

PROYECTO **REPARA**^{2.0}

*Desarrollo de nuevas técnicas y sistemas de información para la
REhabilitación sostenible de PAVimentos y carreteras*

APLICACIÓN DE TECNOLOGÍA BIG DATA PARA MONITORIZACIÓN DE CARRETERAS.

REPARA 2.0 es un proyecto de investigación industrial y desarrollo experimental, llevado a cabo gracias al soporte financiero del CDTI, por las empresas Sacyr Construcción, CHM Obras e Infraestructuras, Repsol, Fractalia, Acciona Infraestructuras, Cemosa, y Solid Forest.

www.proyectorepara.com - email: info@proyectorepara.com - twitter: [@ProyectoREPARA2](https://twitter.com/ProyectoREPARA2)

Informe preparado por Daniela Espinoza y Gema Balbacid del proyecto REPARA 2.0 en el paquete de trabajo 3, liderado por FRACTALIA, correspondiente al Hito 1 Entregable: E3.2.1.1.

Madrid, Julio 2017

© Grupo FRACTALIA S.L 2017

ÍNDICE

1. APLICACIÓN BIG DATA PARA MONITORIZACIÓN DE CARRETERAS.....	1
1.1 ADQUISICIÓN DE DATOS.	1
1.2 ALMACENAMIENTO DE DATOS.....	2
1.3 PROCESADO DE DATOS.....	2
1.4 ANÁLISIS DE DATOS.	3
2. ARQUITECTURA BIG DATA.....	3

1. APLICACIÓN BIG DATA PARA MONITORIZACIÓN DE CARRETERAS.

La principal innovación presente en el proyecto REPARA 2.0 es la gestión de carreteras a partir de la investigación y aplicación de tecnologías Big Data. Esta aplicación supone el desarrollo de un sistema que permite obtener los datos de forma remota y desatendida y que además es capaz de integrar de forma sencilla nuevos elementos en función de la evolución del sistema. Permite la integración de distintas fuentes de datos mejorando el proceso de análisis y monitoriza el sistema de acuerdo a los parámetros contemplados. También desarrolla una plataforma única con tecnología Big Data que integra la adquisición, el almacenamiento y el análisis de toda la información recogida.

Esta solución permitirá combinar valores sensoricos con otros datos de interés respecto al deterioro del firme generando modelos predictivos.

Se han definido los requisitos del sistema: requisitos de usuario, requisitos funcionales (acciones que debe llevar a cabo el sistema) y requisitos no funcionales (cómo deben realizarse las acciones). Dentro de estos últimos se han generado requisitos de desempeño, de desarrollo, de usabilidad, de seguridad y legales. La toma de requisitos es un proceso continuo y debe realizarse en paralelo a otras tareas, por lo que a medida que se vaya avanzado en el proyecto, se podrá introducir información referente a éstos o generar requisitos nuevos.

Tras conocer los requisitos, y previo al desarrollo del sistema, se deben definir las fases que componen el sistema Big Data: adquisición de datos, almacenamiento, análisis de datos y visualización.

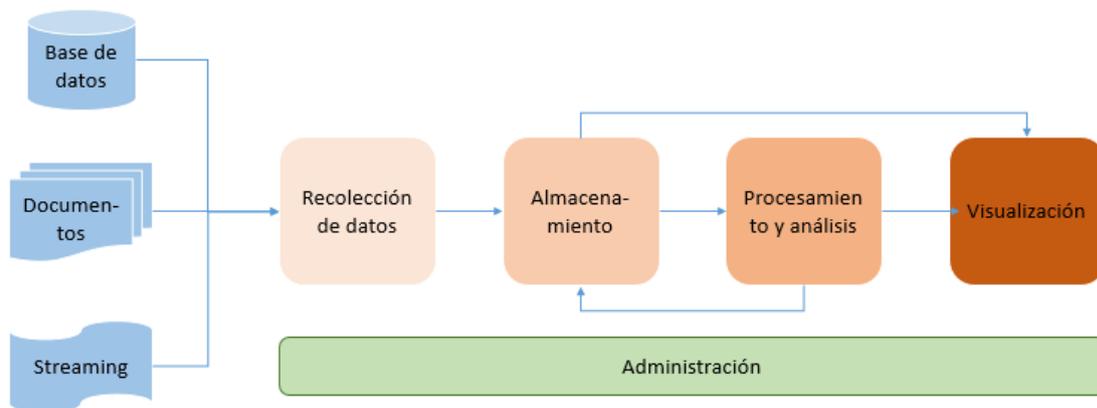


Ilustración 1: Arquitectura básica Big Data.

1.1 ADQUISICIÓN DE DATOS.

Para comenzar el desarrollo de un sistema Big Data es importante identificar el tipo de fuente de datos que se presenta en el sistema. Los tipos de fuentes se pueden clasificar en: **fuentes estructuradas** (datos relacionales que siguen un esquema rígido almacenado en tablas que definen las relaciones entre ellas), **fuentes no estructuradas** (datos que no

siguen ningún esquema, lo que complica la automatización de los procesos) y semiestructuradas (no presenta una estructura rígida pero el esquema puede deducirse de los propios datos).

Una vez identificado el tipo de fuente se debe determinar la manera en que se integrarán esos datos, este punto es de especial importancia para el desarrollo del sistema. Dependiendo del tipo de fuente se pueden distinguir las siguientes soluciones: para el tratamiento de información estructurada y/o semiestructurada se emplean sistemas EII (Enterprise Information Integration), que permite la virtualización de los datos. Para el tratamiento de gestión no estructurada se emplean tanto bases de datos universales como data warehouses o se suele hacer uso de motores de recuperación o sistemas de clasificación automática. Una vez se integran los datos en el sistema se almacenan.

1.2 ALMACENAMIENTO DE DATOS.

Actualmente el almacenamiento de datos puede clasificarse según dos grandes grupos: **almacenamiento completo** (representado por sistemas de ficheros distribuidos) y **almacenamiento en compactaciones** (sistemas de almacenamiento de datos).

Para los sistemas de ficheros distribuidos se emplean tecnologías como:

- HDFS
- Riak-CS
- AWS S3
- Alluxio

En cuanto a sistemas de almacenamiento de datos se emplean:

- Modelos relacionales (Oracle, SQL Server etc.).
- Modelos clave (Redis, Riak, Aerospike).
- Modelos columnares (Cassandra, HBase y Druid).
- Modelos documentales (MongoDB o Elasticsearch).
- Modelos de grafos (Neo4J o Virtuoso).

1.3 PROCESADO DE DATOS.

Tras la recogida de datos y almacenamiento, el sistema debe procesar los datos para la obtención de información relevante. Existen diversas técnicas para el tratamiento de datos cuya aplicación depende del objetivo a alcanzar. El procesamiento por batch habilita la ejecución de un programa sin necesidad de supervisión directa, esto permite acumular y procesar grandes volúmenes de datos, mediante tareas repetitivas, obteniendo de esta forma resultados por lotes. Es escalable y permite el procesamiento distribuido y paralelo. La herramienta Hadoop representa este tipo de procesamiento.

Una de las principales desventajas que presenta el procesamiento batch es la velocidad de procesamiento, lo que da lugar a la creación de un procesamiento en tiempo real o streaming que presenta una buena tolerancia a fallos y a una baja latencia. Las tecnologías más comunes para cada fase son:

- Adquisición de datos (Flume).
- Almacenamiento (Kafka o Kestrel).
- Análisis de datos (Storm, Spark, Streaming, Flume, Tridente o S4).

Existe la posibilidad de combinar los tipos de procesamiento anteriores.

1.4 ANÁLISIS DE DATOS.

Una vez procesados los datos del sistema deben ser analizados. Las técnicas de análisis más comunes son: **análisis visual** (representación gráfica), **análisis estadístico** y **minería de datos** (emplea distintas técnicas de estadística y programación).

Esta clasificación es artificial dado que los límites de cada tipo no están definidos al existir técnicas que, aun cuando inicialmente fueron definidas para un tipo de análisis concreto, finalmente se aplican a otros.

2. ARQUITECTURA BIG DATA.

Tras el estudio de las diferentes fases del sistema se analiza la forma en la que se ofrecen los resultados, también denominada arquitectura del sistema. Para ello se gestiona el sistema como cuatro etapas independientes (recolección de datos, almacenamiento, análisis y visualización) y una transversal (administración del sistema) que combinan las arquitecturas de procesado en batch y en streaming (tiempo real).

a) ARQUITECTURA LAMBDA.

Esta arquitectura presenta tres capas.

- **Capa batch:** Esta capa define una única función que engloba todos los datos existentes. El sistema está enfocado a computar previamente dicha función y calcular cada consulta sobre la marcha basada en una vista precalculada.
- **Capa de servicio:** Permite indexar los resultados obtenidos en la capa batch para poder ser consultados de forma óptima.
- **Capa de velocidad:** Analiza información que aún no ha sido representada. A diferencia de la capa batch, no utiliza todos los datos sino que actualiza los resultados a medida que los recibe.

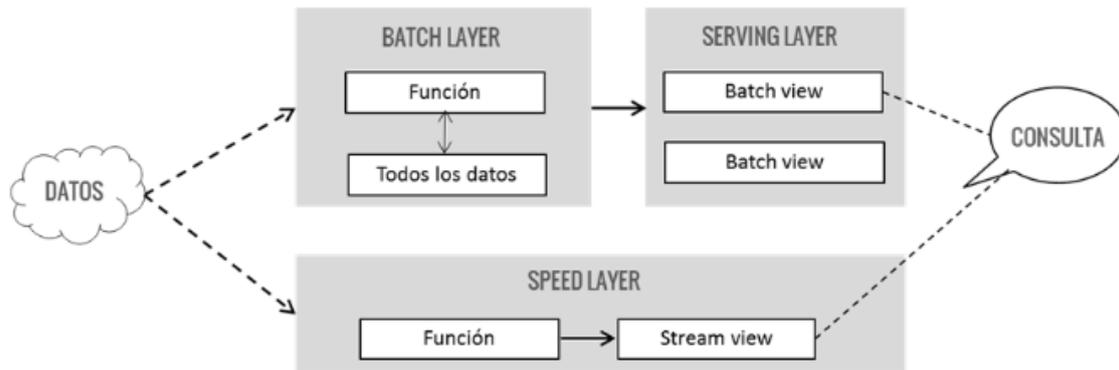


Ilustración 2: Esquema arquitectura Lambda

b) ARQUITECTURA KAPPA.

Se trata de una simplificación de la arquitectura anterior muy similar a la capa de velocidad y, dado que las tecnologías actuales permiten procesar información a tiempo real, esta tecnología es de mayor alcance.

Su hipótesis de partida es la de considerar que las tecnologías asociadas al procesado en streaming son lo suficientemente potentes como para manejar al mismo nivel ambas capas.

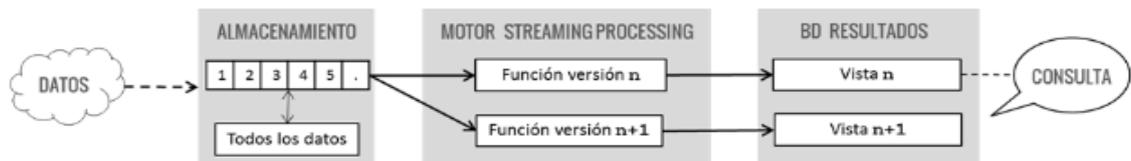


Ilustración 3: Esquema arquitectura Kappa.

Para obtener el documento completo E3.2.1.1: “ESPECIFICACIÓN Y REQUISITOS SW DEL SISTEMA DE MONITORIZACIÓN Y GESTIÓN DE FIRMES” puede ponerse en contacto con los responsables del proyecto en info@proyectorepara.com.