Introducción a arquitecturas y herramientas de Big Data

17 de noviembre de 2020

Formación y experiencia

Nicolás Balparda

INSTRUCTOR

- *Estrategia tecnológica*
- *Arquitectura de datos y sistemas*
- *Big Data*
- *Análisis de datos*

FORMACIÓN

- Ingeniero en sistemas
- MsC en Gestión de Datos e Innovación tecnológica
- MsC en Big Data y Business Analytics (en curso)

Presentación

a. Nombre

b. ¿Qué rol ocupa en su trabajo?

c. ¿Cuál cree que es su relación en el trabajo con los datos?

d. ¿Participaste en los cursos de las semanas pasadas?

Temario del curso

1	**Conceptos de Big Data**
	¿Qué es Big Data? Casos de uso frecuentes
2	**Arquitecturas de Hadoop y su ecosistema de herramientas**
	Apache Hadoop: almacenamiento y cómputo Plataformas HDP y evolución de plataformas de Big Data Ejercicio práctico con HDFS
3	**Procesamiento en paralelo con datos distribuidos**
	Demo con MapReduce
4	**Ingesta y procesamiento de datos**
	Herramientas de procesamiento batch (Spark, Sqoop y Hive) - Ejercicios Herramientas de procesamiento real time (Kafka, NiFi, Druid y Spark Streaming) - Ejercicios
5	**Data Science and Engineering Platform en HDP 3.0**
6	**Análisis y visualización de datos**
	Procesamiento exploratorio en notebooks (demo de Jupyter y ejercicio práctico con Zeppelin) Visualizadores - Herramientas más utilizadas (demo de Superset)

Conceptos de Big Data

Dinámica

Consigna

- ¿Qué entienden por BigData?
- ¿Han trabajado con alguna herramienta de Big Data?

15 minutos

Datos masivos...

¿Qué hacemos con tantos datos?

Conceptos de Big Data

¿Qué es Big Data?

¿Qué es Big Data?

Big Data son activos de información de gran volumen, velocidad y / o variedad que exigen formas rentables e innovadoras de procesamiento de información que permitan una mejor comprensión, toma de decisiones y automatización de procesos.

Muchas veces el mismo término se refiere a las **nuevas tecnologías** que hacen posible el almacenamiento y procesamiento, además de al uso que se hace de la información obtenida a través de dichas tecnologías.

¿Pero de dónde vienen todos esos datos?

Producidos por personas

Mandar un email, escribir un comentario en Facebook, contestar a una encuesta telefónica, ingresar información en una hoja de cálculo, responder a un WhatsApp, hacer clic en un enlace de Internet…

Entre máquinas

Lo que se conoce igualmente como M2M. Así, los termómetros, parquímetros y sistemas de riego automático de las ciudades, los GPS de vehículos y teléfonos móviles, el Wifi, el ADSL, la fibra óptica y el Bluetooth...

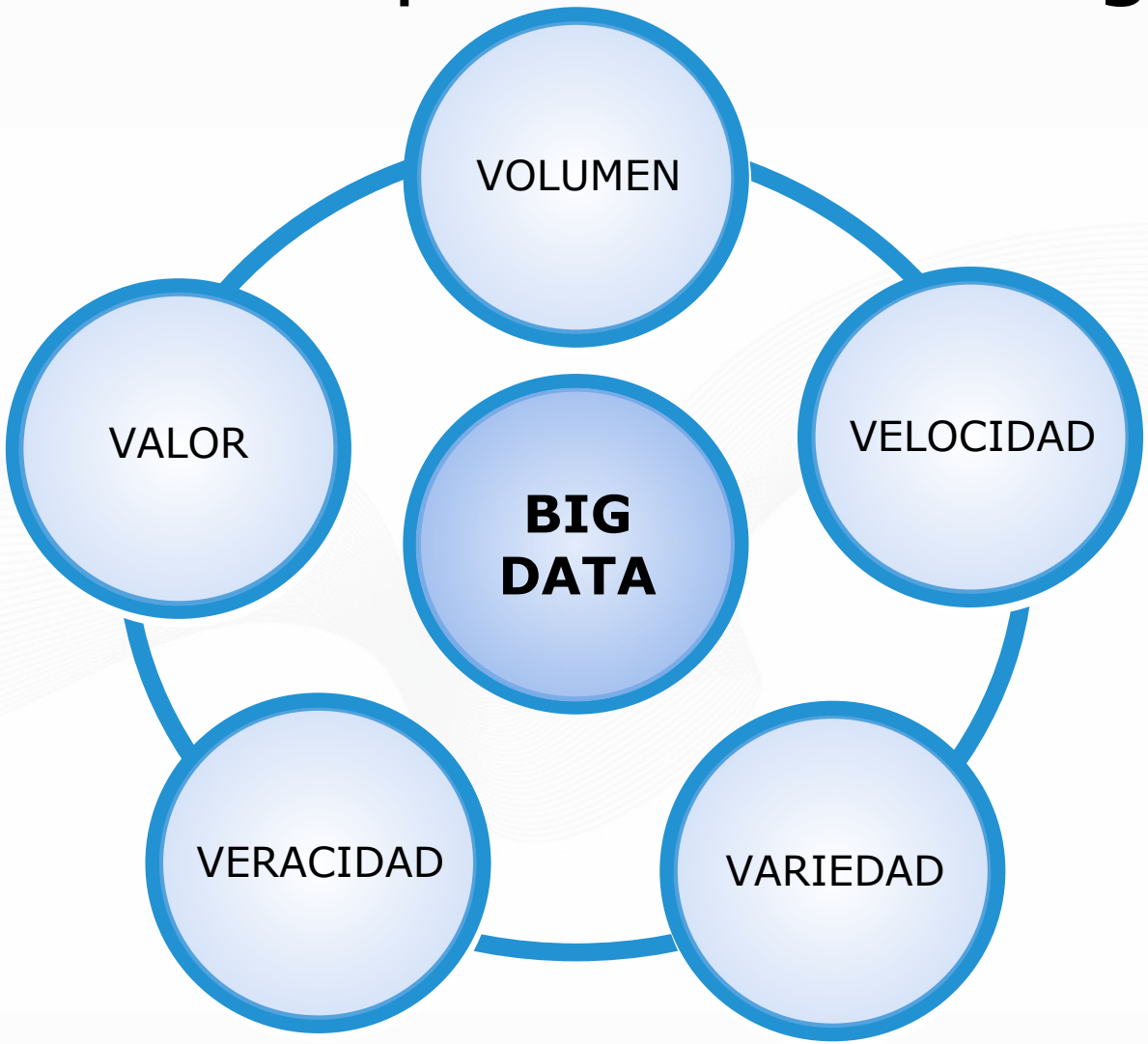
Biométricos

Artefactos como sensores de huellas dactilares, escáneres de retina, lectores de ADN, sensores de reconocimiento facial o reconocimiento de voz.

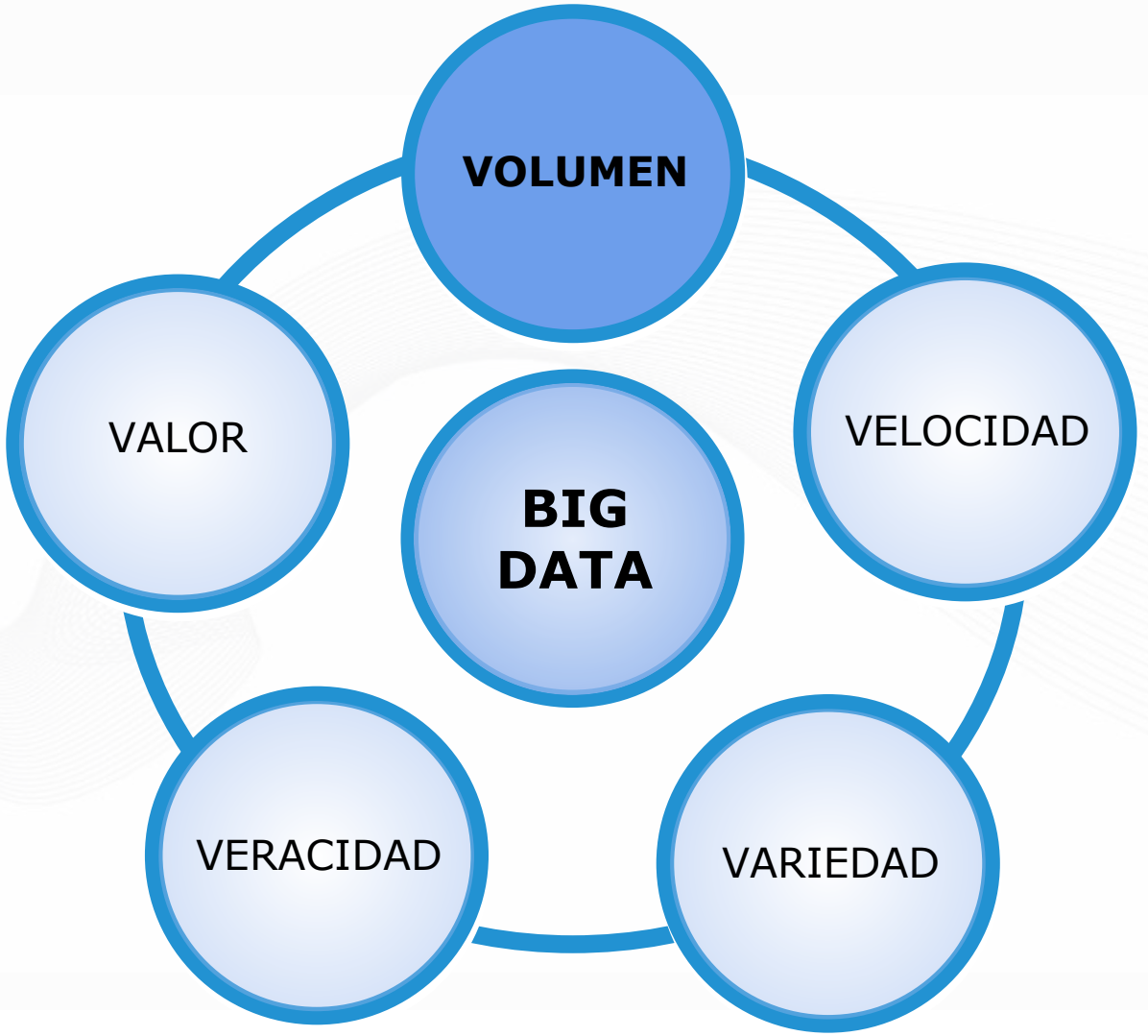
Marketing o transaccionales

Nuestros movimientos en la Red están sujetos a todo tipo de mediciones que tienen como objeto estudios de marketing y análisis de comportamiento. Asimismo, el traspaso de dinero de una cuenta bancaria a otra, la reserva de un billete de avión o añadir un artículo a un carrito de compra virtual de un portal de comercio electrónico, serían algunos ejemplos.

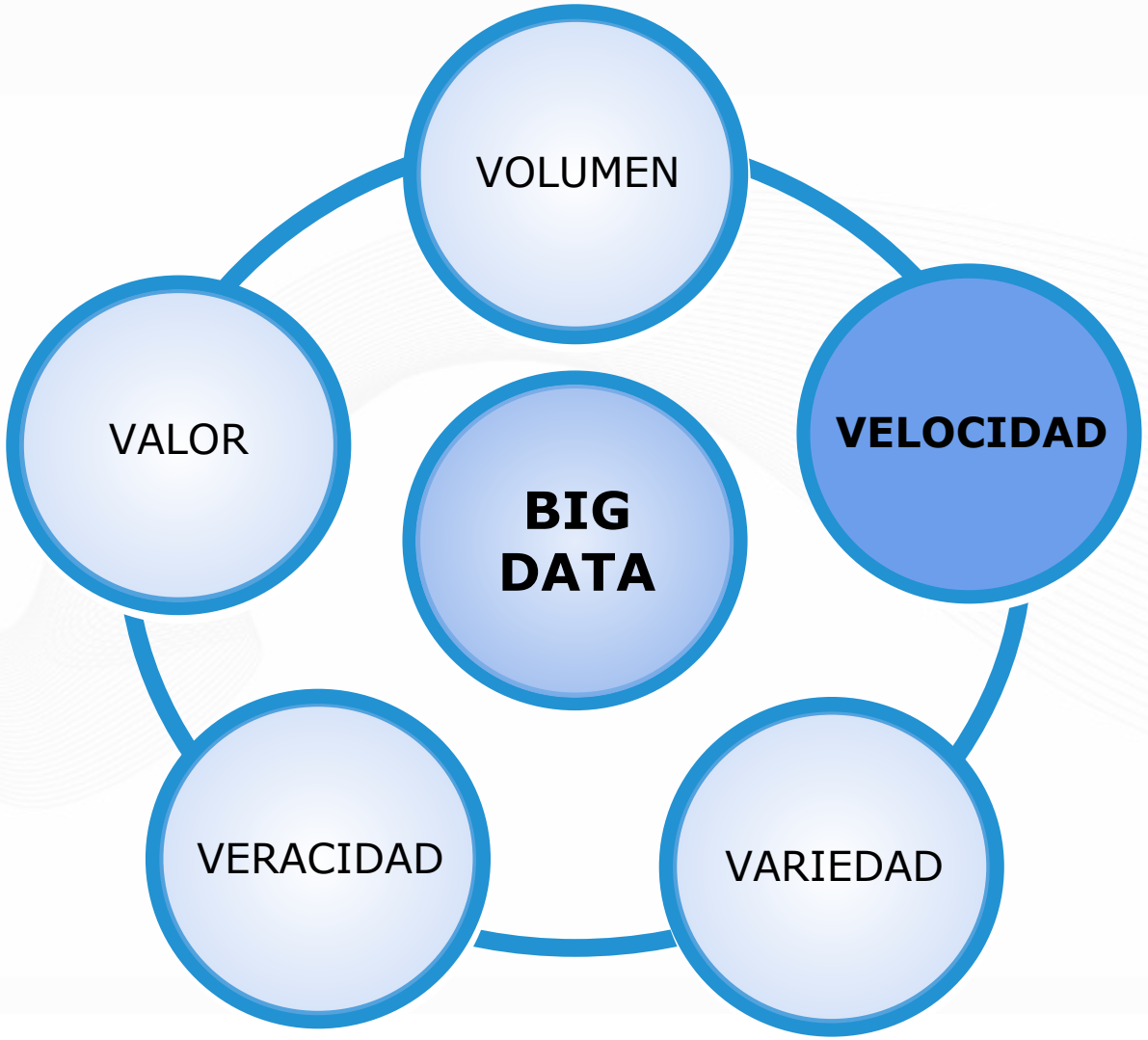
¿Qué hace que Data sea Big Data?

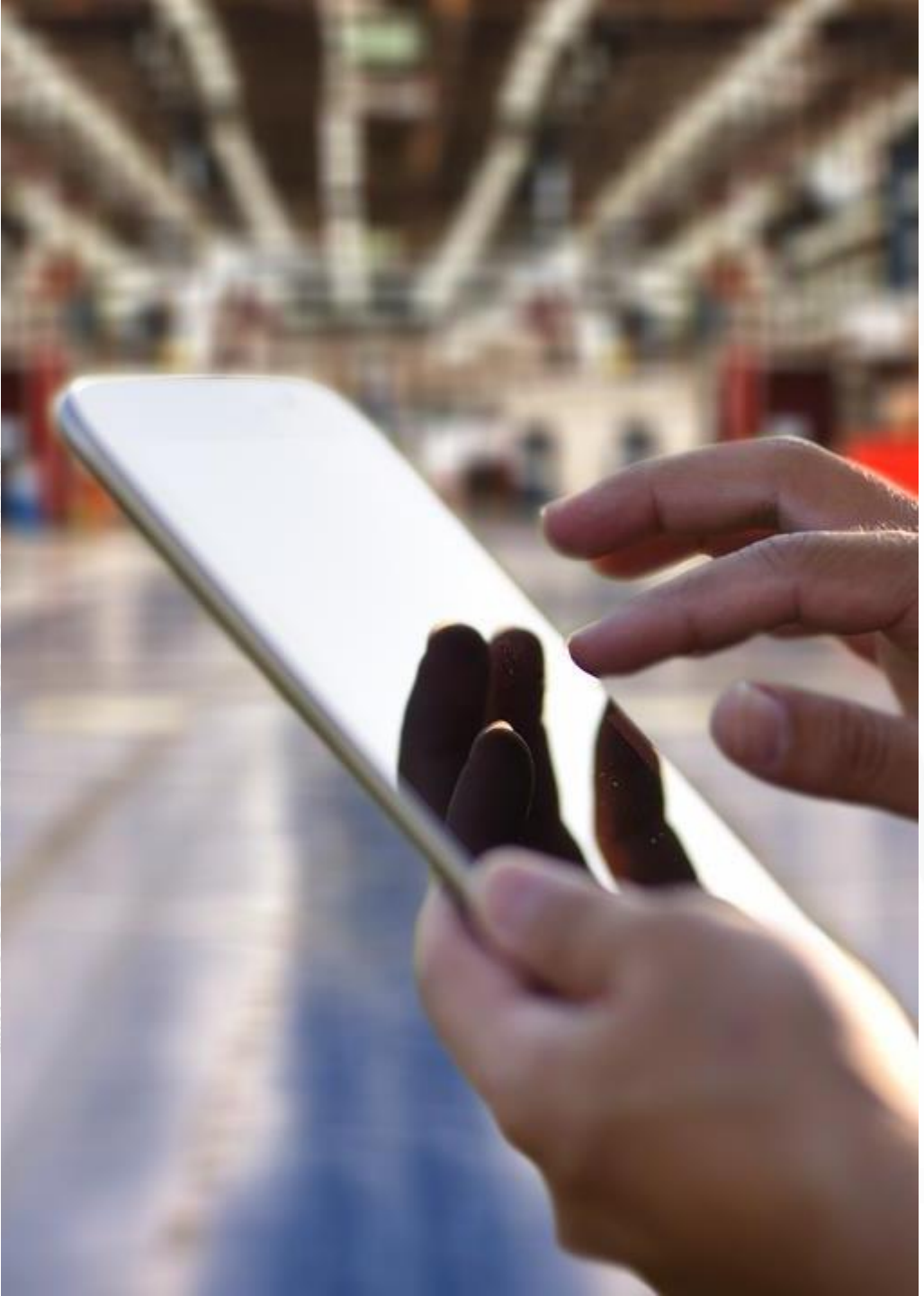


¿Qué hace que Data sea Big Data?

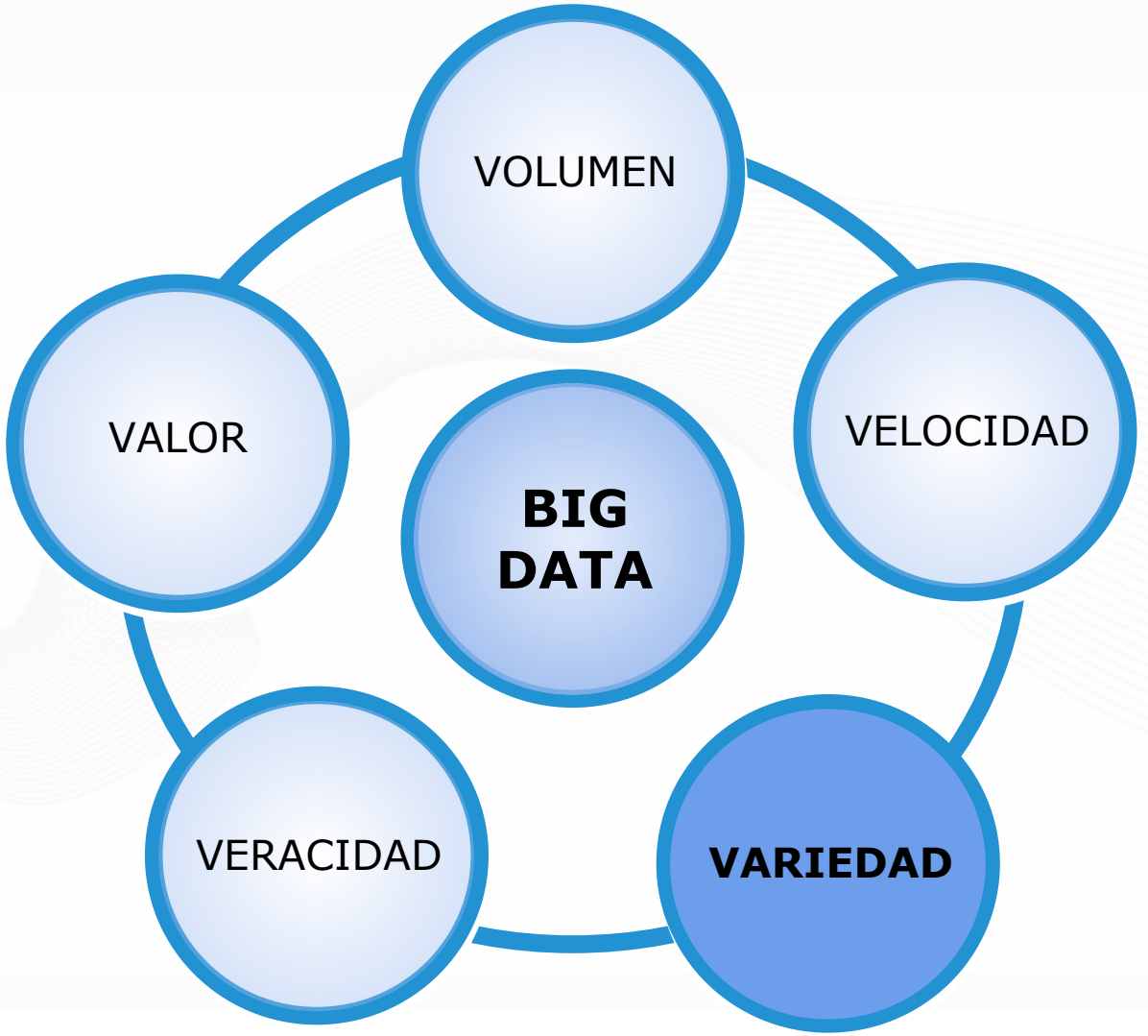


¿Qué hace que Data sea Big Data?

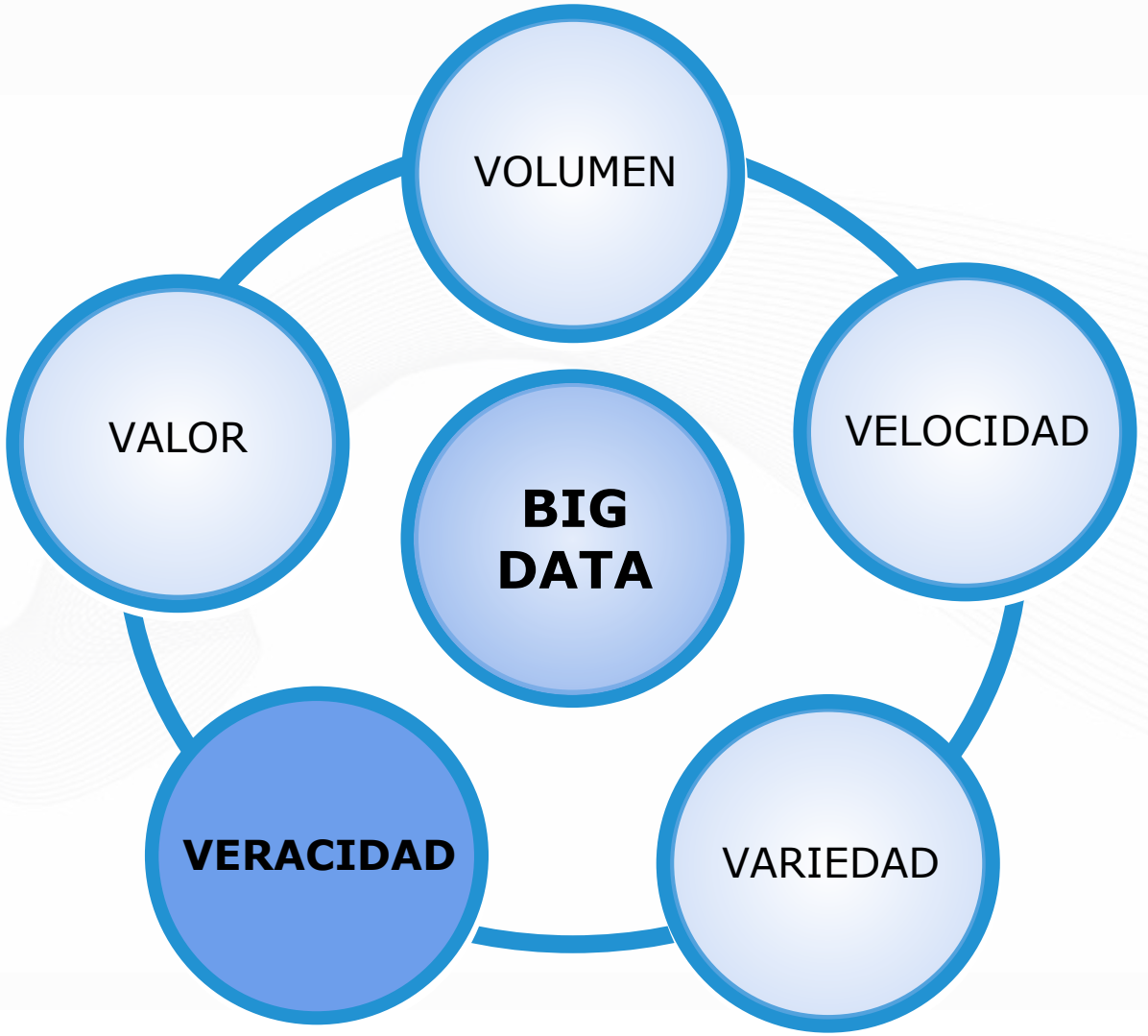




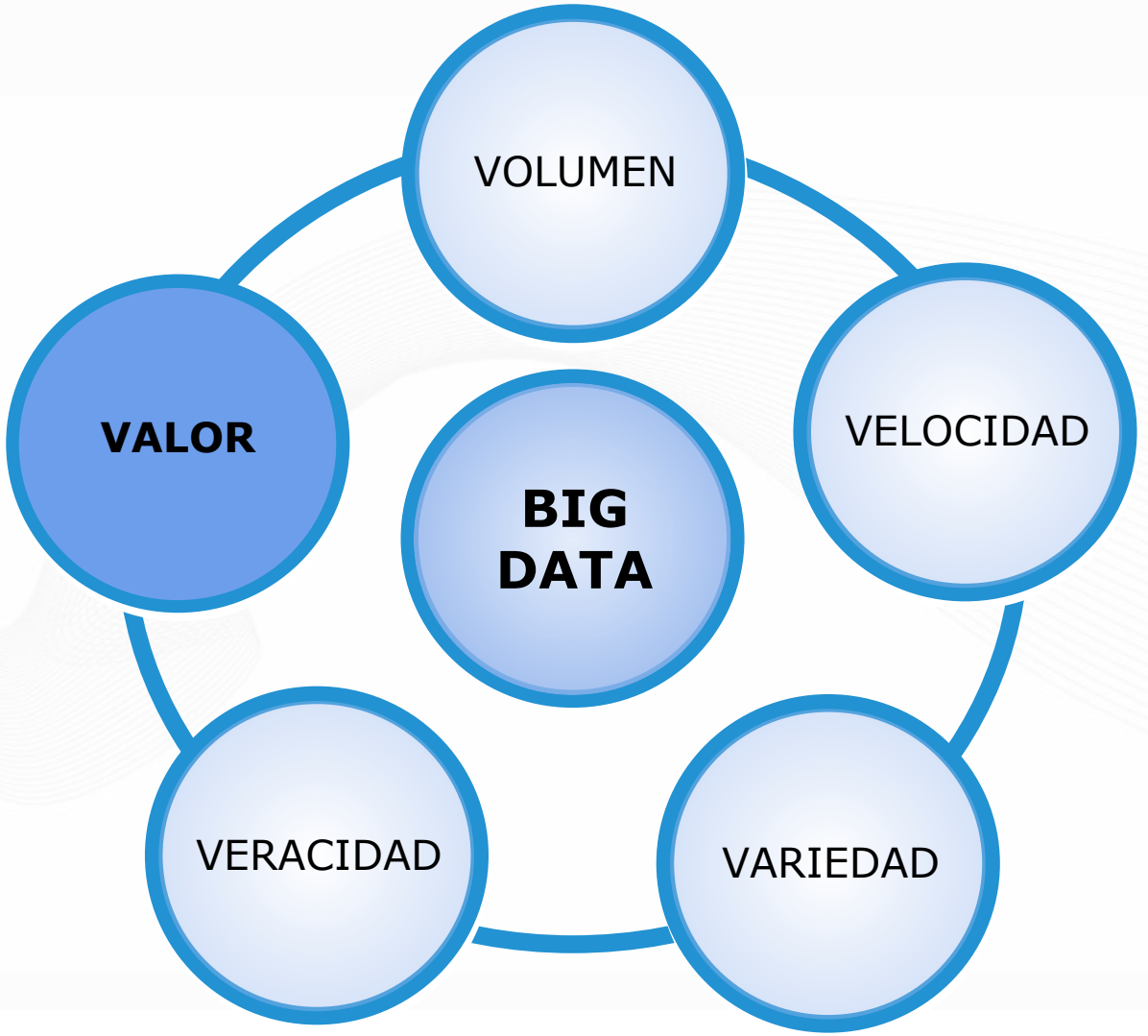
¿Qué hace que Data sea Big Data?



¿Qué hace que Data sea Big Data?



¿Qué hace que Data sea Big Data?



Conceptos de Big Data

Casos de uso frecuentes

Casos de uso frecuentes

Casos de uso en la administración pública

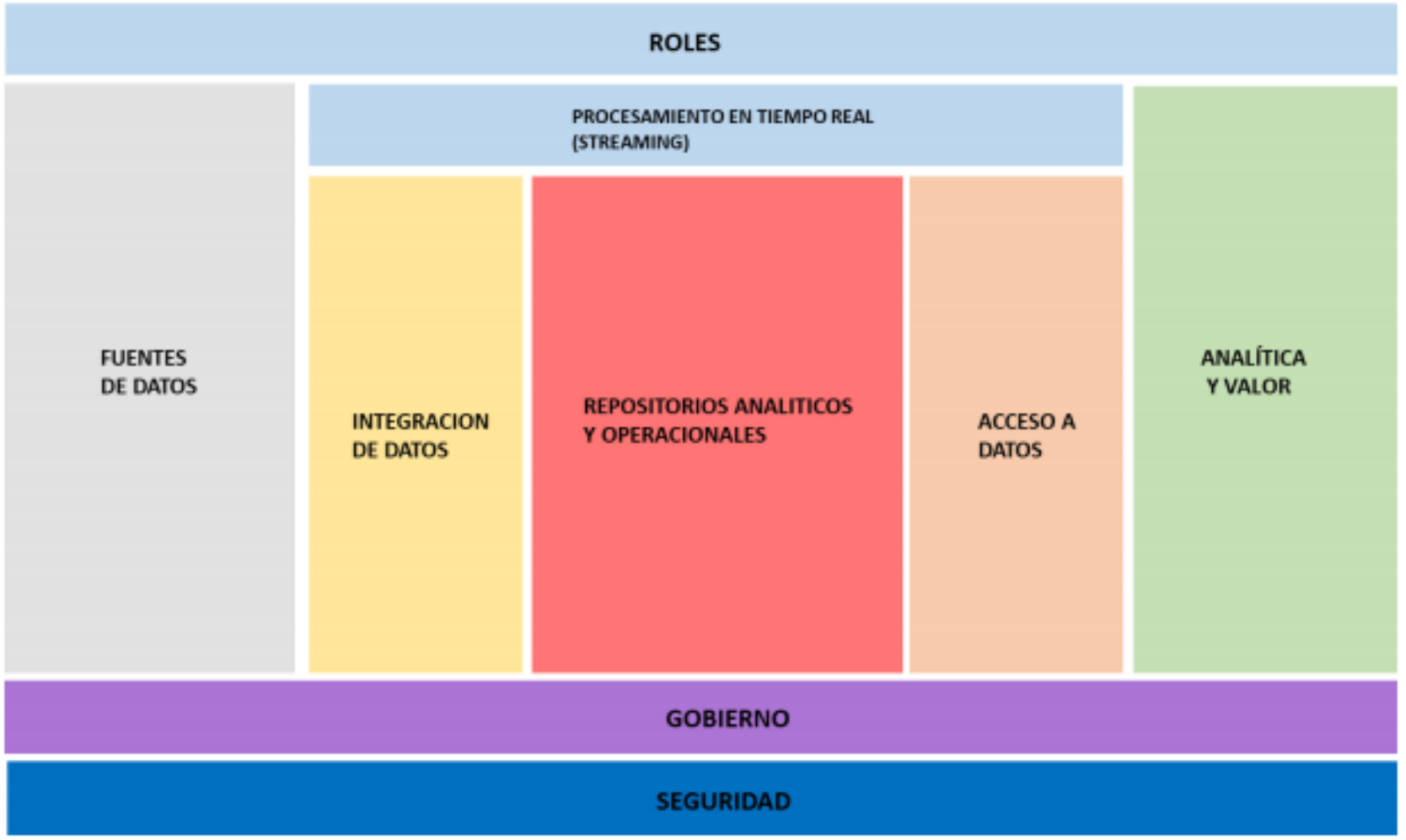
IMM - Movilidad y Transporte
Mediante IoT/Big data desarrollaron la Matriz Origen Destino (MOD) para el transporte público de Montevideo, herramienta clave para la planificación y gestión del transporte. Mediante analítica predictiva determinaron indicadores claves como el porcentaje de ocupación de autobuses para diferentes líneas y horarios, control horario de flota, cantidad de pasajeros que suben por parada, entre otros.

IMM - Turismo
Utilizan Big Data como herramienta para conocer los principales mercados emisores, preferencias y comportamiento de los visitantes. Esto ayuda a identificar y segmentar mejor los perfiles de visitantes que componen la demanda. Con Big Data se podrá mejorar el servicio que Montevideo ofrece a sus visitantes.

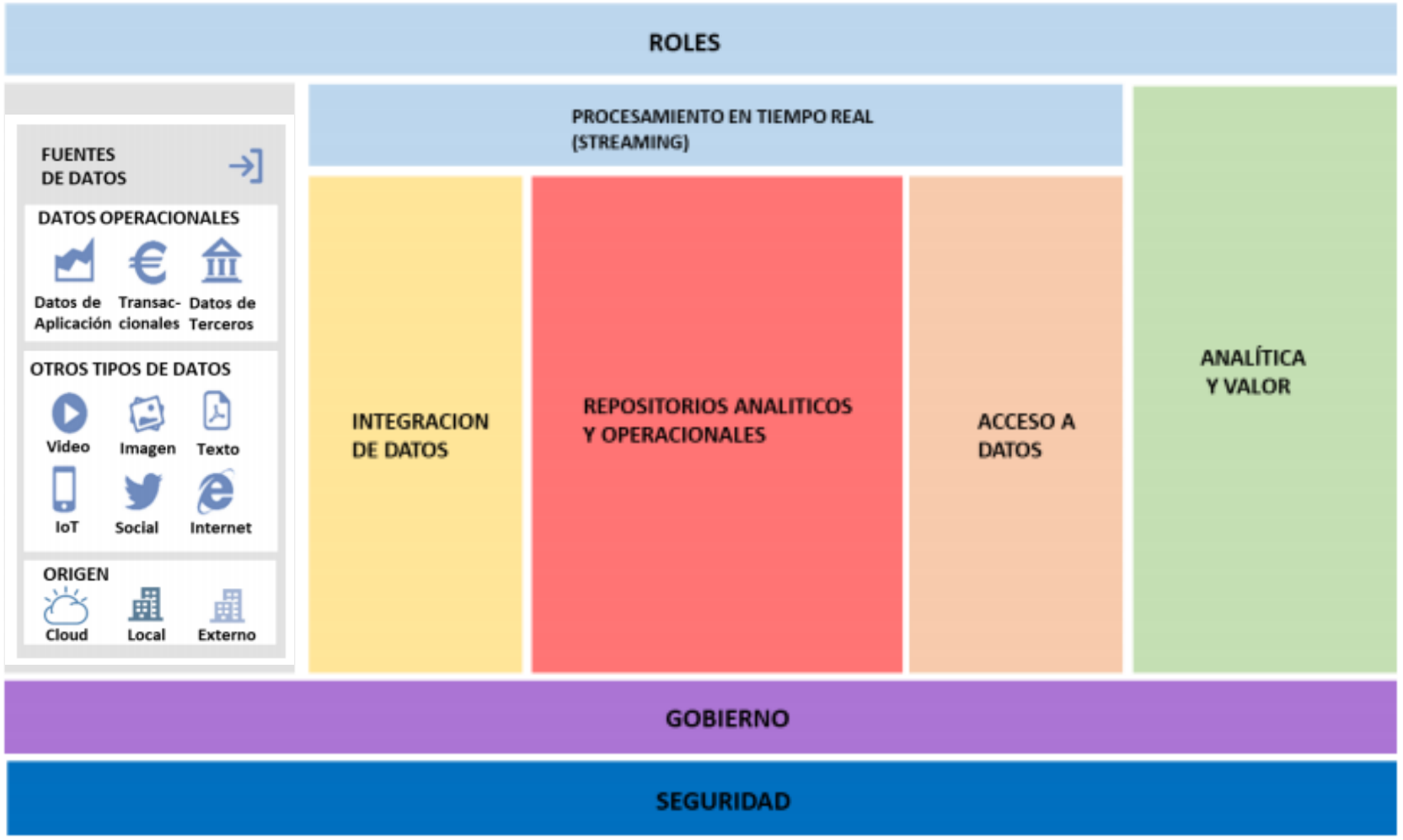
Cuenca Inteligente - Río Santa Lucía
Monitoreo de parámetros medioambientales de la cuenca del río Santa Lucía, con información online e histórica, generada por diferentes organismos del Estado (MVOTMA, MGAP, MIEM, InUMet, OSE y Antel)

Arquitecturas referencia

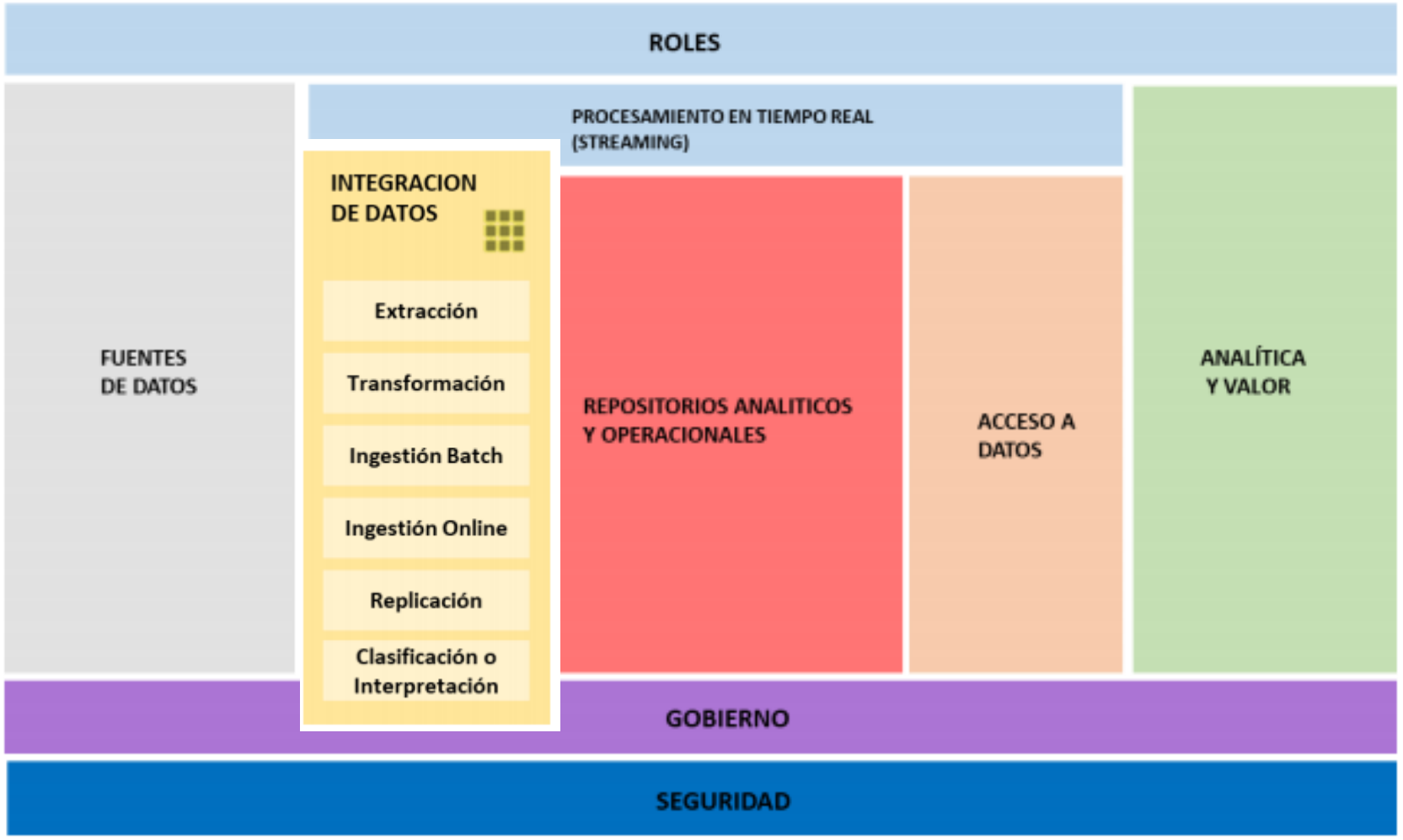
Arquitectura referencia



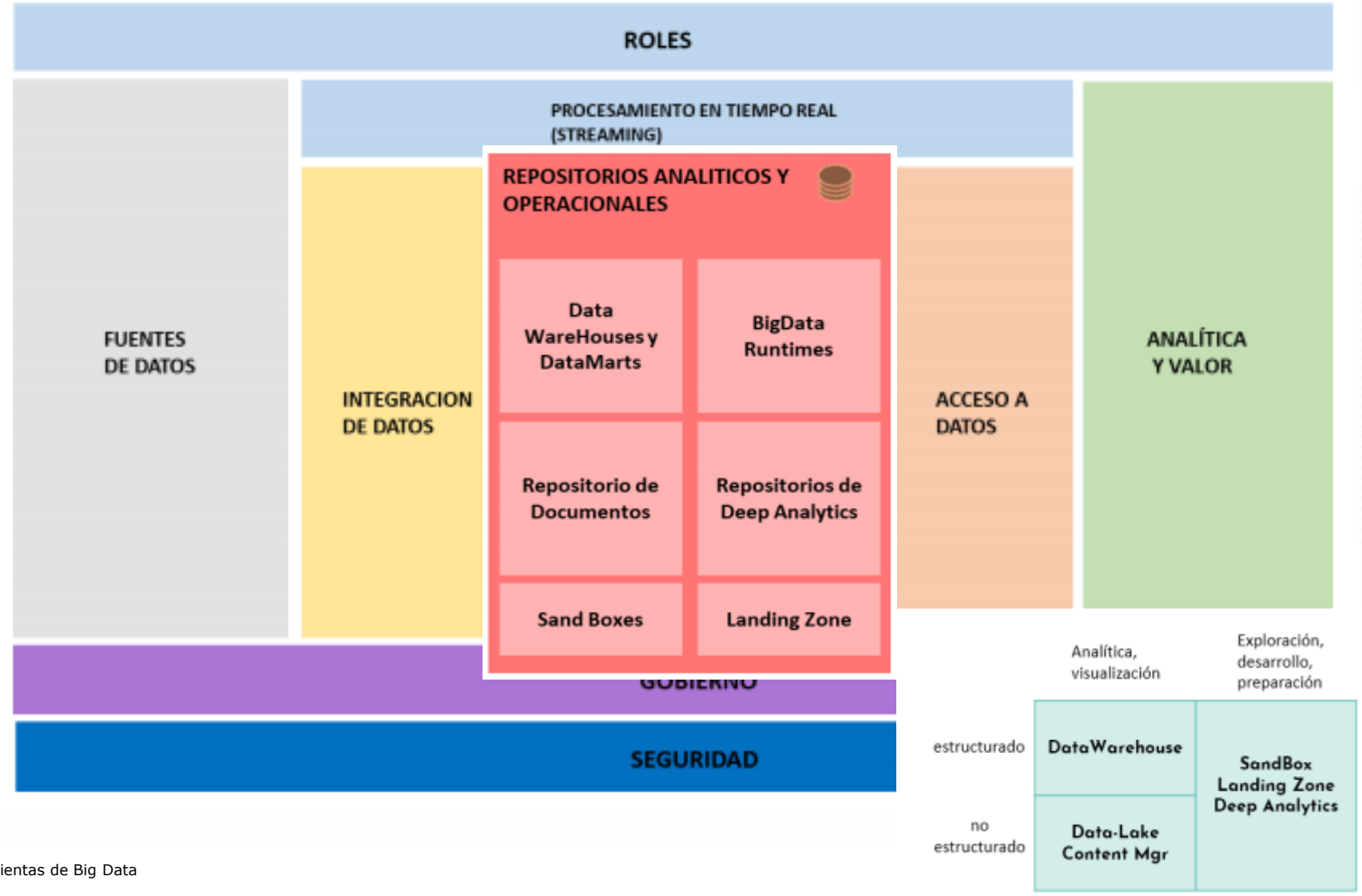
Arquitectura referencia



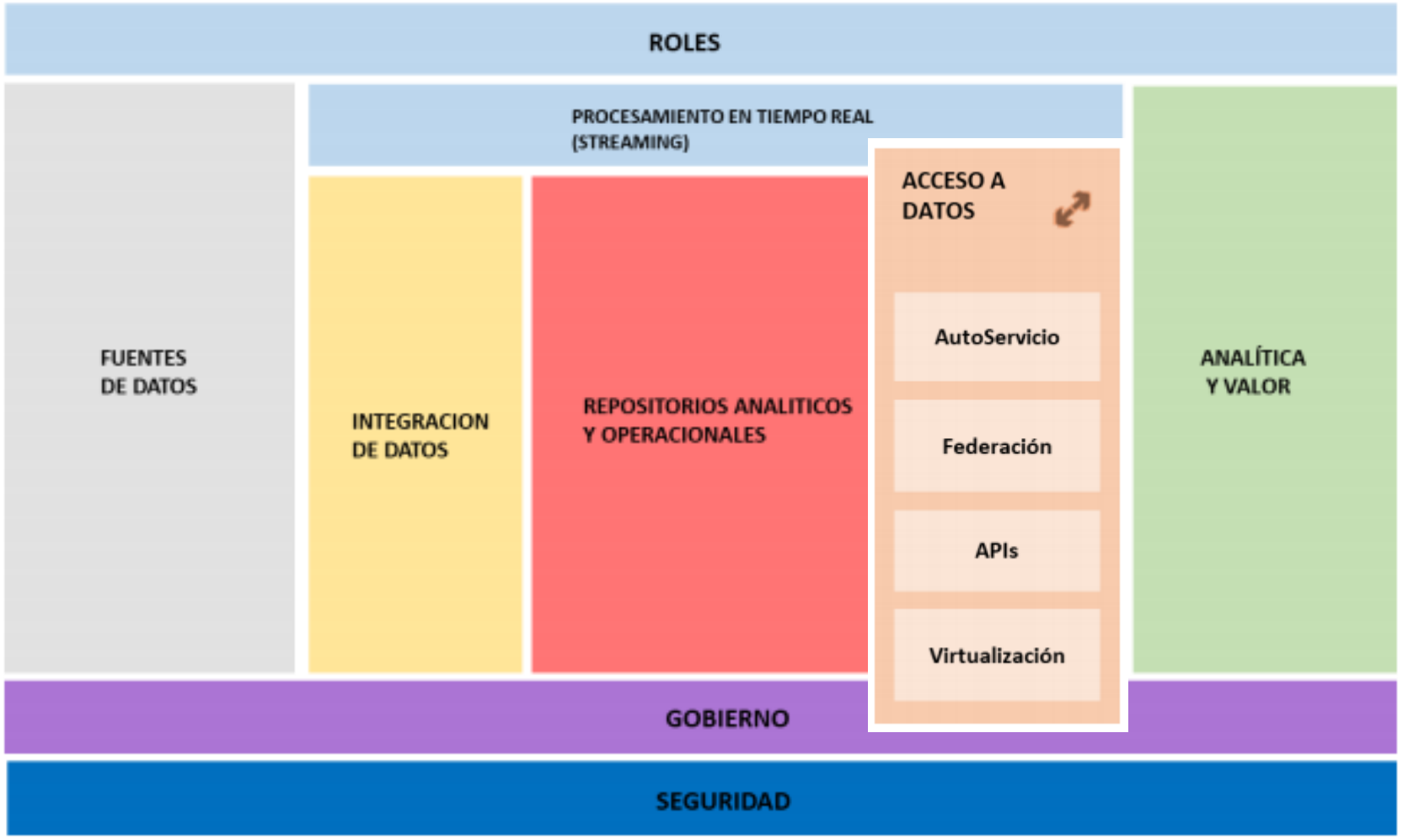
Arquitectura referencia



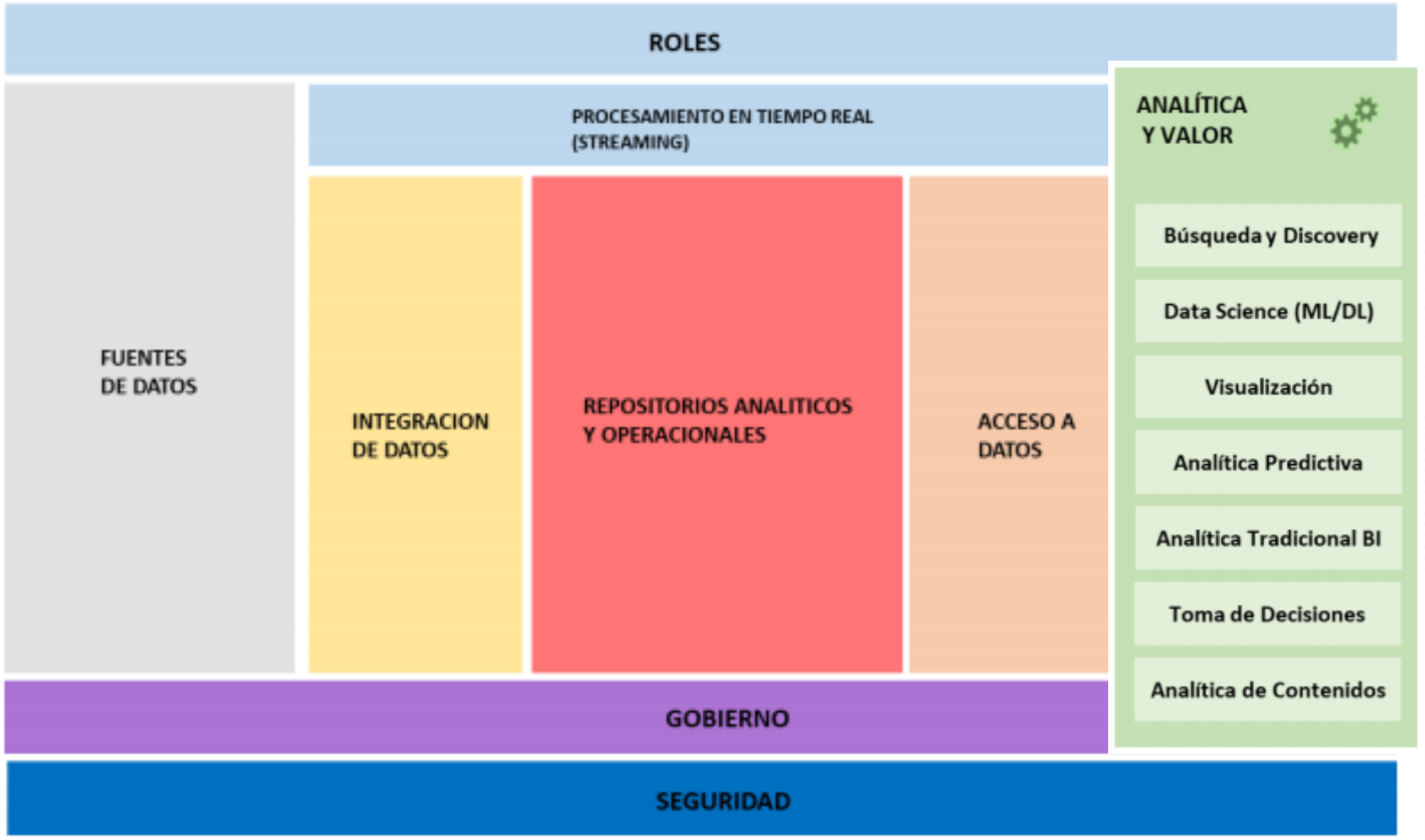
Arquitectura referencia



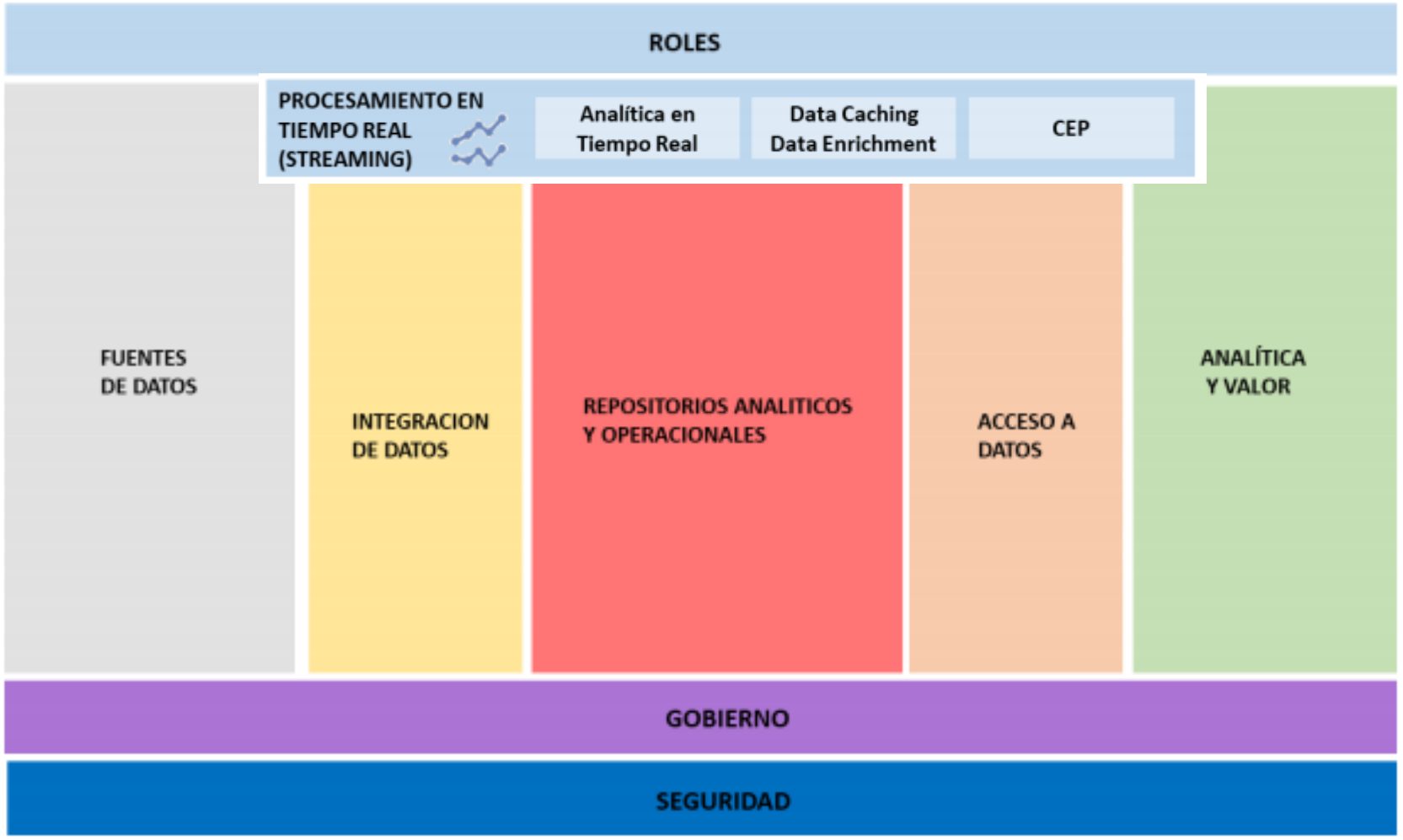
Arquitectura referencia



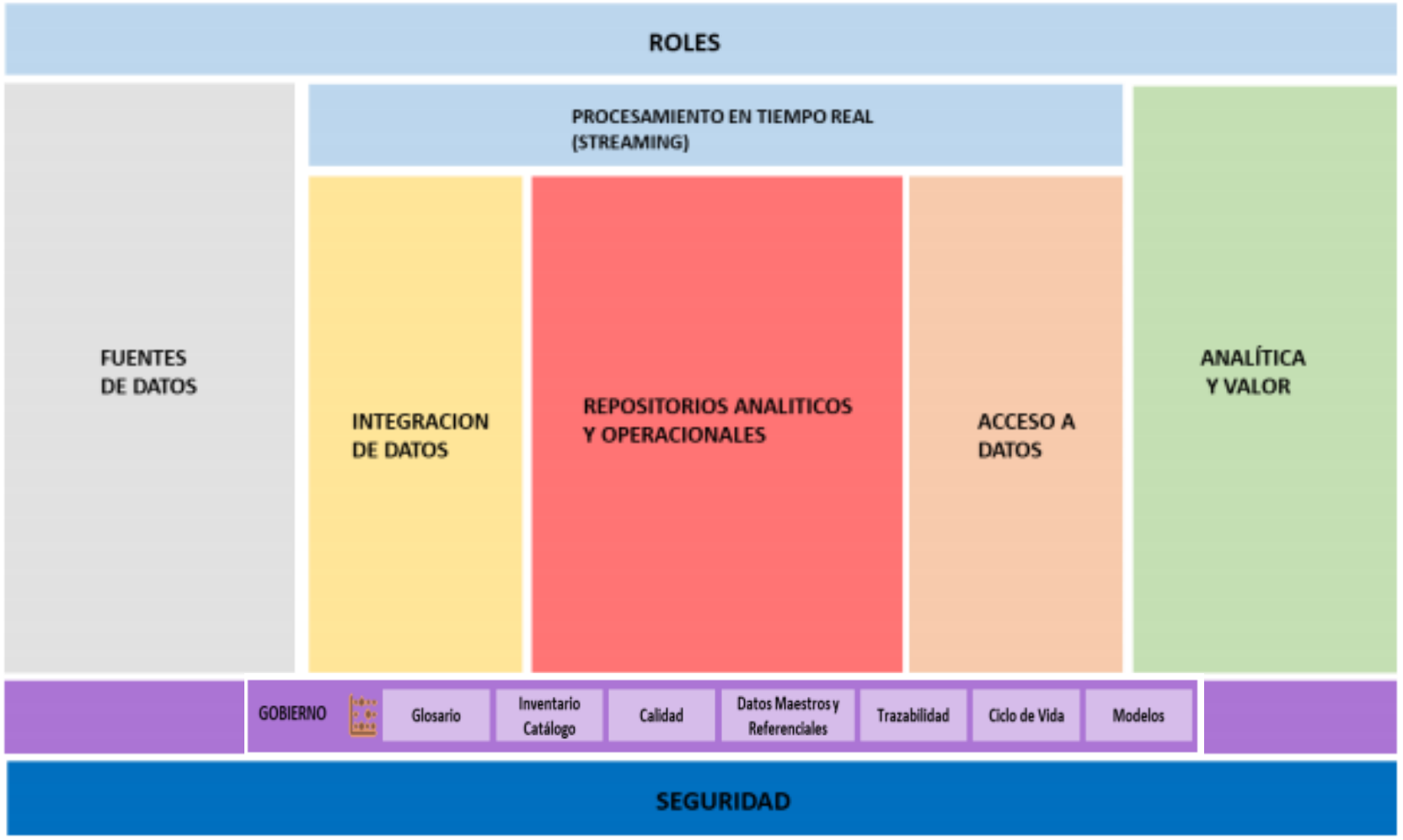
Arquitectura referencia



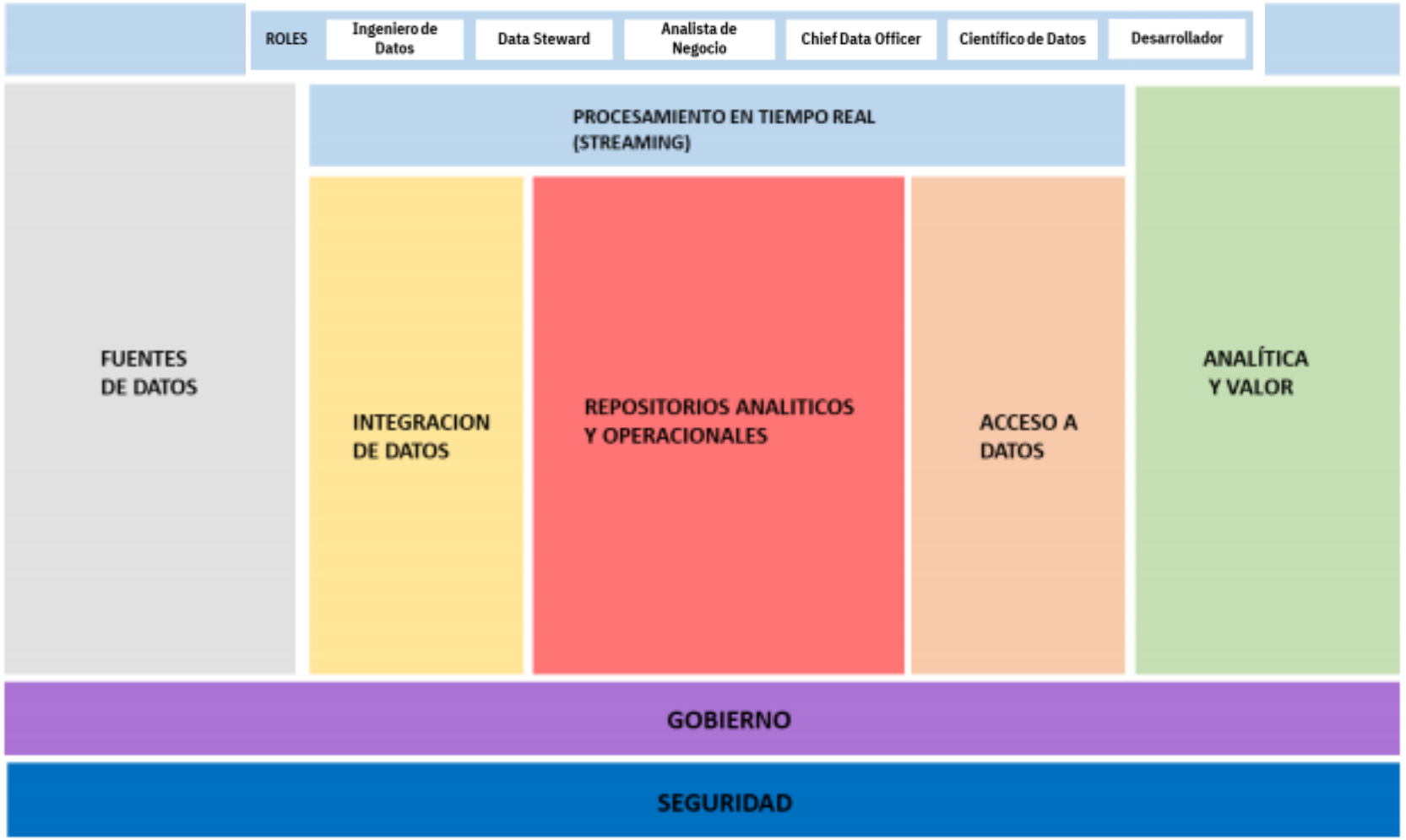
Arquitectura referencia



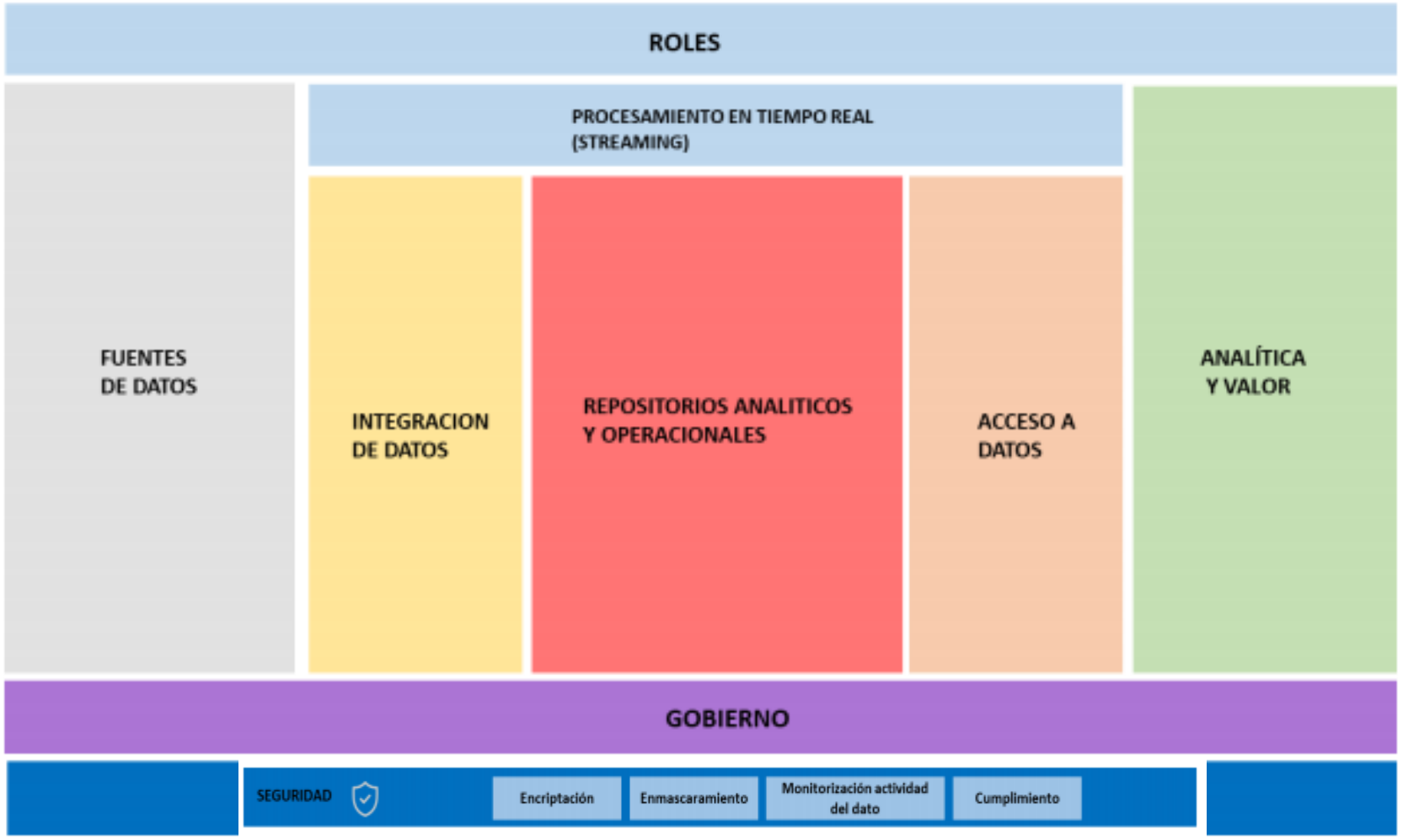
Arquitectura referencia



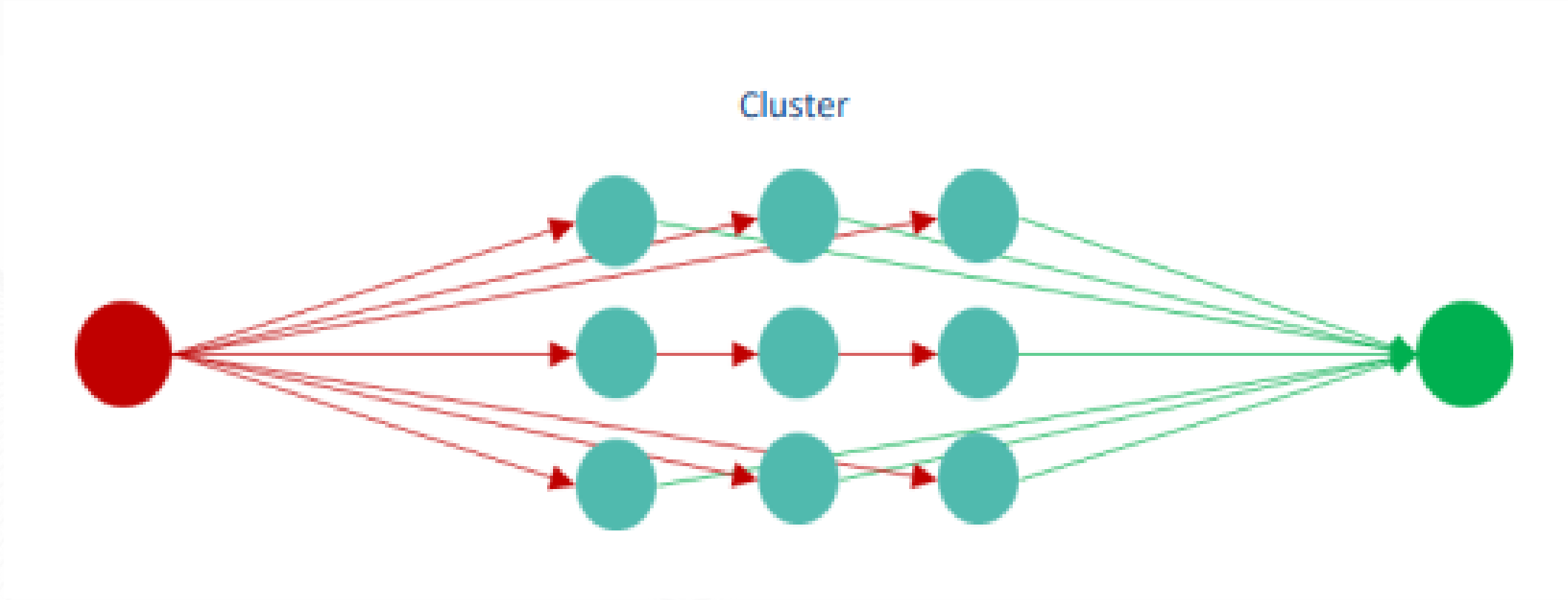
Arquitectura referencia



Arquitectura referencia



Computación distribuida



Arquitecturas de Hadoop y su ecosistema de herramientas

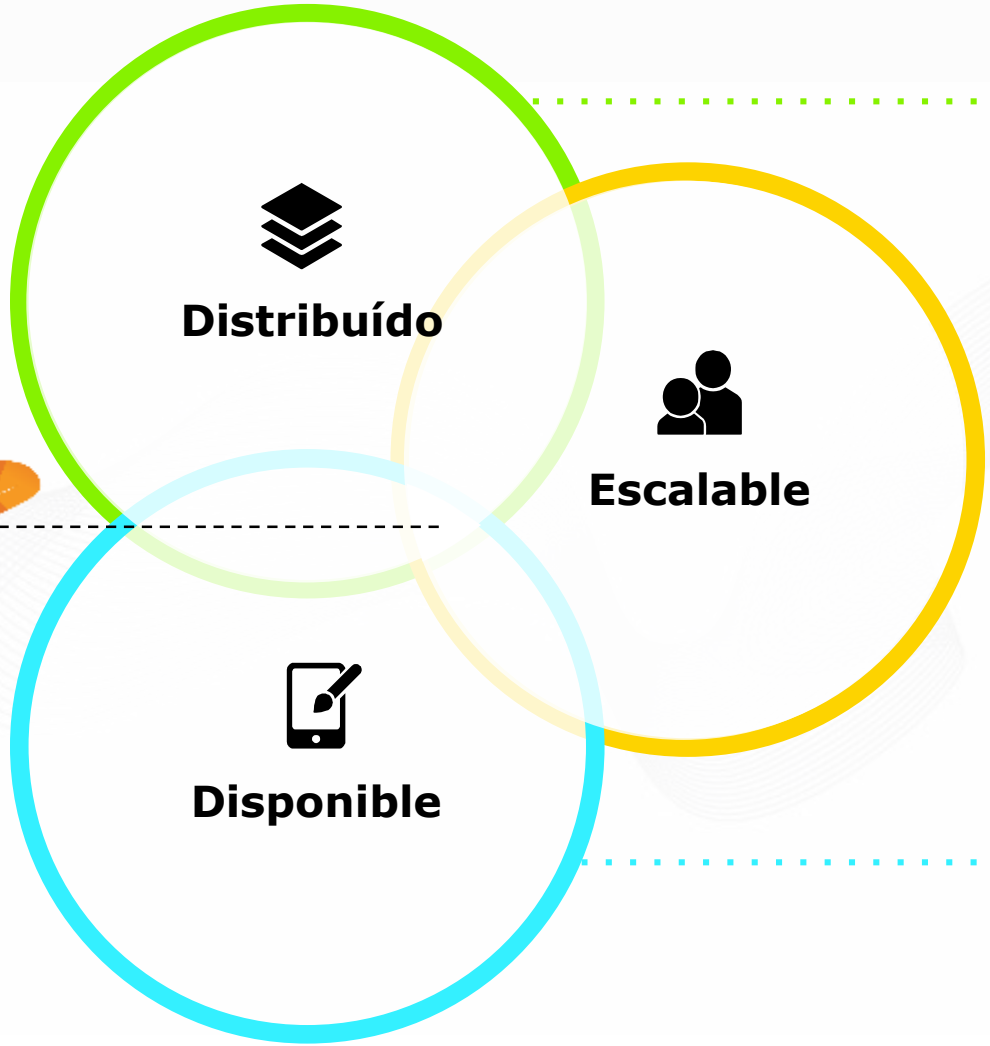
Herramientas

¡Tecnología disponible y **open source**! ¿qué más pedir?

Arquitecturas de Hadoop y su ecosistema de herramientas

Apache Hadoop: almacenamiento y cómputo

Apache Hadoop



Distribuído

Procesamiento distribuido de grandes datasets.

Escalable

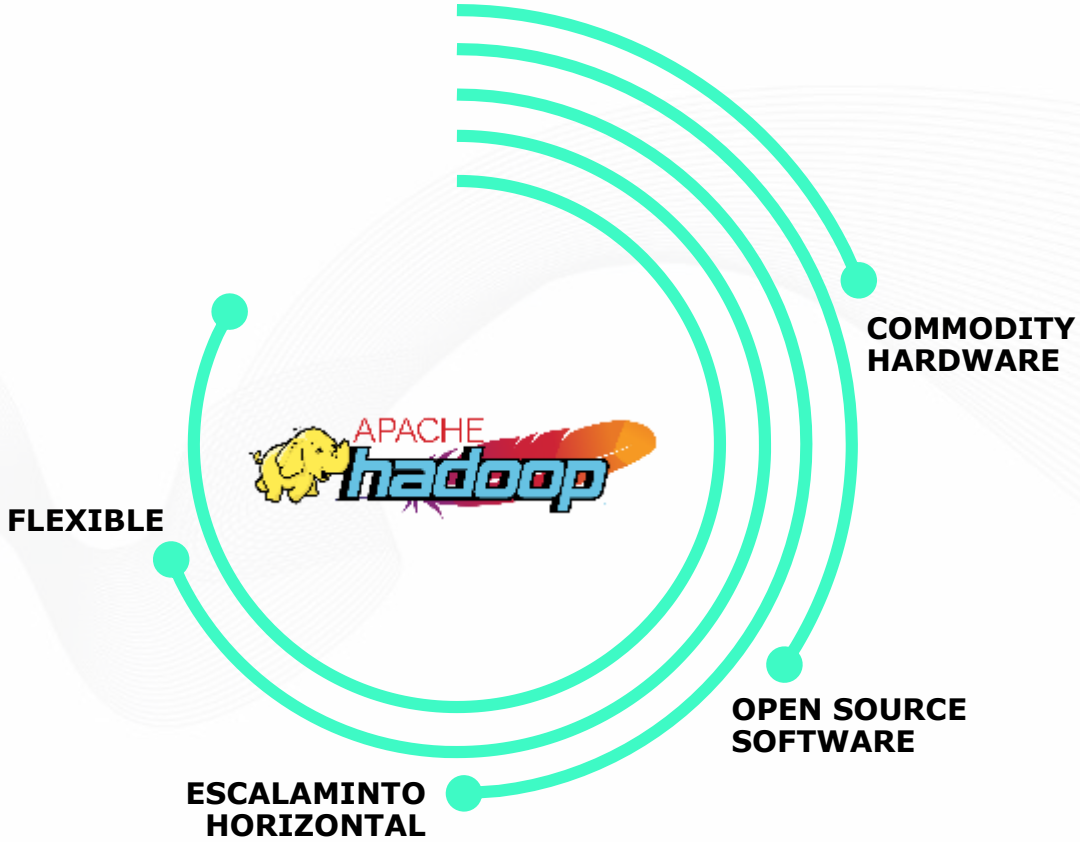
Diseñado para escalar a miles de máquinas.

Disponible

Provee un servicio de alta disponibilidad en un cluster de máquinas.

Apache Hadoop

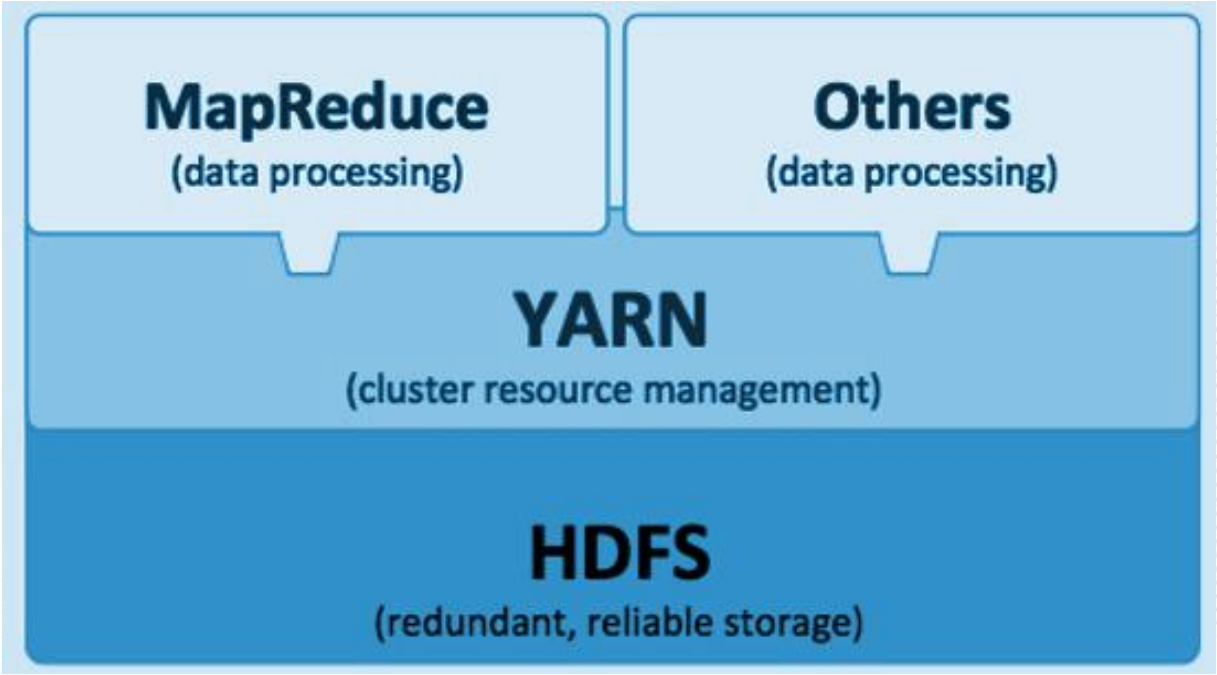
La tecnología es open source y está al **alcance** de la mano de **cualquier** organización.



Hadoop CORE = Almacenamiento + Cómputo

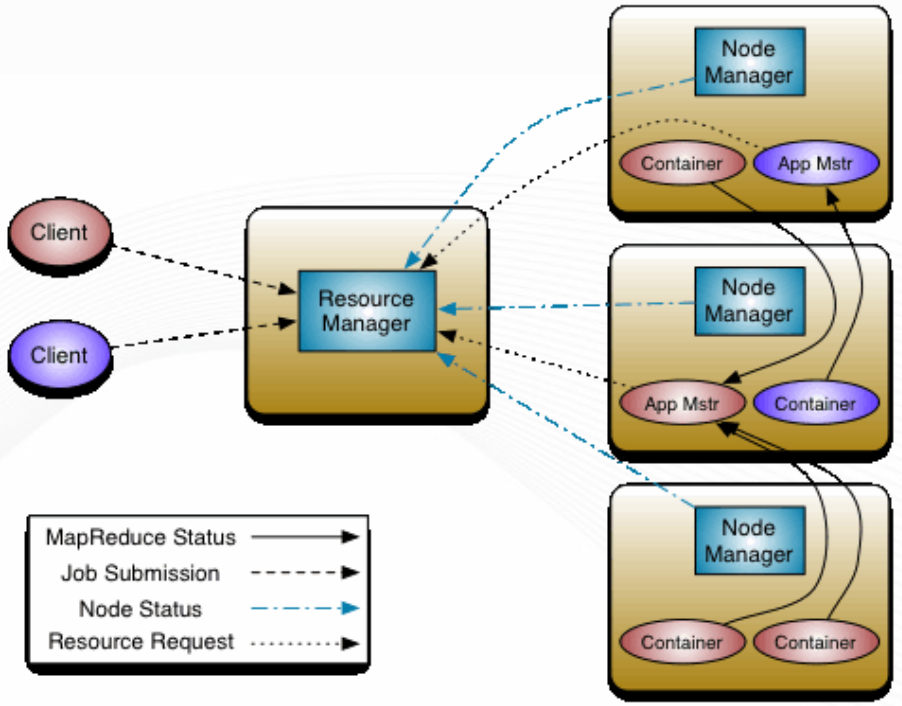
HDFS
Sistema de almacenamiento distribuido

YARN
Administrador de recursos del cluster



Arquitectura de YARN

- ➔ Muchas aplicaciones, una sola plataforma
- ➔ Soporta acceso y procesamiento multi-tenant
- ➔ Cada nodo procesa los bloques locales
- ➔ Batch y real-time



Arquitectura de HDFS

Name Nodes

- Almacena todos los metadatos necesarios para construir el sistema de ficheros a partir de los datos que almacenan los datanodes
- Se tienen nodos primarios y secundarios para la alta disponibilidad
- El failover puede ser manual o automático
- Sisetema de Quorum Journaling

Data Nodes

- Gestiona el almacenamiento de los datos
- Los archivo se divide en uno o más bloques y estos bloques se almacenan en un conjunto de DataNodes
- Realizan la creación, eliminación y replicación de bloques mediante instrucciones del NameNode
- Se utiliza Rack awareness para la tolerancia a fallas

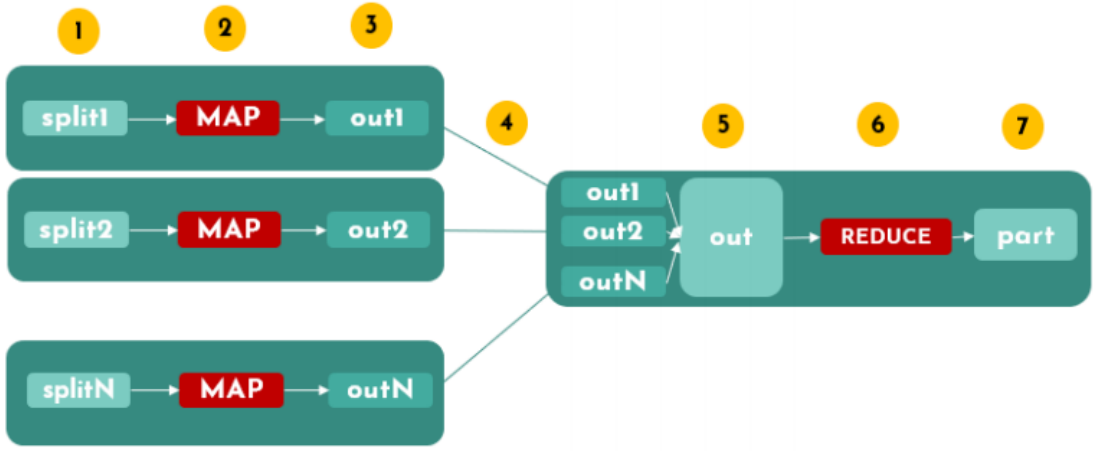
Procesamiento en paralelo con datos distribuidos

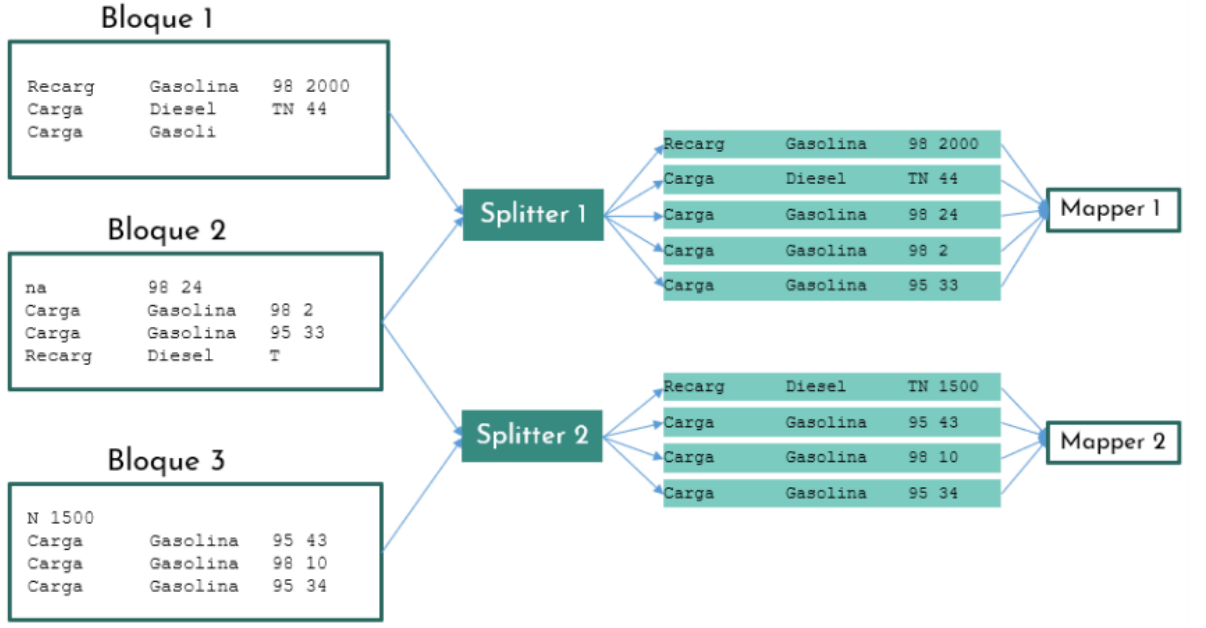

Paralelismo...
¡Distribuir para **maximizar** la eficiencia!

Procesamiento en paralelo con datos distribuidos

MapReduce

MapReduce


1
2
3
split1
MAP
out1
split2
MAP
out2
splitN
MAP
outN
4
5
6
7
out1
out2
outN
out
REDUCE
part


Bloque 1
Recarg Gasolina 98 2000
Carga Diesel TN 44
Carga Gasoli
Bloque 2
na 98 24
Carga Gasolina 98 2
Carga Gasolina 95 33
Recarg Diesel T
Bloque 3
N 1500
Carga Gasolina 95 43
Carga Gasolina 98 10
Carga Gasolina 95 34
Splitter 1
Splitter 2
Recarg Gasolina 98 2000
Carga Diesel TN 44
Carga Gasolina 98 24
Carga Gasolina 98 2
Carga Gasolina 95 33
Recarg Diesel TN 1500
Carga Gasolina 95 43
Carga Gasolina 98 10
Carga Gasolina 95 34
Mapper 1
Mapper 2

MapReduce

Mapper 1:

Diesel TN: **-44 litros**

Gasolina 98: **+1974 litros**

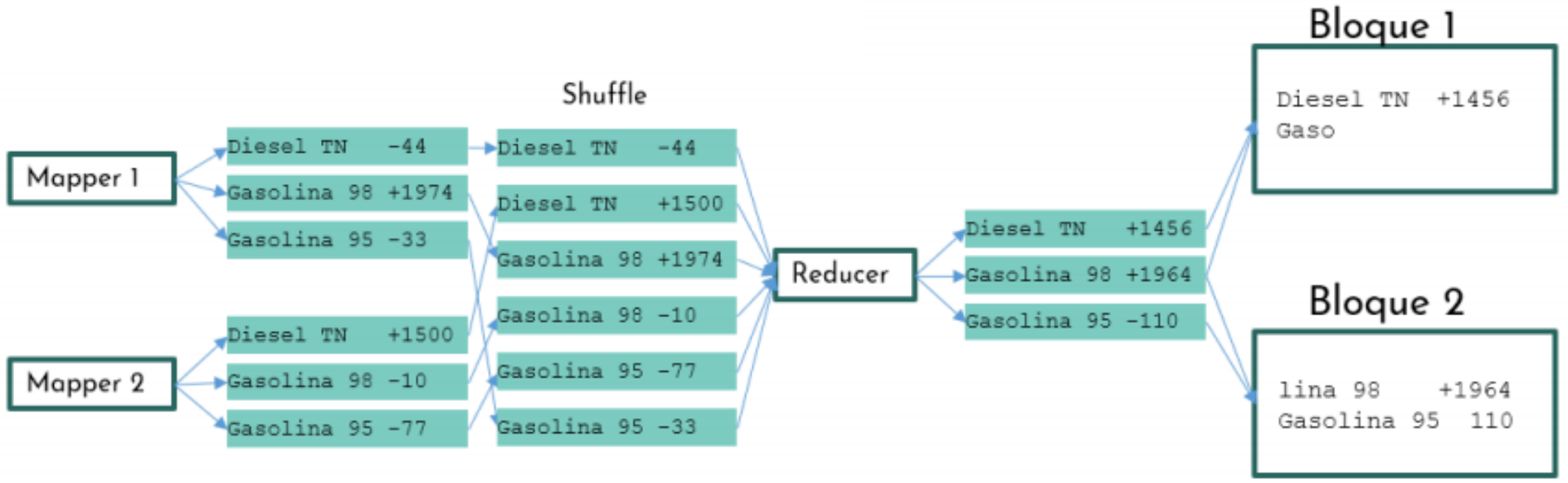
Gasolina 95: -33 litros

Mapper 2:

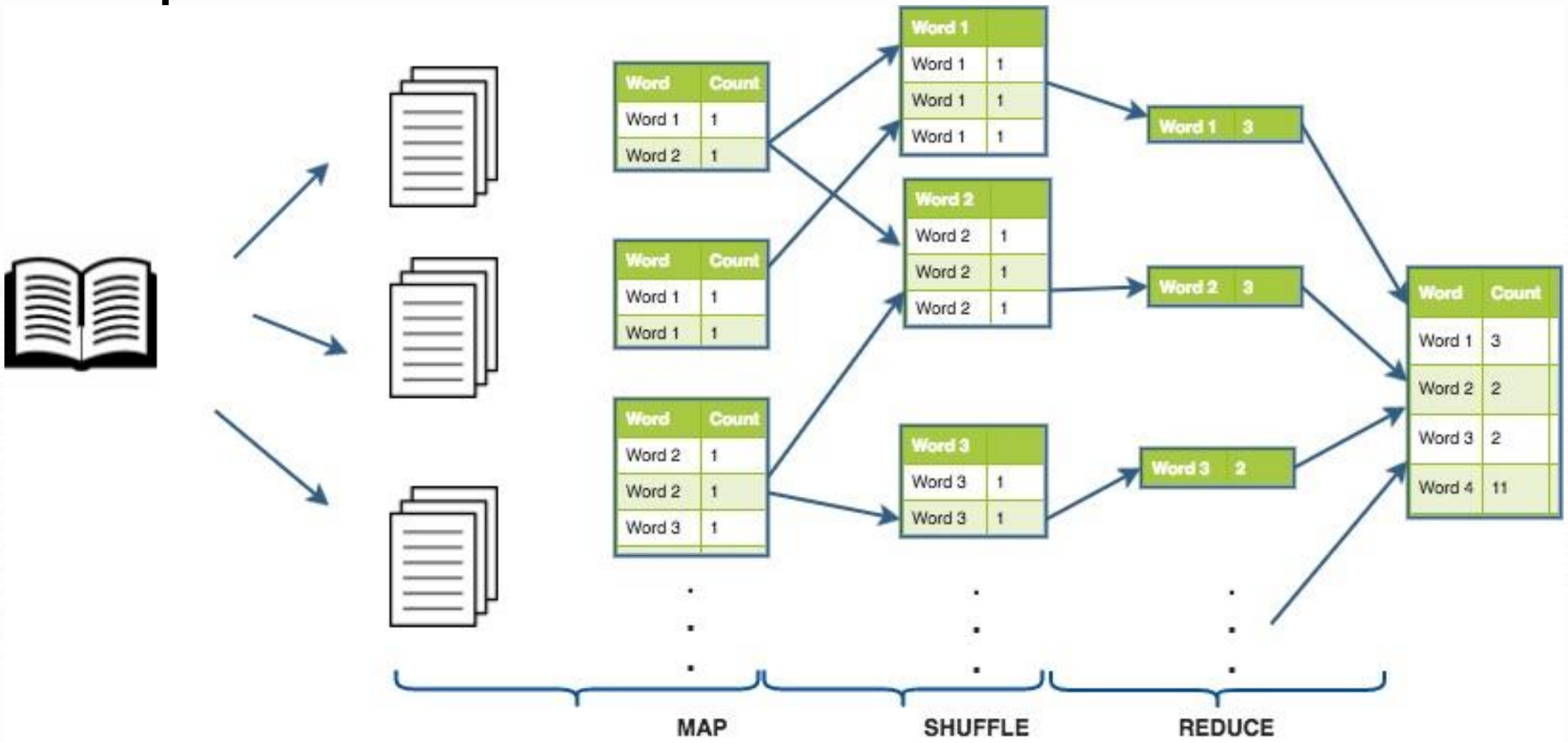
Diesel TN: +1500 litros

Gasolina 98: -10 litros

Gasolina 95: -77 litros

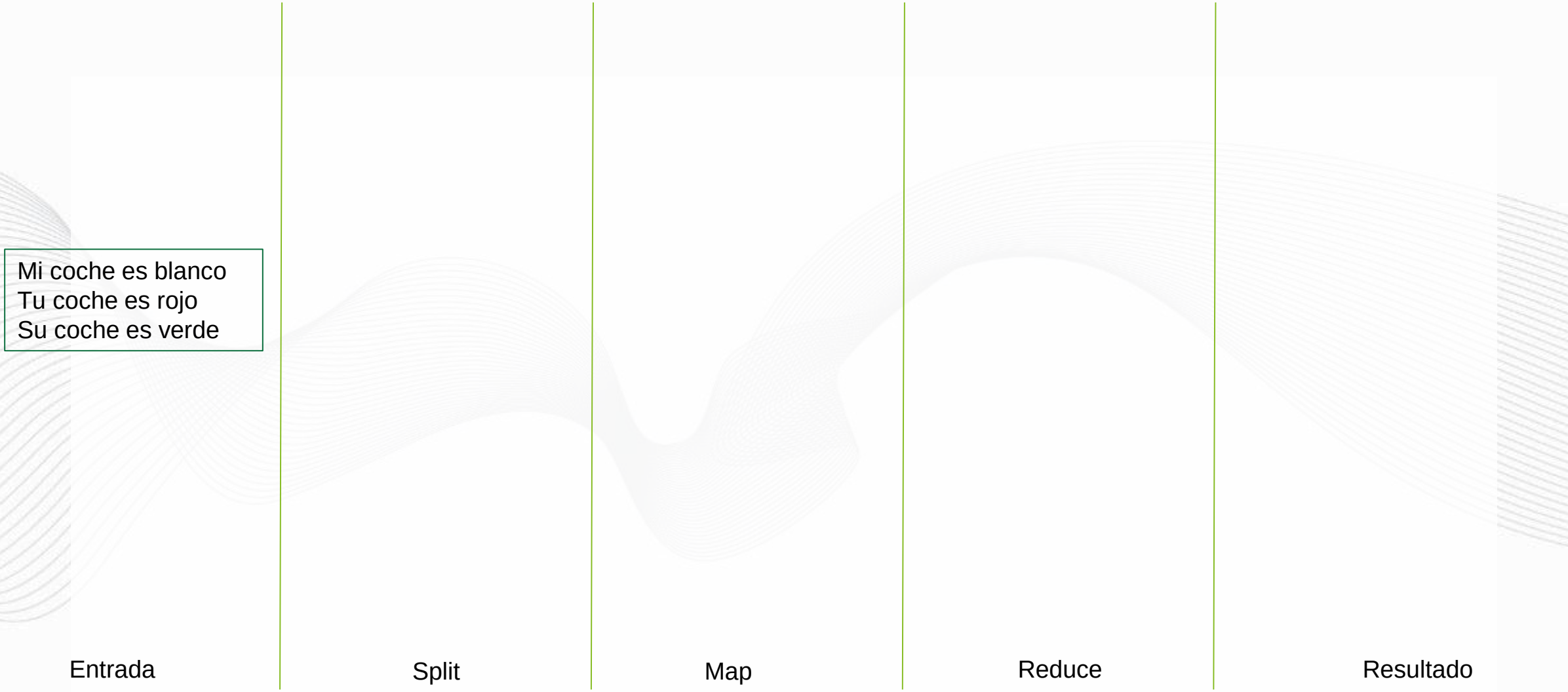


MapReduce

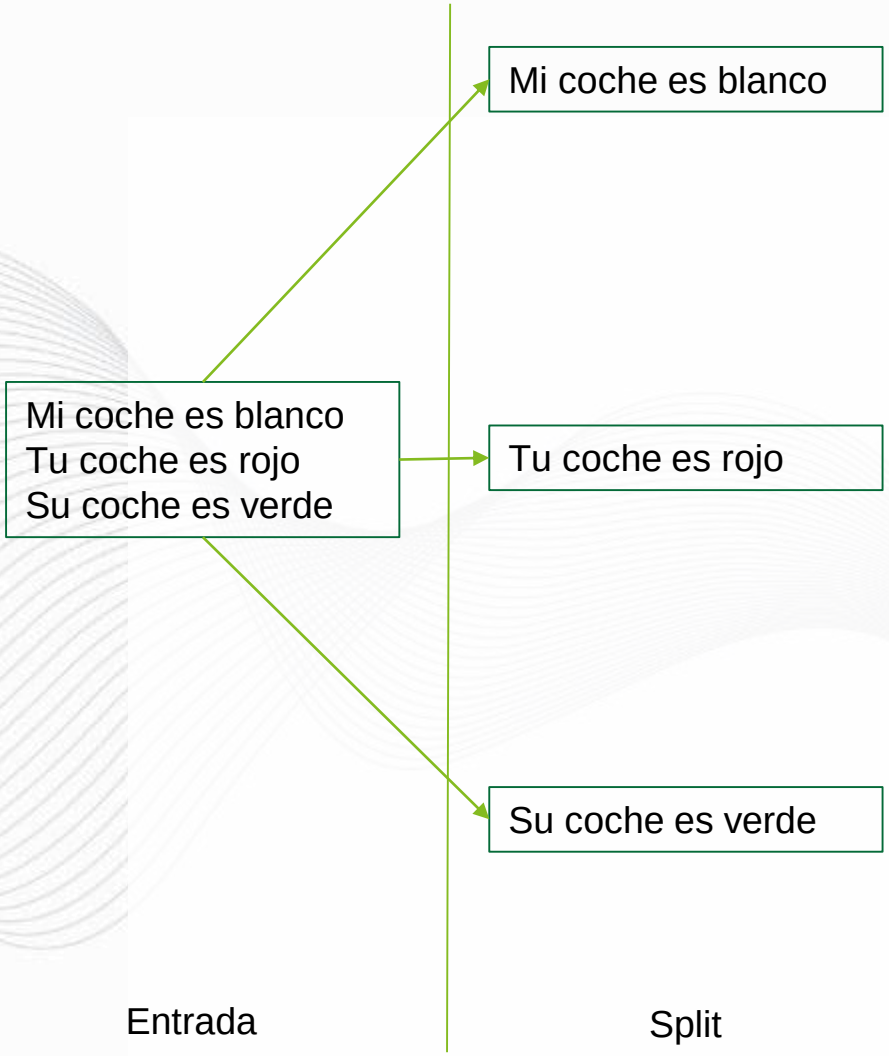

Word Count
Word 1 1
Word 2 1
Word Count
Word 1 1
Word 1 1
Word Count
Word 2 1
Word 2 1
Word 3 1
Word 1
Word 1 1
Word 1 1
Word 1 1
Word 2
Word 2 1
Word 2 1
Word 2 1
Word 3
Word 3 1
Word 3 1
Word 1 3
Word 2 3
Word 3 2
Word Count
Word 1 3
Word 2 2
Word 3 2
Word 4 11
MAP
SHUFFLE
REDUCE

Ejemplo MapReduce

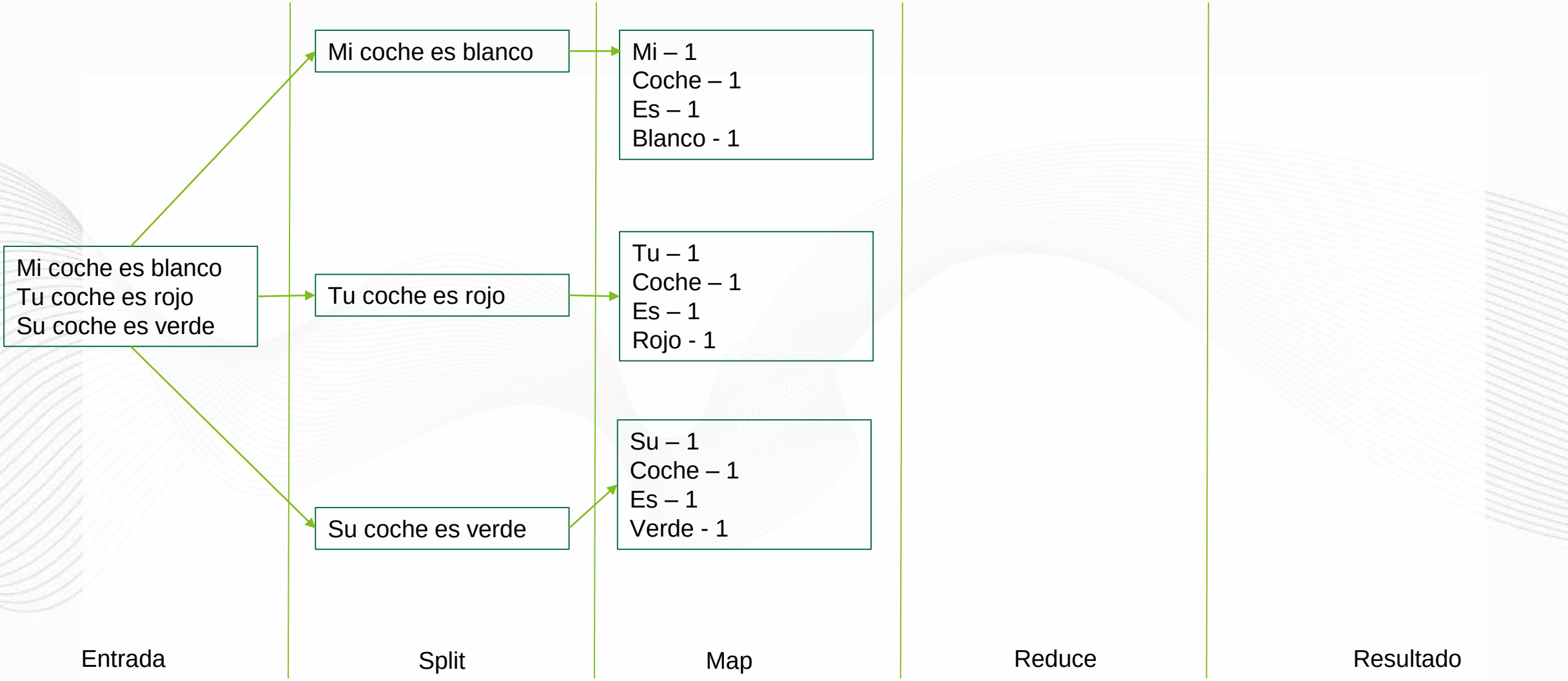
Ejemplo MapReduce : Contar palabras



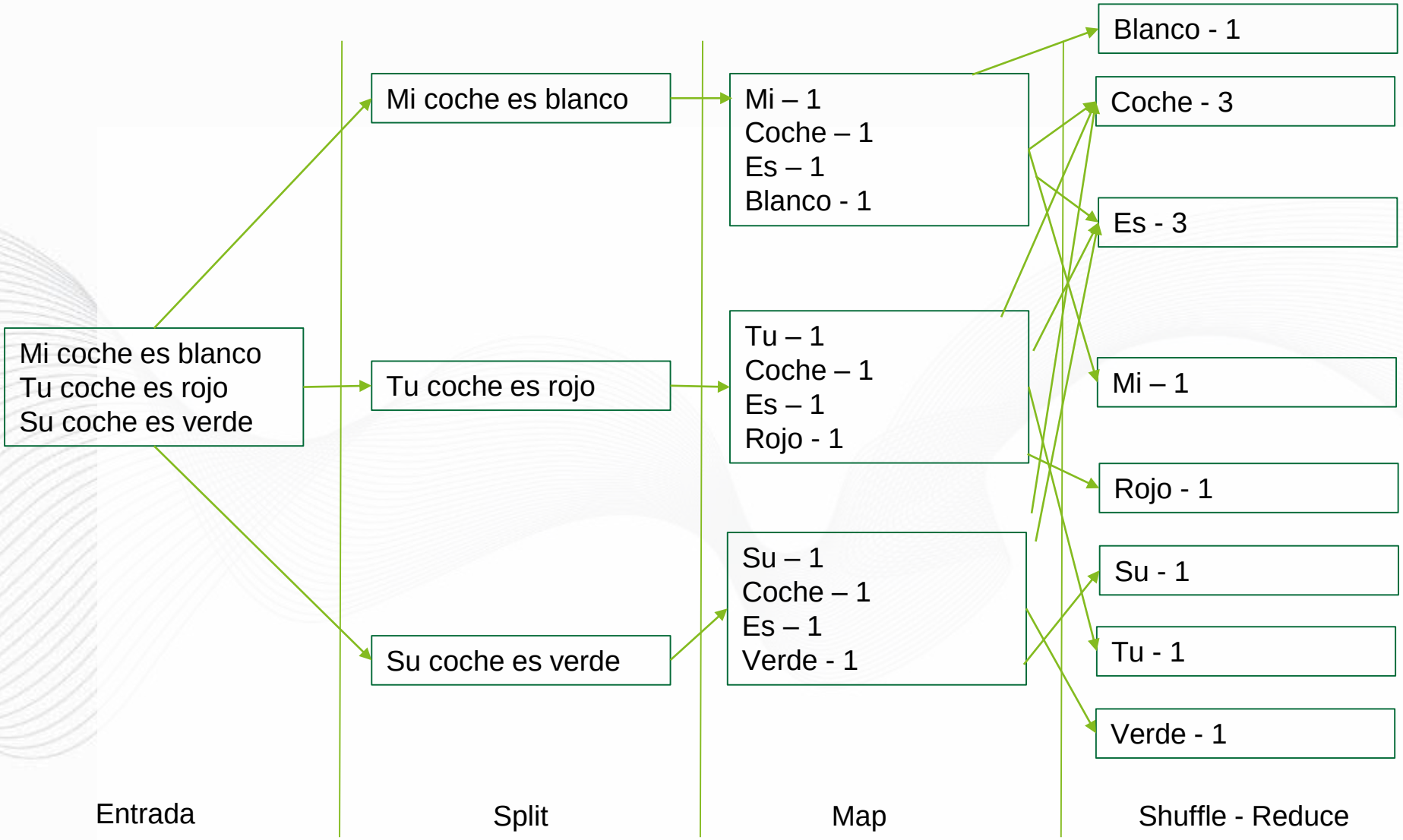
Ejemplo MapReduce : Contar palabras



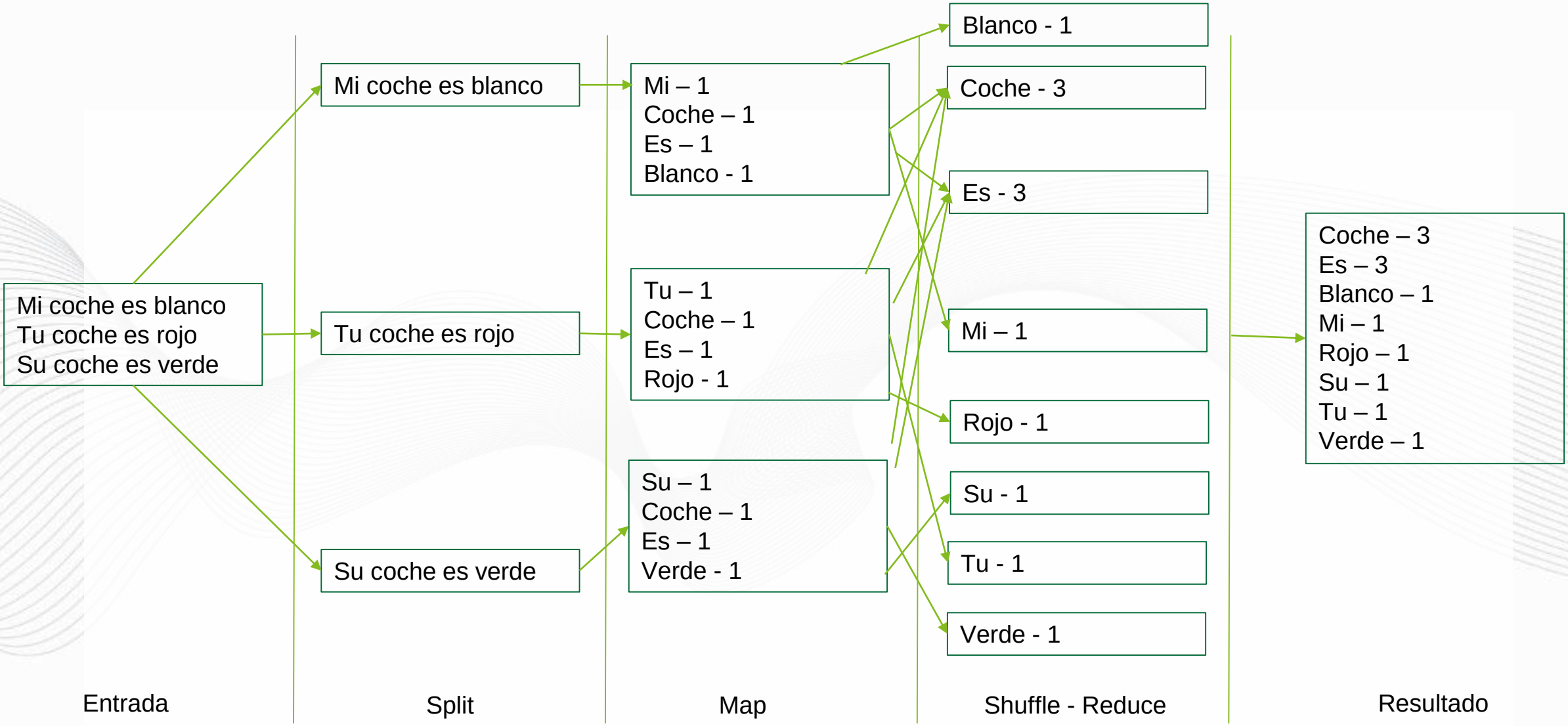
Ejemplo MapReduce : Contar palabras



Ejemplo MapReduce : Contar palabras



Ejemplo MapReduce : Contar palabras

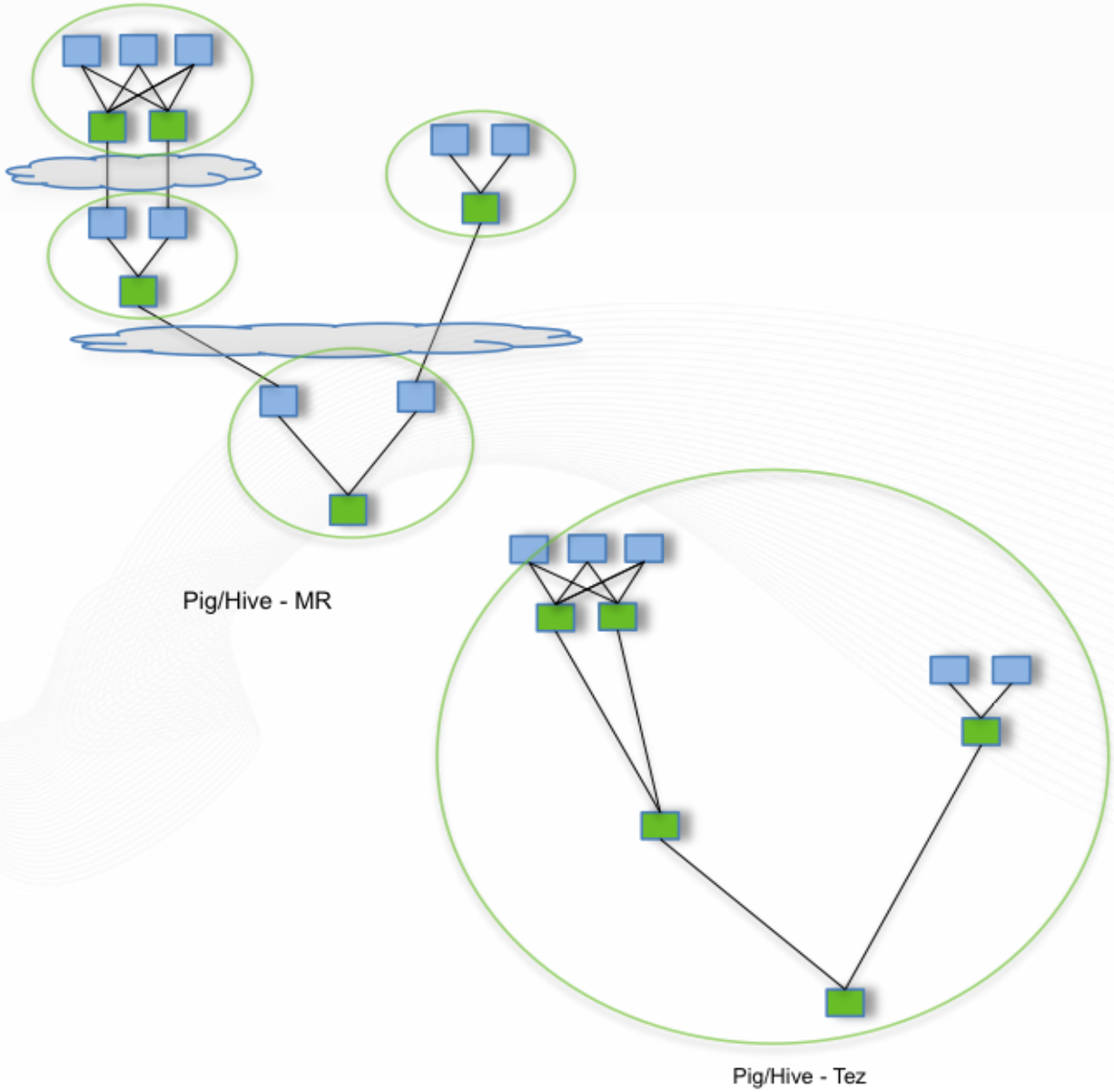


Procesamiento en paralelo con datos distribuidos

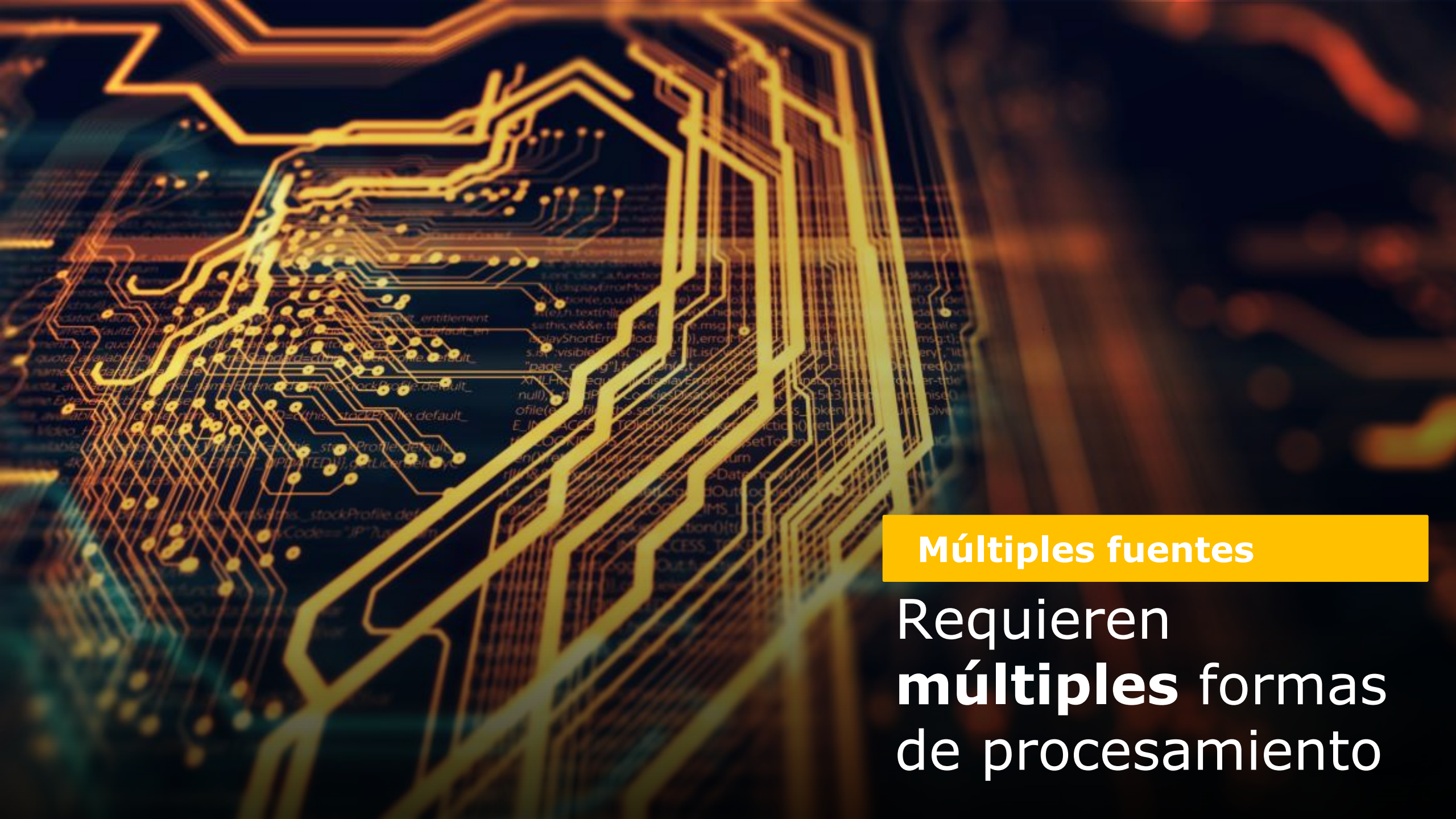
Tez

Tez

- Calcula un grafo dirigido acíclico con las tareas a ejecutar
- Múltiples trabajos de MR se transforman en un solo trabajo de Tez
- Evita escrituras intermedias al HDFS



Ingesta y procesamiento de datos

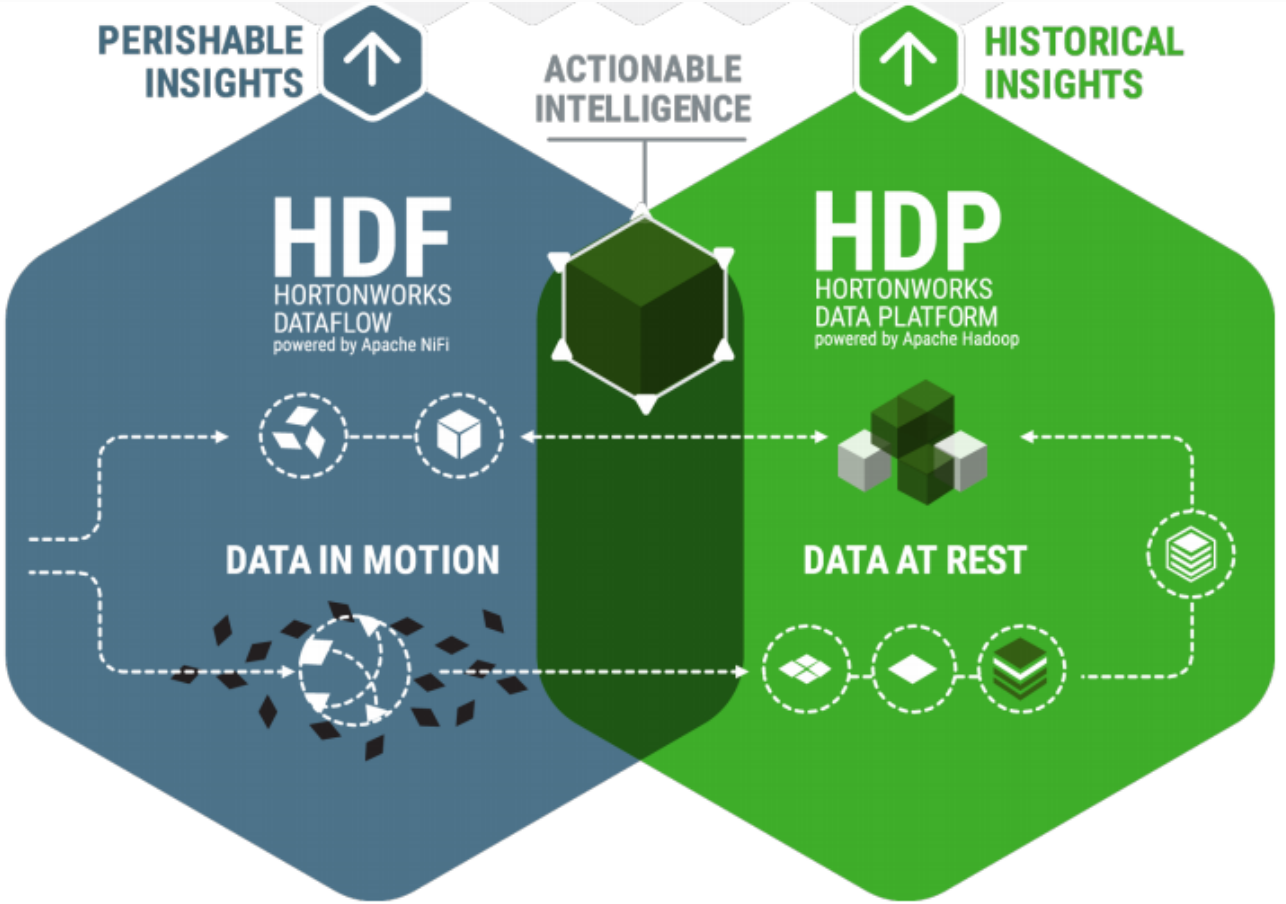


Múltiples fuentes

Requieren **múltiples** formas de procesamiento

Batch vs Real Time

Dos formas de **operar**, una misma forma de inteligencia.



Ingesta y procesamiento de datos

Herramientas de procesamiento batch

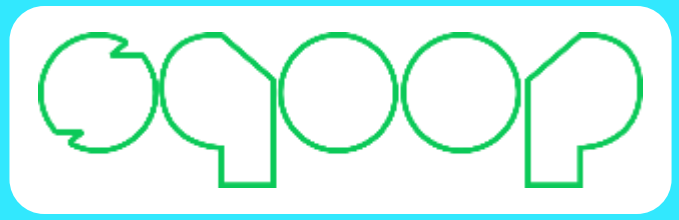
Principales características

- Motor de procesamiento de grandes volúmenes de datos distribuidos sobre la plataforma.
- In-memory.
- APIs: Streaming, SQL, ML y más.
- Empaqueta los libraries para cada invocación, pueden haber entornos o versiones diferentes.
- YARN Ready.

Ejemplo Spark

Ejemplo Spark : Contar palabras

```
# Leer el fichero origen
f = sc.textFile("/var/log/README")

# Definir las transformaciones map y reduce
wc = f.flatMap(lambda x: x.split(' ')).map(lambda x: (x, 1)).reduceByKey(lambda x,y: x+y)

# Trigger action to execute transformations and collect results
wc.collect()
```



Principales características

- Permite transferir grandes volúmenes de datos de forma eficiente entre Apache Hadoop y datos estructurados como bases de datos relacionales.
- Soporta cargas incrementales de datos.
- Proporciona multitud de conectores como FTP, JDBC, Kafka, Kite, SFTP.
- Permite definir jobs que puedan ser ejecutados.

Principales características

- Permite modelar un datawarehouse sobre Hadoop.
- Estándar para hacer consultas SQL sobre Hadoop.
- HiveQL: similar a SQL.
- Permite particiones.
- Managed/External tables.
- Las consultas se convierten a MapReduce/Tez.
- Hive CLI/Beeline, JDBC/ODBC/ WebUI.
- Soporta varios backends o “motores”: ORC, Parquet, Text File.
- Que se pueda no quiere decir que se deba.
- Soporta compresión al vuelo.
- La performance depende de la compresión y del Backend.
- Usualmente apuntamos a ORC + Snappy.
- Los discos son más lentos que la CPU.

Comandos HiveQL

MySQL	HiveQL
SELECT from_columns FROM table WHERE conditions;	SELECT from_columns FROM table WHERE conditions;
SELECT * FROM table;	SELECT * FROM table;
SELECT * FROM table WHERE rec_name = “value”;	SELECT * FROM table WHERE rec_name = "value";
SELECT * FROM table WHERE rec1=”value1” AND rec2=”value2”;	SELECT * FROM TABLE WHERE rec1 = "value1" AND rec2 = "value2";
SELECT column_name FROM table;	SELECT column_name FROM table;
SELECT DISTINCT column_name FROM table;	SELECT DISTINCT column_name FROM table;
SELECT col1, col2 FROM table ORDER BY col2;	SELECT col1, col2 FROM table ORDER BY col2;
SELECT col1, col2 FROM table ORDER BY col2 DESC;	SELECT col1, col2 FROM table ORDER BY col2 DESC;
SELECT COUNT(*) FROM table;	SELECT COUNT(*) FROM table;
SELECT owner, COUNT(*) FROM table GROUP BY owner;	SELECT owner, COUNT(*) FROM table GROUP BY owner;
SELECT MAX(col_name) AS label FROM table;	SELECT MAX(col_name) AS label FROM table;
SELECT pet.name, comment FROM pet, event WHERE pet.name = event.name;	SELECT pet.name, comment FROM pet JOIN event ON (pet.name = event.name);

Ingesta y procesamiento de datos

Herramientas de procesamiento real time

Principales características

- Sistema distribuido de ETL.
- Procesamiento de datos en tiempo real.
- Interface web, potente e intuitiva, que permite diseñar y configurar de forma visual el flujo de datos.
- Variedad de conectores: HDFS, ElasticSearch, FTP, bases de datos SQL, MongoDB,etc.
- Transformación en varios formatos de datos: JSON, XML, Avro, CSV.

Principales características

- Publicación y subscripción: lee y escribe flujos de datos como un sistema de mensajería.
- Almacenamiento: almacena flujos de datos de forma segura en un clúster distribuido, replicado y tolerante a fallas.
- Procesamiento: permite el procesamiento en tiempo real de los streams.

Principales características

- Extensión de la API de Spark
- Procesamiento de flujo de datos en tiempo real de forma escalable, alto rendimiento y tolerante a fallas
- Varias fuentes de datos: Kafka, Flume, Kinesis, or TCP sockets
- Soporta Java, Scala and Python
- Se puede utilizar funciones de alto nivel, algoritmos de ML y procesamiento de grafos
- Los datos procesados se pueden enviar a sistemas de archivos, bases de datos y dashboard

Principales características

- Base de datos columnar para el análisis en tiempo real.
- Sistema distribuido escalable.
- Se pueden ingestar datos en tiempo real y en batch.
- Particionamiento basado en tiempo.
- Sumarización automática al momento de la ingesta.

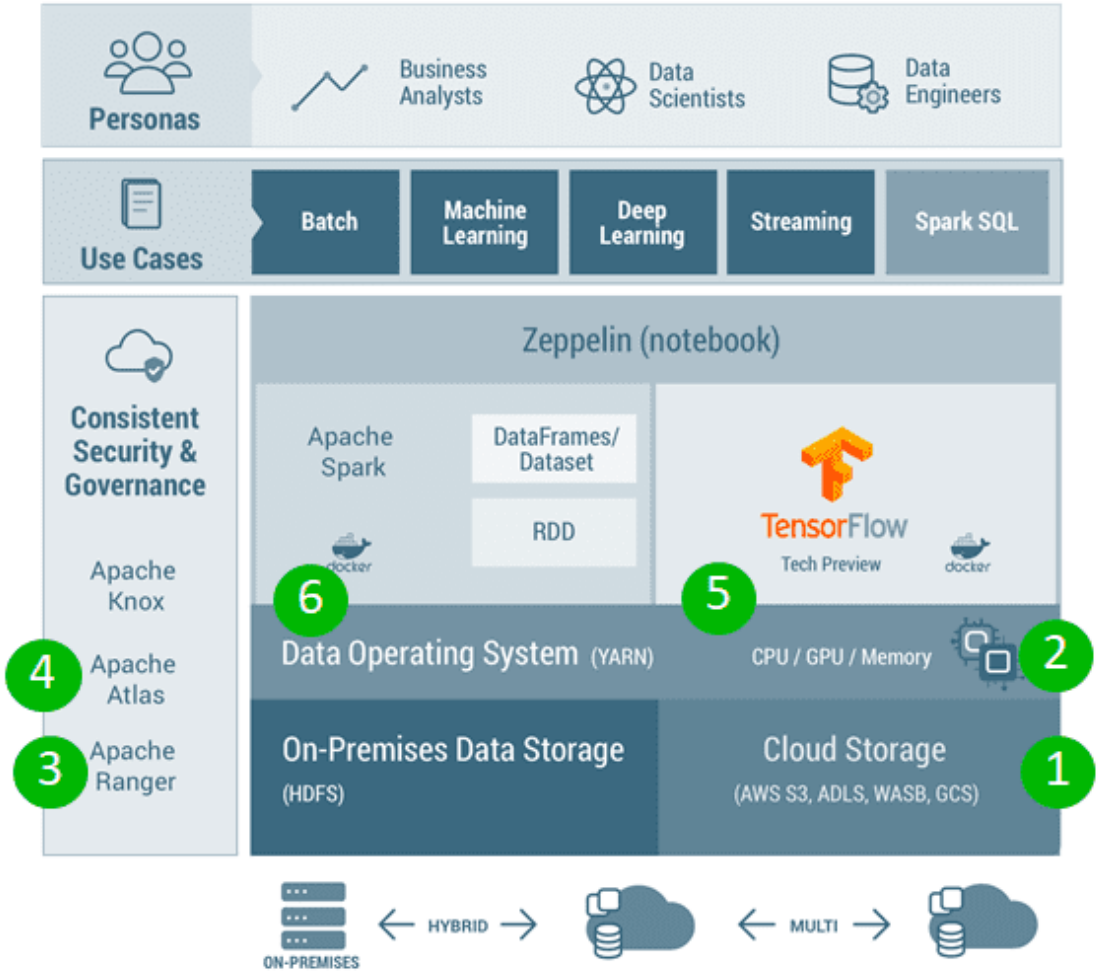
Ejemplo Spark Streaming : Contar palabras

```
1  from pyspark import SparkContext
2  from pyspark.streaming import StreamingContext
3
4  # Creamos un StreamingContext con dos hilos de ejecución y un intervalo batch de 1 segundo
5  sc = SparkContext("local[2]", "NetworkWordCount")
6  ssc = StreamingContext(sc, 1)
7
8  # Creamos un DStream que se conecte al hostname:port, como localhost:9999
9  lines = ssc.socketTextStream("localhost", 9999)
10
11 # Separamos las lineas en palabras
12 words = lines.flatMap(lambda line: line.split(" "))
13
14 # En cada batch contamos las palabras
15 pairs = words.map(lambda word: (word, 1))
16 wordCounts = pairs.reduceByKey(lambda x, y: x + y)
17
18 # Se imprimen los primeros 10 elementos en la consola
19 wordCounts.pprint()
20
21 ssc.start()             # Start the computation
22 ssc.awaitTermination()  # Wait for the computation to terminate
23
24
25
```

Data Science and Engineering Platform en HDP 3.0

Ciencia de datos en HDP 3.0

1. Híbrido y listo para la nube
2. Elastico y escalable
3. Seguridad empresarial
4. Gobernanza integral
5. Deep Learning
6. Agilidad de desarrollo



Ciencia de datos en HDP 3.0

Spark MLlib

- Librería de Spark de Machine Learning
- Algoritmos de ML comunes como clasificación, regresión, clustering y filtrado colaborativo
- Extracción de características, transformación, reducción de dimensionalidad y selección
- Herramientas para construir, evaluar y ajustar ML Pipelines
- Guardar y cargar algoritmos, modelos y pipelines
- Álgebra lineal, estadísticas y manejo de datos

TensorFlow (Tech preview)

- Es una biblioteca de código abierto para ML(Deep Learning)
- Basado en gráficos de flujo de datos computacionales para representar una arquitectura de red neuronal
- Se puede crear y entrenar modelos de ML fácilmente utilizando API intuitivas de alto nivel como Keras

Análisis y visualización de datos

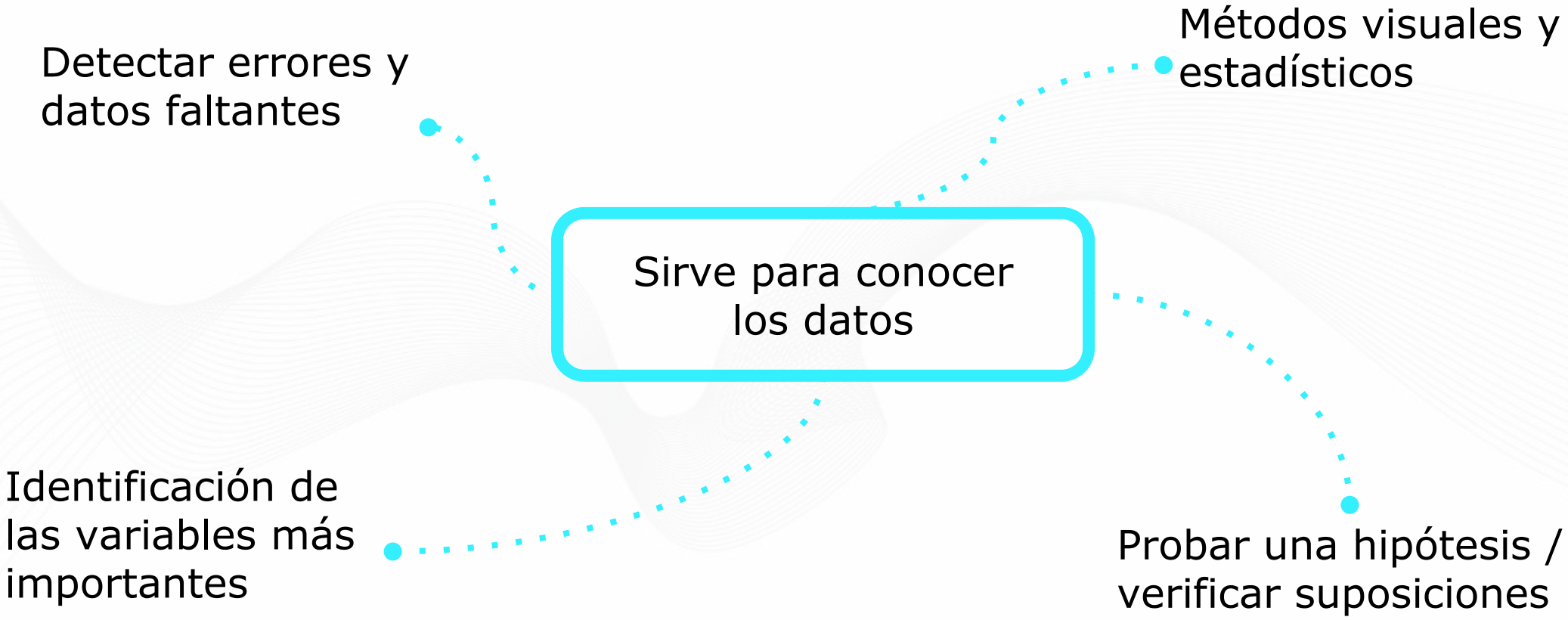
Ver para creer

Buscamos, analizamos… ya es hora de **visualizar**…

Análisis y visualización de datos

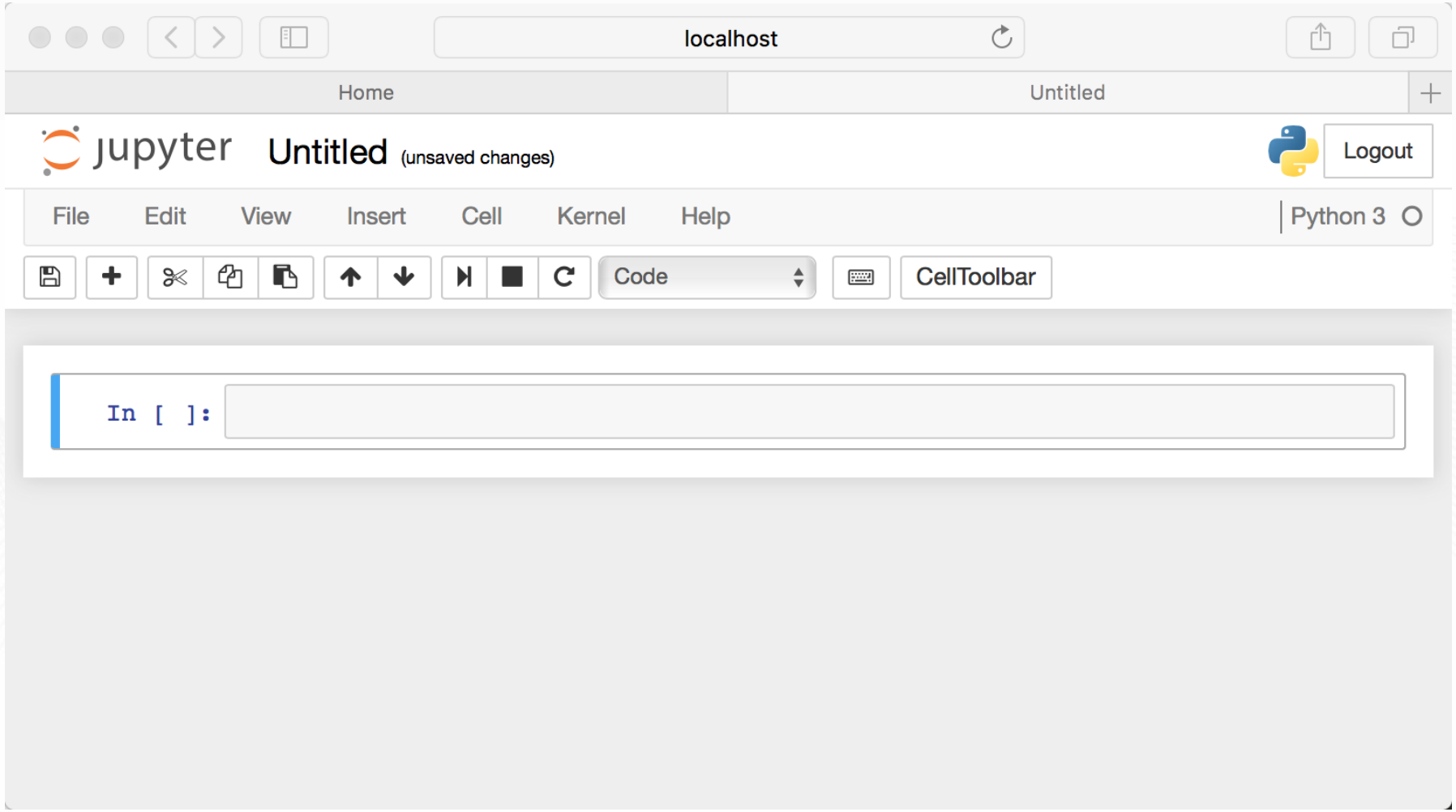
Procesamiento exploratorio en notebooks

Procesamiento exploratorio en notebooks

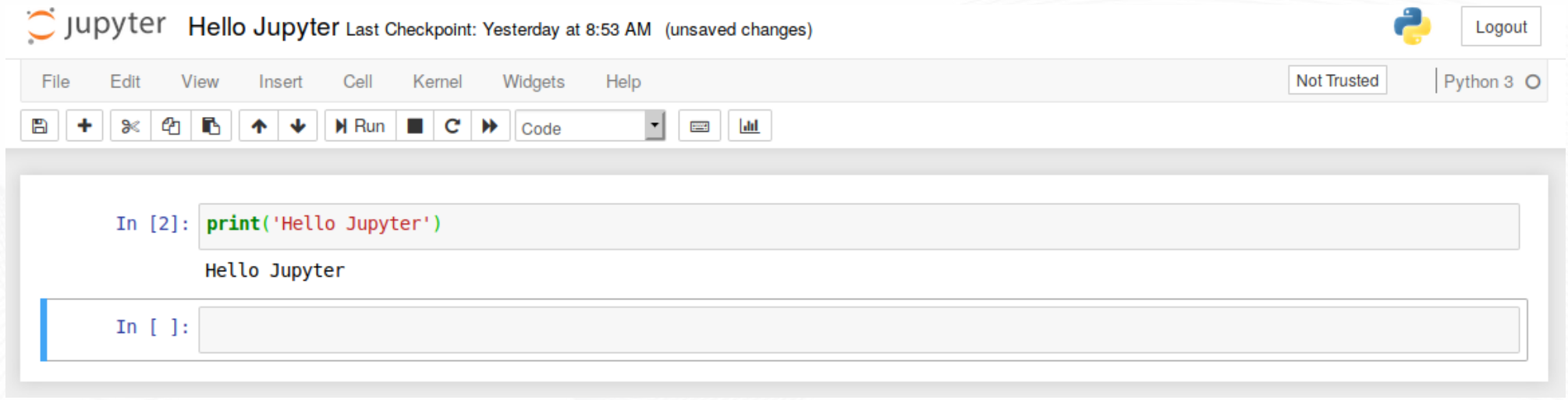


Demostración Notebook

Demostración Notebook



Demostración Notebook



Demostración Notebook

jupyter ipython_galaxy_notebook (unsaved changes)

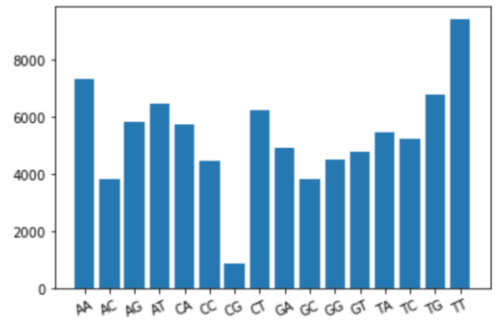
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Code

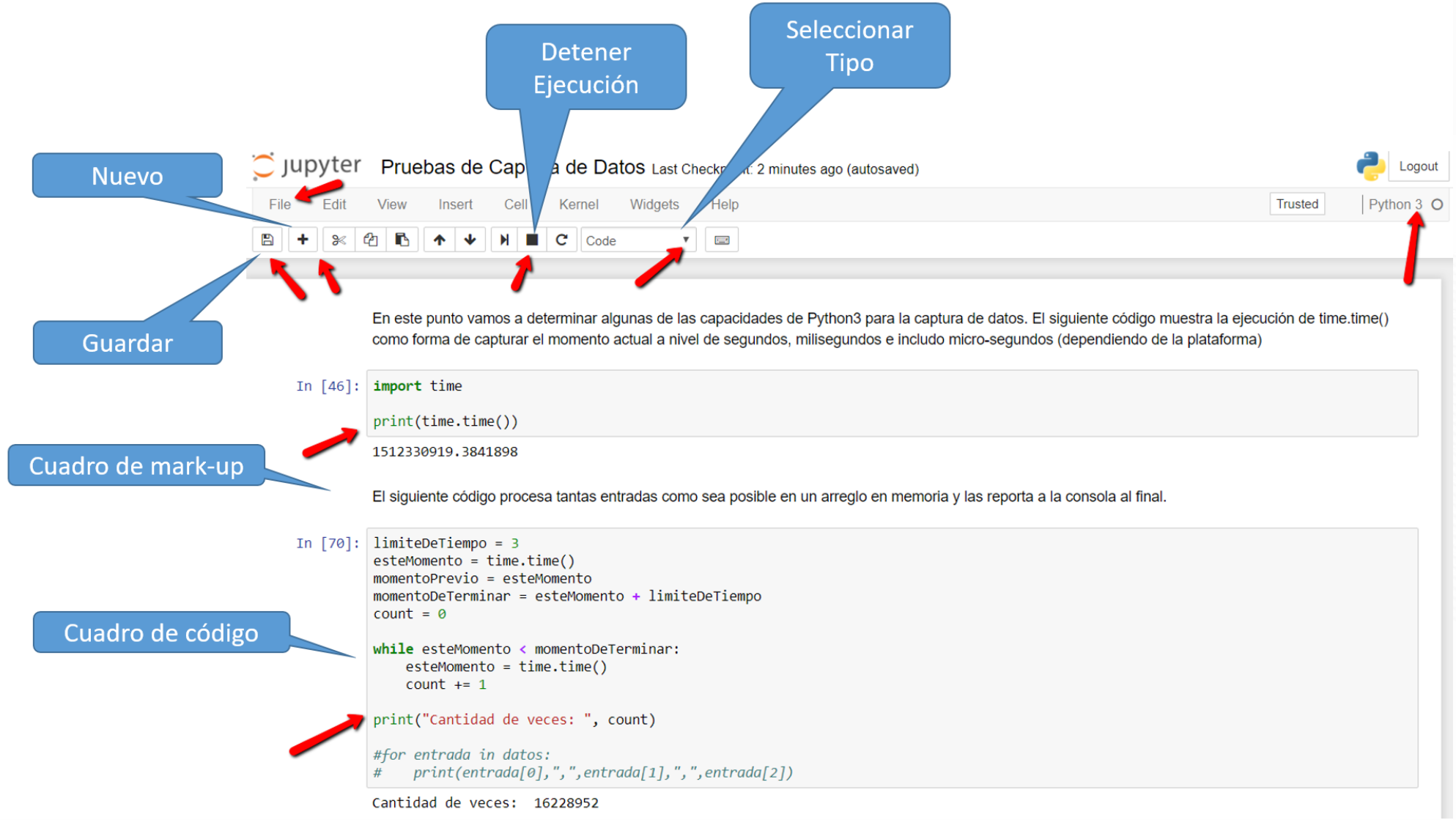
In [4]:
```
import collections

obsrv = dict()

for n in range(len(seq)-1):
    dinuc = seq[n:n+2]
    if not dinuc in obsrv:
        obsrv[dinuc] = 1
    else:
        obsrv[dinuc] += 1

obsrv = collections.OrderedDict(sorted(obsrv.items()))
print(obsrv)
```

```
OrderedDict([('AA', 7316), ('AC', 3823), ('AG', 5796), ('AT', 6444), ('CA', 5734), ('CC', 444
5), ('CG', 884), ('CT', 6210), ('GA', 4894), ('GC', 3795), ('GG', 4475), ('GT', 4763), ('TA',
5435), ('TC', 5210), ('TG', 6771), ('TT', 9384)])
```

In [5]:
```
import matplotlib.pyplot as plt

plt.bar(range(len(obsrv)), obsrv.values(), align='center')
plt.xticks(range(len(obsrv)), obsrv.keys(), rotation=25)
plt.show()
```



Demostración Notebook



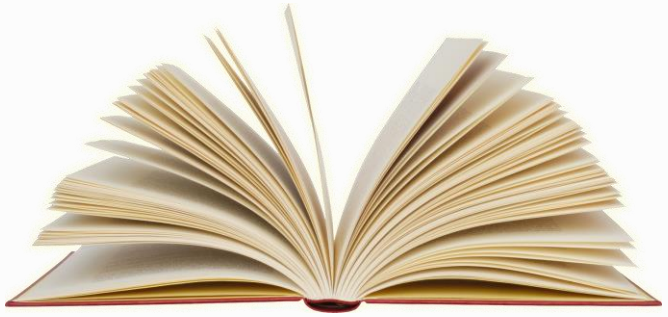
Análisis y visualización de datos

Visualizadores – Herramientas más utilizadas

Visualizadores externos a la plataforma más utilizados

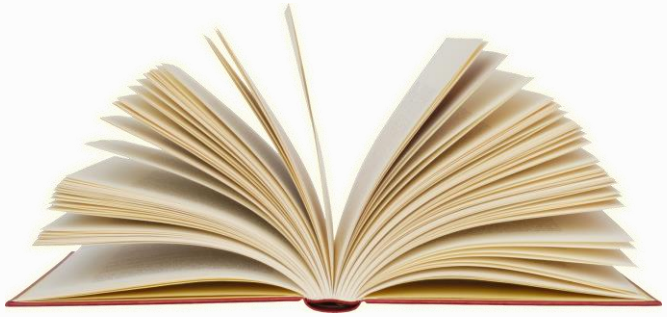
Muchas gracias.

Bibliografía y sitios relacionados



1) Sternkopf H, Mueller R (2018). Doing Good with Data: Development of a Maturity Model for Data Literacy in Non-governmental Organizations. *Proceedings of the 51st Hawaii International Conference on System Sciences*.
2) Ridsdale C, Rothwell J, Smith M, Ali-Hassan, Bliemel M, Irvine D, Kelley D, Matwin S, Wuetherick B. Strategies and Best Practices for Data Literacy Education. *Dalhouse University*.
3) Grillenberger A, Romeike R (2018). Developing a Theoretically Founded Data Literacy Competency Model. *WiPSCE*.
4) Data to the people (2018). Databilities. *Sitio web Wix.*
5) Bonikowska A, Sanmartin C, Frenette M (2019). Data Literacy: What It Is and How to Measure It in the Public Service. *Sitio web Statics Canada*.
6) *Sitio web Data Literacy Project*.
7) Department of the Prime Minister and Cabinet (2016). Data skills and capability in the Australian Public Service. *Sitio web Australian Government*.
8) *Sitio web Open Data Institute.*

Bibliografía y sitios relacionados



9) AGESIC, Presidencia de la República Oriental del Uruguay (2019). Plan de Gobierno Digital 2020: Transformación con equidad. *Sitio web Presidencia de la República Oriental del Uruguay*.

10) DAMA International (2019). DAMA – DMBOK: Data Management Body of Knowledge. *Technics Publications*.

11) AGESIC (2019). Uruguay: Política de Datos para la Transformación Digital*. Sitio web Presidencia de la República Oriental del Uruguay*.

12) Azevedo A, Santos M.F. (2008). KDD, SEMMA and CRISP-DM: A PARALLEL OVERVIEW.

12) IBM Analytics. Foundation Methodology for Data Science IBM.

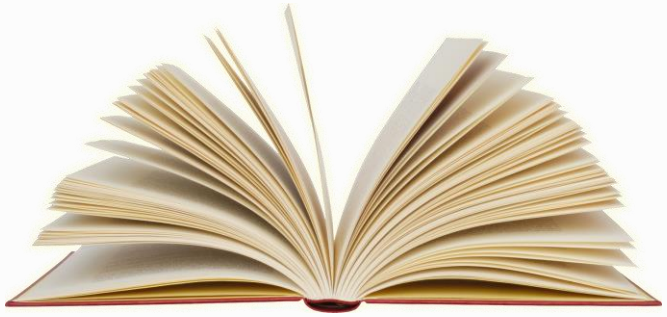
13)AGESIC (2019). Framework de Análisis de Datos. *Sitio web Presidencia de la República Oriental del Uruguay*.

14) AGESIC (2019). Marco de referencia para la gestión de calidad de datos. *Sitio web Presidencia de la República Oriental del Uruguay*.

15)Política y estrategia de datos para la transformación digital (2018). AGESIC. Sitio web Presidencia de la República *Oriental del Uruguay.*

16) AGESIC (2019). Estrategia de Inteligencia Artificial para el Gobierno Digital. *Sitio web Presidencia de la República Oriental del Uruguay .*

Bibliografía y sitios relacionados



17) Documentación Apache Hadoop 3.2.1
18) Documentación Apache Spark
19)Documentación Apache Hive
20)Documentación Apache NIFI
21)Documentación Apache Kafka
22)Documentación Apache Sqoop
23)Documentación Apache Druid
24)https://blog.cloudera.com/data-science-engineering-platform-hdp-3-0-hybrid-secure-scalable/