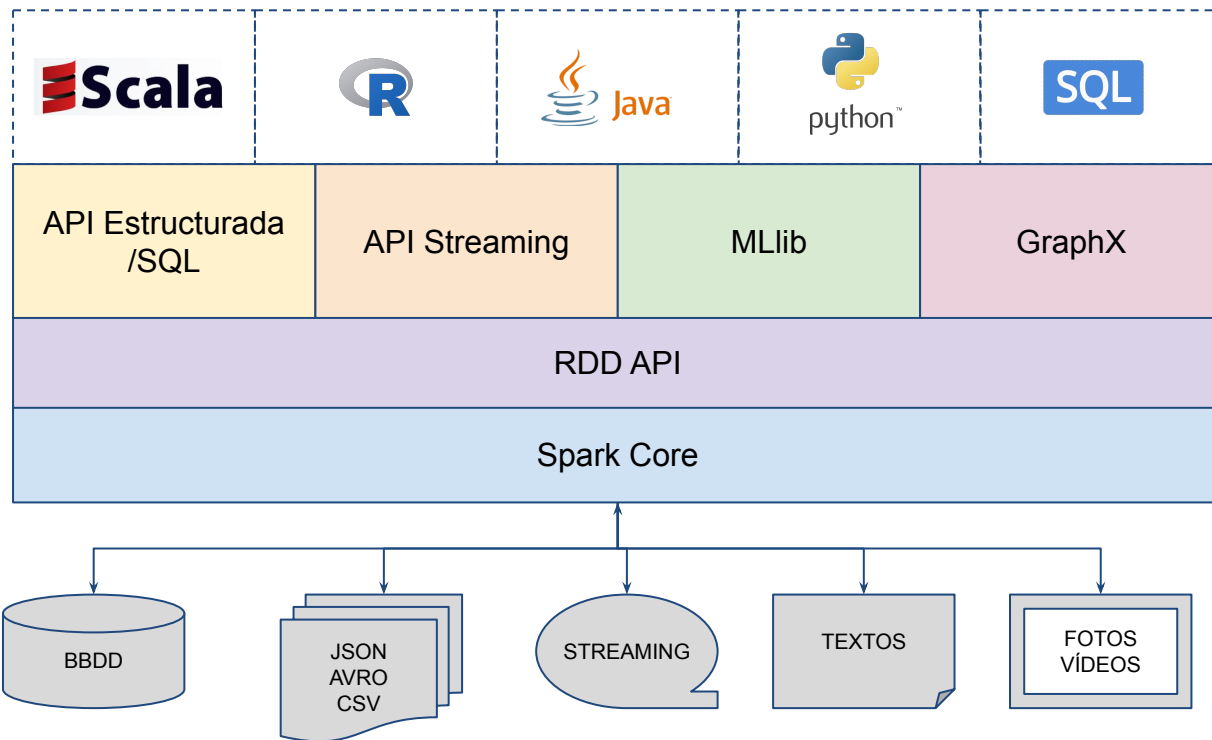
## Enfocado hacia el big data

- **Centrado en la carga y procesamiento de datos**
  - **Múltiples fuentes de datos heterogéneas**
  - **Procesamiento de los datos**
    - Independiente de la fuente/destino
    - Con la misma API
    - En paralelo de forma transparente
    - Múltiples lenguajes de programación
  - **Múltiples destinos de datos heterogéneos**

# Spark 101

- Arquitectura
- DataFrames
- Procesamiento en memoria
- Operaciones
  - Transformaciones
  - Acciones
- Evaluación perezosa

# Spark 101





# DataFrames

```
schema =  
{  
  nombre: String  
  edad: Integer  
  genero: String  
  ciudad: String  
}
```

Nombre	Edad	Género	Ciudad



```
Welcome to
┌───┐
└───┘ version 2.4.4

Using Python version 3.6.9 (default, Nov  7 2019 10:44:02)
SparkSession available as 'spark'.
>>> movieRatings = spark.read.csv("/tmp/movielens/ratings.csv", header=True)
>>> movieRatings.show(10)
+-----+
|userId|movieId|rating|timestamp|
+-----+
| 1|      1|  4.0|964982703|
| 1|      3|  4.0|964981247|
| 1|      6|  4.0|964982224|
| 1|     47|  5.0|964983815|
| 1|     50|  5.0|964982931|
| 1|     70|  3.0|964982400|
| 1|    101|  5.0|964980868|
| 1|    110|  4.0|964982176|
| 1|    151|  5.0|964984041|
| 1|    157|  5.0|964984100|
+-----+
only showing top 10 rows

>>> movieRatings.printSchema()
root
 |-- userId: string (nullable = true)
 |-- movieId: string (nullable = true)
 |-- rating: string (nullable = true)
 |-- timestamp: string (nullable = true)
>>> 
```

```
>>> from pyspark.sql.types import *
>>> ratingSchema = StructType([
...     StructField("userId", IntegerType()),
...     StructField("movieId", IntegerType()),
...     StructField("rating", FloatType()),
...     StructField("timestamp", IntegerType())
... ])
>>> movieRatings = spark.read.csv("/tmp/movielens/ratings.csv", schema=ratingSchema, header=True)
>>> movieRatings.show(10)
+-----+-----+-----+-----+
|userId|movieId|rating|timestamp|
+-----+-----+-----+-----+
|      1|       1|     4.0|964982703|
|      1|       3|     4.0|964981247|
|      1|       6|     4.0|964982224|
|      1|      47|     5.0|964983815|
|      1|      50|     5.0|964982931|
|      1|      70|     3.0|964982400|
|      1|     101|     5.0|964980868|
|      1|     110|     4.0|964982176|
|      1|     151|     5.0|964984041|
|      1|     157|     5.0|964984100|
+-----+-----+-----+-----+
only showing top 10 rows

>>> movieRatings.printSchema()
root
 |-- userId: integer (nullable = true)
 |-- movieId: integer (nullable = true)
 |-- rating: float (nullable = true)
 |-- timestamp: integer (nullable = true)
```

# Trabajando con DataFrames

- **Transformaciones**
  - Filtrar filas y/o columnas
  - Aplicar una operación a filas y/o columnas
  - Agregar filas y/o columnas
  - 1 a 1, 1 a N, N a 1
- **Los DataFrames son inmutables**

# Trabajando con DataFrames

```
>>>
>>>
>>> ratingsFiltered = movieRatings.filter("rating >= 4.5")
>>> ratingsFiltered.show(10)
+-----+-----+-----+-----+
|userId|movieId|rating|timestamp|
+-----+-----+-----+-----+
|      1|      47|    5.0|964983815|
|      1|      50|    5.0|964982931|
|      1|     101|    5.0|964980868|
|      1|     151|    5.0|964984041|
|      1|     157|    5.0|964984100|
|      1|     163|    5.0|964983650|
|      1|     216|    5.0|964981208|
|      1|     231|    5.0|964981179|
|      1|     260|    5.0|964981680|
|      1|     333|    5.0|964981179|
+-----+-----+-----+-----+
only showing top 10 rows
```

# Trabajando con DataFrames

- **Acciones y evaluación perezosa**
  - Las transformaciones configuran un plan de ejecución
  - Las acciones desencadenan la ejecución del plan
  - Tipos de acciones:
    - Visualizar los resultados
    - Escribir los resultados en destino
    - Convertir datos objetivos nativos

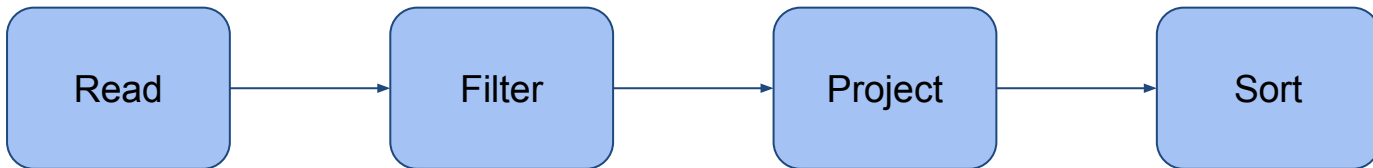
# Trabajando con DataFrames

```
>>>
>>>
>>> ratingsFiltered = movieRatings.filter("rating >= 4.5")
>>> ratingsFiltered = ratingsFiltered.sort("rating")
>>> ratingsFiltered.explain()
== Physical Plan ==
*(2) Sort [rating#87 ASC NULLS FIRST], true, 0
+- Exchange rangepartitioning(rating#87 ASC NULLS FIRST, 200)
   +- *(1) Project [userId#85, movieId#86, rating#87, timestamp#88]
      +- *(1) Filter (isnotnull(rating#87) && (cast(rating#87 as double) >= 4.5))
         +- *(1) FileScan csv [userId#85,movieId#86,rating#87,timestamp#88] Batched: false, Format
: CSV, Location: InMemoryFileIndex[file:/tmp/movielens/ratings.csv], PartitionFilters: [], PushedF
ilters: [IsNotNull(rating)], ReadSchema: struct<userId:int,movieId:int,rating:float,timestamp:int>
>>> ratingsFiltered.show(10)
+-----+-----+-----+
|userId|movieId|rating| timestamp|
+-----+-----+-----+
| 2| 1704| 4.5|1445715228|
| 7| 50| 4.5|1106635993|
| 2| 58559| 4.5|1445715141|
| 2| 68157| 4.5|1445715154|
| 2| 80489| 4.5|1445715340|
| 3| 1587| 4.5|1306464003|
| 3| 3024| 4.5|1306464054|
| 3| 5764| 4.5|1306464021|
| 3| 7899| 4.5|1306464036|
| 3| 26409| 4.5|1306463993|
+-----+-----+-----+
only showing top 10 rows
```

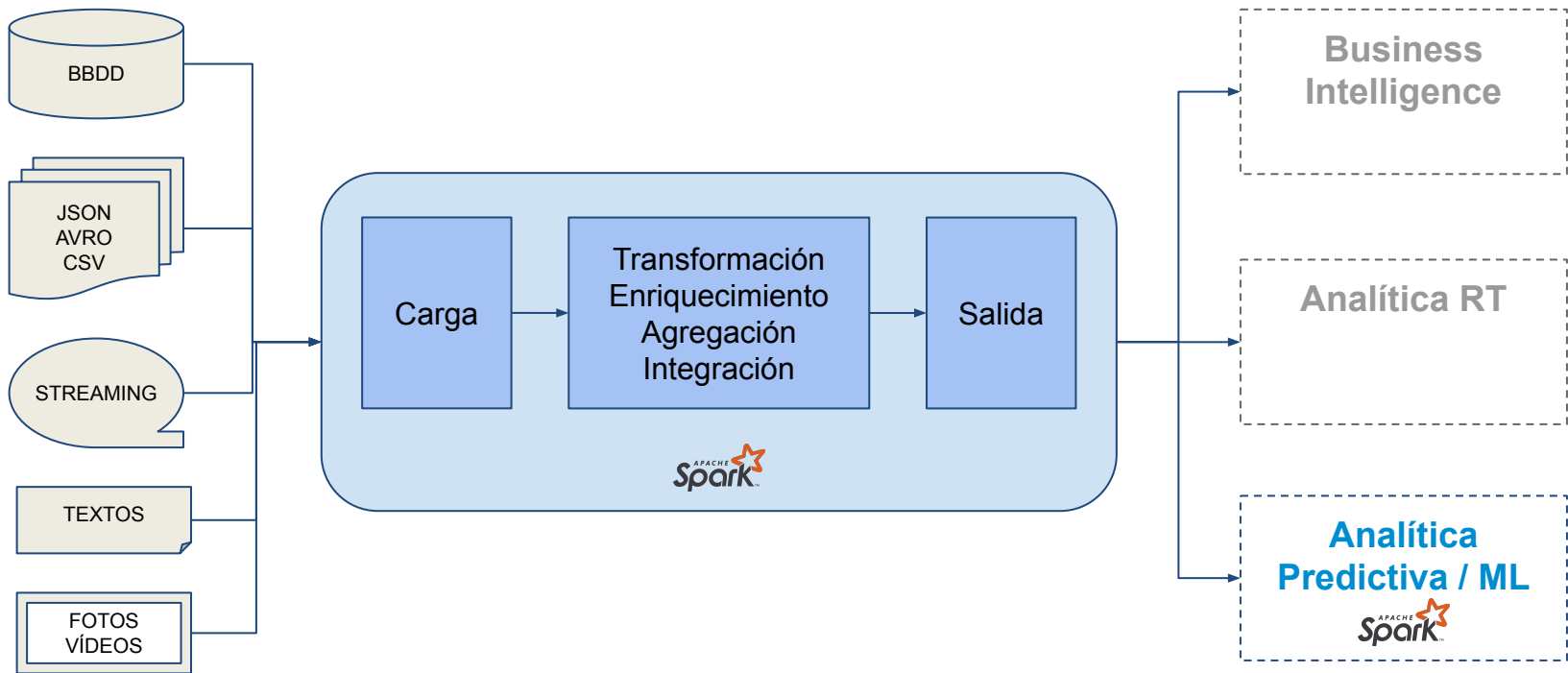


# Trabajando con DataFrames

```
movieRatings.filter("rating >= 4.5").sort("ratings")
```

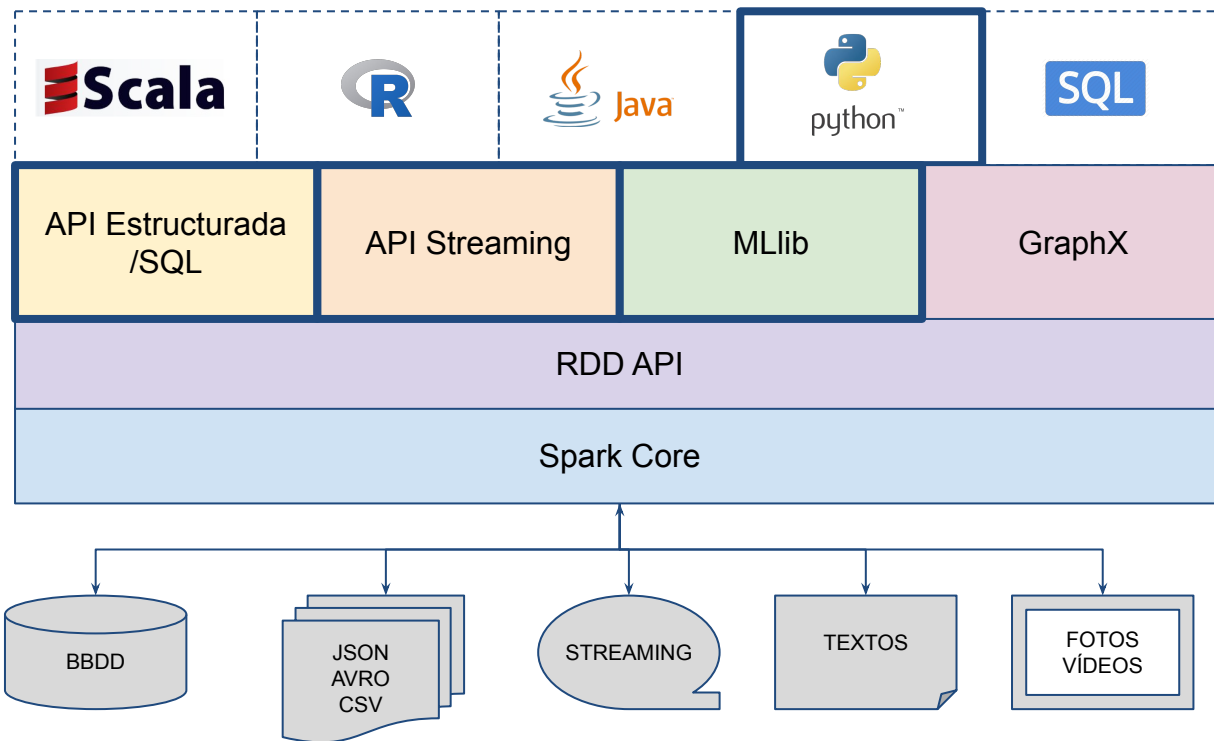


# Big Data con Spark



# Tour rápido de Spark

## ¿Qué vamos a ver?



## ¿Qué vamos a ver?

- **API Estructurada / SQL**
  - Procesamiento de datasets en batch
  - API programática inspirada en operaciones SQL
- **API Streaming**
  - Procesamiento de datos en streaming
  - Misma API que batch con algunas limitaciones
- **MLlib**
  - Librerías de machine learning paralelizadas

## API Estructurada

- **Es una API de alto nivel para la manipulación de dataframes**
  - Permite manipular cualquier dataframe, sea de origen estructurado (BBDD relacional, Parquet, etc.) o no (JSON, CSV, etc.)
  - Es la base del resto de APIs de Spark, incluyendo Streaming y ML
  - Si sabéis SQL os resultará familiar

# API Estructurada: Operaciones

- **Crear Dataframes**

- Importar datos para procesarlos.
- Soporta diferentes fuentes de entrada mediante conectores.

```
jsonDF = spark.read
    .format("json")
    .load("/data/flight/2015-summary.json")

mongoDF = spark.read
    .format("mongo")
    .option("uri",
        "mongodb://127.0.0.1/people.contacts")
    .load()
```

```
jdbcDF = spark.read
    .format("jdbc")
    .option("url", "jdbc:postgresql:dbserver")
    .option("dbtable", "schema.tablename")
    .option("user", "username")
    .option("password", "password")
    .load()
```



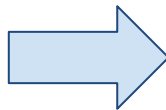
# API Estructurada: Operaciones

- **Proyección**

- Manipular las columnas: seleccionar, eliminar, renombrar, aplicar funciones, etc.

Nombre	Ciudad	Edad
Leela	NNY	25
Branigan	NNY	42
Hermes	NNY	56
Fry	NNY	26
Bender	NNY	5

```
df.select(  
    "nombre", "edad",  
    year(current_date()) - col("edad") )
```



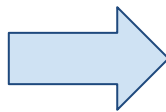
Nombre	Edad	Nacimiento
Branigan	42	1978
Hermes	56	1964

# API Estructurada: Operaciones

- **Filtrado**

- Seleccionar filas en base a condiciones

Nombre	Ciudad	Edad
Leela	NNY	25
Branigan	NNY	42
Hermes	NNY	56
Fry	NNY	26
Bender	NNY	5



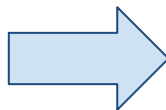
```
df.filter("edad > 40")
```

Nombre	Ciudad	Edad
Branigan	NNY	42
Hermes	NNY	56

# API Estructurada: Operaciones

- Agrupación y Agregación

Nombre	Ciudad	Edad
Leela	NNY	25
Lisa	Spring.	8
Homer	Spring.	43
Fry	NNY	26
Bender	NNY	5



```
df.select("ciudad", "edad")  
  .groupBy("ciudad")  
  .agg(avg("edad"))
```

Ciudad	Media Edad
NNY	18.67
Spring.	25.5

# API Estructurada: Operaciones

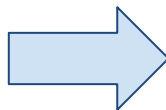
- Joins

Nombre	Género
Leela	M
Lisa	M



Nombre	Ciudad	Edad
Leela	NNY	25
Lisa	Spring.	8

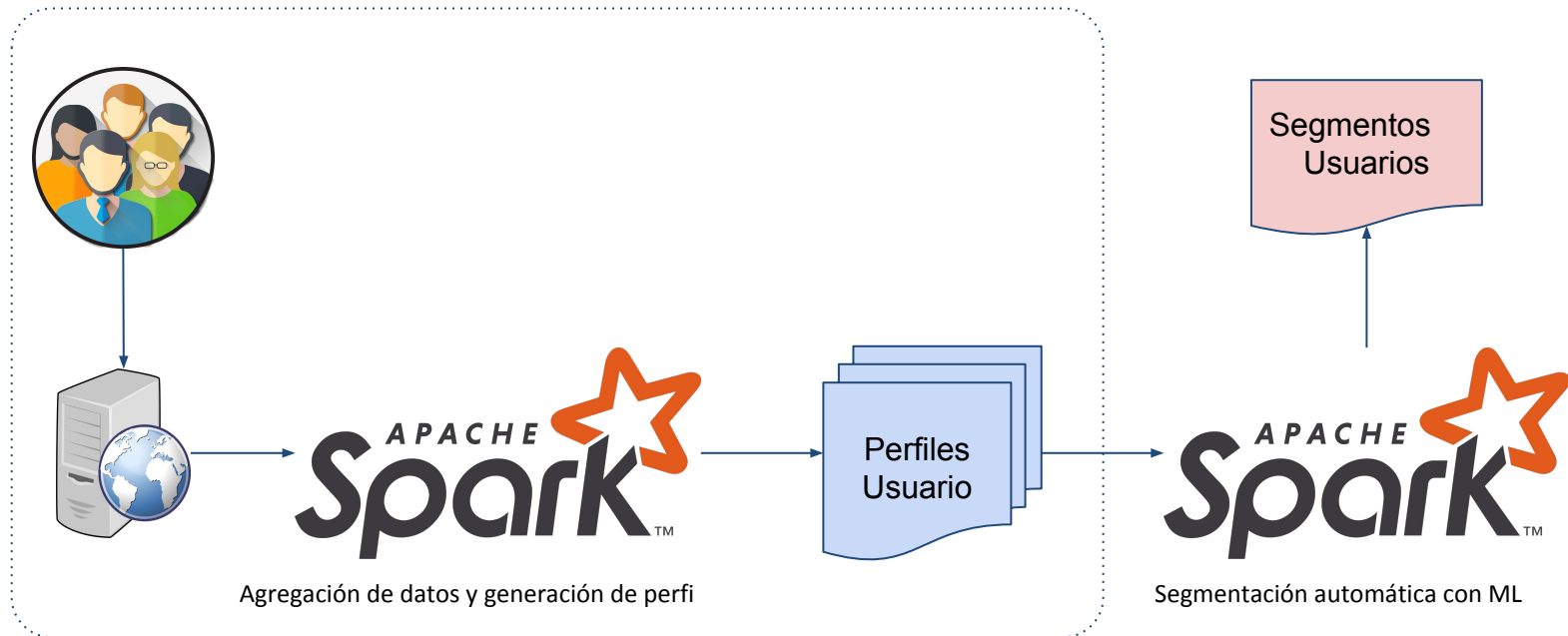
```
df.join(df2, df["nombre"] == df2["nombre"], "left_outer")
```



Nombre	Ciudad	Edad	Género
Leela	NNY	25	M
Lisa	Spring.	8	M

- <https://github.com/gervarela/spark-101>
- **Movielens Dataset**
  - <https://grouplens.org/datasets/movielens/>
  - Movielens Latest Small
    - 100,000 ratings and 3,600 tag applications applied to 9,000 movies by 600 users.
- **Jupyter Notebook**
  - Programación interactiva
  - Visualización de resultados

## Ejemplo: Perfil de usuario a partir de eventos

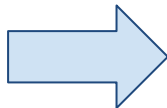


# Ejemplo: Perfil de usuario a partir de eventos

movieId	title	genres
1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy
10	GoldenEye (1995)	Action Adventure Thriller



userId	movieId	rating	timestamp
1	1	4.0	964982703
1	3	4.0	964981247
467	10	4.5	964981249



userId	num_Adventure	rating_Adventure	num_Action	rating_Action
1	45	4.56	30	4.07
467	20	3.78	78	4.12



# Spark Streaming

- **Posibilita el procesamiento de streams continuos de datos**
  - Ingesta de datos desde diversas fuentes de datos en streaming como: Apache Kafka, sockets TCP o Amazon Kinesis.
  - Procesamientos complejos utilizando la API estándar de Spark.
  - Salida hacia diversos destinos, BBDD, dashboards en tiempo real, otros sistemas de streaming, etc.

# Spark Streaming

- **Procesamiento en micro-batches de datos**
  - Recibe como entrada streams de datos en vivo, divide los datos en batches, que son procesados por spark para generar el un stream de salida en batches.



Fuente: <https://spark.apache.org/docs/latest/streaming-programming-guide.html#overview>

## Ejemplo: Dashboard en tiempo real



- **Librería de machine learning paralelizada sobre Spark**
  - Extracción, transformación y selección de características.
  - Clasificación y Regresión
    - Árboles de decisión, random forest, SVMs, RRNNs, etc.
  - Clustering
    - k-means, bisecting k-means, GMM, etc.
  - Collaborative filtering
  - Frequent Pattern Mining

## Ejemplo: Recomendación de películas



# Q & A

<https://github.com/gervarela/spark-101>

@gervarela  
gervasio.varela@redegal.com



Avda. Horacio 930, Polanco.  
Miguel Hidalgo 11560, Ciudad de México.  
55 5280 2992

Carrera 9A, N°99-07, Torre 'La Equidad',  
Oficina 901, Bogotá.  
+571 5209851

Avda. de Santiago, 9 Bajo. 32001,  
Ourense, España  
988 54 98 58

C/ Argensola, 17-1º izq. 28001  
Madrid, España  
910 800 966

Avenida de Linares Rivas, 56 Bajo. 15005  
A Coruña, España  
902 09 15 14





[info@redegal.com](mailto:info@redegal.com)  
[www.redegal.com](http://www.redegal.com)

España | México | Colombia