Tecnologies Big Data: Recol·lecció, Transformació, Visualització i Anàlisis de dades en temps real

Guillem Medina Cuadradas

Resum—Actualment, vivim rodejats de grans quantitats de dades i informació digital. Tota aquest gran volum de dades a condicionat el mercat actual creant nous reptes tecnològics i noves propostes de negoci. A partir d'aquest article, es detallarà el concepte de Big Data: com el conjunt d'eines capaces de tractar grans volums de dades i s'analitzaran diferents tecnologies a partir d'una proposta pràctica. Es proposa la recerca i l'estudi de diferents tecnologies Hadoop les quals permeten participar dins els processos de recol·lecció, transformació , anàlisis i visualització de dades en temps real. Com a font de dades de l'estudi, s'utilitzen dades no-massives i open-source les cuals analitzen l'estat actual del transit en la ciutat de Barcelona. Avui en dia, aquestes dades han estat tractades únicament per a la visualització en temps real del transit a Barcelona. Per aquest motiu, a partir d'aquest projecte, es proposa la realització d'una arquitectura basada en tecnologies BigData , una visualització i un posterior anàlisis històric de les dades amb l'objectiu d'extreure informació sobre l'estat del trànsit de Barcelona.

Paraules Clau — Big Data, Kafka, Storm, HortonWorks, Hadoop, Hbase, Data analítics,Spotfire…

Abstract— Nowadays, we life surrounded by a big quantity of data and digital information. All of this volume of this data conditoned the actual market creating news technologies challenges and news bussines proposals. With this article, will be detailed the concept of Big Data: as a combination of tools to be able to manage big data volumes. In additon, will be analyzed diferent kind of technologies since a proposal practice. Proposed the research and study of different kind of Hadoop technolgies on which are allowed to participate into colection, transformation, analysis and visualization of data on real time. As the main study data, will use no-massive and open-source data wich analize the actual state of traffic jams in Barcelona. With this project, is proposed the creation of an architecture based on the BigData technologies, the visualization and the finall analysis of all the historic data with the main goal to extract relevant information about the trafic jams in Barcelona.

Index Terms — Big Data, Kafka, Storm, HortonWorks, Hadoop, Hbase, Data analítics,Spotfire…

1 Introducció

Desde fa uns anys, amb el creixent ús dels dispositius mòbils i el desenvolupament de les xarxes socials, han donat lloc a una segona revolució industrial, produint un creixement exponencial de dades disponibles a través de contingut multimèdia. Aquestes dades, poder sorgir de diferents fonts d'informació com poden ser xarxes socials, blogs, enquestes, anàlisis de negocis..., amb formats diferents: imatges,textos,dades estadístiques, vídeos... que avui en dia estan essent estudiades, analitzades, tractades i comparades per diferents companyies tecnologies com Google, Facebook, Amazon, Ebay..., per tal de millorar les diferents decisions de negoci de manera quasi instantània. A partir de aquest repte per emmagatzemar, quantificar i analitzar aquest gran volum de dades(principalment entre els **TeraBytes-PetaBytes)** ha donat lloc a l'era del **Big Data** hi ha l'evolució de l'Internet de les Coses *(Internet of the Things)*.

Aquesta contribució a l'acumulació de dades massives la podem trobar en diverses industries, companyies amb grans dades transaccionals, reunint així dades dels seus clients, proveïdors, operacions... En molts països també trobem enormes bases de dades capaces de contenir dades de cens de poblacions, registres mèdics...I a més a més si a tot això afegim transaccions financeres o anàlisis de xarxes socials (Twitter,Facebook...) estem parlant de que es generar al voltant de 2,5 quintillons de bytes diàriament en el mon.

Segons un estudi realitzar per ***Cisco***[1] entre el 2011 i el 2016 la quantitat de tràfic de dades mòbils creixerà anualment un 78%.

Així doncs, un dels grans reptes d'avui en dia es poder controlar de manera efectiva aquests grans volums de dades amb l'objectiu de poder analitzar-les de manera eficient i en el menor temps possible, per tal de prendre millors decisions als diferents problemes que sorgeixin, no tan sols en el món empresarial, sinó també en tots els sectors com poder ser els sectors públics, sanitaris...

Aquest article, esta organitzat com es defineix a continuació:

Es detalla un apartat **d'estat de l'art** on es defineixen els problemes que trobem avui en dia amb la gestió de dades massives i una possible solució. A més a més, s'exposa breument el context del cas pràctic al qual va dirigit aquest

- *E-mail de contacte: guillem_medina@hotmail.com*
- Menció realitzada: Enginyeria del Software.
- Treball tutoritzat per: Katerine Diaz (CVC).
- Curs 2015/2016

projecte.

Seguidament, es proposen uns **objectius** a complir i s'exposa la **metodologia** utilitzada per realitzar-los.

Finalment, es conclueix amb el apartat de **resultats**, on es detallen els resultats obtinguts i l'apartat de **conclusions** on s'exposa les conclusions finals del projecte respecte els objectius plantejats.

1.1 Estat de l'art

El principal problema que es troba avui en dia en el mercat es l'incapacitat de poder gestionar dades que ocasionen a les empreses dificultats alhora del seu processament, tractament, anàlisis...com s'ha comentat anteriorment, entre els **TeraByes i PetaBytes** de dades.

Això es degut a que les tecnologies que s'utilitzen: com bases de dades relacionals o arquitectures no distribuïdes, arriben a un punt que no poden gestionar correctament aquestes grans quantitats de dades.

Aquestes dades es poden diferenciar en 3 tipus[8]:

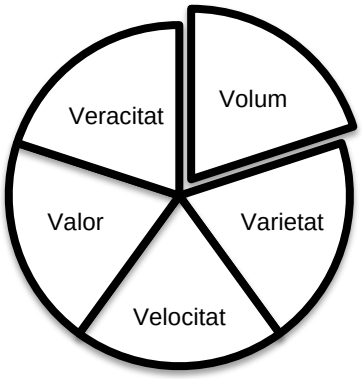
- Dades estructurades: Tenen ben definits els seus formats i longituds, com les dates, números o cadenes de caràcters.
- Dades no estructurades: No tenen un format específic, com poden ser PDF, dades multimèdia, documents de text...
- Dades semi-estructurades: Dades que no delimiten formats determinats, però que contenen marcadors per tal de poder diferenciar els formats de les dades, com poden ser l'HTML, JSON, XML...

Donat un increment de dades no-estructurades en l'ultima dècada, degut a les xarxes socials... treballar amb estructures de bases de dades estructurades poden ocasionar ineficiències ja que no **son capaces de realitzar transaccions de grans volums de dades en temps real** , degut a que aquestes bases de dades no van estar concebudes per el tractament de dades no estructurades i per tant, impliquen limitacions per la gestió d'aquestes.

Per aquest motiu, l'objectiu del Big Data, es poder solucionar aquesta manca de tractament de grans volums de dades utilitzant noves tecnologies capaces de complir les **5V**[10],com es mostra en la figura 1:

- **Volum**: Volum d'informació (TeraBytes, PetaBytes). Moltes taules o files dins una BD, moltes transaccions simultànies...
- **Varietat**: Diferents dades que poden estar representats de manera diversa en qualsevol aplicació, com poden ser dispositius mòbils, àudios, vídeos...
- **Velocitat**: Amb la capacitat de processar la informació amb un temps de resposta relativament baix, 2-5 segons (temps real, batch...)
- **Valor**: Les dades tenen que aportar un valor, com pot ser l'estadístic, o veure correlacions entre dades.
- **Veracitat**: Que les dades aportin confiança i autenticitat, que es puguin confirmar i per tant estar segurs que tenen lògica.

Algunes de les grans preguntes que avui en dia encara sorgeixen respecte la utilització del Big Data són: "*Perquè tenim que utilitzar aquestes noves tecnologies? Perquè no podem seguir utilitzant les nostres bases de dades tradicionals*".



1.Les 5 V del Big Data

La majoria de la gent, coneix les anomenades bases de dades basades en SQL (*Structured Query Language*) gestionades típicament per sistemes com Oracle, MySQL, DB2, Microsoft SQL Server, entre d'altres...seguint doncs, les regles ACID[9](Atomicitat, Consistència, Aïllament, Durabilitat). Això comporta, que les instruccions siguin d'una visió senzilla en la qual una dada s'emmagatzema de manera única i amb unes relacions definides a partir d'unes taules creades per files i columnes.

En canvi, en les bases de dades no tradicionals, o no estructurals com son les bases de dades No-SQL, on l'idea principal es l'emmagatzematge de dades **massives (TeraBytes, Petabytes)** en col·leccions de documents, on l'analisis d'aquestes dades no tenen perquè seguir els estàndards que no necessàriament s'adapten a ells. Ja que on les bases de dades SQL són lentes, les No-SQL resulten més eficients ja que no fa falta manipular les dades tenint que adaptant-les a una estructura rígida.

Tot això, no vol dir que les bases de dades SQL siguin pitjors a les bases de dades No-SQL, sinó que per els casos de grans volums de dades on la majoria d'aquestes son dades no estructurades, com poden ser por exemple: *la gestió dels tweets que Twitter processa durant un dia;* la utilització de bases de dades on l'emmagatzematge es distribuït com poden ser Mongol-DB, Cassandra, Hbase... proporciona una estructura mes flexible, sense la necessitat d'aplicar a les nostres operacions costos i rendiments innecessaris.

En 2012 **Gartner**[2] va definir el **Big Data** com: "*actius d'informació caracteritzats per el seu volum i velocitat elevats i alta varietat que demanen solucions innovadores i eficients de processat per a la millora del coneixement i presa de decisions en les organitzacions*". Actualment l'objectiu principal del Big Data es proporcionar oportunitats de negoci, com per exemple per entendre el perfil d'un usuari, les seves necessitats, les diferents objeccions respecte un producte...Es a dir, es una nova forma amb la qual l'empresa pot interactuar amb els seus clients.

Avui en dia, xarxes socials com Facebook, Twitter, Linkedin etc.. son unes de les principals empreses que generen dades massives i , que al mateix temps, permet l'obtenció d'aquestes. Obrint una línia d'investigació molt important com es l'anàlisi del sentiment.
Tan mateix, tota aquesta volumetria d'informació s'ha de processar correctament. Un conjunt de dades amb un volum de **10TB** planteja la utilització de sistemes distribuïts en lloc de la utilització d'un sol node, per tal de complir una de les característiques de les **5V** vistes anteriorment, la velocitat.

Una de les principals organitzacions que ofereixen solucions per tal de tractar aquesta gran quantitat de dades es **Apache Hadoop,** que proporciona un software lliure per el càlcul distribuït, fiable i escalable**.**
Es tracta d'una plataforma software que permet escriure amb facilitat i executar aplicacions que processen grans quantitats de dades.

En aquest estudi, s'aplicarà un cas pràctic en el qual s'analitzaran les dades del trànsit de la ciutat de Barcelona. Aquestes dades actualment, estan essent utilitzades per **Barcelona.cat[7]** on mostren l'estat actual del trànsit de Barcelona. Per la realització d'aquesta visualització, utilitzen tecnologies com **JavaScript**, **PHP i Google Maps**.
A partir d'aquest estudi llavors, es proposa una nova arquitectura basada en tecnologies BigData i no únicament una visualització final, sinó també una visualització i anàlisis històric de les dades amb l'objectiu de proporcionar informació de valor respecte l'estat dels tràfic de Barcelona.

1.2 Objectius

Com s'ha detallat en l'apartat anterior, per tal del processament de grans volums de dades, es necessari tenir una gran arquitectura capaç de suportar aquesta gran quantitat d'informació. En aquest cas, per **falta de recursos**, l'objectiu principal de l'estudi es centrarà en la investigació de les diferents tecnologies i la realització del cas pràctic a partir d'un tractament de dades **no-massives**.

Per tal de detallar i aclarir millor els objectius que s'exposen a continuació, s'han **dividit els objectius** en **sub-tasques** en funció l'entorn funcional al qual afecten:
Com a **objectius principals**:

1. Creació d'una arquitectura BigData per el processament de dades en temps real dins d'un entorn Hadoop.
2. Visualització de les dades.
3. Anàlisis estadístic/històric final de les dades de l'estat del trànsit de Barcelona.

Per tal de complir el primer objectiu (1.creació de l'arquitectura Big Data) es van planificar les següents tasques:

- **Preparació i configuració de l'entorn de programació** a partir de la màquina Virtual Sandbox 2.3 de HortonWorks.
- **Estudi de la tecnologia Hadoop:** Es detallarà com funciona Hadoop, motors de càlculs, sistema de fitxers...

- **Estudi de diferents tecnologies per tal de dissenyar una arquitectura fiable i consistent per el flux de les dades:** En aquest apartat s'analitzaran diferents tecnologies que incorpora la màquina virtual basades en tecnologies relacionades amb el Big Data, i se'n descriuran les característiques principals per tal de poder dissenyar una arquitectura capaç de gestionar i processar un flux de dades.

Un cop vist l'entorn:

- **Contextualització del cas pràctic:** Es detallarà el cas pràctic.
- **Recopilació, extracció de dades open-source:** S'analitzarà com i quan extraiem les dades.
- **Transformació de les dades:** Es treballarà amb un format únic de dades.

Seguidament, per tal de complir el segon objectiu (2.Visuaització de dades):

- **Visualització de les dades en temps real:** Finalment, es realitzarà una interfície capaç de visualitzar les dades amb l'objectiu de generar informació visual per l'usuari.

I finalment, **l'analisis estadístic de l'històric obtingut** per tal d'obtenir la informació per extreure uns resultats quantificats.

2 METODOLOGIA

En aquest apartat s'analitzarà el context de cas pràctic proposat, s'estudiarà l'entorn de programació Apache **Hadoop** i s'anaitzaran les diferents etapes en les quals s'organitza el projecte.

2.1 Estudi: Tràfic Trams de Barcelona.

Es proposa un cas pràctic on s'estudiarà **la densitat del tràfic de Barcelona dins uns determinats trams**.
Per tal de poder realitzar aquest estudi, principalment es dividirà l'estructura del nostre estudi en 4 parts, com es mostra en la figura 2:

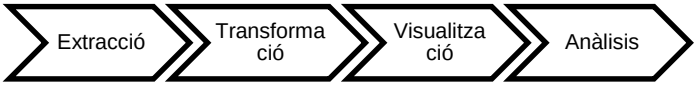


Figura 2: Estats Metodologia

El primer procés a realitzar es tracta de **l'Extracció de dades**. En aquest procés es veurà:

- d'on i com extraurem les dades.
- quines dades.
- període de extracció.

Seguidament, un cop obtenim aquestes dades s'analitzarà dins el procés de **Transformació**, l'arquitectura proposada per ta de poder crear un flux de dades òptim utilitzant les tecnologies Kafka, Storm i Hbase.

I finalment, la **Visualització i l'anàlisi** d'aquestes dades.

Aquest cas pràctic s'analitzarà amb més detall en el punt [*3.Resultats i Discussió*].

2.2 Entorn de programació

L'entorn de programació que s'utilitzarà es troba sota el framework software **Apache Hadoop**, que suporta aplicacions distribuïdes open-source. Aquest framework es va inspirar a partir de document Google com el MapReduce i Google File System (GFS).

2.2.1 MapReduce

Es un model de programació i implantació amb l'objectiu de poder processar grans quantitats de dades. Aquest procés, es realitza a partir de l'especificació d'un conjunt de clau→ valor (*Map*). On k1 i k2 son claus i v1,v2 son valors.

$$Map\ (k1\ ,\ v1\ ) \rightarrow list\ (k2\ ,\ v2)$$

Un cop realitzat aquest procés, actuen una sèrie de funcions *Reduce* els cuals barreja tots els valors obtinguts anteriorment agrupant-los dins de la seva clau. Com es veu en la figura 2. On finalment, genera un output únic per a cada clau.

$$Reduce\ (k2\ ,\ list\ (v2\ )) \rightarrow list\ (v3)$$

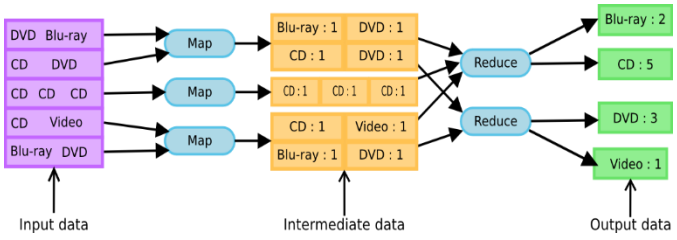


Figura 3: Exemple MapReduce

En la figura 3, es proposa un exemple on s'observa com donats uns inputs d'entrada es realitza un Map on s'assignen unes claus (Blu-ray,DVD,CD,Video) les qual tenen un valor, en aquet cas numèric segons la quantitat que trobem. Seguidament es realitza el Reduce on, a partir de les claus assignades es retorna una llista dels resultats de cada clau i la suma del valor corresponent.

2.2.2 Hadoop Distributed File System (HDFS)

Es un sistema de arxius distribuït i escalable escrit en el llenguatge de programació Java. Aquest sistema de fitxers es molt important ja que per sobre d'aquests fitxers estan construïdes diverses eines. Aquest sistema de fitxers proporciona a l'estructura de tecnologies **Big Data** una gran **escalabilitat**, es a dir, un sistema que pugui variar en el seu volum, segons les necessitats del moment, sense que això afecti a la resta del sistema.

En la figura 4 s'observen quatre fitxers que estan distribuïts en 4 nodes (sistema físic), però que l'usuari quan entra dins el directori els veu dins el mateix directori (sistema lògic).

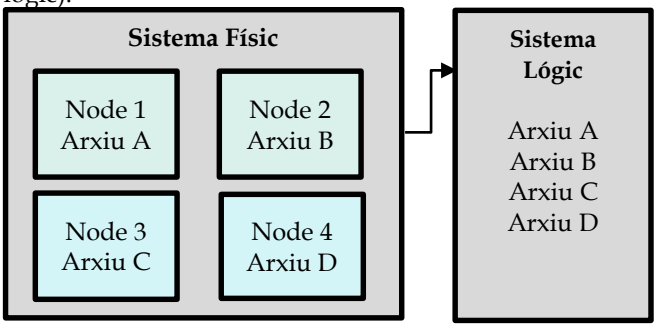


Figura 4: Arquitectura HDFS

2.3 Tecnologies Hadoop

En aquest apartat, es descriuran un seguit de tecnologies que formen part de l'entorn Hadoop, i que part tant tenen la mateixa estructura que la detallada en l'apartat [2.1].
Es definirà les tecnologies que s'ha utilitzat durant el cas pràctic proposat per tal de poder entendre millor la metodologia final utilitzada.

2.3.1 Apache Pig

Apache **Pig[3],** es una eina per tal d'analitzar grans volums de dades mitjançant un llenguatge d'alt nivell anomenat PigLatin. Mitjançant el compilador, es capaç de traduir aquestes sentencies PigLatin a tasques MapReduce.
Per aquest motiu, Pig proporciona un aprenentatge molt àgil ja que no es te que conèixer grans nocions de programació en MapReduce per aconseguir processar correctament scripts que englobin grans volums de dades.

PigLatin, funciona principalment a partir de sentencies que reben i que produeixen una relació, es a dir, una col·lecció de dades.

2.3.2 Apache Kafka

Kafka[4], es un sistema d'emmagatzematge que utilitza el protocol de publicador/subscriptor, representat en la figura 4. Aquests sistema, al treballar sota un entorn Hadoop, també proporciona un emmagatzematge particionat i duplicat. Gràcies a això, pot treballar grans volums de dades en temps real.

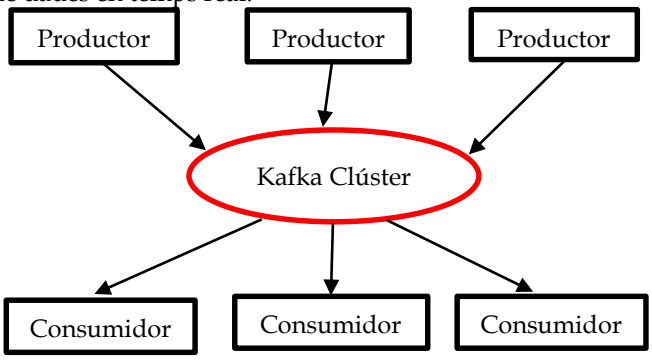


Figura 4: Sistema Publicador/Subscriptor de Kafka

Els principals conceptes que es troben en kafka són els seguents:

- **Tòpic**: Son categories on el clúster de Kafka guarda els missatges.
- **Productors**: Son els encarregats de publicar un missatge en el tòpic de Kafka.
- **Consumidors**: Son els qui consumeixen aquests missatges.

Tot això s'executa com un clúster o un conjunt de servidors els quals s'anomenen **Broker**.
Gracies a aquesta estructura, permet un **ràpid** temps d'execució de lectures i escriptures, es **escalable**, ja que aquests clúster es **flexible** i per tant, permet la divisió de les dades en diferents fluxos dins aquest clúster, garantint sempre una **tolerància a fallades** reals.

2.3.3 Apache Storm

Apache Storm[5], es una tecnologia que permet recuperar Streams de dades en temps real a partir de diverses fonts d'informació de manera distribuïda, tolerant a fallades i amb una disponibilitat elevada.

La seva arquitectura esta formada principalment per 2 components:

- **Spouts**: És l'encarregat de recollir el flux de dades d'entrada.
- **Bolts**: L'encarregat de processar aquest flux de dades.

En la figura 6, es representa un esquema on els Spouts són aixetes que simulen l'entrada d'un stream de dades i on els Bolts són els llamps que generen les transformacions i els processament d'aquestes dades entre altres Bolts.

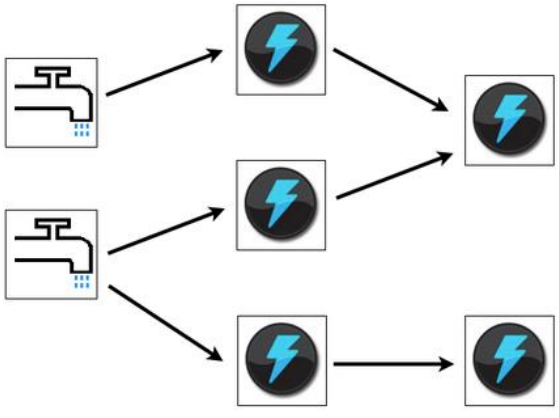


Figura 6: Esquema Spout-Bolts Storm

A més a més, l'arquitectura Storm presenta **topologies**. Aquestes topologies, permeten crear diferents instancies entre els Bolts i Spouts. Això, permet que el sistema sigui molt més **escalable** permeten que les dades es distribueixin de forma particionada entre els diferents components(Spouts,Bolts) que formen una topologia.

Gràcies a la utilització d'aquesta tecnologia es possible processar dades en temps real. Avui en dia, existeixen grans quantitats de projectes que utilitzen Storm. Un dels més rellevants es el projecte de **Twitter[6]**, com es pot observar en la figura 7.

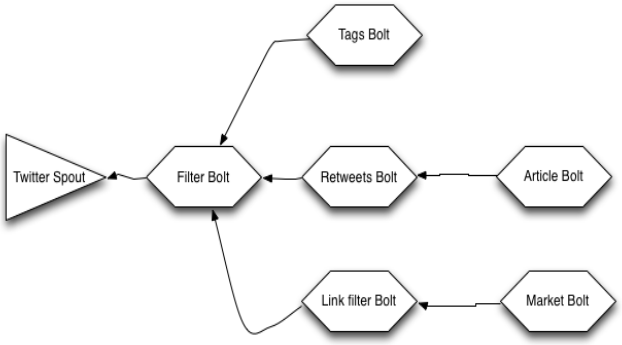


Figura7: Esquema Storm de Twitter

Twitter, utilitza Storm com a eina per tal de **processar i distribuir els tweets** que es publiquen en temps real.

2.3.4 HBase

Com s'ha comentar en apartats anteriors, un dels principals problemes alhora de obtenir grans volums de dades es la seva escriptura dins bases de dades que permetin obtenir temps de latència escriptura/lectura reduïts.

Hbase[7], es una base de dades distribuïda open-source sota de l'entorn **HDFS Hadoop**. Com s'ha comentat en punt *2.1.2 HDFS.* Aquest sistema de fitxers en el qual es basa tota l'arquitectura hadoop, proporciona a que Hbase tingui alts nivells de rendiment i de redundància de dades, ja que es distribueix cada dada dins de HDFS.

Hbase es una base de **dades columnar** que sota l'entorn HDFS permet diversos beneficis com la baixa latència en l'accés de les dades de petit tamany i en la cerca de registres a partir d'indexacions claus-columna.
Com s'ha anomenat anteriorment, Hbase es tracta de una "columna de bases de dades". On s'estructura de la següent forma:

- **Taula**: Es un conjunt de files.
- **Fila**: Es una col·lecció d'una columna.
- **Columna**: Es una recopilació dels principals valors.
  Columna Família: Es una col·lecció de columnes

Com s'observa en la taula 1 les diferents característiques que proporciona Hbase respecte altres bases de dades relacionals (RDBMS):

HBASE	RDBMS
No te concepte de columnes fixes; nomes defineix columnes de famílies.	**Es caracteritza per el seu esquema, en el qual es descriu un conjunt de estructura de taules.**
Està construïda en taules amples. Hbase es escalable horitzontalment	**Es prima i construïda per petites taules. Difícil d'ampliar.**
No hi ha transaccions.	**Es transaccional.**

Es bo per a dades semi-estructurades i estructurades.	**Es bo per a dades estructurades.**

Taula1: Hbase Vs RDBMS

En aquesta secció, finalment s'ha detallat l'entorn de programació en el qual es realitzarà el cas pràctic. S'observa que tecnologies Hadoop com HDFS, Kafka, Storm i Hbase treballen totes sota MapReduce i per tant es centren en la distribució i partició de dades (divideix i guanyarás) amb l'objectiu de dotar al sistema d'escalabilitat de dades i de tolerància a fallades.

3 RESULTATS I DISCUSSIÓ

En aquest apartat, s'aprofundirà en el cas pràctic proposat en relació l'estat del trànsit dins de la ciutat de Barcelona. Es detalla com s'ha aconseguit crear, a partir de tecnologies que formen part de l'entorn Hadoop com son: **Pig,Kafka,Storm, Hbase** un flux de dades capaç de gestionar les dades per tal de poder visualitzar-les posteriorment i extreure conclusions.
Un cop vist l'arquitectura proposada, es realitzarà un petit test d'execucions per comprovar el rendiment del sistema.

3.1 Estudi: Lògica del Sistema

Es representa aquest sistema lògic a partir de 3 diagrames dividits a partir de l'estructura plantejada anteriorment: **Extracció** corresponent a la figura 9, **Transformació** i **processament** corresponent a la figura 10 i **Visualització i anàlisis**, figura 11.

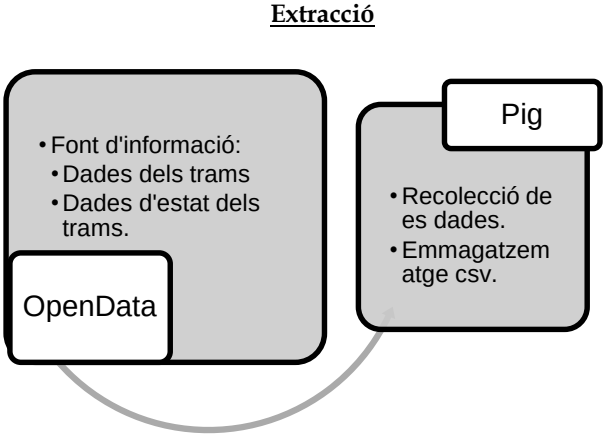


Figura 9: Extracció de dades

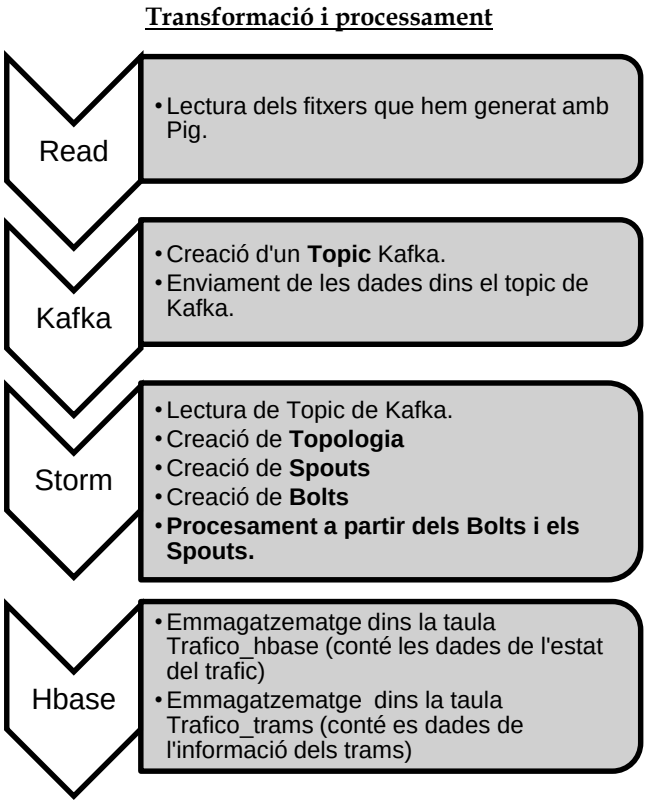


Figura 10:Transformació processament

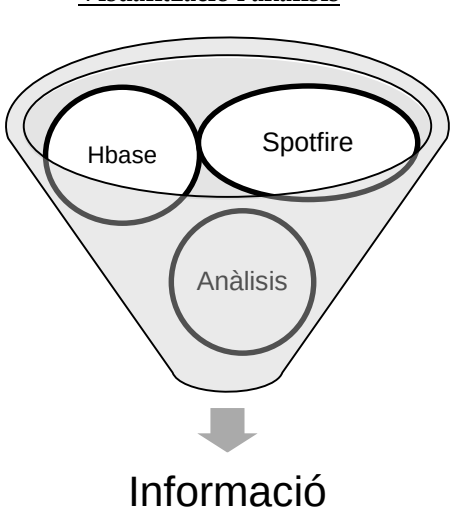


Figura 11: Visualització i anàlisis

3.2 Estudi: Extracció de dades

Aquest es el punt de sortida del cas pràctic. Primer de tot s'ha de trobar una font d'informació d'on recuperar les dades de l'estat del transit de Barcelona.

S'ha utilitzat les dades obertes que proporciona **Open Data Barcelona.**

3.2.1 Dades Trànsit Barcelona

A continuació s'exposaran els detalls de la informació de

proporciones les dades del Transit de Barcelona.
Contem amb 2 fitxers de dades, amb un format **BIN** el qual conté **dades sobre tots els trams de Barcelona**, amb la següent estructura:

- **Tram**: Identificador numèric del tram.
- **Descripció**: Nom del tram.
- **Coordenada**: Coordenades composades per longitud i latitud del tram físic.

1 Diagonal (Ronda de Dalt a Doctor Marañón)
2.11203535639414,41.3841912394771,0
2.101502862881051,41.3816307921222,0

I el segon fitxer, que conté informació sobre l'estat del trànsit dins un tram específic:

- Id: identificador del tram.
- Data: formada per hora, minuts i segons.
- Estat del tram: Aquest indicador, proporciona l'estat actual del tram basant-se en una numeració del 0-6.
    - 0: Sense dades
    - 1: Molt Fluid.
    - 2: Fluid.
    - 3: Dens.
    - 4: Molt Dens
    - 5: Congestionat.
    - 6: Tallat.
- Temps previst: descriu l'estat del tràfic previst passats 15 minuts utilitzant la nomenclatura anterior.

1#20151114182057#2#2

Finalment, s'obté un fitxer de dades de 12KB , on cadascuna d'elles representa el tram i el seu estat.
Així doncs, obtindrem cada 10 minuts un total de 534 estats.

3.3 Estudi: Transformació i processament

Com hem analitzat en els apartats anteriors, utilitzarem les tecnologies Kafka, Storm y Hbase per el processament del flux de dades corresponent a l'estat dels trams de tràfic de Barcelona.
Per tal de processar aquestes dades, es realitzen un seguit de pasos:

1. Creació d'un tòpic de Kafka.
2. Creació d'un Productor de dades fins al Topic de Kafka.
3. Creació d'un Consumidor Storm, que consumeixi les dades del tòpic.
4. Tractament de dades i Inserció en Hbase.

El primer pas es la **creació del tòpic**, per tal de poder emmagatzemar la informació corresponent als fitxers:

En aquest cas, s'ha definit un topic anomenat **tràfic_data_tfg**, serà l'encarregat de produir els missatges perquè seguidament la tecnologia **Storm** pugui consumirlos.

Un cop realitzat aquest pas, s'ha de enviar les dades cap a aquest tòpic.
Per la realització d'aquest pas, es va realitzar un projecte en **Java-Maven** on s'incloïen les dependències Maven, detallades en la figura 12, necessàries per tal de poder treballar amb les classes que proporciona Kafka, ja que aporten les configuracions necessàries per establir una connexió entre Java i Hadoop.

```
<dependency>
<groupId>org.apache.kafka</groupId>
<artifactId>kafka_2.10</artifactId>
<version>0.8.2.1</version>

<dependency>
    <groupId>org.apache.kafka</groupId>
    <artifactId>kafka-clients</artifactId>
    <version>0.8.2.0</version>
</dependency>
```

Figura 12: Depedències Kafka

Seguidament s'ha d'establir la configuració per tal de poder tenir una connexió amb la nostra tecnologia dins la Maquina virtual:

- El port del servidor: **sandbox.hortoworks.com:6667**

Es declaren les propietats necessàries per poder enllaçar la nostre connexió amb el tòpic a partir de la classe **KafkaProducer.** Un cop realitzat aquest procedí ment ja es pot tractar els arxius i enviar les dades al tòpic.

Kafka ens proporciona, com s'ha esmentat anteriorment, una capacitat d'escriptura rapida. Per aquest motiu, el següent procediment es aconseguir consumir aquestes dades
Per tal de consumir les dades del tòpic de Kafka per el seu posterior tractament, s'utilitza la **tecnologia Storm**.
Es necessari la realització d'un altre projecte Java-Maven el qual inclogui les dependències específiques per tal d'integrar les dues tecnologies.

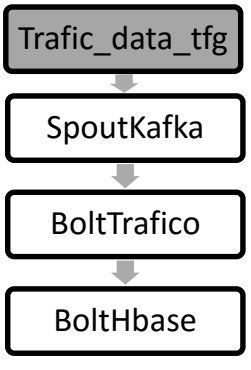


Figura 13: Arquitectura Storm

Com s'observa en la figura 13 l'arquitectura Storm proposada consta de **1 Spout** encarregat de consumir les dades que es troben dins del tòpic de Kafka. I **2 Bolts**, el **boltTrafico** es l'encarregat de rebre les dades de **l'SpoutKafka** i transformar-les. En aquest cas, es van generar transformacions de tipus de dades, i es va partir les dates (dia,mes,any,hora,minut,segon) que ens arribaven per tal de obtenir informació més específica d'un tram.

I finalment, el BoltHbase encarregat de rebre les dades del BoltTrafico i emmagatzemar-les dins de **Hbase**.

3.3.1 Test de rendiment

Durant el procés d'extracció realitzat amb les tecnologies BigData, s'han recol·lectat i emmagatzemat un total de **2.259.552 rows,** amb un tamany total de **100MB** de dades. Per tal de comprovar la fiabilitat i el rendiment de l'arquitectura proposada, es detalla un petit test de rendiments.

Per aquest projecte, s'ha utilitzat únicament una màquina amb les següents característiques:

- 16GB RAM
- 1TB disc
- Procesador Intel i7
- 1,8GHz – 3GHz

Per tal de comprovar el rendiment de l'execució, s'analitza els temps que es triga en extreure un arxiu, i processar-lo fins la inserció final en la base de dades, dins de l'arquitectura MapReduce:

Tamany Input	Número de arxius	Repeticions	Total (s)	Mitja (s)
12KB	1	1	150s	**153s**
		2	146s	
		3	163s	

Taula 2: Test d'execució

A més, s'ha d'afegir a aquest temps final de 153s, els temps que triga la maquina virtual en accedir a Hbase, entorn als 3 minuts.
S'observa com els temps d'execucions no son molt bons. Ja que el temps d'accés a Hbase (3 minuts) supera el temps real que triga en processar i emmagatzemar les dades (153 segons).

3.4 Estudi: Visualització i Anàlisi de les dades

Dins aquest apartat es detallarà la tecnologia que s'ha utilitzar per tal de crear la visualització de les dades i es veuran els resultats dels anàlisis estadístics/històrics proposats.

3.4.1 Visualització

Per tal mostrar les dades s'utilitza l'eina de visualització **Spotfire**. Aquesta eina es capaç de connectar-se al nostre entorn Hadoop i rebre directament les dades que es troben en la nostra base de dades.

A partir de les dades que es va recopilar **entre el 2 Novembre del 2015 fins el 26 de Gener del 2016.**
S'ha pogut realitzar diferents anàlisis per tal de extreure informació més rellevant de l'estat de trànsit de Barcelona.

Gràcies a l'adquisició d'aquest històric de dades, es possible realitzar anàlisis no només per la situació actual (com en el projecte de Barcelona.cat), en aquest projecte s'inclou la realització de filtres per tal de poder especificar períodes de temps, estats..alhora de realitzar les diferents visualitzacions per el posterior anàlisis.

Els filtres definits són els següents:

- Filtres per la visualització per data:
  - Mes (11,12,1)
  - Any (2015,2016)
  - Dia (1-31)
- Filtres per la visualització dins un període de temps:
  - Hores
  - Minuts
- Filtres per la visualització d'un tram específic:
  - Direcció del tram.

Les visualitzacions realitzades són els següents:

- **Visualització de l'estat actual i el previst del trànsit de Barcelona**: Com es mostra en la figura 14, es realitza una visualització de l'últim registre que es troba dins la base de dades per cada tram, amb l'objectiu de mostrar l'estat actual/previst del transit de Barcelona.

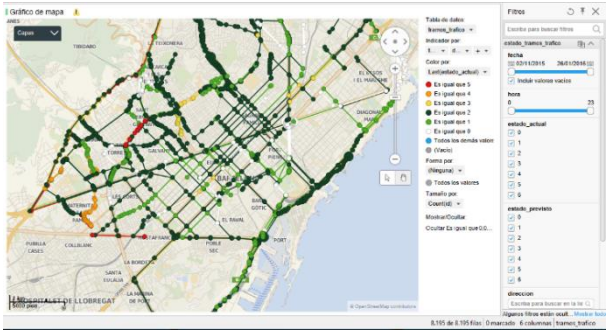


Figura 14: Estat actual del transit

- **Visualització de l'estat d'un tram específic**:
  Es possible veure el % de l'estat d'un únic tram, tant per data com per període de temps.
  Com es mostra en la figura 15, s'especifica amb colors els diferents estats (1-tràfic molt fluid fins al 5-Congestionat o 6-Tallat).

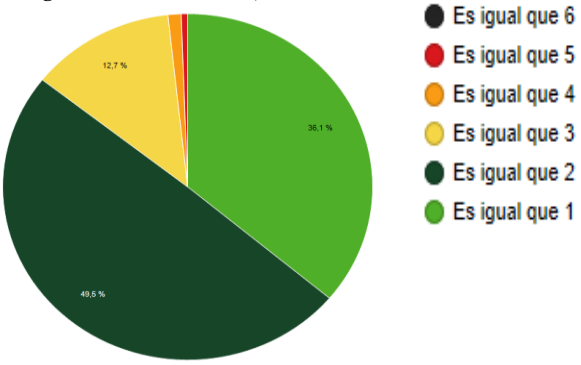


Figura 15: % d'estats Balmes (Aragó a Gràcia) - dins un període de temps

- **Visualització de la quantitat dels diferents estats produïts per mes:**
  Per tal de poder visualitzar i comparar tant per **tota la ciutat de Barcelona**, com per **trams específics** la variació d'estats produïda. Com s'observa en la figura 16.

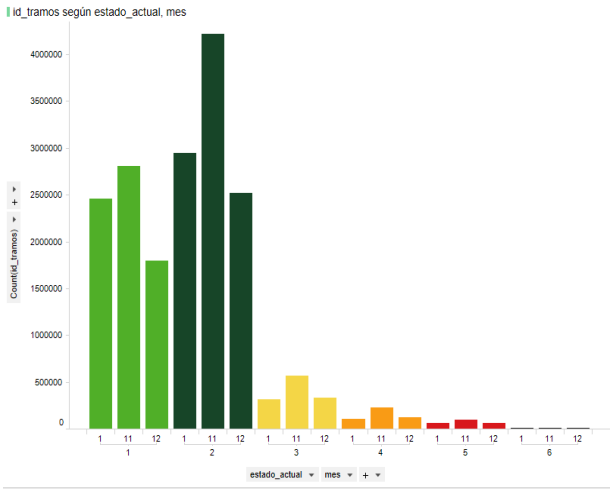


Figura 16: Sumatori d'estats de tots trams durant els mesos 11,12 i 1.

- **Visualització de els trams amb major densitat de transit per mes:**
  En aquest cas, es representa un gràfic corresponent a la figura 17, on s'observa els mesos distingits per colors (1-blau, 11-verd i 12-vermell).
  Es filtra per estats de trams, per veure els trams més congestionats.

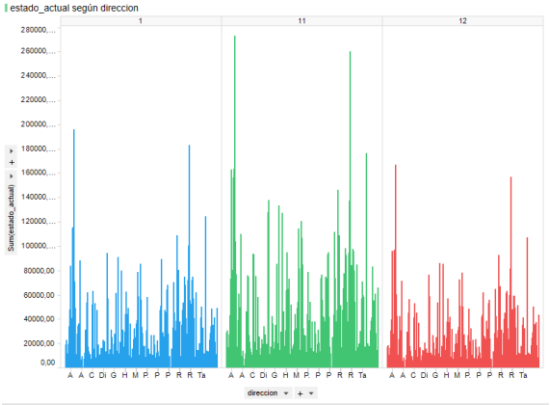


Figura 17: Gràfic Densitat per tots els trams/mes

- **Visualització de la mitjana d'estats dins un tram, per mes/hora:**
  En la figura 18, s'observa com es pot analitzar l'estat mitjà que té un tram específic i filtrar-ho per dates i hores.

Per aquest gràfic també es poden aplicar filtres per tal de veure una densitat mes específica per exemple, a partir d'un únic mes, o entre unes hores concretes.

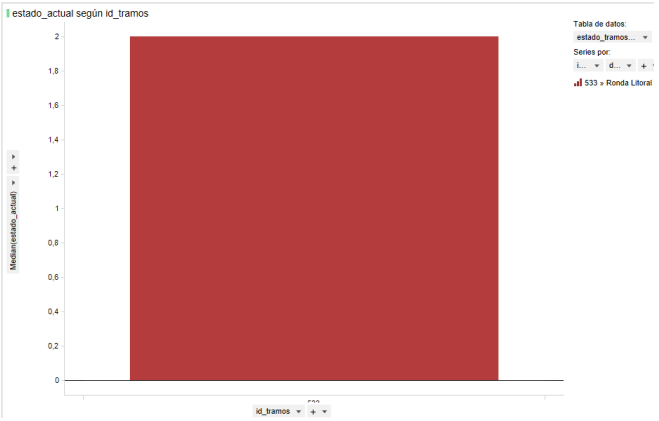


Figura 18: Mitjana d'estat per el tram de Ronda Litoral (Polos-Nus de la Trinitat
Estat mitjà = 2)

3.4.2 Anàlisis

Com a resultat de visualitzacions que s'han mostrat, s'ha pogut analitzar, a partir dels filtres per mes i per el filtre d'estats, el Top 3 trams amb més congestió/mes:

MES	TOP1	TOP2	TOP3
11	Av.Foix (Pg.Reina Elisenda a pg. Manuel de Girona).	Ronda Litoral (Passeig de la Zona franca a Miramar).	Túnel de la Rovira (Rambla del Carmel a Ronda Guinardó).
12	Av.Foix (Pg.Reina Elisenda a pg. Manuel de Girona).	Ronda Litoral (Passeig de la Zona franca a Miramar).	Gran Via (Tetuán-Marina)
1	Av.Foix (Pg.Reina Elisenda a pg. Manuel de Girona).	Passeig Sant Gervasi (Bonanova-Balmes)	Ronda Litoral (Passeig de la Zona franca a Miramar).

Taula 3: Resultats Top 3 Congestió

- % d'estats durant un mes sobre el Top 1 de trams amb més densitat:
  En aquest cas, s'ha analitzat el cas el qual apareixia comú durant tots els mesos, el tram corresponent a **Av.Foix (Pg.Reina Elisenda a Manuel Girona).**
  S'observa a la figura 19.
- % d'estats de la ciutat de Barcelona, per mes com per els últims 3 mesos, es detalla en la taula 4.

Mes	Molt Fluid	Fluid	Dens	Molt Dens	Con-ges-tio-nat	Ta-llat
11	35,4%	53,2%	7,2%	2,9%	1,3%	-
12	32,7%	52,1%	6,9%	2,5%	1,3%	-
1	41,8%	50,1%	5,4%	1,7%	1%	-
11,12,1	37,9%	51,9%	6,5%	2,4%	1,2%	-

Taula 4: % Estats ciutat Barcelona total i mes

- Intervals d'hores amb més congestió de tràfic:
  Per aquest cas, analitzarem en intervals de 4 hores, per tots els trams de Barcelona, la congestió màxima, es a dir, l'interval de temps on trobem estats on el % de congestió sigui el més elevat.

Intervals	Molt Dens	Congestionat
00-03	0,5%	0,2%
04-07	1,3%	0,6%
08-11	3,3%	1,6%
12-15	2,7%	1,1%
16-19	3,8%	1,9%
20-23	1,0%	0,6%

Taula 5: % Intervals de temps amb més congestió

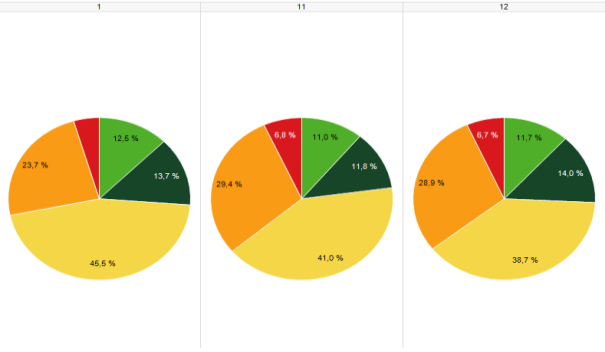


Figura 19: % estats Tram Top 1 / mesos

4.CONCLUSIONS

A partir d'aquest projecte, s'ha analitzat les diferents tecnologies que avui dia engloben el terme de BigData dins de l'entorn proporcionat per la Sandbox 2.3 deHortonWorks. A més a més s'han aplicat directament a un cas pràctic.
Durant el principi de desenvolupament de projecte, va ser una mica confús, tant per la quantitats d'eines noves que hi han ,com per els nous paradigmes de desenvolupament que tractaven.
A més a mes, s'ha de tenir en compte que per les limitacions de treballar amb un entorn on només es comptava amb una màquina, no s'aprofita el rendiment que ofereix Hadoop i que, per aquest motiu, el resultats que es veuen reflexats en la taula 2, no superen els temps d'execucció que avui dia s'ofereix dins del projecte de Barcelona.cat, el cuals proposen una visualització en temps real (entre 2-10 segons).
Finalment, encara que els resultats en quant a temps d'execució no han sigut òptims, les tecnologies utilitzades han permès la creació d'una estructura flexible a canvis, ja que gràcies a la utilització del tòpic de Kafka per produir les dades, com els diferents Spouts i Bolts de Storm per consumir-les i tractar-les, permeten la incorporació d'altres fonts de dades les cuals es poden adaptar fàcilment per tal de utilitzar la mateixa arquitectura vista en el cas pràctic, i ens han permès analitzar i visualitzar les dades fiables d'una forma molt senzilla, per tal de poder adquirir informació important com, per exemple, els trams que han originat més congestió en els últims 3 mesos, o els intervals de temps amb major congestió.

Es pot concloure llavors, segons els resultats obtinguts en aquest projecte, donen lloc a suposar que la utilització del BigData dins de projectes amb pocs recursos no es una bona opció, degut a que la utilització d'aquestes tecnologies es molt costosa ja que estan pensades per tal la distribució de grans volums de dades sota clústers de màquines.

Personalment, a partir d'aquest projecte he pogut veure tecnologies molt interessants, les quals durant els 4 anys de grau mai havia escoltat i que avui en dia, estan essent molt importat per les empreses com per l'evolució de la tecnologia en general. Per aquest motiu, gràcies a aquest projecte, he pogut plantejar-me una orientació laboral encaminada a les tecnologies Hadoop i en la consultoria del Big Data.

5.POSSIBLES MILLORES

La principal idea es millorar el rendiment i que ofereixen aquestes noves tecnologies adaptant-les dins un entorn amb varis clústers. D'aquesta manera es podria corroborar, el benefici d'aquestes tecnologies.
Es poden veure projectes com Linkedin o Twitter (referencia), els cuals proposen aquestes arquitectures, per tractar milions de dades per segon.
En aquest cas, un cop es verifiqués el bon rendiment, es podria aplicar no només a la ciutat de Barcelona sinó a un estudi centrat en dades de ciutats amb un tràfic dens, com pot ser Madrid, Sevilla, Valencia...

6 BIBLIOGRAFIA

[1] Cisco, "Internet será cuatro veces más grande en 2016", Artícle Web, http://www.cisco.com/web/ES/about/press/2012/2012-05-30-internet-sera-cuatro-veces-mas-grande-en-2016--informe-vini-de-cisco.html
[2] Gartner, "Big data", Article Web, http://www.gartner.com/it-glossary/big-data/
[3] Pig, "Official Site Pig-HortonWorks "http://hortonworks.com/hadoop/pig/
[4] Kafka, "Official Site Kafka-HortonWorks" http://hortonworks.com/hadoop/kafka/
[5] Storm, "Official Site Storm-HortonWorks" http://hortonworks.com/hadoop/storm/
[6] Storm & Twitter Example, " Introducción a Apache Storm" http://www.adictosaltrabajo.com/tutoriales/introduccion-storm/
[7] Transit Barcelona, "Estat actual trànsit Barcelona "http://com-shiva.lameva.barcelona.cat/es/transit
[8] Estructura de dades,"Tipus de Dades" http://sisinfo-sri.blogspot.com.es/2011/10/los-documentos-estructurados.html
[9] ACID,"Acid en les bases de dades" http://www.dosideas.com/noticias/base-de-datos/973-acid-en-las-bases-de-datos.html
[10] 5V Big Data, "5V" http://www.quanticsolutions.es/big-data/las-5-v-big-data/