Procedimiento para el Análisis de Comportamiento Meteorológico Utilizando el Módulo Big Data de la Herramienta Pentaho

Ronald Mauricio Gelvez Ruiz

Universidad de Pamplona

Directora: Luz Marina Santos Jaimes

Doctora en ciencias

Ronald Mauricio Gelvez Ruiz, Facultad de Ingenierías y Arquitectura, Universidad de Pamplona

Universidad de Pamplona, Norte de Santander

Resumen

En la actualidad la gran cantidad de datos en diferentes áreas crece en forma continua a una alta velocidad, con gran variedad en sus datos y en donde se tiene que conservar la integridad de los mismos para que su valor no se pierda. Aquí el análisis de Big Data juega un papel fundamental aplicando modernas herramientas de software para almacenar, procesar y descubrir información valiosa que contribuya a la solución de diversos problemas.

Este trabajo contiene la descripción detallada de todas las actividades que deben seguirse para llevar a cabo un procedimiento de tratamiento y procesamiento de grandes cantidades de información para el análisis de datos climáticos, siguiendo una serie de métodos y estrategias Big Data. El trabajo se desarrolló principalmente en la herramienta Pentaho como entorno de trabajo central, en la cual se investigó los módulos para tratamiento de datos y las demás funciones que se puedan utilizar. Se tomó como referencia un modelo ETL (Extract, Transform, Load) para la extracción, transformación y carga de datos ambientales en bodegas de datos DATA WAREHOUSE (Data WareHouse) o NoSQL (No solo relacionales) con el fin de automatizar y mejorar el proceso de consulta y análisis. Los datos de prueba fueron adquiridos del IDEAM (Instituto de Hidrología, Meteorología y Estudios Ambientales) para el desarrollo y validación del procedimiento para el análisis de comportamientos de cambios climáticos.

Abstract

Currently the large amount of data in different areas is growing continuously at a high speed, with great variety in your data and where you have to preserve the integrity of them so that their value is not lost. Here the analysis of Big Data plays a fundamental role by applying modern software tools to store, process and discover valuable information that contributes to the solution of various problems.

This work contains a detailed description of all the activities that must be followed to carry out a process of treatment and processing large amounts of information for the analysis of climatic data, following a series of Big Data methods and strategies. The work was mainly developed in the Pentaho tool as a central work environment, in which the modules for data processing and other functions that could be used were investigated. An ETL model (Extract, Transform, Load) was taken as reference for the extraction, transformation and loading of environmental data in data warehouses or NoSQL (Not only relational) in order to automate and improve the process of consultation and analysis. The test data was acquired from the IDEAM (Instituto de Hidrología, Meteorología y Estudios Ambientales) for the development and validation of the procedure for the analysis of climate change behavior.

Tabla de contenidos

Tabla de Figuras

1 Descripción del proyecto

1.1 Planteamiento del problema.

¿Es posible mejorar el análisis del comportamiento del clima usando Big Data? ¿Si es así, cuál sería el procedimiento?

Las repercusiones generadas por la falta de predicciones correctas y oportunas influyen críticamente en finanzas, seguridad alimentaria y prevención de desastres en nuestra región y el país. Según *Weather Analytics*, una compañía que proporciona datos climáticos, estima que el 33% del PIB mundial se ve afectado por el clima (Fernández, 2018). Este problema se debe a que el estudio para llevar a cabo una predicción es un proceso arduo y que necesita el análisis de una gran cantidad de datos recolectados a través de los años.

Para hacer una correcta predicción del clima o cualquier estudio climático existen grandes volúmenes de datos ambientales, generados por estaciones de monitoreo ambiental, estos datos se utilizan para: describir las variables en el tiempo según la localidad de la toma de datos, conocer el comportamiento de las medidas, descubrir la relación entre los datos y la dinámica de los fenómenos climáticos. Las herramientas que se usan actualmente en la meteorología no son suficientes para analizar la gran cantidad de datos acumulados a través de los años, esto actualmente representa un problema que requiere el uso de herramientas informáticas más avanzadas para su almacenamiento, tratamiento, análisis y extracción de los datos que será relevante para mejorar la predicción del tiempo.

Mediante el uso de Big Data, la información histórica resulta tan valiosa como la información más reciente. Permite mapear distintas tendencias y patrones que se pueden utilizar para predecir

mejor qué ocurrirá en un futuro. Y conocer lo que está por venir implica la existencia de soluciones más viables para gestionar potenciales problemas (Nathan Sykes, 2018).

1.2 Justificación

A inicios de marzo del 2017, la iniciativa de innovación de datos masivos Big Data de las Naciones Unidas, y *Western Digital Corporation*, un líder global en soluciones y tecnologías de almacenamiento de datos, anunciaron una alianza para lanzar el desafío "*Data for Climate Action*" (Datos para una Acción Climática). La iniciativa tiene como objetivo generar trabajos de investigación originales y herramientas que demuestren cómo la innovación orientada a los datos pueda aportar información a las soluciones y transformar los esfuerzos para combatir el cambio climático (TAMAYO NEYRA ANTONIO, 2017).

En Colombia el IDEAM desde el año 2017 le apuesta a estas tecnologías para poder medir la actual deforestación y emisiones de carbono presentadas en el país, por eso renovaron su centro de datos para el procesamiento de variables ambientales (TECNÓSFERA, 2017). La presente propuesta se encuentra alineada tanto a los intereses nacionales como internacionales, con lo que se pretende dar un inicio en nuestra región a la investigación sobre herramientas Big Data aplicado al tratamiento de datos climáticos que servirá para futuros trabajos fructíferos acerca de predicción del clima mediante la realización de este procedimiento.

1.3 Delimitación

1.3.1 Objetivo General:

- Elaborar un procedimiento para el análisis de comportamientos de cambio climático utilizando el modulo Big Data de la herramienta Pentaho Data Integration y datos climáticos de la ciudad de Cúcuta – Norte de Santander

1.3.2 Objetivos específicos:

- Estudiar tecnologías asociadas a Big Data para la construcción de modelos de análisis de datos.
- Examinar un modelo de predicción de comportamiento climático a partir de fuentes de datos ambientales.
- Desarrollar un procedimiento de aplicación de Big Data en el caso de estudio de análisis de cambio climático

1.3.3 Acotaciones:

El trabajo usará la herramienta de software libre *Pentaho Data Integration*, herramientas relacionadas con Big Data y datos climatológicos suministrados por el IDEAM.

Variables disponibles de la fuente de datos suministrada por el IDEAM:

- Temperatura mínima: es la temperatura mínima promediada de las lecturas de temperatura registradas, estos datos pueden promediarse por mes, o día.
- Temperatura máxima: es la temperatura máxima promediada de las lecturas de temperatura registradas, estos datos pueden promediarse por mes, o día.

- Humedad relativa mínima: es la humedad relativa mínima promediada de las lecturas de humedad relativa obtenidas, estos datos pueden promediarse por mes o día.
- Humedad relativa máxima: es la humedad relativa máxima promediada de las lecturas de humedad relativa obtenidas, estos datos pueden promediarse por mes o día
- Precipitación pluvial o de granizada: son las medidas del volumen o cantidad de precipitación que se llevó en un determinado tiempo.
- Velocidad del viento: son los datos de la velocidad del viento registrada, pueden promediarse ya sea mensualmente o diariamente de las lecturas de velocidad de viento recopiladas.

Es importante resaltar que no se tiene esperado obtener un pronóstico del clima sino un documento con el procedimiento a realizar para mejorar el procesamiento del análisis de datos climáticos usando herramientas Big Data.

1.4 Metodología

El tipo de metodología escogida es descriptiva experimental debido a que se desarrolla un procedimiento partiendo de un modelo propuesto. El procedimiento incluyó en su estudio tres fases principales en donde se defienden las variables independientes integridad de los datos y la confianza. A lo largo del trabajo se describen las actividades hechas, los archivos de entrada, archivos de salida y los posibles variantes a cada actividad. Por último, se anexan los archivos que se usaron.

Para la ejecución del procedimiento se plantean tres ciclos, cada ciclo está asociado al desarrollo de uno de los objetivos específicos presentados en la sección 1.3.2, estos son a su vez

caracterizados en una serie de actividades. Los ciclos y actividades de la metodología se resumen en la siguiente tabla.

Tabla 1:
Metodología propuesta

Ciclo	**Objetivo asociado**	**Actividades**
Revisión y refinamiento del marco conceptual y preparación de los datos ambientales	• Estudiar tecnologías asociadas a Big Data para la construcción de modelos de análisis de datos.	• Capacitación en conceptos básicos de Big Data mediante un curso online. • Exploración de las herramientas. • Instalación de las herramientas a utilizar. • Elaboración del marco conceptual. • Recolección de los datos.
Búsqueda y estudio del modelo ETL para el procesamiento de datos ambientales	• Examinar un modelo de predicción de comportamiento climático a partir de fuentes de datos ambientales.	• Exploración de modelos ETL. • Selección del modelo. • Apropiación del modelo. • Revisión de conceptos básicos acerca del modelo.
Aplicación de las fases del modelo y descripción de las actividades realizadas	• Desarrollar un procedimiento de aplicación de Big Data en el caso de estudio de análisis de cambio climático	• Elaboración de la fase de prerrequisitos o extracción de los datos. • Construcción de la fase principal • Analizar el proceso, mostrar resultados y concluir.

Fuente: elaboración propia.

2 Marco teórico y estado del arte

2.1 Marco Conceptual

- Dato: En programación, un dato es la expresión general que describe las características de las entidades sobre las cuales opera un algoritmo. En estructura de datos, es la parte mínima de la información ("DATA IS THE NEW OIL," 2016).
- Información: Está constituida por un grupo de datos ya supervisados y ordenados, que sirven para construir un mensaje basado en un cierto fenómeno o ente. La información permite resolver problemas y tomar decisiones, ya que su aprovechamiento racional es la base del conocimiento (Julián Pérez Porto & Gardey, 2012).
- Business Intelligence (BI): Según Gartner "La Inteligencia de Negocio (BI) es un término genérico que incluye las aplicaciones, la infraestructura, las herramientas y las mejores prácticas que permiten el acceso y el análisis de la información para mejorar y optimizar las decisiones y rendimiento" (Laney, 2012).
- Sistema Transaccional: Un sistema transaccional es un tipo de sistema de información diseñado para recolectar, almacenar, modificar y recuperar todo tipo de información que es generada por las transacciones en una organización. Una transacción es un evento o proceso que genera o modifica la información que se encuentra eventualmente almacenada en un sistema de información ("Definición de Sistema transaccional (sistema de procesamiento de transacciones)," 2018)

- Sistema BI (Business Intelligence): Un sistema de Business Intelligence es un software o aplicativo que tiene como finalidad transformar los datos de una compañía en información y conocimiento para obtener una ventaja competitiva. Estas herramientas de Business Intelligence permiten reunir, depurar, transformar los datos obtenidos de operaciones diarias como transacciones en información estructurada lista para su explotación directa mediante análisis y conversión en conocimiento que sirva para el soporte a la toma de decisiones de negocio. Permite adaptar la frecuencia y el formato de la información al gusto de nuestras necesidades. Así la herramienta Business Intelligence debe adaptarse a las necesidades de un director de almacén, a un director comercial o a un director general proveyendo indicadores de utilidad para cada uno de ellos ("¿Para qué sirve un sistema de Business Intelligence?," 2017).
- Big data y BI: Big Data nos permite tratar un gran volumen de datos, tanto estructurados como desestructurados, y que se reciben a una velocidad muy alta. Con esta información, las herramientas de BI nos permiten realizar análisis predictivos y avanzados, que nos ayudan en la toma de decisiones estratégicas en función de una predicción de comportamiento basada en datos reales que permiten reducir el umbral de error. Es decir, son términos complementarios, la definición de Big data tiene varias variantes que se verán más adelante. ("Business Intelligence y Big Data. ¿Son lo mismo?," 2018)
- Data Warehouse: Es un repositorio de datos que proporciona una visión general, común, integrada y centralizada de los datos de la organización. Además, permite el

análisis de grandes cantidades de información de forma rápida (Corporación Colombia Digital, 2017).

- ETL (*Extract, Transform, Load*): Es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en otra base de datos o Data Warehouse (Pinto, 2017).
- SQL: *(Structured Query Language*) es un lenguaje de consultas estándar e interactivo para la obtención de información desde una base de datos y para actualizarla (Rouse, 2016b).
- NoSQL (*Not Only SQL*): Abarca una amplia gama de tecnologías y arquitecturas, busca resolver los problemas de escalabilidad y rendimiento de Big data que las bases de datos relacionales no fueron diseñadas para abordar. Contrariamente a las ideas falsas causadas por su nombre, NoSQL no prohíbe el lenguaje estructurado de consultas (SQL). Si bien es cierto que algunos sistemas NoSQL son totalmente no-relacionales, otros simplemente evitan funcionalidades relacionales seleccionadas como esquemas de tablas fijas y operaciones conjuntas (Rouse, 2016a) .
- *Web logs*: Un Web log se compone de diferentes piezas de información, cada una de las cuales sirve para un propósito específico. Ejemplos típicos son el texto de etiquetas de lenguajes XML y HTML (Aguilar, 2013).
- *Map Reduce*: Es un modelo de programación fuertemente orientado a la ejecución paralela y distribuida entre múltiples computadoras, que se utiliza para trabajar con grandes colecciones de datos. La idea de MapReduce es ofrecer una forma simple, rápida, escalable y resistente a fallos para manipular enormes cantidades de datos. En

la terminología de MapReduce a estas manipulaciones se les conoce como trabajos (Feregrino, 2018).

- *Hadoop*: Apache Hadoop es un framework de código abierto que permite el procesamiento distribuido de grandes conjuntos de datos en varios clusters de ordenadores, pero que a ojos del usuario parece un único ordenador. Hadoop separa y distribuye automáticamente los archivos que contienen los datos, además de dividir el trabajo en tareas más pequeñas y ejecutarlas de manera distribuida y recuperarse de posibles fallos automáticamente y de forma transparente al usuario ("Fundamentos de Apache Hadoop y MapReduce," 2018).
- *Cloud Computing*: La computación en la nube (cloud computing) es una tecnología que permite acceso remoto a softwares, almacenamiento de archivos y procesamiento de datos por medio de Internet, siendo así, una alternativa a la ejecución en una computadora personal o servidor local. En el modelo de nube, no hay necesidad de instalar aplicaciones localmente en computadoras (SalesForce, 2018).
- *Zettabytes*: Un zettabyte es una unidad de almacenamiento de información cuyo símbolo es el ZB, equivale a 10^{21} bytes.
- *Petabytes*: Un petabyte es una unidad de almacenamiento de información cuyo símbolo es PB, y equivale a 10^{15} bytes.
- Paso (*Step*): Los pasos están agrupados por categorías y cada uno está diseñado para cumplir una función determinada. Cada paso tiene una ventana de configuración específica, donde se determinan los elementos a tratar y su forma de comportamiento (Hernandez, 2015).

- Transformación (*Transform*): La transformación es el elemento básico de diseño de los procesos ETL en Pentaho. Una transformación se compone de pasos o steps, que están enlazados entre sí a través de los saltos o hops, ver Figura 1 (Espinosa, 2015).

Figura 1. Descripción de transformación

Fuente: (Roberto Espinosa, 2010)

- Trabajo (*Job*): Un trabajo o job es similar al concepto de proceso. Un proceso es un conjunto sencillo o complejo de tareas con el objetivo de realizar una acción determinada. En los trabajos se pueden utilizar pasos específicos (que son diferentes a los disponibles en las transformaciones) como para recibir un fichero vía ftp, mandar un email y ejecutar un comando. Además, se pueden ejecutar una o varias transformaciones que se haya diseñado y orquestar una secuencia de ejecución de ellas, ver Figura 2. Los trabajos estarían en un nivel superior a las transformaciones (Espinosa, 2015).

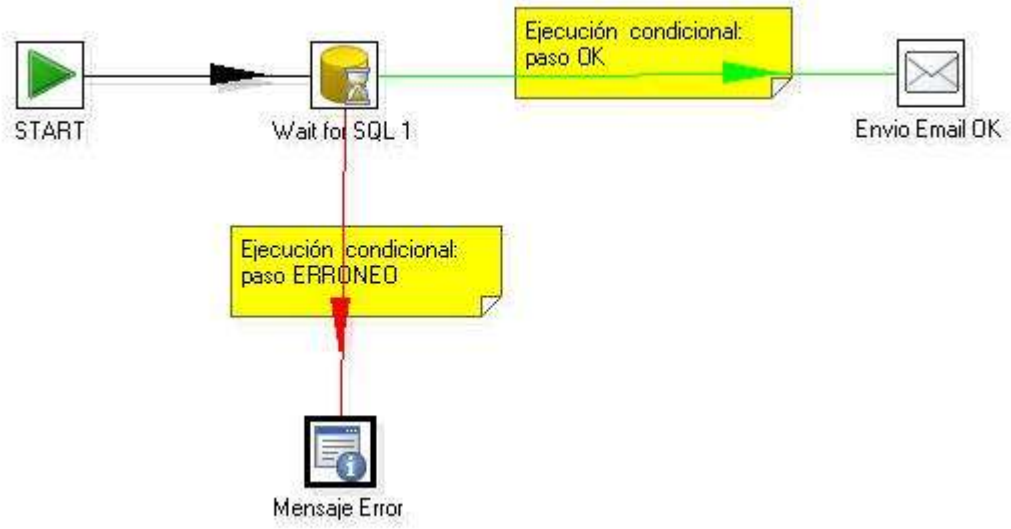


Figura 2. Descripción de Trabajo

Fuente: (Espinosa, 2015)

2.2 ¿Qué es Big Data?

Big Data (grandes datos, grandes volúmenes de datos o macrodatos como recomienda utilizar la Fundación Fundéu BBVA "Fundación del español urgente") supone la confluencia de una multitud de tendencias tecnológicas que venían madurando desde la primera década del siglo XXI, y que se han consolidado durante los últimos años, cuando han explosionado e irrumpido con fuerza en organizaciones y empresas, en particular, y en la sociedad, en general: movilidad, redes sociales, aumento de banda ancha y reducción de su coste de conexión a internet, medios sociales (en particular las redes sociales), internet de las cosas, geolocalización, meteorología , y de modo muy significativo la computación en la nube (*cloud computing*) (Aguilar, 2013).

Se están generando grandes cantidades de datos de forma exponencial, pero el verdadero reto está en poder analizar y extraer valor de los mismos ya que no se están aprovechando ni un

pequeño porcentaje de lo que se tiene a nuestro alcance. En un mundo que está siendo impulsado por los datos. Estos deben ser aprovechados ya que permiten a las empresas poder tomar mejores decisiones para mejorar en todos los aspectos. También es una información valiosa para poder estudiar y buscar curas para enfermedades (Mesa, 2018).

La consultora IDC (*International Data Corporation*), realiza un estudio anual sobre estadísticas referentes al crecimiento de los datos, dicho estudio llamado "Estudio del Universo Digital" estima que en el 2013 habían 4.4 zettabytes de datos digitales en todo el mundo, pero para el 2020 habrán 44 zettabytes o lo que es lo mismo 44 trillones de gigabytes, es decir, desde el año 2013 al 2020 la data global crecerá en un factor de 10, por lo que se está duplicando su tamaño cada dos años (Mesa, 2018).

Según (Yllanes, 2012) otro dato importante suministrado por IDC es que se estima que "Menos del 0,5% de todos los datos del mundo están siendo analizados". Por otra parte, se pueden observar estadísticas sobre lo que ocurre en internet en tan solo 1 minuto y cómo se ha ido incrementado desde el año 2016 al 2017 en Figura 3.

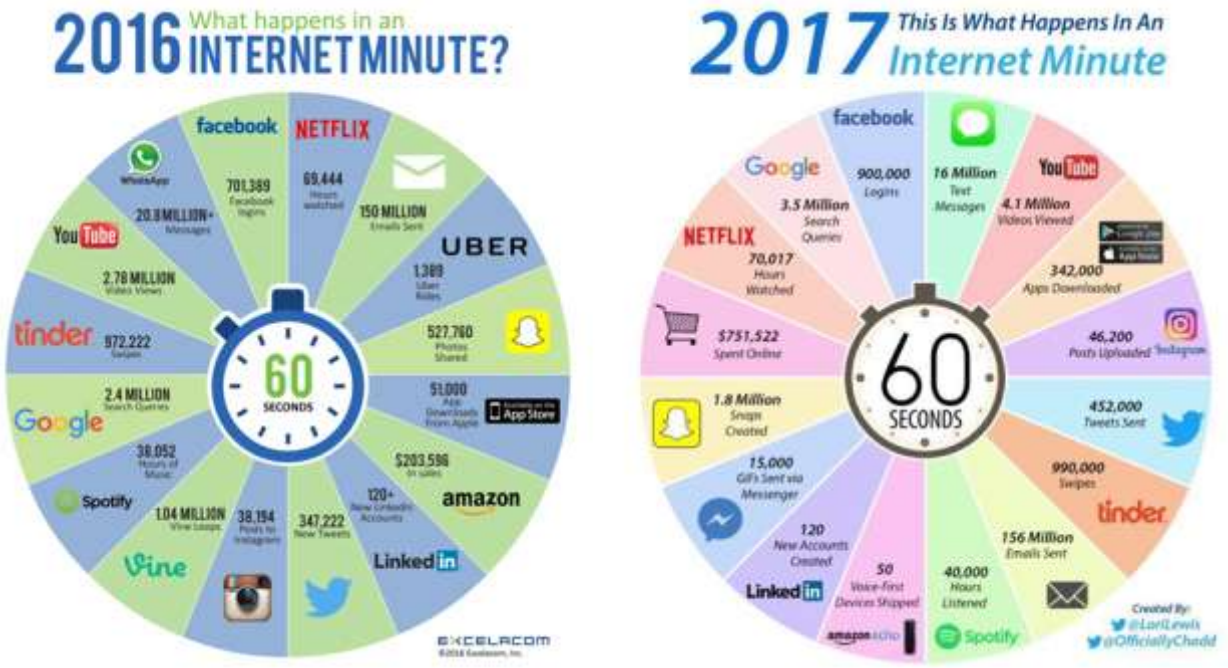


Figura 3. Estadísticas de crecimiento de la data mundial

Fuente: (Mesa, 2018)

Según (Aguilar, 2013) un dato significativo, Walmart la gran cadena de almacenes de los Estados Unidos, posee bases de datos con una capacidad de 2,5 petabytes, y procesa más de un millón de transacciones cada hora. Los Big Data están brotando por todas partes y utilizándolos adecuadamente proporcionaran una gran ventaja competitiva a las organizaciones y empresas. En cambio, su ignorancia producirá grandes riesgos en las organizaciones y no las hará competitivas. Para ser competitivas en el siglo actual, como señala Franks (2012): "Es imperativo que las organizaciones persigan agresivamente la captura y análisis de estas nuevas fuentes de datos para alcanzar los conocimientos y oportunidades que ellas ofrecen".

2.3 Definición de Big Data

Según (Aguilar, 2013) no existe unanimidad en el concepto de Big Data, aunque si un cierto consenso en la fuerza disruptiva que supone los grandes volúmenes de datos y su necesidad de captura, almacenamiento y análisis. Se han seleccionado aquellas definiciones realizadas por instituciones relevantes y con mayor impacto mediático y profesional. En general existen diferentes aspectos donde casi todas las definiciones están de acuerdo y con conceptos consistentes para capturar la esencia de Big Data: crecimiento exponencial de la creación de grandes volúmenes de datos, orígenes o fuentes de datos distintas y la necesidad de su captura, almacenamiento y análisis para conseguir el mayor beneficio para organizaciones y empresas junto con las oportunidades que ofrecen y los riesgos de su no adopción.

La primera definición que se tiene es la de Adrian Merv, vicepresidente de la consultora Gartner, que en la revista Teradatta Magazine, del primer semestre de 2011, define este término como "Big Data excede el alcance de los entornos de hardware de uso común y herramientas de software para capturar, gestionar y procesar los datos dentro de un tiempo transcurrido tolerable para su población de usuarios" (Merv, 2011) .

Otra definición muy significativa es del *Mckinsey Global Institute* que define el término del siguiente modo: "Big Data se refiere a los conjuntos de datos cuyo tamaño está más allá de las capacidades de las herramientas típicas de software de bases de datos para capturar, almacenar, gestionar y analizar". Esta definición puede variar para cada sector, dependiendo de cuales sean los tipos de herramientas de software normalmente disponibles; y cuales, los tamaños típicos de los conjuntos de datos en ese sector o industria. Big Data en muchos sectores hoy en día, varía

desde decenas de terabytes a petabytes y ya casi exabytes (James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, 2011).

Otra fuente de referencia es la consultora tecnológica lDC anteriormente nombrada, que apoyándose en estudios suyos propios, considera que: “Big Data es una nueva generación de tecnologías, arquitecturas y estrategias diseñadas para capturar y analizar grandes volúmenes de datos provenientes de múltiples fuentes heterogéneas a una alta velocidad con el objeto de extraer valor económico de ellos” (Yllanes, 2012).

La empresa multinacional de auditoria Deloitte lo define como: “El término que se aplica a conjuntos de datos cuyo volumen supera la capacidad de las herramientas informáticas (computación) de uso común, para capturar, gestionar y procesar datos en un lapso de tiempo razonable. Los volúmenes de Big Data varían constantemente, y actualmente oscilan entre algunas decenas de terabytes hasta muchos petabytes para un conjunto de datos individual” (Erika Díaz de Argandoña, 2016).

Otra definición muy acreditada por venir de la mano de la consultora Gartner es: "Big Data son los grandes conjuntos de datos que tiene tres características principales: volumen (cantidad), velocidad (velocidad de creación y utilización) y variedad (tipos de fuentes de datos no estructurados, tales como la interacción social, video, audio, cualquier cosa que se pueda clasificar en una base de datos)". Estos factores, naturalmente, conducen a una complejidad extra de los Big Data; en síntesis "‘Big Data’ es un conjunto de datos tan grandes como diversos que rompen las infraestructuras de TI (Tecnologías de la Información) tradicionales”.Gartner considera que la esencia importante de Big Data no es tanto el tema numérico, sino todo lo que

se puede hacer si se aprovecha el potencial y se descubren nuevas oportunidades de los grandes volúmenes de datos (Hung LeHong, 2012).

En suma, la definición de Big Data puede variar según las características de las empresas.

Para unas empresas prima el volumen; para otras, la velocidad; para otras, la variabilidad de las fuentes. Las empresas con mucho volumen o volumetría van a estar interesadas en capturar la información, guardarla, actualizarla e incorporarla en sus procesos de negocio; pero hay empresas que, aunque tengan mucho volumen, no necesitan almacenar, sino trabajar en tiempo real y a gran velocidad. Otras, por el contrario, pueden estar interesadas en gestionar diferentes tipos de datos (Aguilar, 2013).

Un ejemplo clásico son los sistemas de recomendación: sistemas que en tiempo real capturan información de lo que está haciendo el usuario en la Web, lo combina con la información histórica de ventas, lanzando en tiempo real las recomendaciones. Otras empresas tienen otro tipo de retos como fuentes heterogéneas, y lo que necesitan es combinarlas. La captura es más compleja, ya que hay que combinar en un mismo sitio y analizarla (Aguilar, 2013).

Otras áreas están usando el Big Data para mejorar sus procesos de análisis, como es el caso de redes sociales, estudios de mercadeo, investigaciones criminalísticas en ciertas ciudades grandes y la meteorología, estas áreas de estudio centran sus esfuerzos en mejorar los clásicos procedimientos que debido al crecimiento y variabilidad de los datos ya están quedándose atrás en la búsqueda de soluciones a problemas emergentes.

2.4 Tipos de datos en Big Data

Según (Aguilar, 2013) en Big Data se consideran los datos en diferentes tipos según su organización y estructura, estos tipos pueden ser estructurados, semiestructurados y no

estructurados. Las bases de datos relacionales se consideran estructuradas, pero aún no son Big Data propiamente, necesitan un tratamiento para llegar a serlo llamado ETL. Como resultado de este proceso se obtiene una Data Warehouse o una base de datos NoSQL, que serán explicados más adelante pero básicamente son almacenes de datos optimizados para datos masivos. Los datos semiestructurados y no estructurados necesitan un tratamiento más exhausto y complejo que se verá en la sección 3.1 y se encarga de darle la estructura a los datos, lo cual es necesario para poder ser manipulados eficazmente.

2.4.1 Datos estructurados.

Son los datos de las bases de datos relacionales, las hojas de cálculo y los archivos, fundamentalmente. Los datos estructurados se componen de piezas de información que se conocen de antemano, vienen en un formato especificado, y se producen en un orden especificado. Estos formatos facilitan el trabajo con dichos datos. Formatos típicos son: fecha de nacimiento (DD, MM, AA); documento nacional de identidad o pasaporte (por ejemplo, 8 dígitos y una letra); número de la cuenta corriente en un banco (20 dígitos).

Según (Aguilar, 2013), la mayoría de las fuentes de datos tradicionales son datos estructurados, datos con formato o esquema fijo que poseen campos fijos. En estas fuentes, los datos vienen en un formato bien definido que se especifica en detalle, y que conforma las bases de datos relacionales.

2.4.2 Datos semiestructurados.

Según (Aguilar, 2013), son los datos que tienen un flujo lógico y un formato que puede ser definido, pero no es fácil su comprensión por el usuario. Datos que no tienen formatos fijos, pero contienen etiquetas y otros marcadores que permiten separar los elementos dato. La lectura de

datos semiestructurados requiere el uso de reglas complejas que determinan cómo proceder después de la lectura de cada pieza de información. Un ejemplo típico de datos semiestructurados son los registros Web logs de las conexiones a Internet. Un Web log se compone de diferentes piezas de información, cada una de las cuales sirve para un propósito específico. Ejemplos típicos son el texto de etiquetas de lenguajes XML y HTML.

2.4.3 Datos no estructurados.

Según (Aguilar, 2013), los datos no estructurados son datos sin tipos predefinidos. Se almacenan como "documentos" u "objetos" sin estructura uniforme, y se tiene poco o ningún control sobre ellos. Datos de texto, video, audio, fotografía son datos no estructurados. Por ejemplo, las imágenes se clasifican por su resolución en píxeles. Datos que no tienen campos fijos; ejemplos típicos son: audio, video, fotografías, documentos impresos, cartas, hojas electrónicas, imágenes digitales, formularios especiales, mensajes de correo electrónico y de texto, formatos de texto libre como correos electrónicos, mensajes instantáneos, artículos, libros, mensajes de mensajería instantánea tipo WhatsApp, Line, Joyn, Viber, Line, WeChat, Spotbros. Al menos, el 80% de la información de las organizaciones no reside en las bases de datos relacionales o archivos de datos, sino que se encuentran esparcidos a lo largo y ancho de la organización; todos estos datos se conocen como datos no estructurados.

Datos más difíciles de dominar por los analistas son los datos no estructurados, pero su continuo crecimiento ha provocado el nacimiento de herramientas para su manipulación como es el caso de *MapReduce*, *Hadoop* o bases de datos *NoSQL*. Se almacenan como Ejemplos típicos de datos que no tienen campos fijos: audio, video, fotografías, o formatos de texto libre cómo

correos electrónicos, mensajes instantáneos, artículos, libros, mensajes de mensajería instantánea tipo *WhatsApp* o *Wear*.

2.5 Características de Big Data

Según (Fragoso, 2018) en un artículo escrito en la página de IBM cada día se crean 2,5 quintillones de bytes de datos, de forma que el 90% de los datos del mundo actual se han creado en los últimos dos años. Estos datos proceden de todos los sitios: sensores utilizados para recoger información del clima, entradas (posts) en sitios de medios sociales, imágenes digitales, fotografías y videos, registros de transacciones comerciales y señales GPS de teléfonos celulares, por citar unas pocas referencias.

Big Data al igual que la nube (*cloud*) abarca diversas tecnologías. Los datos de entrada a los sistemas de Big Data pueden proceder de redes sociales, logs, registros de servidores Web, sensores de flujos de tráfico, imágenes de satélites, flujos de audio y de radio, transacciones bancarias, MP3 de música, contenido de páginas Web, escaneado de documentos de la administración, caminos o rutas GPS (Sistema de Posicionamiento Global), telemetría de automóviles, datos de mercados financieros. IBM plantea como también hizo Gartner que Big Data abarca tres grandes dimensiones, conocidas como el "Modelo de las tres V" (3 V o V3): volumen, velocidad y variedad (Fragoso, 2018).

Existe un gran número de puntos de vista para visualizar y comprender la naturaleza de los datos y las plataformas de software disponibles para su explotación; la mayoría incluirá una de estas tres propiedades V en mayor o menor grado. Sin embargo, algunas fuentes contrastadas, como es el caso de IBM, cuando tratan las características de los Big Data también consideran una cuarta característica que es la veracidad, y que se analizará también para dar un enfoque más

global a la definición y características de los Big Data. Otras fuentes notables añaden una quinta característica, valor.

Según Doug Laney (considerado padre del modelo de las 3v para Big Data) existen unas características esenciales que encajan en todas las definiciones y él las describió de esta manera en el blog *Gartner Blog Network*. (Laney, 2012)

2.5.1 Volumen.

Las empresas generan grandes volúmenes de datos, desde terabytes hasta petabytes. Las cantidades que hoy parecen enormes, en pocos años serán normales. Se está pasando de la era del petabyte a la era del exabyte, y del 2015 a 2020, se espera entrar en la era del zettabyte. IBM da el dato de 12 terabytes para referirse a lo que crea Twitter cada día solo en el análisis de productos para conseguir mejoras en la eficacia (Aguilar, 2013).

Según (Vega, Ortega, & Aguilar, 2016), en el año 2000 se almacenaron en el mundo 800.000 petabytes. Se espera que en el año 2020 se alcancen los 35 zettabytes (ZB). Solo Twitter genera más de 9 terabytes (TB) de datos cada día. Facebook, 10 TB; y algunas empresas generan terabytes de datos cada hora de cada día del año. Las organizaciones se enfrentan a volúmenes masivos de datos. Las organizaciones que no conocen cómo gestionar estos datos están abrumadas por ello. Sin embargo, la tecnología existe, con la plataforma tecnológica adecuada para analizar casi todos los datos (o al menos la mayoría de ellos, mediante la identificación idónea) con el objetivo de conseguir una mejor comprensión de sus negocios, sus clientes y el marketplace.

IBM plantea que el volumen de datos disponible en las organizaciones hoy día está en ascenso mientras que el porcentaje de datos que se analiza está en disminución.

2.5.2 Velocidad.

Según (Aguilar, 2013) la importancia de la velocidad de los datos o el aumento creciente de los flujos de datos en las organizaciones junto con la frecuencia de las actualizaciones de las grandes bases de datos son características importantes a tener en cuenta. Esto requiere que su procesamiento y posterior análisis, normalmente, ha de hacerse en tiempo real para mejorar la toma de decisiones sobre la base de la información generada. A veces, cinco minutos es demasiado tarde en la toma de decisiones; los procesos sensibles al tiempo como pueden ser los casos de fraude obligan a actuar rápidamente. Suponga los millones de escrutinios de los datos de un banco con el objetivo de detectar un fraude potencial o el análisis de millones de llamadas telefónicas para tratar de predecir el comportamiento de los clientes y evitar que se cambien de compañía.

La importancia de la velocidad de los datos se une a las características de volumen y variedad, de modo que la idea de velocidad no se asocia a la tarea de crecimiento de los depósitos o almacenes de datos, sino que se aplica la definición al concepto de los datos en movimiento, es decir, la velocidad a la cual fluyen los datos. Dado que las empresas están tratando cada día con mayor intensidad, petabytes de datos en lugar de terabytes, y el incremento en fuentes de todo tipo como sensores, chips RFID (Identificación por Radiofrecuencia), chips NFC (Comunicación de Campo Cercano), datos de geolocalización y otros flujos de información que conducen a flujos continuos de datos, imposibles de manipular por sistemas tradicionales.

2.5.3 Variedad.

Según (Aguilar, 2013) las fuentes de datos son de cualquier tipo. Los datos pueden ser estructurados y no estructurados (texto, datos de sensores, audio, video, flujos de clics, archivos

Logs), y cuando se analizan juntos se requieren nuevas técnicas. Suponga el registro en vivo de imágenes de las cámaras de video de un estadio de fútbol o de vigilancia de calles y edificios.

En los sistemas de Big Data las fuentes de datos son diversas y no suelen ser estructuras relacionales típicas. Los datos de redes sociales, de imágenes pueden venir de una fuente de sensores y no suelen estar preparados para su integración en una aplicación. En el caso de la Web, la realidad de los datos es confusa. Los navegadores envían datos diferentes; los usuarios pueden ocultar información, pueden utilizar diferentes versiones de software, bien para comunicarse entre ellos, o para realizar compras, o leer un periódico digital. Sin embargo, los riesgos por la no adopción de las tendencias de Big Data son grandes, ya que:

- La voluminosa cantidad de información puede llevar a una confusión que impida ver las oportunidades y amenazas dentro de nuestro negocio y fuera de él, y perder así competitividad.
- La velocidad y flujo constante de datos en tiempo real puede afectar a las ventas y a la atención al cliente.
- La variedad y complejidad de datos y fuentes puede llevar a la vulneración de determinadas normativas de seguridad y privacidad de datos.

El volumen asociado con Big Data conduce a nuevos retos para los centros de datos que intentan tratar con su variedad. Con la explosión de sensores y dispositivos inteligentes, así como las tecnologías de colaboración sociales, los datos en la empresa llegan a ser muy complejos. Estos incluyen los datos relacionales tradicionales, datos semiestructurados y no estructurados procedentes de páginas Web, archivos de registros Web (*Web log*), datos de los flujos de clics, índices de búsquedas queda, foros de medios sociales, correo electrónico, documentos, datos de

sensores de sistemas activos y pasivos, entre otros. Es decir, variedad representa todos los tipos de datos, y supone un desplazamiento fundamental en el análisis de requisitos desde los datos estructurados tradicionales hasta la inclusión de los datos en bruto, semiestructurados y no estructurados como parte del proceso fundamental de la toma de decisiones. Las plataformas de analítica tradicionales no pueden manejar la variedad. Sin embargo, el éxito de una organización dependerá de su capacidad para resaltar el conocimiento de los diferentes tipos de datos disponibles en ella, que incluirá tanto los datos tradicionales como los no tradicionales. Por citar un ejemplo, el video y las imágenes no se almacenan fácil ni eficazmente en una base de datos relacional, mucha información de sucesos de la vida diaria como el caso de los datos climáticos cambian dinámicamente. Por todas estas razones, las empresas deben capitalizar las oportunidades de la gran cantidad de datos, y deben ser capaces de analizar todos los tipos de datos mencionados.

2.5.4 Veracidad.

Según IBM en su definición de Big Data, al comentar la característica de veracidad proporciona un dato estremecedor: "Uno de cada tres líderes de negocio (directivos) no se fía de las informaciones que utilizan para tomar decisiones". ¿Cómo puede, entonces, actuar con esta información si no se fía de ella? El establecimiento de la veracidad o fiabilidad de Big Data supone un gran reto a medida que la variedad y las fuentes de datos crecen (Fragoso, 2018).

2.5.5 Valor.

Además de las 3 V clásicas con las que todas las fuentes coinciden, y la cuarta que suele señalar IBM, existe una quinta característica que también se suele considerar: el valor. Las organizaciones estudian obtener información de los grandes datos de una manera rentable y

eficiente. Aquí es donde las tecnologías de código abierto tales como *Apache Hadoop* se han vuelto muy populares. *Hadoop*, que se estudia más adelante en el documento, es un software que procesa grandes volúmenes de datos a través de un clúster de centenares, o incluso millares de computadores de un modo muy económico (Fragoso, 2018).

En Figura 4 se representan las V's que caracterizan a Big Data y sus propiedades con las que se asocian:

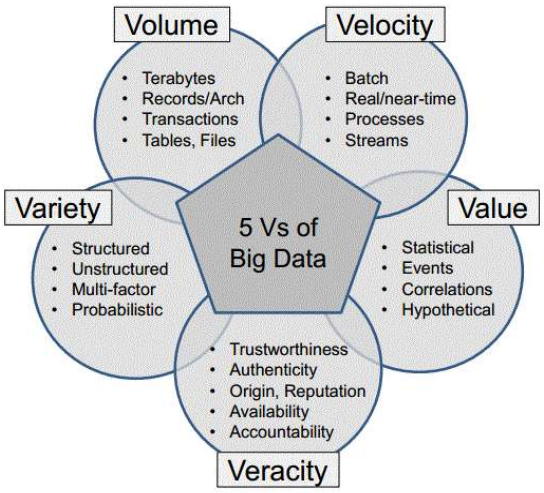


Figura 4. Diagrama las 3 v de Big Data

Fuente: (Webinar, 2015)

2.6 Redes Neuronales Artificiales (ANN)

Según (Salas, n.d.) una red neuronal artificial (ANN *Artificial Neural Network*) es un esquema de computación distribuida inspirada en la estructura del sistema nervioso de los seres humanos.

La característica más importante de las redes neuronales artificiales es su capacidad de aprender a partir de un conjunto de datos de entrenamiento, es decir, es capaz de encontrar un

modelo que ajuste los datos. El proceso de aprendizaje también conocido como entrenamiento de la red que puede ser supervisado o no supervisado.

- El aprendizaje supervisado: consiste en entrenar la red a con un conjunto de datos de entrenamiento compuesto por patrones de entrada y salida. El objetivo del algoritmo de aprendizaje es variar los pesos y ajustarlos de manera tal que la salida generada por la ANN sea lo más cercanamente posible a la verdadera salida dada una cierta entrada. Es decir, la red neuronal trata de encontrar un modelo que se ajuste al proceso que generó la salida y este aprendizaje se llama supervisado pues se conoce el patrón de salida el cual hace el papel de supervisor de la red (Salas, n.d.).
- Aprendizaje no supervisado: se presenta sólo un conjunto de datos a la ANN, y el objetivo del algoritmo de aprendizaje es ajustar los pesos de la red de manera tal que la red encuentre algún patrón, o estructura presente en los datos (Salas, n.d.).

2.6.1 Scikit-Learn

Scikit-learn es la principal librería en Python que existe para trabajar con redes neuronales y *machine learning*, incluye la implementación de un gran número de algoritmos de aprendizaje. La podemos utilizar para clasificaciones, extracción de características, regresiones, agrupaciones, reducción de dimensiones, selección de modelos y pre-procesamiento. Posee una API que es consistente en todos los modelos y se integra muy bien con el resto de los paquetes científicos que ofrece Python. Esta librería también facilita las tareas de evaluación, diagnóstico y validaciones cruzadas ya que proporciona varios métodos de fábrica para poder realizar estas tareas en forma muy simple (Briega, 2015).

Los algoritmos que más se suelen utilizar en los problemas de Machine Learning son los siguientes:

- Regresión Lineal
- Regresión Logística
- Arboles de Decisión
- Random Forest
- SVM o Máquinas de vectores de soporte.
- KNN o K vecinos más cercanos.
- K-means

Todos ellos están todos implementados en la librería librería de Python, Scikit-learn.

2.7 ¿Qué es Pentaho Data Integration?

También se la denomina PDI o Kettle.

Según (Brathwaite & Doug Moran, 2015) Pentaho es una herramienta de las que se denominan ETL. Es decir, una herramienta de extracción de datos de una fuente, transformación de esos datos, y carga de esos datos en otro sitio.

Estas tareas son típicas en procesos de migración, integración con terceros, explotación de Big Data y en general se podría decir que son necesarias en casi cualquier proyecto mediano o grande. Por eso Pentaho nace con la intención de facilitar este trabajo, de forma que no se tenga que entrar en el detalle de la implementación de como se hace cada una de estas tareas, sino que simplemente se especifica qué es lo que se quiere hacer. Por eso en muchos sitios se califica a este tipo de herramientas, herramientas de metadatos, ya que trabajan a nivel de definición

diciendo qué hay que hacer, pero no el detalle del cómo se hace, éste queda oculto a nuestros ojos, lo cual resulta muy interesante en la mayoría de los casos.

Es la herramienta de código abierto más popular disponible. PDI admite una amplia gama de formatos de entrada y salida, incluidos archivos de texto, hojas de datos y motores de bases de datos comerciales y gratuitos. Además, las capacidades de transformación de PDI le permiten manipular datos con muy pocas limitaciones.

La plataforma ha sido desarrollada bajo el lenguaje de programación *JAVA* y tiene un ambiente de implementación también basado en *JAVA*, haciendo así que Pentaho sea una solución muy flexible al cubrir una alta gama de necesidades empresariales.

Si bien las herramientas ETL se utilizan con mayor frecuencia en entornos de almacenes de datos, PDI también se puede usar para otros fines:

- Migración de datos entre aplicaciones o bases de datos.
- Exportación de datos desde bases de datos a archivos planos.
- Cargando datos masivamente en bases de datos
- Limpieza de datos
- Aplicaciones integradoras

PDI es fácil de usar. Cada proceso se crea con una herramienta gráfica donde se especifica qué hacer sin escribir código para indicar cómo hacerlo (Brathwaite & Doug Moran, 2015).

2.7.1 ¿Qué es Spoon?

Spoon es el entorno gráfico estándar de Pentaho, mediante esta Interface Gráfica (UI) se puede diseñar todos los KTR (archivos generados de Pentaho) basados en una tecnología *Rapid*

Application Development (RAD). Las tareas son modeladas tipo *Workflow* ó flujo de trabajo para coordinar recursos, ejecución y dependencias de actividades ETL (Leninmhs, 2018).

La Empresa Gravitar se especializa en Bussiness Intelligence (BI) y es una organización donde se utiliza diariamente de manera profesional la herramienta Pentaho, según la empresa Gravitar, Pentaho es una herramienta de BI desarrollada bajo la filosofía del software libre para la gestión y toma de decisiones empresariales. Es una plataforma compuesta de diferentes programas que satisfacen los requisitos de BI. Ofreciendo soluciones para la gestión y análisis de la información, incluyendo el análisis multidimensional OLAP (*On-Line Analytical Processing*), presentación de informes, minería de datos y creación de cuadros de mando para el usuario (gravitar, 2017).

2.8 ¿Qué es una ETL?

ETL son las siglas en inglés de Extraer, Transformar y Cargar (Extract, Transform and Load). Es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en otra base de datos o Data Warehouse para analizar, o en otro sistema operacional para apoyar un proceso de negocio.

Los procesos ETL también se pueden utilizar para la integración con sistemas heredados o aplicaciones antiguas existentes en las organizaciones que se han de integrar con los nuevos aplicativos (Pinto, 2017).

2.8.1 Extracción.

La primera parte del proceso ETL consiste en extraer los datos desde los sistemas de origen. La mayoría de los proyectos de almacenamiento de datos fusionan datos provenientes de diferentes sistemas de origen. Cada sistema separado puede usar una organización diferente de

los datos o formatos distintos. Los formatos de las fuentes normalmente se encuentran en bases de datos relacionales o ficheros planos, pero pueden incluir bases de datos no relacionales u otras estructuras diferentes. La extracción convierte los datos a un formato preparado para iniciar el proceso de transformación.

2.8.2 Transformación.

La fase de transformación de un proceso de ETL aplica una serie de reglas o funciones sobre los datos extraídos para convertirlos en datos que serán cargados. Algunas fuentes de datos requerirán alguna pequeña manipulación de los datos.

2.8.3 Carga.

La fase de carga es el momento en el cual los datos de la fase anterior (transformación) son cargados en el sistema de destino. Dependiendo de los requerimientos de la organización, este proceso puede abarcar una amplia variedad de acciones diferentes. En algunas bases de datos se sobrescribe la información antigua con nuevos datos. Los Data Warehouse mantienen un historial de los registros de manera que se pueda hacer una auditoría de los mismos y disponer de un rastro de toda la historia de un valor a lo largo del tiempo.

2.8.4 Beneficios de un procedimiento ETL.

A cualquier empresa u organización le beneficia poner en marcha un proceso ETL para mover y transformar los datos que maneja por los siguientes motivos:

- Poder crear una Data Warehouse, es decir, un repositorio central estandarizado de todos los datos de la organización. Por ejemplo, si se tiene un objeto cliente en una base de datos de créditos y otro objeto cliente en la base de datos de tarjetas de crédito,

lo que haría el sistema sería definir, de forma concreta e inequívoca, un registro cliente único con su nombre y apellidos para la organización.

- Posibilita a los directivos tomar decisiones estratégicas basadas en el análisis de los datos cargados en las bases nuevas y actualizadas: la Datamart o Data Warehouse.
- Sirve para integrar sistemas. Las organizaciones crecen de forma orgánica y cada vez se van agregando más fuentes de datos. Esto provoca que comience a surgir nuevas necesidades, como por ejemplo integrar los datos de un banco online con los datos antiguos de un sistema legado.

2.9 Data Warehouse

Es un repositorio de datos que proporciona una visión general, común e integrada (centralizada) de los datos de la organización.

Muchos tipos de datos de negocio se analizan a través de un Data Warehouse. La necesidad de contar con este sistema se hace evidente cuando los requerimientos analíticos de la organización entran en conflicto con el rendimiento de las bases de datos operacionales o transaccionales. Sobre todo, debido a la ejecución de consultas complejas que muchas veces son imposibles para esas bases de datos (Corporación Colombia Digital, 2017). Por lo tanto:

- Una Data Warehouse se emplea para hacer el trabajo analítico, dejando las bases de datos transaccionales libres para centrarse en las transacciones.
- Tiene la capacidad de analizar datos de múltiples fuentes y puede negociar las diferencias en cuanto a esquemas de almacenamiento utilizando procesos de ETL.
- Al integrar datos de múltiples sistemas de origen, permite una visión central en toda la empresa.

- Mantiene el historial de datos incluso si los sistemas transaccionales de origen no lo hacen.
- Mejora los datos, proporcionando códigos y descripciones coherentes e incluso arreglando datos erróneos.
- Presenta la información de la organización de forma coherente.
- Proporciona un único modelo de datos común para todos los datos de interés independientemente de la fuente de los datos.
- Reestructura los datos de manera que tienen sentido para los usuarios de negocios.
- Reestructura los datos de modo que ofrece un excelente rendimiento para consultas analíticas complejas, sin afectar a los sistemas operativos.
- Añade valor a las aplicaciones de negocio operativas, en especial a las de gestión de relaciones con clientes.

2.9.1 Elementos de un Data Warehouse.

- Tabla de hecho (*Fact Table*): es la representación en la Data Warehouse de los procesos de negocio de la organización. Por ejemplo, una venta puede identificarse como un proceso de negocio de manera que es factible, si corresponde en nuestra organización, considerar la tabla de hecho ventas.
- Dimensión (*Datamarts*): es la representación en la Data Warehouse de una vista para un cierto proceso de negocio. Si se regresa al ejemplo de una venta, para la misma se tiene el cliente que ha comprado, la fecha en la que se ha realizado. Estos conceptos pueden ser considerados como vistas para este proceso de negocio. Puede ser interesante recuperar todas las compras realizadas por un cliente. Ello hace entender por qué se identifica como

una dimensión. Las Datamarts se pueden cargar de manera asincrónica y posee datos atómicos.

- Métrica: son los indicadores de negocio de un proceso de negocio. Aquellos conceptos cuantificables que permiten medir nuestro proceso de negocio. Por ejemplo, en una venta se tiene el importe de la misma.

2.9.2 Una clave subrogada.

Según (Bussiness Inteligence Facil, 2015) Una clave subrogada es un identificador único que se asigna a cada registro de una tabla de dimensión. Esta clave, generalmente, no tiene ningún sentido específico de negocio. Son siempre de tipo numérico y preferiblemente un entero autoincremental. Habitualmente, el sistema operacional ya utiliza sus propias claves, aunque suelen ser de tipo carácter y tienen un sentido específico para los componentes de la compañía. Por ejemplo, el código BCN puede utilizarse para referirse a Barcelona, o la cedula de ciudadanía de cada empleado puede ser la clave única de la tabla de empleados. O el código de barras para referirse a un producto. ¿Por qué se necesita, entonces, crear unos nuevos identificadores en el sistema Big Data?

Por varios motivos:

- Fuentes heterogéneas. La Data Warehouse suele alimentarse de diferentes fuentes, cada una de ellas con sus propias claves, por lo que es arriesgado asumir un código de alguna aplicación en particular. ¿Qué ocurriría si en el futuro se añade información de una aplicación que tiene llaves similares a estas? Seguro que esto puede ser un problema, aparecerán datos que no pertenecen a ninguna tabla maestra. ¿Cómo se gestiona? Lo mejor es crear propias claves subrogadas desde el inicio del proyecto.

- Cambios en las aplicaciones origen. Puede ocurrir que cambie la lógica operacional de alguna clave que se hubiese supuesto única, o que siempre debería estar informada. ¿Qué pasará cuando llegue un empleado sin cedula de ciudadanía o con un DNI de otro país? ¿Qué pasará cuando se dé de alta una ciudad extranjera con el código BCN? Lo mejor es crear propias claves subrogadas desde el inicio del proyecto.
- Rendimiento. En la base de datos, ocupa menos espacio un entero que una cadena. Identificar una ciudad con 5 bytes, o una persona con 9 bytes es un desperdicio considerable de espacio. De hecho, no debe tener preocupación del espacio que ocupa, sino el tiempo que se pierde en leerlo. Hay que Recordar que las claves subrogadas las a las tablas de hechos, por lo que cada código es susceptible de repetirse cientos de millones de veces. Conviene optimizarlo al máximo. Lo mejor es crear nuestras propias claves subrogadas desde el inicio del proyecto.

En ocasiones, puede parecer útil aprovechar la lógica que subyace a los elementos para generar nuestras propias claves. Por ejemplo, si se quiere crear una jerarquía de ciudades, nunca se debe caer en la tentación de crear una simple concatenación (y generar el código ESP-CAT-BCN para referirse a Barcelona). También es un error crear una clave compuesta de varios campos. Aunque en el operacional la terna país-CCAA-ciudad sea única, en el Data Warehouse se debe crear una clave subrogada formada por un solo campo entero autoincremental.

2.9.3 Metodologías para la elaboración de una Data Warehouse.

2.9.3.1 Metodología Bill Inmon.

Según (ZALDÍVAR, 2014) Bill Inmon ve la necesidad de transferir la información de los diferentes OLTP (*OnLine Transaction Processing* - Sistemas Transaccionales) de las

organizaciones a un lugar centralizado donde los datos puedan ser utilizados para el análisis. Insiste en conservar las siguientes características.

- Orientado a temas: Los datos en la base de datos están organizados de manera que todos los elementos de datos relativos al mismo evento u objeto del mundo real queden unidos entre sí.
- Integrado: La base de datos contiene los datos de todos los sistemas operacionales de la organización, y dichos datos deben ser consistentes.
- No volátil: La información no se modifica ni se elimina, una vez almacenado un dato, éste se convierte en información de sólo lectura, y se mantiene para futuras consultas.
- Variante en el tiempo: Los cambios producidos en los datos a lo largo del tiempo quedan registrados para que los informes que se puedan generar reflejen esas variaciones.

La información ha de estar a los máximos niveles de detalle. Los Data Warehouse departamentales o Datamarts son tratados como subconjuntos de este Data Warehouse corporativo, que son construidos para cubrir las necesidades individuales de análisis de cada departamento, y siempre a partir de este Data Warehouse Central (del que también se pueden construir los ODS (Operational Data Stores) o similares) (Roberto Espinosa, 2010) .

En la figura 5 se representa el esquema para el uso de esta metodología:

Figura 5 Esquema Metodología Bill Inmon

Fuente: (Roberto Espinosa, 2010)

El enfoque Inmon también se referencia normalmente como Top-down. Los datos son extraídos de los sistemas operacionales por los procesos ETL y cargados en las áreas pequeñas, donde son validados y consolidados en el Data Warehouse corporativo, donde además existen los llamados metadatos que documentan de una forma clara y precisa el contenido del Data Warehouse. Una vez realizado este proceso, los procesos de refresco de los Datamarts departamentales obtienen la información de él, y con las consiguientes transformaciones, organizan los datos en las estructuras particulares requeridas por cada uno de ellos, refrescando su contenido (ZALDÍVAR, 2014).

Al tener este enfoque global, es más difícil de desarrollar en un proyecto sencillo (pues se está intentando abordar el "todo", a partir del cual luego se irá al "detalle").

2.9.3.2 Metodología Ralph Kimball.

Según (Roberto Espinosa, 2010) El Data Warehouse es un conglomerado de todos los Datamarts dentro de una empresa, siendo una copia de los datos transaccionales estructurados de una forma especial para el análisis, de acuerdo al Modelo Dimensional (no normalizado), que incluye las dimensiones de análisis y sus atributos, su organización jerárquica, así como los diferentes hechos de negocio que se quieren analizar. Por un lado, se tienen tablas para las representar las dimensiones y por otro lado tablas para los hechos (las facts tables). Los diferentes Datamarts están conectados entre sí por la llamada (bus structure), que contiene los elementos anteriormente citados a través de las dimensiones conformadas (que permiten que los usuarios puedan realizar consultas de conjuntos sobre las diferentes Datamarts, pues este bus contiene los elementos en común que los comunican). Una dimensión conformada puede ser, por ejemplo, la dimensión cliente, que incluye todos los atributos o elementos de análisis referentes a los clientes y que puede ser compartida por diferentes datas marts (ventas, pedidos, gestión de cobros, etc) (Roberto Espinosa, 2010).

En la figura 6 se representa el esquema para el uso de esta metodología:

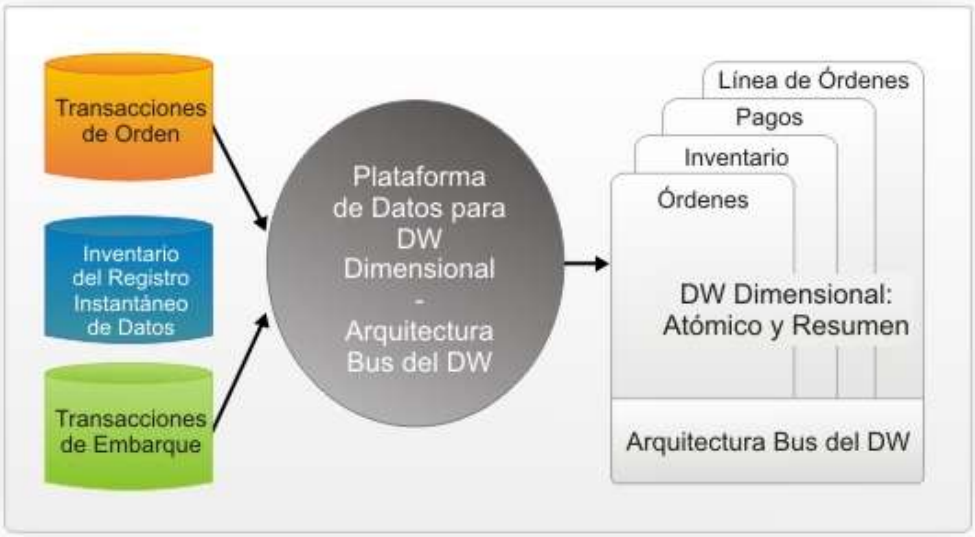


Figura 6 Esquema Metodologia Ralph Kimball

Fuente: (Roberto Espinosa, 2010)

Este enfoque también se referencia como Bottom-up, pues al final el Datawarehouse Corporativo no es más que la unión de los diferentes Datamarts, que están estructurados de una forma común a través del bus structure. Esta característica le hace más flexible y sencillo de implementar, pues se puede construir un Datamarts como primer elemento del sistema de análisis, y luego ir añadiendo otros que comparten las dimensiones ya definidas o incluyen otras nuevas. En este sistema, los procesos ETL extraen la información de los sistemas operacionales y los procesan igualmente en el area stage, realizando posteriormente el llenado de cada uno de los Datamarts de una forma individual, aunque siempre respetando la estandarización de las dimensiones (dimensiones conformadas) (Roberto Espinosa, 2010).

La metodología para la construcción del Data Warehouse incluye las 4 fases que son: Selección del proceso de negocio, definición de la granularidad de la información, elección de

las dimensiones de análisis e identificación de los hechos o métricas. Igualmente define el tratamiento de los cambios en los datos a través de las Dimensiones Lentamente Cambiantes (SCD slowly changing dimensions) (Roberto Espinosa, 2010).

2.10 Estado del arte

2.10.1 Internacional:

BIG DATA EN EL COMPORTAMIENTO DE DATOS CLIMATOLÓGICOS Y ESTRATEGIAS INTERNACIONALES DE REDUCCIÓN DE DESASTRES PARA LA GESTION DE RIESGO AMBIENTAL

Este trabajo introduce el concepto de Big Data para la manipulación y análisis de datos climatológicos, utilizando para su representación y análisis sistemas de información geográfica, y para su gestión y control las EIRD (estrategias internacionales de reducción de desastres). De esta manera se pretende abordar el tema sobre la gestión de riesgo ambiental (Fernando, 2016).

Razón:

En este trabajo se hace una introducción al concepto de big data y su utilización en el área meteorológica planteando propuestas y mecanismos de solución a problemas de riesgo ambiental teniendo en cuenta las estrategias internacionales de reducción de desastres

2.10.2 Nacional:

MODELO PARA EL PROCESO DE EXTRACCIÓN, TRANSFORMACIÓN Y CARGA EN BODEGAS DE DATOS. UNA APLICACIÓN CON DATOS AMBIENTALES

La administración de Data Warehouse requiere de un procesamiento para garantizar la veracidad, integridad y centralización de los datos cuando existen diversas fuentes de información, haciendo necesario utilizar aplicativos especializados para la Extracción,

Transformación y Carga de datos (ETL). Estos aplicativos presentan conflictos en su parametrización, carecen de la implementación de filtros de corrección adaptables a las características de los datos y pueden demandar altos costos para su implementación. El artículo plantea un modelo genérico que aplica las etapas de ETL y permite realizar seguimiento del proceso al mantener un registro histórico de errores filtrados y calcular indicadores para identificar la calidad en el procesamiento. La validación del modelo fue realizada sobre un caso de estudio con datos ambientales. El modelo demostró obtener resultados satisfactorios. Se plantea realizar más validaciones del modelo, en otros ámbitos, incluyendo nuevos tipos y estructuras de datos (Duque Méndez, Hernández Leal, Pérez Zapata, Arroyave Tabares, & Espinosa Gómez, 2016).

Razón:

Se tomará como referencia el modelo realizado en este artículo y el procedimiento que se va a elaborar garantizará la veracidad, integridad y centralización de los datos mediante procesos de filtrado y limpieza.

2.10.3 Regional:

MODELO COMPUTACIONAL PARA EL PRONÓSTICO DEL COMPORTAMIENTO METEOROLÓGICO EN LA CUENCA DEL RÍO PAMPLONITA

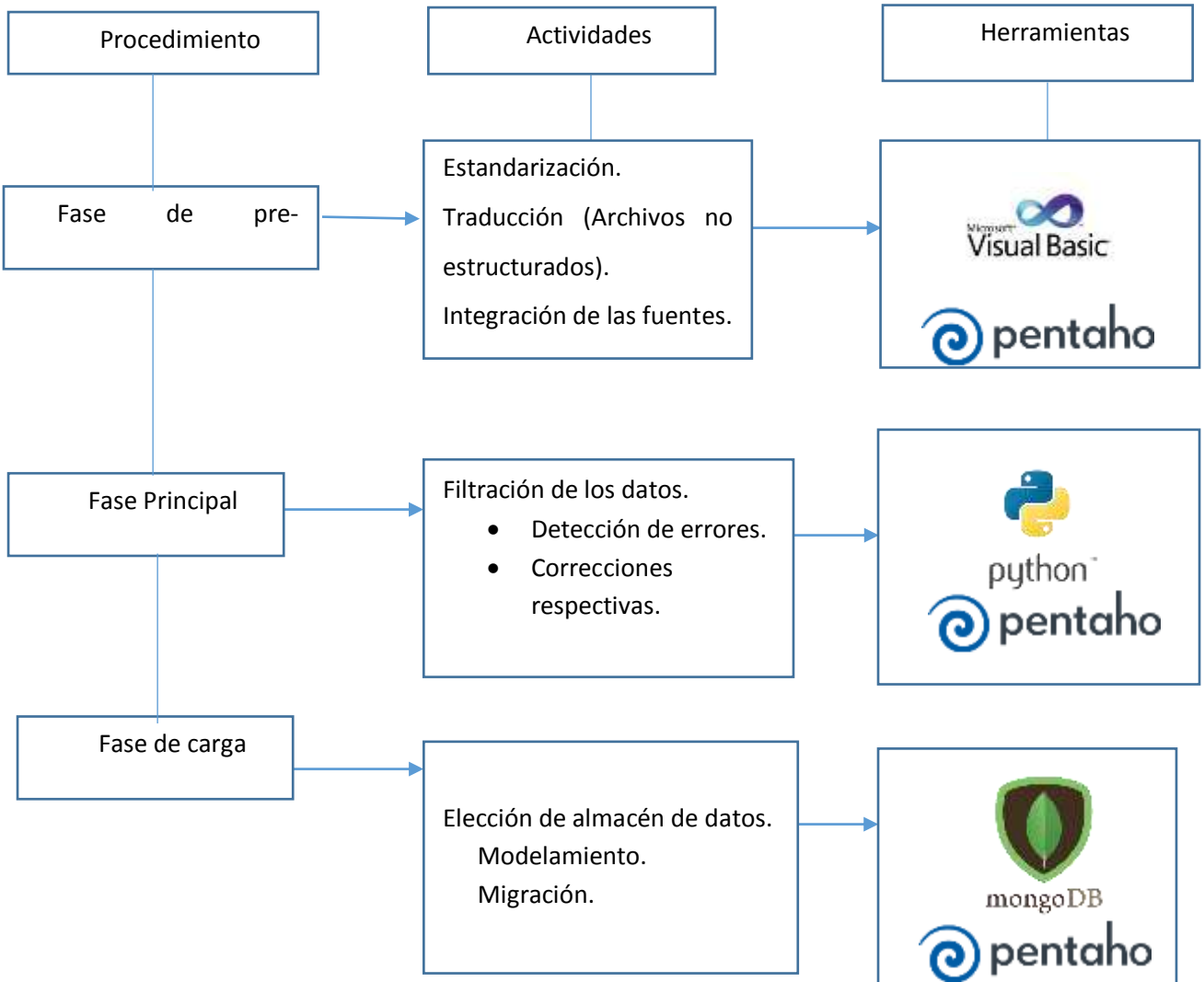
Este trabajo consiste en el desarrollo de un modelo computacional para el pronóstico del comportamiento de la atmósfera partiendo de la identificación de las variables principales que actúan en la atmósfera, estableciendo relaciones entre ellas mediante sistemas de ecuaciones diferenciales en derivadas parciales, las cuales son discretizadas mediante el método de

diferencias finitas y base a estas ecuaciones se desarrolla un código en el lenguaje de la suite MatLab (Lenguaje M), para obtener la solución numérica. Una vez hecho lo anterior se seleccionó una serie temporal de datos correspondientes al año 2011, se tomaron como condiciones iniciales del modelo los datos de los días 1 y 2 de enero a diferentes alturas (2, 1829, 2743, 3658 y 4572 metros), se obtuvo pronóstico para los siguientes días, en particular para los días 3, 6 y 12 de enero. Estos resultados fueron comparados con los datos reales (descargados de bases de datos y obtenidos de satélites, los cuales están previamente validados), lo cual arrojó óptimos resultados en comparación a modelos comerciales globales. Finalmente se desarrolló una interfaz de usuario para facilitar el manejo del modelo (GUILLEN BETANCOURT, 2014).

3 Procedimiento

En este capítulo se explica el modelo utilizado en el procedimiento y el porqué de cada fase, además los objetivos de las actividades que se realizaran en la sección 3.2.

3.1 Procedimiento General



En este segmento se expone el desarrollo de un modelo que aplica las etapas ETL para el tratamiento de grandes cantidades de datos usando variables ambientales. El modelo incluye la posibilidad de tomar diferentes fuentes de datos, también cuenta con la capacidad de hacer cambios en la estructura de los datos sin alterar su contenido, y garantiza la integridad y consistencia de los datos que se almacenan en una Data Warehouse o Base de datos NoSQL que será el objetivo.

3.1.1 Fase de prerrequisitos.

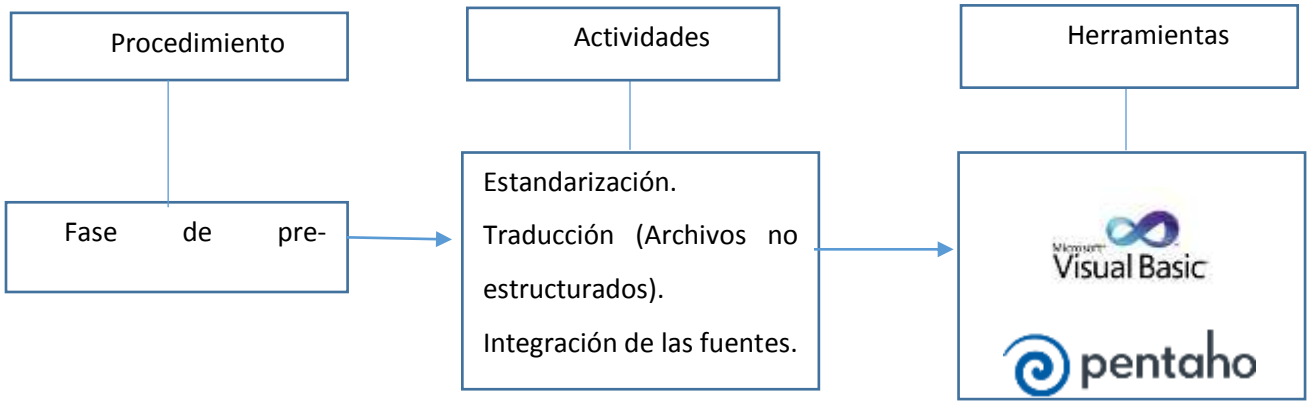


Figura 8 Esquema fase de pre-requisitos

Fuente: Elaboración propia

Los datos pueden ser entregados en varias formas de almacenamiento, como por ejemplo archivos planos y repositorios estructurados (bases de datos), los cuales pueden presentar problemas y errores en su adquisición. Por lo anterior es necesario contar con un proceso llamado ‘traducción’; este proceso consiste en la toma y estandarización de la estructura de los datos, para luego proceder a la carga de los datos en el entorno de la herramienta que se utilice.

3.1.1.1 Traducción

Este proceso tiene como objetivo unificar o tipificar las fuentes de datos obtenidas, esto se hace comparando los diferentes archivos y elaborando una estructura general. Este proceso es lento debido a que hay operaciones que se hacen de una manera manual o utilizando herramientas no optimizadas para manejar tal cantidad de datos como por ejemplo Visual Basic.

Para fuentes de datos de estaciones ambientales este proceso no lleva mucho trabajo, ya que todos los datos están bajo una misma estructura, sin embargo, es necesario realizarla ya que son datos provenientes de archivos planos que hay que limpiar y contiene información extra que no necesitará la próxima fase.

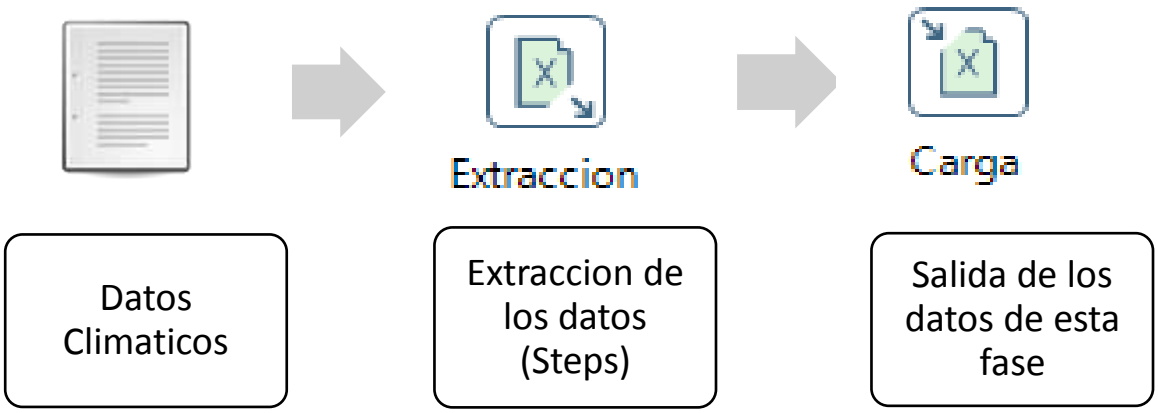


Figura 9 Pasos de extracción

Fuente: Elaboración propia

3.1.1.2 Pasos (Steps) a utilizar en Pentaho

A continuación, se muestra las opciones de extracción de datos que tiene la suite Pentaho y se detallan brevemente en la Tabla 2.

Tabla 2 Pasos de carga en Pentaho

	Nombre del paso	Descripción
1	CSV File input	Archivo de entrada de tipo CSV
2	Data Grid	Ingresa filas de datos estáticos en una cuadrícula, generalmente con fines de prueba, referencia o demostración.
3	De-serialize from file	Leer filas de datos de un cubo de datos.
4	ESRI Shapefile Reader	Lee datos de un archivo de formas ESRI y archivos DBF vinculados
5	Email messages input	Leer servidor POP3 / IMAP y recuperar mensajes
6	Fixed file input	Entrada de archivo fija
7	GZIP CSV Input	Lector paralelo de entrada de archivos GZIP CSV
8	Generate Rows	Generar un número de filas vacías o iguales.
9	Generate random credit card numbers	Generar números de tarjeta de crédito válidos al azar (cheque luhn)
10	Generate random value	Generar valor aleatorio
11	Get File Names	Obtenga nombres de archivos del sistema operativo y envíelos al siguiente paso.
12	Get Files Rows Count	Obtener cuantos registros de un archivo
13	Get SubFolder names	Lee una carpeta principal y devuelve todas las subcarpetas
14	Get System Info	Obtenga información del sistema como fecha del sistema, argumentos, etc.
15	Get data from XML	Obtenga datos de un archivo XML utilizando XPath. Este paso también le permite analizar XML definido en un campo anterior.
16	Get repository names	Enumera información detallada sobre transformaciones y / o trabajos en un repositorio
17	Get table names	Obtenga nombres de tablas de la conexión de la base de datos y envíelos al siguiente paso
18	Google Analytics	Obtiene datos de la cuenta de Google Analytics
19	HL7 Input	Leer datos de flujos de datos HL7.
20	JSON Input	Extraiga partes relevantes de estructuras JSON (archivo o campo entrante) y filas de salida
21	LDAP Input	Leer datos del host LDAP
22	LDIF Input	Leer datos de archivos LDIF
23	Load file content in memory	Cargar contenido del archivo en memoria
24	Microsoft Access input	Leer datos de un archivo de Microsoft Access
25	Microsoft Excel input	Lea datos de Excel y OpenOffice Workbooks (XLS, XLSX, ODS).
26	Mondrian Input	Ejecute y recupere datos utilizando una consulta MDX contra un servidor OLAP de Pentaho Analyzes (Mondrian)
27	Olap input	Ejecute y recupere datos usando una consulta MDX contra cualquier fuente de datos OLAP / XML usando olap4j
28	Property input	Leer datos (clave, valor) de archivos de propiedades.
29	RSS input	Leer feeds RSS
30	S3 CSV input	Entrada CSV S3
31	SAP input	Leer datos de SAP ERP, opcionalmente con parámetros.
32	SAS input	Este paso lee archivos en formato nativo sas7bdat

		(SAS)
33	Salesforce input	Lee información de SalesForce
34	Table input	Leer información de una tabla de base de datos.
35	Text file input	Leer datos de un archivo de texto en varios formatos. Estos datos se pueden pasar a los siguientes pasos ...
36	XBase input	Lee registros de un tipo de base de datos XBase (DBF)
37	XML input stream (StAX)	Este paso es capaz de procesar archivos XML muy grandes y complejos muy rápido.
38	Yaml input	La fuente de YAML leída (archivo o secuencia) los analiza, los convierte en filas y los escribe en una o más salidas.

Fuente: Elaboración propia

3.1.2 Fase principal.

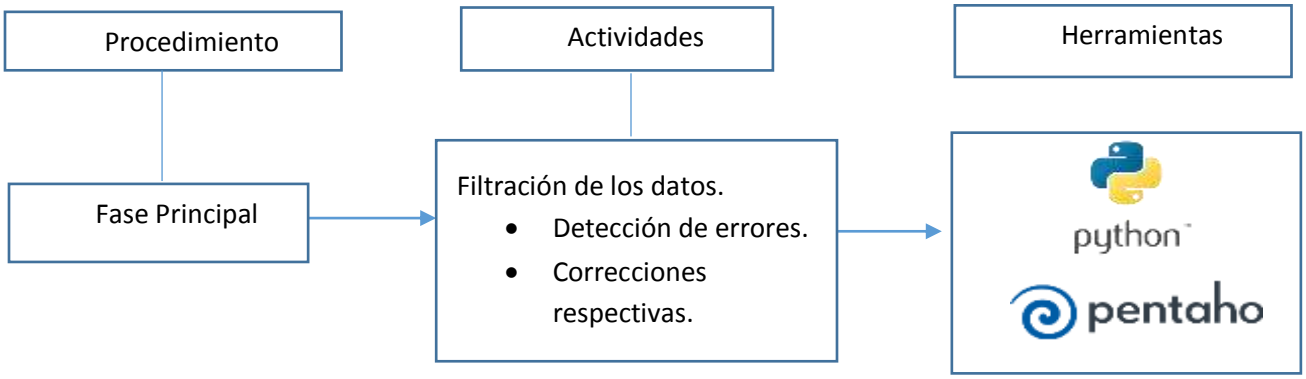


Figura 10 Esquema fase principal

Fuente: Elaboración propia

Consta de detección y corrección, es la fase encargada de recibir los datos organizados en una estructura estándar, construida en la fase de prerrequisitos, y a partir de esto realizar las transformaciones necesarias para que esta información tenga un valor agregado.

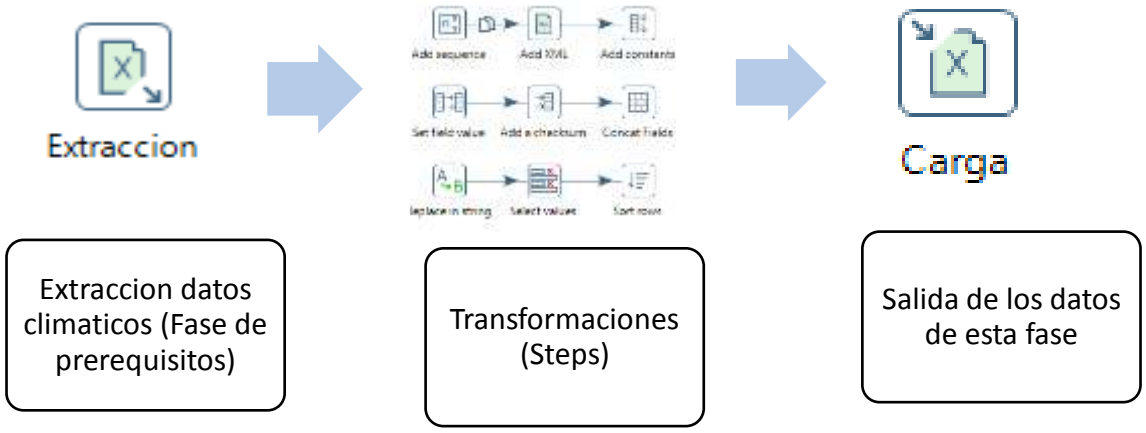


Figura 11 Paso de transformación

Fuente: Elaboración propia

3.1.2.1 Pasos (Steps) a utilizar en Pentaho

A continuación, en la Tabla 3, se muestra y se explica brevemente las opciones de transformación de datos que tiene la suite Pentaho, también se pueden usar pasos de otros módulos de la herramienta para casos específicos.

Tabla 3 Descripción pasos de transformación

Nombre del paso	Descripción

1. Add XML	Codificar varios campos en un fragmento XML
2. Add a checksum	Agregar una columna de suma de comprobación para cada fila de entrada
3. Add a constants	Agregue una o más constantes a las filas de entrada
4. Add sequence	Obtener el siguiente valor de una secuencia
5. Add value fields changing sequence	Añadir secuencia dependiendo del cambio de valor de los campos. Cada valor de tiempo de al menos un cambio de campo, Pentaho restablecerá la secuencia.
6. Calculator	Crea nuevos campos realizando cálculos simples.
7. Closure Generator	Este paso le permite generar una tabla de cierre utilizando relaciones padre-hijo.
8. Concat Fields	El paso Campos concat se utiliza para concatenar múltiples campos en un campo objetivo. Los campos se pueden separar con un separador y la lógica del gabinete es completamente compatible con el paso de salida del archivo de texto.
9. Get ID from slave server	Recupera identificadores únicos en bloques de un servidor esclavo. La secuencia a la que se hace referencia debe configurarse en el servidor esclavo en el archivo de configuración XML.
10. Number range	Crear rangos basados en campo numérico.
11. Replace in string	Reemplaza todas las ocurrencias de una palabra en una cadena con otra palabra.
12. Row Normaliser	La información des-normalizada puede normalizarse usando este tipo de paso.
13. Row Denormaliser	Desnormaliza las filas buscando pares clave-valor y asignándolos a nuevos campos en las filas de salida. Este método agrega y necesita que las filas de entrada se ordenen en los campos de agrupación
14. Row flattener	Acopla filas consecutivas según el orden en que aparecen en el flujo de entrada
15. Select values	Seleccione o elimine campos en una fila. Opcionalmente, configure los metadatos de campo: tipo, longitud y precisión.
16. Set field value	Reemplazar el valor de un campo con otro campo de valor
17. Set field value to constant	Reemplazar el valor de un campo a una constante
18. Sort rows	Ordenar filas según los valores de campo (ascendente o descendente)
19. Split fields	Cuando desee dividir un solo campo en más de uno, use este tipo de paso.
20. Split field to rows	Divide un solo campo de cadena por delimitador y crea una nueva fila para cada término dividido
21. String operations	Aplica ciertas operaciones como recorte, relleno y otras al valor de la cadena.
22. Strings cut	Cadenas cortadas (subcadenas).
23. Unique rows	Elimine las filas dobles y deje solo ocurrencias únicas. Esto funciona solo en una entrada ordenada. Si la entrada no está ordenada, solo se manejan correctamente las filas dobles consecutivas.
24. Unique rows (HashSet)	Elimine filas dobles y deje solo ocurrencias únicas utilizando un HashSet.
25. Value Mapper	Asigna los valores de un campo determinado de un valor a otro
26. XSL Transformation	Transforme el flujo de XML utilizando XSL (lenguaje extensible de hojas de estilo).

Fuente: Elaboración propia

Cabe resaltar que el módulo de transformación no es el único que se utiliza, hay transformaciones en otros módulos de la herramienta.

3.1.2.2 Tarea de filtrado

Está compuesta por dos actividades, las cuales son: filtrado de detección y filtrado de corrección de fallas.

- Filtrado de detección: en esta actividad se detectan los errores presentes en las mediciones de cada una de las variables; es la encargada de detectar los valores atípicos, valores faltantes e inconsistencias en los datos fuente.
- Filtrado de corrección de fallas: en esta actividad se reciben los datos con los errores detectados y organizados y se sigue un estándar específico. Los filtros de corrección deben ser investigados en el ámbito del área de aplicación del modelo. Por ejemplo, sí se encuentran valores negativos o nulos en alguna variable se debe investigar si esos valores están en el rango permitido, si no, se procede a la corrección. Después de aplicar las acciones correctivas correspondientes a las mediciones de cada variable, los datos están listos para la siguiente tarea.

Uno de los mayores problemas que se encuentran en estaciones de monitoreo ambiental son las pérdidas de información, suceden por errores humanos o por lo más común que son fallas de lectura de los sensores que se encuentran allí. Este problema es una oportunidad para usar tecnologías Big Data y resolverlo.

3.1.2.3 Completación de Datos Meteorológicos.

Los estudios de datos ambientales requieren de datos de precipitación, caudal, temperatura y radiación a escala diaria. Los datos requeridos por los modelos deben ser confiables y estar

completos en el periodo de estudio. Muchas veces los datos de estaciones de precipitación, aforo, temperatura entre otros se presentan incompletos en varias partes siendo posible su completación mediante métodos numéricos, regresiones o algoritmos de inteligencia artificial.

Keras es una plataforma de alto nivel para redes neurales escrita en Python. Esta plataforma está enfocada en permitir una experimentación rápida de los datos de entrada. Keras soporta redes convulsiónales, recurrentes y combinaciones de ellas; además está diseñada para correr tanto en computadores personales como en computadores avanzados de multiprocesadores.

La ventaja de utilizar inteligencia artificial en scripts y librerías como Keras, es la practicidad en el manejo de los datos, las opciones de configuración de las redes neuronales dependiendo de los datos de entrada, la capacidad de procesamiento y representación de grandes series de datos.

Para el siguiente ejemplo se usan varias estaciones que es lo más recomendado para llenar datos, los datos de las estaciones faltantes son utilizadas como entrenamiento para la perdida de datos que tuvo la estación objetivo, las fechas de la toma de los datos deben ser las mismas y estar en el mismo formato para no tener errores. Por ejemplo: los datos de la estación 1 son de precipitación tomadas en la ciudad de Bogotá en el 2015, los datos de la estación 2 son de precipitación tomadas en la ciudad de Bogotá en el 2015, los datos de la estación 3 son de precipitación tomadas en la ciudad de Bogotá en el 2015. Si algunos de los conjuntos de datos no pertenecen al mismo lugar, al rango de tiempo o a la variable de la toma de datos no se puede usar una red neuronal de este tipo.

A continuación, en Figura 12 se muestra la perdida de datos de precipitación para la estación 2, el cual tiene un vacío de datos desde diciembre del año 2014 hasta comienzos del 2015.

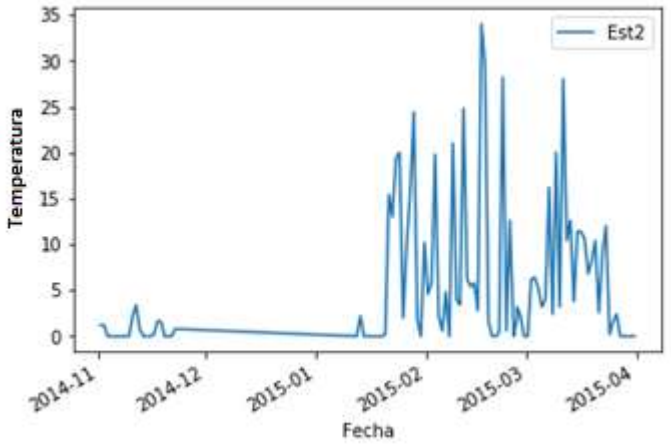


Figura 12 Est2 estación de perdida

Fuente: Elaboración propia

Los datos de las estaciones 1 y 3 son mostrados en las Figuras 13 y 14, se observa que no tienen perdida de información, por lo que los datos de estas estaciones servirán como entrenamiento para la red neuronal.

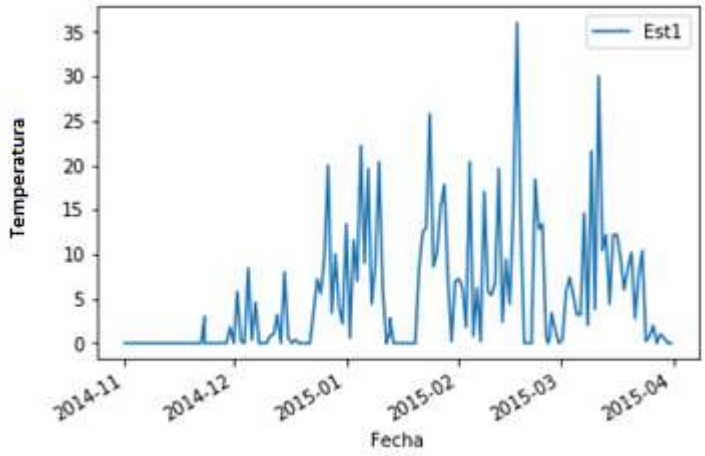


Figura 13 Est1 entrenamiento1

Fuente: Elaboración propia

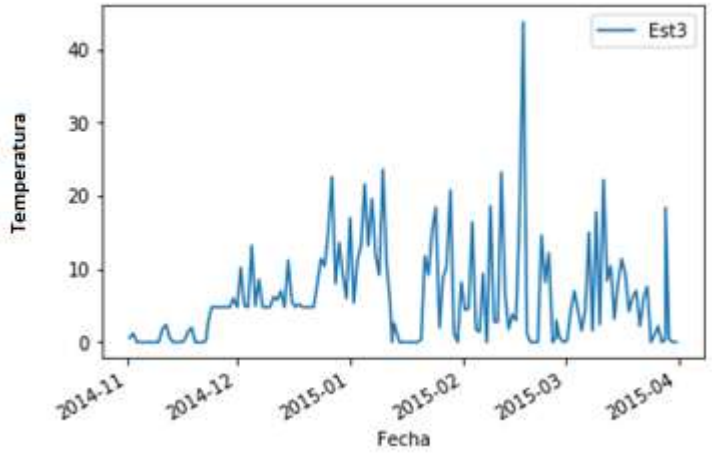


Figura 14 Est3 entrenamiento 2

Fuente: Elaboración propia

Se crea un objeto único en pandas con los datos de las 3 estaciones con las instrucciones detalladas en Figura 15.

Creation of a unique Panda Dataframe

Join the 3 stations into one dataframe and plot it. Our study dataframe will be from November 1st 2014 to March 31th 2015 or: "2014-11-01":"2015-03-31"

In [8]:
```
TodasEstaciones = Estacion_01.resample('24H').sum()
TodasEstaciones['Est2']=Estacion_02['Est2'].resample('24H').sum()
TodasEstaciones['Est3']=Estacion_03['Est3'].resample('24H').sum()
TodasEstaciones.head()
```

Out[8]:

Fecha	Est1	Est2	Est3
2014-07-25	0.0	NaN	0.0
2014-07-26	0.6	NaN	0.0
2014-07-27	0.0	NaN	0.0
2014-07-28	0.0	NaN	0.0
2014-07-29	0.2	NaN	0.2

Figura 15 Creación de un Dataframe

Fuente: Elaboración propia

A continuación, se construye la red neural siguiendo las instrucciones mostradas en Figura 16. La red neuronal es de tipo multilayer perceptron o perceptron multicapa, de la librería sknn (scikit neural network) de Python, usando: 1000 neuronas, una tasa de aprendizaje de 0.00001 y 9000 iteraciones. Luego se crea un array llamado valortest y se le agregan los valores predichos por la red neuronal.

Construction of a Neural Network

Use of MultiLayerPerceptron from the scikit neural network and define a function for our train set:
'2015-01-12':'2015-03-31'

In [15]:
```
from sknn.mlp import Regressor, Layer

capasinicio = TodasEstaciones.loc['2015-01-12':'2015-03-31'].as_matrix()[:,[0,2]]
capasalida = TodasEstaciones.loc['2015-01-12':'2015-03-31'].as_matrix()[:,1]
neurones =  1000
tasaaprendizaje = 0.00001
numiteraciones = 9000


#Definition of the training for the neural network
redneural = Regressor(
    layers=[
        Layer("ExpLin", units=neurones),
        Layer("ExpLin", units=neurones), Layer("Linear")],
    learning_rate=tasaaprendizaje,
    n_iter=numIteraciones)
redneural.fit(capasinicio, capasalida)

#Get the prediction for the train set
valortest = ([])

for i in range(capasinicio.shape[0]):
    prediccion = redneural.predict(np.array([capasinicio[i,:].tolist()]))
    valortest.append(prediccion[0][0])
```

Figura 16 Código de Red Neuronal

Fuente: Elaboración propia

Se grafican los datos reales con los predichos y se observan sí convergen, ver Figura 17.

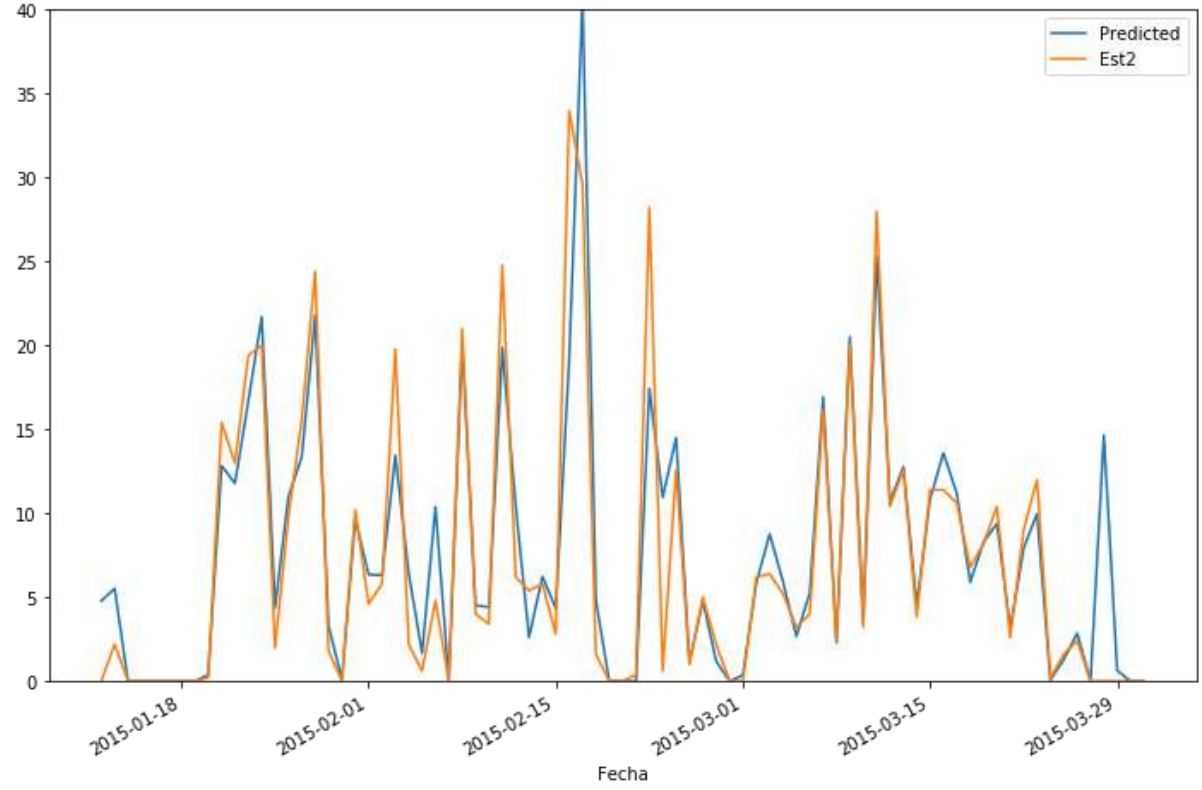


Figura 17 Resultados Red Neuronal

Fuente: Elaboración propia

3.1.3 Fase de carga al almacén de datos.

Esta fase es una de las más importantes del modelo, se encarga de llevar los datos a un almacén de datos, éste será diseñado para optimizar el trabajo de consultas de análisis según el área que se busque en este caso serán los estudios de las variables ambientales a través de los años.

Hay dos posibilidades de hacer un almacén de datos para Big Data, la primera es utilizar una base de datos no relacional, como ya se mencionó estas bases de datos son las mejores al momento de manejar un gran volumen de datos, ya que se tienen respuestas muy rápidas al momento de hacer consultas, dentro de las bases de datos no relacionales hay varias alternativas

que se evalúan según la que más beneficia. La segunda opción es más compleja y está relacionada a sistemas que no quieren hacer cambios bruscos a su centro de datos, y es usar una base de datos relacional, pero usando alguna de las metodologías expuestas para que funcione como Data Warehouse.

3.1.3.1 Carga hacia Base de datos Data WareHouse según Metodologia Ralph Kimball.

Una Data Warehouse puede ayudar a las organizaciones a extraer el máximo valor de los datos que se generan en el día a día, permitiendo además analizarlos y compararlos con sus valores históricos y datos actuales.

Según la guía (Sánchez, 2017) existen principalmente dos tipos de esquemas para estructurar los datos de una base de datos relacional en un almacén de datos: esquema estrella (Ver Figura 18) y esquema copo de nieve (Ver Figura 19).

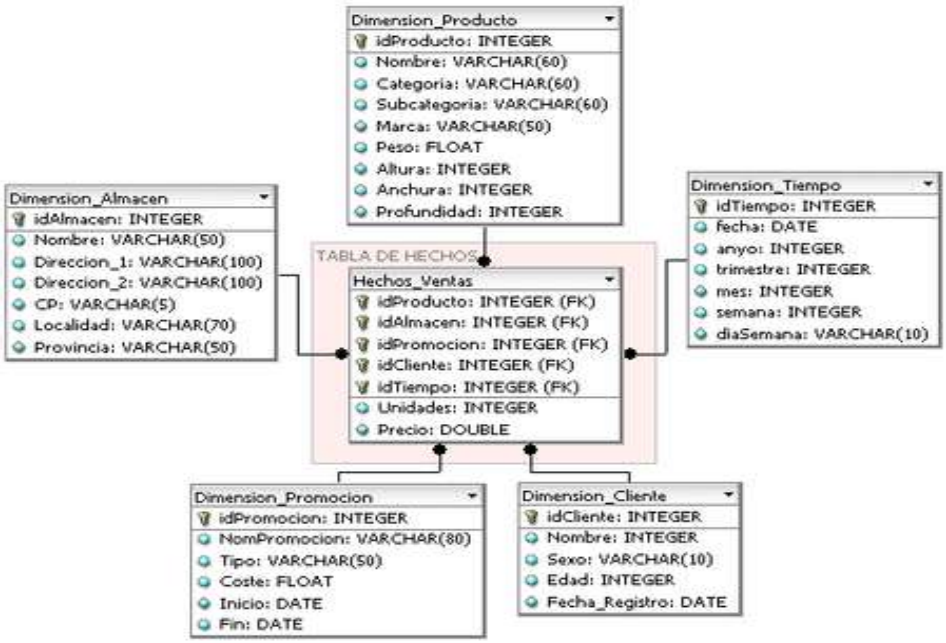


Figura 18 Esquema estrella

Fuente: (Sánchez, 2017)

Para la construcción del esquema "estrella" se deben distinguir entre las tablas de hechos (aquello que se quiere medir o analizar) y las tablas de dimensiones (cómo se quiere medir). Las características del esquema estrella son:

- Una tabla de hechos que contiene los datos sin redundancias.
- Una sola tabla por dimensión.
- La tabla de hechos (Fact table) tiene un atributo columna que forma la clave de cada dimensión.
- Cada tabla de dimensión (Dimension table) es una tabla simple desnormalizada.
- Puede haber redundancia de datos.
- Existen menos joins para las consultas

Cuando se unen distintos esquemas “estrella” que tienen distintas tablas de hechos, pero comparten las de las dimensiones, se habla de constelaciones de hechos; algunos autores hablan incluso de esquema “galaxia”.

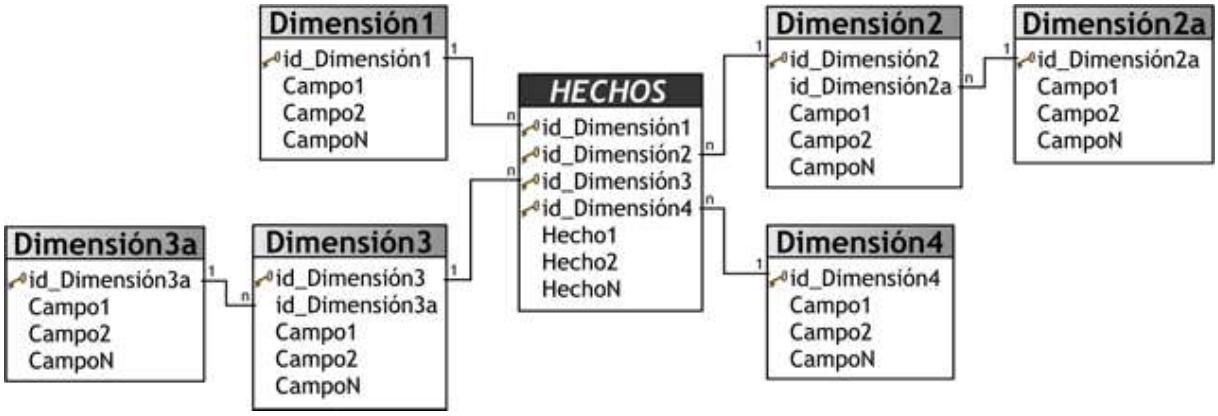


Figura 19 Esquema copo de nieve

Fuente: (Sánchez, 2017)

El esquema copo de nieve es una representación derivada del esquema en estrella, en el que las tablas de dimensión se normalizan en múltiples tablas. Por esta razón, la tabla de hechos deja de ser la única tabla del esquema que se relaciona con otras tablas, y aparecen nuevas uniones. Es posible distinguir dos tipos de esquemas en copo de nieve:

- Completo: en el que todas las tablas de dimensión en el esquema en estrella aparecen ahora normalizadas.
- Parcial: sólo se lleva a cabo la normalización de algunas de las tablas.

Estrategias para modelar un óptimo Data Warehouse:

- Antes de modelar debe tener claro los requerimientos del negocio

- Tener claro la granularidad de los datos
- Diseñar una data con información flexible y reusable.
- En la tabla de hechos los atributos que se usaran para agrupación o filtrado deben ser un campo numérico. Ejemplo: si en la tabla de hechos existen un campo sexo, este no debe ser un tipo texto para incluir femenino o masculino, sino por el contrario debería ser un campo numérico donde “1” sea masculino y “2” femenino.
- No se deben abreviar las descripciones en las tablas dimensionales. Ejemplo: si el campo es nombre no digitar Luigi A. Contreras, sino el nombre completo con los dos apellidos.
- Se debe evitar superar la tercera forma normal, pues el crecimiento exagerado de tablas hace inmanejable su administración. Ejemplo: una tabla para municipios, otra para departamentos, otra para veredas, otra para corregimientos y otra para caseríos.
- No se debe añadir dimensiones en una tabla de hechos antes de definir su granularidad.
- Evite crear un modelo dimensional para resolver un informe en particular.
- No debe mezclar hechos de diferente granularidad en una misma tabla de hechos.
- Unifique los hechos entre distintas tablas de hechos
- Sería necesario incluir campos que permitieran la trazabilidad del dato, por ejemplo, fecha de carga, fecha de modificación, autor, fuente de origen.

Para un correcto modelamiento de una Data WareHouse se sugiere ver la guía (Sánchez, 2017).

3.1.3.2 Carga hacia Base de datos NoSQL.

Pentaho da diferentes opciones para hacer una carga de datos a una base de datos no relacional como muestra la Figura 20:

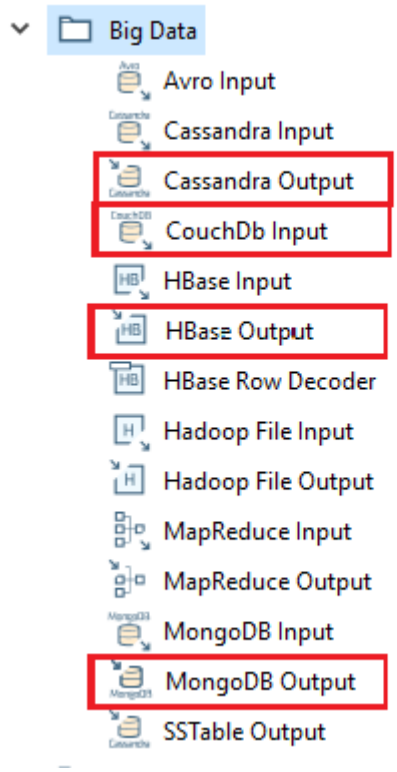


Figura 20 Salidas Big Data

Fuente: Elaboración propia

Según (Rouse, 2016a) existen diferentes tipos de bases de datos NoSQL estás se distinguen principalmente en su tipología (orientadas a Objetos, Columnas y Clave-Valor). A continuación, se mencionan algunas de las bases de datos con las que Pentaho tiene relación.

Cassandra: almacena la información en "Columns Families" que son objetos de base de datos similares a las tablas de un sistema de base de datos relacional (como MySQL), las cuales contienen filas y columnas, cada fila con una clave única. A diferencia de un sistema de base de datos relacional, todas las filas de una tabla no están obligadas a tener las mismas columnas. Estas columnas también se pueden agregar sobre la marcha y se accede a ellas utilizando el Lenguaje de Consulta de Cassandra (CQL= Cassandra Query Language). Mientras que CQL es

similar a SQL en sintaxis, Cassandra no es relacional, por lo que tiene diferentes formas de almacenar y recuperar datos.

Si necesita una base de datos que sea fácil de instalar y mantener, independientemente de cuánto crezca su base de datos, Cassandra puede ser una buena opción. Además, que su lenguaje de consulta es fácil de aprender si ya se tiene conocimiento en bases de datos relacionales.

CouchDb: es una base de datos NoSQL de código abierto basada en estándares comunes para facilitar la accesibilidad y compatibilidad web con una diversidad de dispositivos. Los datos en CouchDB se almacenan en el formato de notificación de objetos JavaScript (JSON), y están organizados en pares de valor clave. La clave es un identificador único de los datos, y el valor es el propio dato o un apuntador a la ubicación de los datos. Las funciones estándar de la base de datos son realizadas por JavaScript. Los estándares web agnósticos de sistema operativo, e independientes de dispositivos permiten a las bases de datos desempeñarse bien en la variedad más amplia de usuarios. Además de hacer disponibles las bases de datos y documentos a una amplia audiencia de usuarios, CouchDB también facilita el desarrollo de aplicaciones web y hace posible servir apps directamente desde la base de datos.

Hbase: HBase es un proyecto open-source mantenido por la Apache Foundation que proporciona una base de datos columnar distribuida creada sobre el sistema de ficheros de Hadoop que puede escalar horizontalmente. HBase utiliza un modelo de datos muy similar al de Google Big Table diseñado para proporcionar acceso aleatorio a una cantidad muy grande de datos estructurados.

El objetivo del proyecto HBase es el almacenamiento de tablas muy grandes, de billones de filas por millones de columnas, para ello almacena los datos por pares de clave-valor. Buscar por

claves en HBase es muy rápido. La escritura también porque se realiza prácticamente en memoria.

En HBase los datos se guardan en tablas que tienen filas y columnas, puede parecer que se habla de una base de datos relacional, pero no tiene nada que ver, sería más acertado pensar en HBase como un mapa multidimensional.

MongoDb: almacena la información en documentos similares a JSON que pueden tener estructuras variadas. Utiliza un lenguaje propio de consulta para permitir el acceso a los datos almacenados. Como no tiene esquemas, puede crear documentos sin tener que crear primero la estructura del documento. Puede ser una gran opción si necesitas escalabilidad y almacenamiento en caché para análisis en tiempo real; sin embargo, no está diseñado para datos transaccionales.

MongoDB se usa con frecuencia para aplicaciones móviles, administración de contenido, análisis en tiempo real y aplicaciones relacionadas con el IoT (Internet de las cosas). Si tiene una situación en la que no tiene una definición de esquema clara, MongoDB puede ser una buena opción.

Si se tiene una situación en la que se está des-normalizando el esquema de la base de datos, los documentos MongoDB se pueden usar para almacenar los datos no estructurados de una manera que es más fácil de actualizar. En una situación donde la carga de escritura es alta, MongoDB puede ser una buena opción.

3.2 Caso de estudio

Los datos fueron recibidos del instituto de hidrología, meteorología y estudios ambientales IDEAM el 3 de abril del 2018, pertenecen a la estación 16015010 de la ciudad de Cúcuta – Norte de Santander, los datos han sido recolectados desde el año 1989 hasta el 2003.

Los datos suministrados cuentan con varias variables como son temperatura, velocidad de vientos y precipitación como detalla la Tabla 4. Para la realización del procedimiento sólo se utilizó la variable temperatura. Esta cuenta con 11237 líneas en un archivo de texto, y describe 24 variables respectivamente.

Tabla 4 Variables

Columna	Variable
1-	Día
2-	Temperatura Máxima
3-	Temperatura Mínima
4-	Amplitud
5-	Termómetro seco 07
6-	Termómetro seco 13
7-	Termómetro seco 19
8-	Termómetro seco media
9-	Termómetro húmedo 07
10-	Termómetro húmedo 09
11-	Termómetro húmedo 13
12-	Termómetro húmedo media
13-	Humedad relativa 07
14-	Humedad relativa 09
15-	Humedad relativa 13
16-	Humedad relativa media
17-	Tensión de vapor en milibares 07
18-	Tensión de vapor en milibares 09
19-	Tensión de vapor en milibares 13
20-	Tensión de vapor en milibares media
21-	Puntos de rocío 07
22-	Puntos de rocío 09
23-	Puntos de rocío 13

24-	Puntos de rocío media

Fuente: Elaboración propia

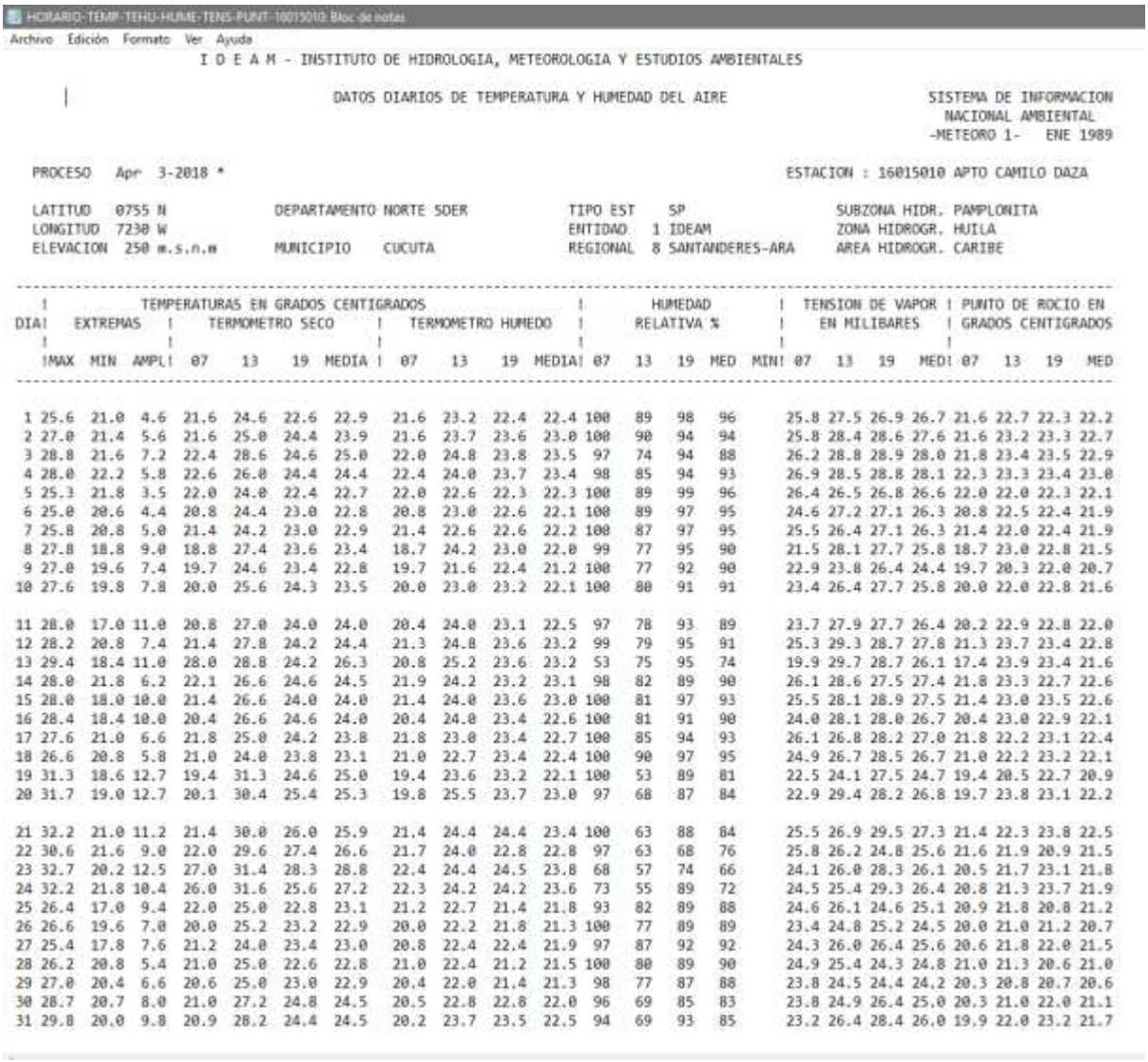
3.2.1 Fase de Prerrequisitos.

Como ya se mencionó las fuentes de los datos pueden ser distintas, aquí se cumple con la tercera "gran v" que caracteriza Big Data del resto de tecnologías, esto es la variedad. La herramienta principal usada para esta fase es Pentaho input, que cuenta con muchas opciones para cargar los datos al entorno de trabajo.

3.2.1.1 Traducción

Esta fase se encarga de estandarizar los datos en el caso en que se tenga archivos de la forma no estructurada, para este particular caso de estudio es completamente necesario ya que se recibieron archivos planos que contienen tablas, pero mucho contenido sobrante que no serán comprendidos por ninguna base de datos o herramienta.

La figura 21 muestra los Datos entrantes tomados de los archivos originales del IDEAM:



HORARIO-TEMP-TEHU-HUME-TENS-PUNT-16015010: Bloc de notas

Archivo Edición Formato Ver Ayuda

I D E A M - INSTITUTO DE HIDROLOGIA, METEOROLOGIA Y ESTUDIOS AMBIENTALES

DATOS DIARIOS DE TEMPERATURA Y HUMEDAD DEL AIRE

SISTEMA DE INFORMACION NACIONAL AMBIENTAL
-METEORO 1- ENE 1989

PROCESO Apr 3-2018 *

ESTACION : 16015010 APTO CAMILO DAZA

LATITUD 0755 N DEPARTAMENTO NORTE SDER TIPO EST SP SUBZONA HIDR. PAMPLONITA
LONGITUD 7230 W ENTIDAD 1 IDEAM ZONA HIDROGR. HUILA
ELEVACION 250 m.s.n.m MUNICIPIO CUCUTA REGIONAL 8 SANTANDERES-ARA AREA HIDROGR. CARIBE

DIA	TEMPERATURAS EN GRADOS CENTIGRADOS											HUMEDAD RELATIVA %				TENSION DE VAPOR EN MILIBARES				PUNTO DE ROCIO EN GRADOS CENTIGRADOS			
	EXTREMAS			TERMOMETRO SECO				TERMOMETRO HUMEDO															
	MAX	MIN	AMPL	07	13	19	MEDIA	07	13	19	MEDIA	07	13	19	MED	MIN	07	13	19	MED	07	13	19	MED
1	25.6	21.0	4.6	21.6	24.6	22.6	22.9	21.6	23.2	22.4	22.4	100	89	98	96		25.8	27.5	26.9	26.7	21.6	22.7	22.3	22.2
2	27.0	21.4	5.6	21.6	25.0	24.4	23.9	21.6	23.7	23.6	23.0	100	90	94	94		25.8	28.4	28.6	27.6	21.6	23.2	23.3	22.7
3	28.8	21.6	7.2	22.4	28.6	24.6	25.0	22.0	24.8	23.8	23.5	97	74	94	88		26.2	28.8	28.9	28.0	21.8	23.4	23.5	22.9
4	28.0	22.2	5.8	22.6	26.0	24.4	24.4	22.4	24.0	23.7	23.4	98	85	94	93		26.9	28.5	28.8	28.1	22.3	23.3	23.4	23.0
5	25.3	21.8	3.5	22.0	24.0	22.4	22.7	22.0	22.6	22.3	22.3	100	89	99	96		26.4	26.5	26.8	26.6	22.0	22.0	22.3	22.1
6	25.0	20.6	4.4	20.8	24.4	23.0	22.8	20.8	23.0	22.6	22.1	100	89	97	95		24.6	27.2	27.1	26.3	20.8	22.5	22.4	21.9
7	25.8	20.8	5.0	21.4	24.2	23.0	22.9	21.4	22.6	22.6	22.2	100	87	97	95		25.5	26.4	27.1	26.3	21.4	22.0	22.4	21.9
8	27.8	18.8	9.0	18.8	27.4	23.6	23.4	18.7	24.2	23.0	22.0	99	77	95	90		21.5	28.1	27.7	25.8	18.7	23.0	22.8	21.5
9	27.0	19.6	7.4	19.7	24.6	23.4	22.8	19.7	21.6	22.4	21.2	100	77	92	90		22.9	23.8	26.4	24.4	19.7	20.3	22.0	20.7
10	27.6	19.8	7.8	20.0	25.6	24.3	23.5	20.0	23.0	23.2	22.1	100	80	91	91		23.4	26.4	27.7	25.8	20.0	22.0	22.8	21.6
11	28.0	17.0	11.0	20.8	27.0	24.0	24.0	20.4	24.0	23.1	22.5	97	78	93	89		23.7	27.9	27.7	26.4	20.2	22.9	22.8	22.0
12	28.2	20.8	7.4	21.4	27.8	24.2	24.4	21.3	24.8	23.6	23.2	99	79	95	91		25.3	29.3	28.7	27.8	21.3	23.7	23.4	22.8
13	29.4	18.4	11.0	28.0	28.8	24.2	26.3	20.8	25.2	23.6	23.2	53	75	95	74		19.9	29.7	28.7	26.1	17.4	23.9	23.4	21.6
14	28.0	21.8	6.2	22.1	26.6	24.6	24.5	21.9	24.2	23.2	23.1	98	82	89	90		26.1	28.6	27.5	27.4	21.8	23.3	22.7	22.6
15	28.0	18.0	10.0	21.4	26.6	24.0	24.0	21.4	24.0	23.6	23.0	100	81	97	93		25.5	28.1	28.9	27.5	21.4	23.0	23.5	22.6
16	28.4	18.4	10.0	20.4	26.6	24.6	24.0	20.4	24.0	23.4	22.6	100	81	91	90		24.0	28.1	28.0	26.7	20.4	23.0	22.9	22.1
17	27.6	21.0	6.6	21.8	25.0	24.2	23.8	21.8	23.0	23.4	22.7	100	85	94	93		26.1	26.8	28.2	27.0	21.8	22.2	23.1	22.4
18	26.6	20.8	5.8	21.0	24.0	23.8	23.1	21.0	22.7	23.4	22.4	100	90	97	95		24.9	26.7	28.5	26.7	21.0	22.2	23.2	22.1
19	31.3	18.6	12.7	19.4	31.3	24.6	25.0	19.4	23.6	23.2	22.1	100	53	89	81		22.5	24.1	27.5	24.7	19.4	20.5	22.7	20.9
20	31.7	19.0	12.7	20.1	30.4	25.4	25.3	19.8	25.5	23.7	23.0	97	68	87	84		22.9	29.4	28.2	26.8	19.7	23.8	23.1	22.2
21	32.2	21.0	11.2	21.4	30.0	26.0	25.9	21.4	24.4	24.4	23.4	100	63	88	84		25.5	26.9	29.5	27.3	21.4	22.3	23.8	22.5
22	30.6	21.6	9.0	22.0	29.6	27.4	26.6	21.7	24.0	22.8	22.8	97	63	68	76		25.8	26.2	24.8	25.6	21.6	21.9	20.9	21.5
23	32.7	20.2	12.5	27.0	31.4	28.3	28.8	22.4	24.4	24.5	23.8	68	57	74	66		24.1	26.0	28.3	26.1	20.5	21.7	23.1	21.8
24	32.2	21.8	10.4	26.0	31.6	25.6	27.2	22.3	24.2	24.2	23.6	73	55	89	72		24.5	25.4	29.3	26.4	20.8	21.3	23.7	21.9
25	26.4	17.0	9.4	22.0	25.0	22.8	23.1	21.2	22.7	21.4	21.8	93	82	89	88		24.6	26.1	24.6	25.1	20.9	21.8	20.8	21.2
26	26.6	19.6	7.0	20.0	25.2	23.2	22.9	20.0	22.2	21.8	21.3	100	77	89	89		23.4	24.8	25.2	24.5	20.0	21.0	21.2	20.7
27	25.4	17.8	7.6	21.2	24.0	23.4	23.0	20.8	22.4	22.4	21.9	97	87	92	92		24.3	26.0	26.4	25.6	20.6	21.8	22.0	21.5
28	26.2	20.8	5.4	21.0	25.0	22.6	22.8	21.0	22.4	21.2	21.5	100	80	89	90		24.9	25.4	24.3	24.8	21.0	21.3	20.6	21.0
29	27.0	20.4	6.6	20.6	25.0	23.0	22.9	20.4	22.0	21.4	21.3	98	77	87	88		23.8	24.5	24.4	24.2	20.3	20.8	20.7	20.6
30	28.7	20.7	8.0	21.0	27.2	24.8	24.5	20.5	22.8	22.8	22.0	96	69	85	83		23.8	24.9	26.4	25.0	20.3	21.0	22.0	21.1
31	29.8	20.0	9.8	20.9	28.2	24.4	24.5	20.2	23.7	23.5	22.5	94	69	93	85		23.2	26.4	28.4	26.0	19.9	22.0	23.2	21.7

Figura 21 Input Fase de Traducción

Fuente: Captura de pantalla de los datos obtenidos del IDEAM

El procedimiento de traducción se hizo en tres pasos:

1. Se limpiaron encabezados que se repetían miles de veces durante todos los registros.
2. Se agregó mes/año a cada uno de los registros usando macros en visual basic (no fue posible usar ningún paso en Pentaho para resolver este problema).

3. Se eliminaron miles de líneas vacías sin afectar registros.

Se detalla a continuación el proceso usado:

1. Se utilizó el paso (Search and Replace) para buscar el texto completo del encabezado que tenía cada mes y se reemplazó por filas vacías, cabe resaltar que en el encabezado había un dato importante que era el mes y el año al que pertenecían los datos, este se conservó.
2. Se eliminaron las filas vacías con el paso (Non Empty).
3. Se usó la herramienta macros de visual basic para recorrer todo el documento, tomar el año y mes perteneciente al registro y añadírselo a la columna 25. Cabe resaltar que la ejecución del macro tarda un tiempo ya que son muchos registros y se sobrescribe uno por uno y la herramienta no está optimizada para tal cantidad de datos. En la Figura 21 está el macro usado para el proceso:

```
Sub fechas()
Dim N As Integer
Dim texto As String
Dim valor As String
'Nos situamos en la celda con el primer dato
Range("U1").Select
'Bajamos hasta la última fila adyacente, que contenga datos
Selection.End(xlDown).Select
'Mostramos el número de líneas
N = ActiveCell.Row
respuesta = MsgBox("Esta hola de cálculo tiene " & N & " filas con datos contínuos.")
For i = 1 To N
    Cells(i, 21).Select
    texto = ActiveCell.Text
    celda = InStr(" ", texto)
    If celda <> 0 Then
    'Si la celda esta vacia no hacemos nada
    Else
    texto = ActiveCell.Text
    valor = "ENE-FEB-MAR-ABR-MAY-JUN-JUL-AGO-SEP-OCT-NOV-DIC"
    celda = InStr(valor, texto)
    If celda <> 0 Then

    For j = 1 To 31
    Cells(i + j, 25).Value = texto + Cells(i, 22).Value

    Next
    Else

    End If
    End If

Next
End Sub
```

Figura 22 Macro Para Formato de Fechas

Fuente: Elaboración propia

La figura 23 muestra el resultado de la fase de traducción

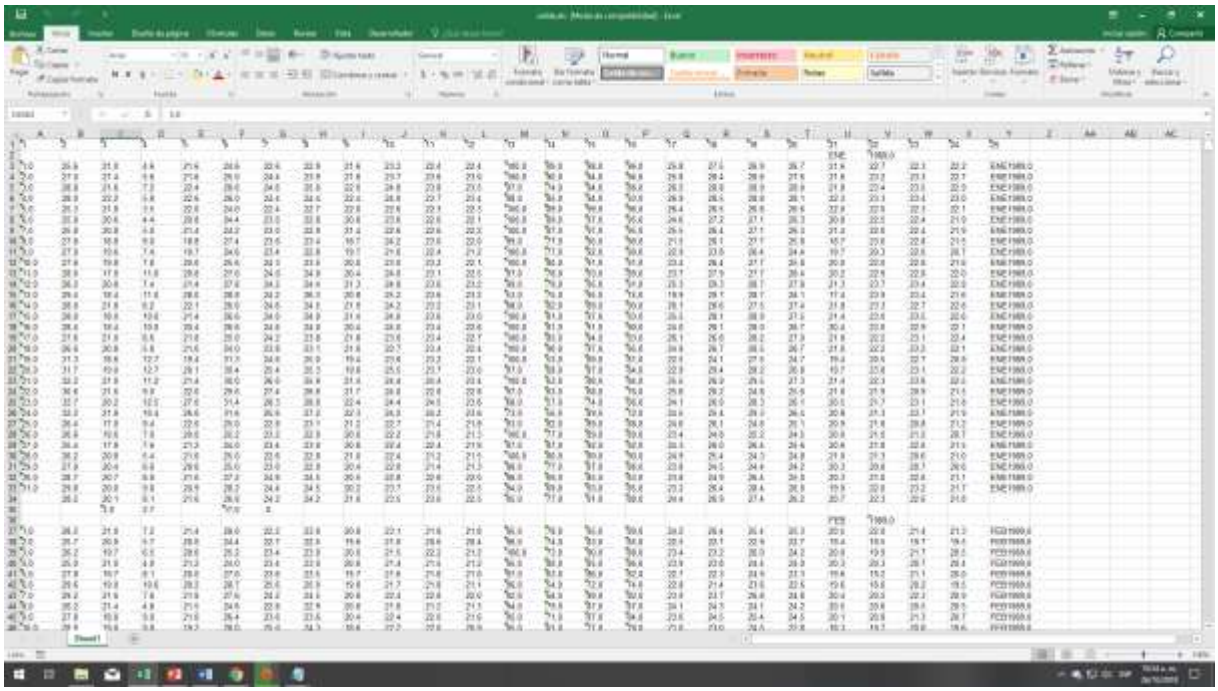


Figura 23 Output Fase de Traducción

Fuente: Elaboración propia

En este particular caso de estudio como lo son las fuentes de datos ambientales, solo se tienen archivos planos generados por las estaciones de monitoreo ambiental aquí en Colombia, sin embargo, se pueden tener muchos otros tipos de archivos o bases de datos que se tendrían usar para la extracción.

En la fase de traducción se hizo una limpieza con el fin de cargar los datos por medio del paso "Microsoft Excel Input" al entorno de trabajo de pentaho, una vez ahí se tienen varias opciones de cargarlo según sea el caso:

1. Cargar mes por mes: Esta opción es viable y muy efectiva cuando el propósito de uso de los datos es el análisis mensual, pero en un rango corto de años, ya que se necesitan

conocer las coordenadas de cada tabla mensual para poderla montar ver (Figura 24).

Figura 24 Extracción por meses

Fuente: Elaboración propia

2. Cargar todos los datos en un paso: Esta opción es la más recomendada cuando el análisis se va a hacer en todo el rango de años que se tienen como es el caso de estudio que tiene 14 años de muestras ver (Figura 25).

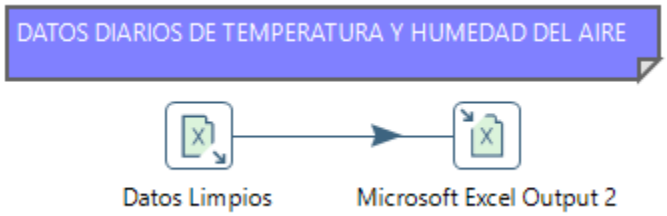


Figura 25 Extracción General

Fuente: Elaboración propia

3.2.2 Fase Principal

3.2.2.1 Tarea de Filtrado

Detección de errores

1. Se detectan registros nulos pertenecientes a los pies de cada mes, en donde estaban los máximos de cada día, que se pueden consultar en otro momento desde la base de datos
2. El formato de la fecha no es un formato valido, se necesita un formato DD/MM/YY para una posible carga hacia Python y trabajar otras herramientas
3. Se detecta perdidas de datos en pequeños grupos y en una pérdida importante en el año 1989
4. Faltan encabezados para reconocer que significa el valor de la columna

Corrección de errores

1. Se eliminan los registros nulos
2. Los registros de fecha se cambian a formato DD/MM/YY

3. Se estudian los datos originales obtenidos y se nombra cada columna
4. Se evaluaron varias estrategias para el rellenado de datos faltantes, para la perdida de datos ocurrida en el año 1989 se decidió utilizar una red neuronal ya que se pueden utilizar los años anteriores y posteriores como entrenamiento para la red neuronal, y obtener un resultado aproximado a los reales.

   A continuación, se detallan los pasos utilizados en la herramienta Pentaho, en el workflow 1 se busca solución a los errores encontrados, en los workflow 2 y 3 se extraen los archivos de entrenamiento para la red neuronal que busca completar los datos faltantes ver (Figura 26).

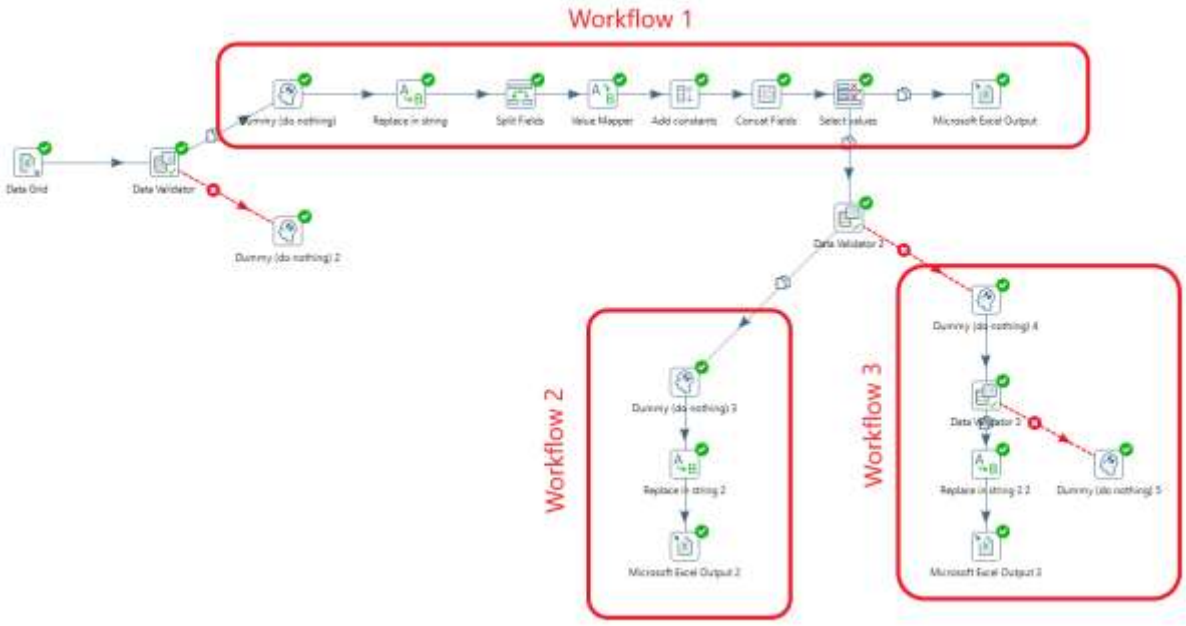


Figura 26 Workflow, Fase Principal

Fuente: Elaboración propia

WorkFlow 1

1. Microsoft Excel Input: Se cargan los datos desde el archivo output de la traducción
2. Data Validator: Se usa para poner un rango de 1-31 en la columna de días y así eliminar todos aquellos registros que no pertenecen a un mes
3. Dummy (1): Recibe los datos validados del paso anterior.
4. Dummy (2): Recibe los datos no validados del paso anterior.
5. Replace In String: Se busca (19 y 20) en la columna de la fecha que se tiene y se reemplaza por (19- y 20-) esto con el fin de tener un delimitador.
6. Split-Flieds: Se dividen campos con el delimitador puesto anteriormente (-) y se forman dos nuevas columnas mes y año.
7. Value Mapper: Con este paso se le da el formato al mes ya que está de la siguiente manera: ENE, FEB, MAR, ABR, etc. Después del paso quedará de la siguiente manera: 01, 02, 03, 04.
8. Select Values: Se selecciona las columnas que se quieren en el documento de salida y se agregan los encabezados correspondientes.
9. Microsoft Excel Output: Se le da la salida a la fase del proceso.

Workflow 2

1. Data validator: Se usa para validar que los registros sean del año 1990 (un año después de la perdida de datos)
2. Dummy (1): Recibe los datos validados del paso anterior.
3. Dummy: (2) Recibe los datos no validados del paso anterior.

4. Replace in String: Si se pasó por el paso 2, se busca 1990 en la columna fecha y se reemplaza por 1989
5. Microsoft Excel Output: Se cargan los datos en un archivo Excel de nombre "entrenamiento1".

Workflow 3

1. Data validator: Se usa para validar que los registros sean del año 1991 (dos años después de la perdida de datos)
2. Dummy (1): Recibe los datos validados del paso anterior.
3. Dummy (2) Recibe los datos no validados del paso anterior.
4. Replace in String: Si pasó por el paso 2, se busca 1991 en la columna fecha y se reemplaza por 1989
5. Microsoft Excel Output: Se cargan los datos en un archivo Excel de nombre "entrenamiento2"

Datos entrantes a la fase principal

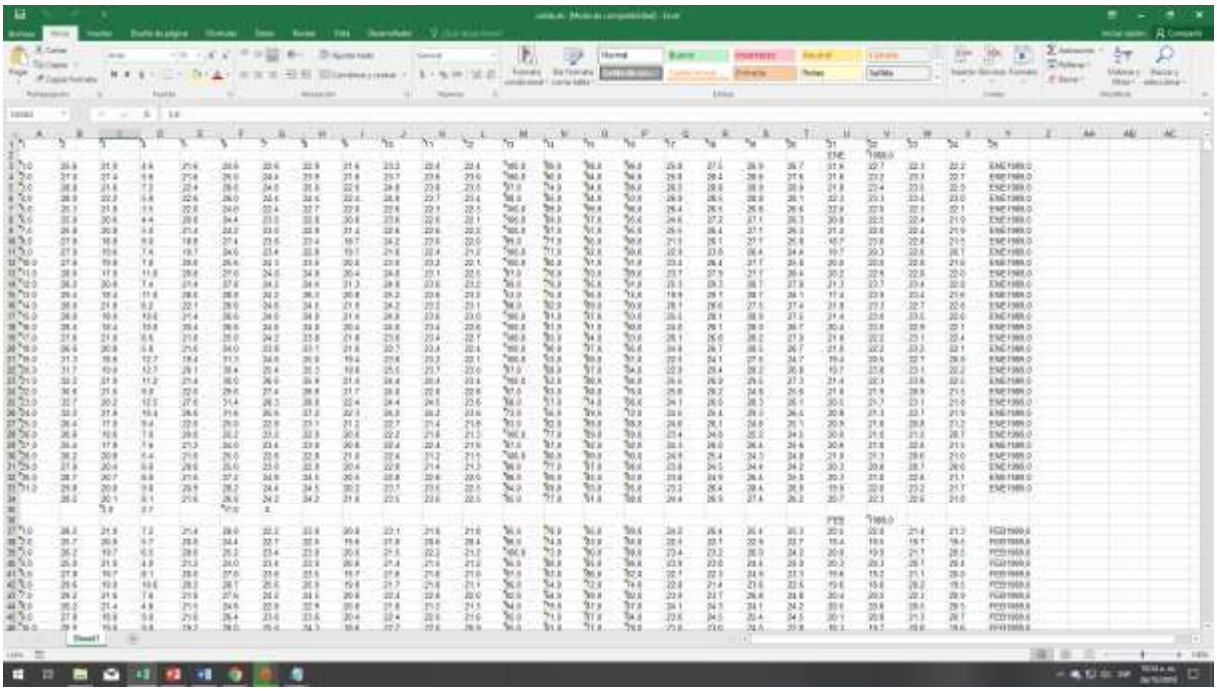


Figura 27 Input Fase Principal

Fuente: Elaboración propia

Datos salientes una vez completado la fase principal

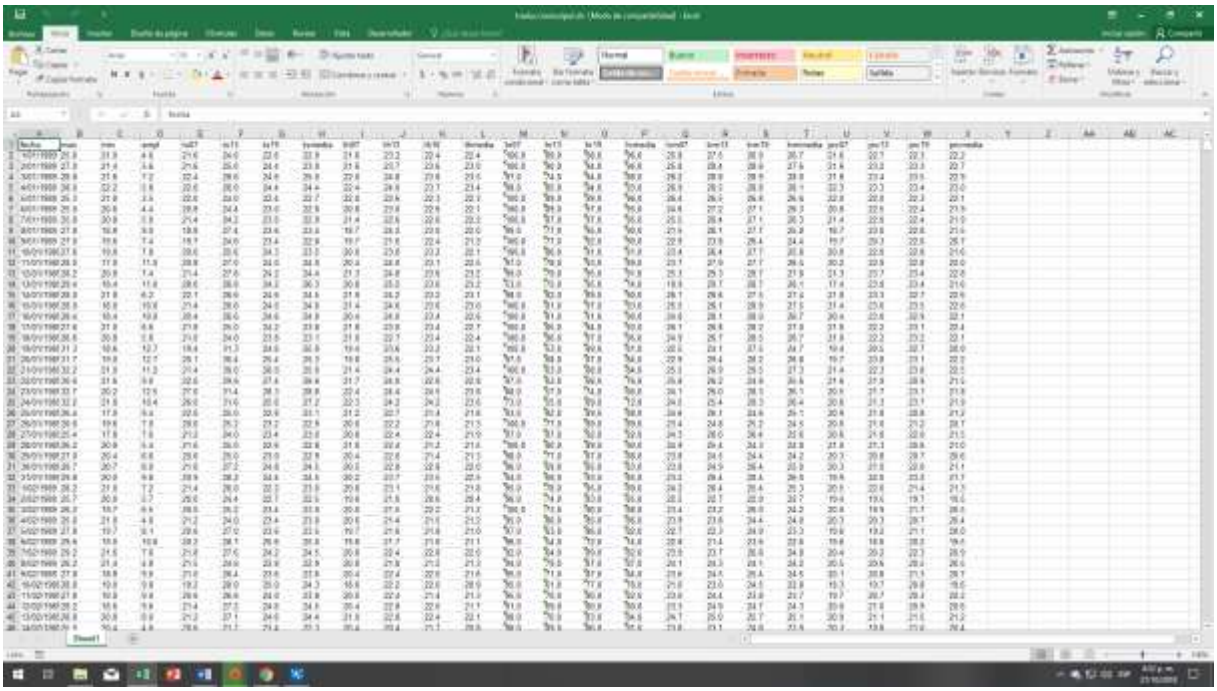


Figura 28 Output Fase Principal

Fuente: Elaboración propia

Archivos de Entrenamiento:

Entrenamiento 1: Constituye los datos del año 1990 que serán usados para el entrenamiento.

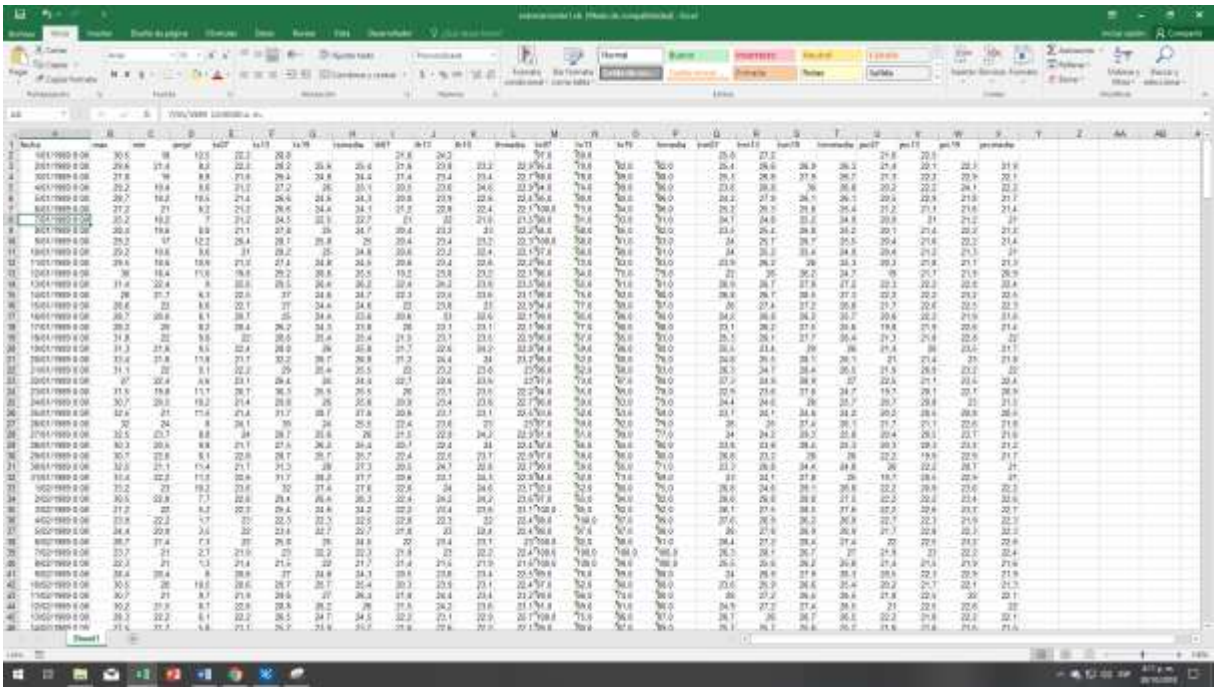


Figura 29 Entrenamiento 1

Fuente: Elaboración propia

Entrenamiento 2: Constituye los datos del año 1991 que serán usados para el entrenamiento

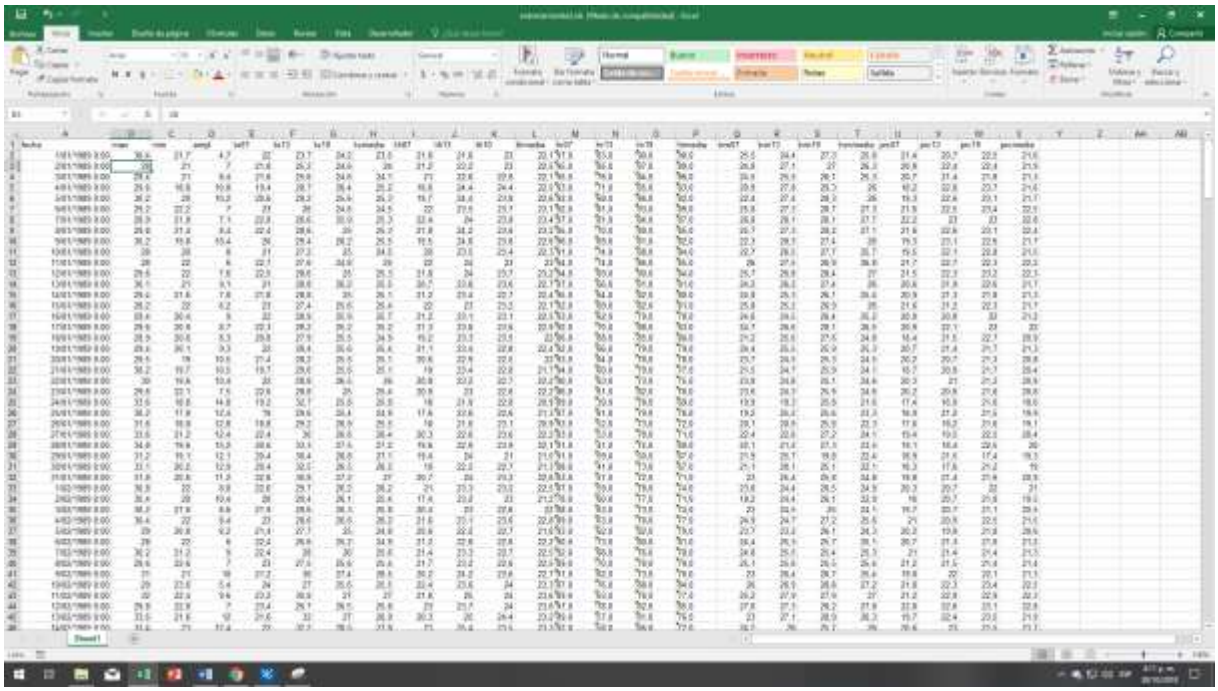


Figura 30 Entrenamiento 2

Fuente: Elaboración propia

3.2.2.2 Implementación de una red neuronal para el rellenado de datos en el caso de estudio.

En este caso de estudio se tiene una pérdida de datos considerable desde mayo del 1989 (ver figura 33) hasta octubre del mismo año en la variable de temperatura, está perdida pudo ser causada por mantenimiento o fallas en la estación de monitoreo ya que son bastante frecuentes.

El proceso de predicción de los datos fue el mismo utilizado en el ejemplo de precipitación, pero en este no se utilizan varias estaciones como entrenamiento para el algoritmo, sino se utilizaron datos de los años anteriores y posteriores ya que son bastante similares.

Para utilizar redes neuronales no se recomienda utilizar meses, ya que los cambios de las estaciones influyen en los valores de la mayoría de las variables.

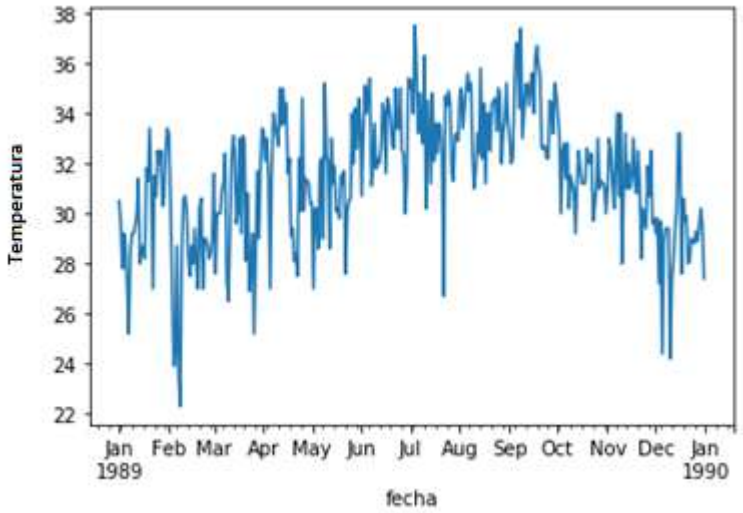


Figura 31 Variable 1 – Temperatura 1990

Fuente: Elaboración propia

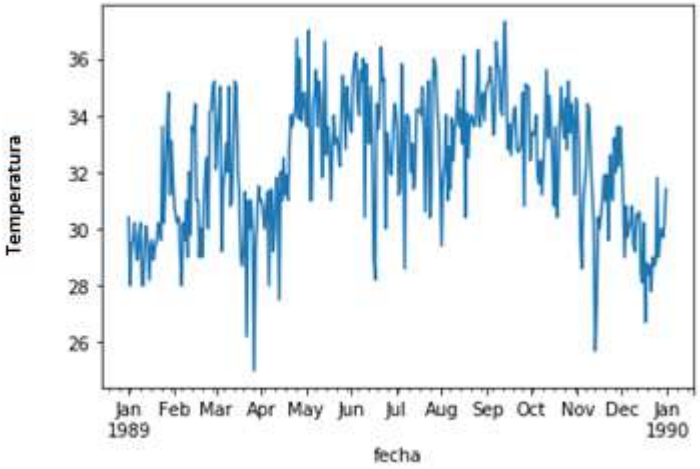


Figura 32 Variable 1 – Temperatura 1991

Fuente: Elaboración propia

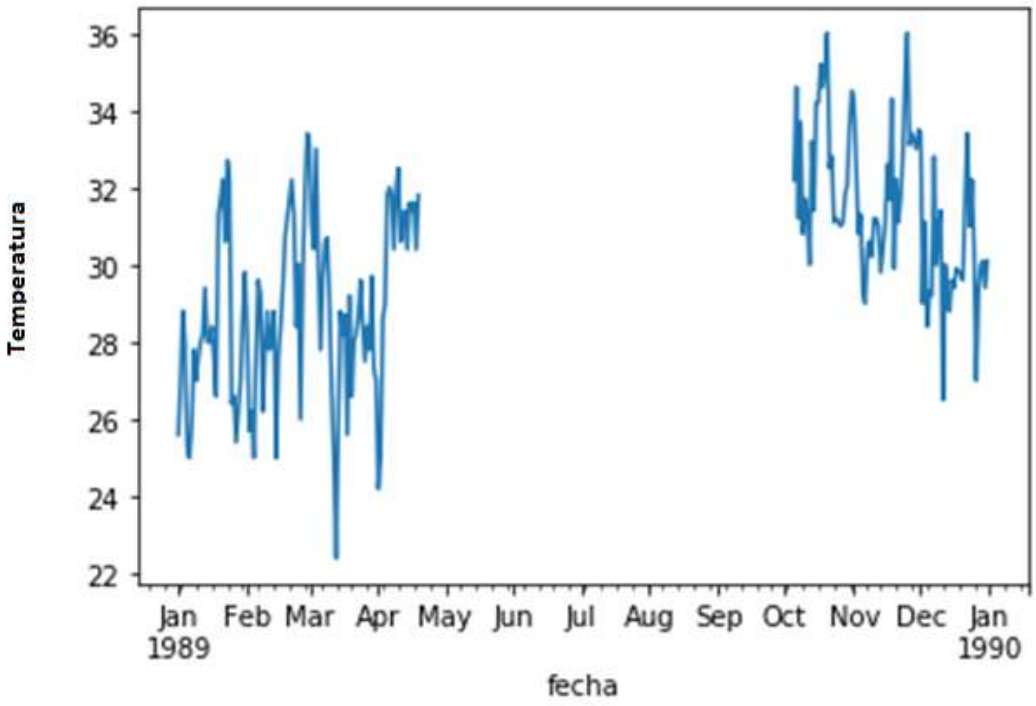


Figura 33 Variable 1 – Temperatura 1989 (Objetivo de rellenado de datos)

Fuente: Elaboración propia

Se crea un dataframe con pandas para la manipulación de los datos de entrenamiento y la tabla objetiva de predicción ver (Figura 34).

```
In [6]: TodasEstaciones = datos.resample('24H').sum()
        TodasEstaciones['Est1'] = datos['Est1'].resample('24H').sum()
        TodasEstaciones['Est2'] = entrenamiento1['max'].resample('24H').sum()
        TodasEstaciones['Est3'] = entrenamiento2['max'].resample('24H').sum()
        TodasEstaciones.head()
```

Out[6]:

fecha	Est1	mm	ampl	ts07	th07	th13	th19	thmedia	tvm13	tvm19	tvmmedia	prc07	prc13	prc19	prcmedia	Est2	Est3
1989-01-01	25.6	21.0	4.6	21.6	21.6	23.2	22.4	22.4	27.5	26.9	26.7	21.6	22.7	22.3	22.2	30.5	30.4
1989-01-02	27.0	21.4	5.6	21.6	21.6	23.7	23.6	23.0	28.4	28.6	27.6	21.6	23.2	23.3	22.7	29.6	28.0
1989-01-03	28.8	21.6	7.2	22.4	22.0	24.8	23.8	23.5	28.8	28.9	28.0	21.8	23.4	23.5	22.9	27.8	29.4
1989-01-04	28.0	22.2	5.8	22.6	22.4	24.0	23.7	23.4	28.5	28.8	28.1	22.3	23.3	23.4	23.0	29.2	29.6
1989-01-05	25.3	21.8	3.5	22.0	22.0	22.8	22.3	22.3	26.5	26.8	26.6	22.0	22.0	22.3	22.1	28.7	30.2

Figura 34 Creación Dataframe Caso de Estudio

Fuente: Elaboración propia

- Est1: Datos de 1989 (Faltantes).
- Est2: Datos de 1990 (Entrenamiento).
- Est3: Datos de 1991 (Entrenamiento).

Se grafican en paralelo los archivos de entrenamiento para visualizar su similitud ver (Figura 35)

```
In [12]: entrenamiento2.loc['1/01/1989':'31/12/1989', 'max'].plot()
         entrenamiento1.loc['1/01/1989':'31/12/1989', 'max'].plot()
Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x1709e358>
```

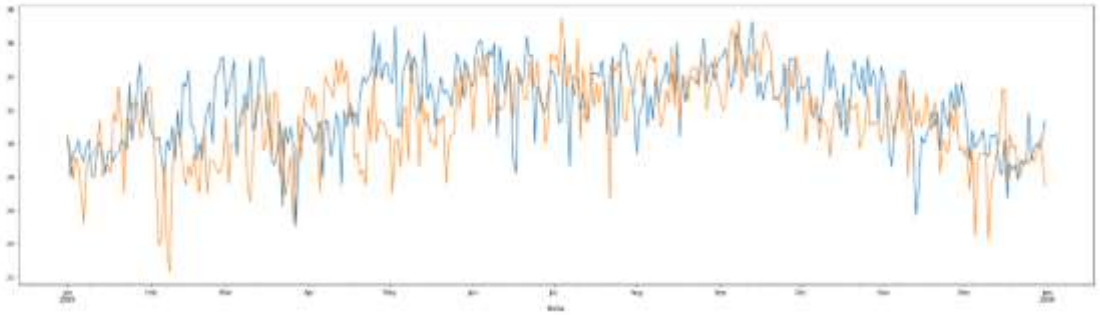


Figura 35 Grafico Compartido De Entrenamiento 1 y 2

Fuente. Elaboración propia

Se observa que los datos usados para el entrenamiento siguen un patrón, sin embargo, tienen variaciones muy bruscas en algunas secciones del periodo seleccionado, lo que podría afectar los resultados.

Elaboración de la red neuronal:

Como en el ejemplo de precipitación (sección 3.1.2.3) se usa la red neuronal de tipo multilayer-perceptron de la librería sknn, para este caso se usan 800 neuronas, una tasa de aprendizaje de 0,00001 y 8000 iteraciones, ver (figura 36).

```
In [8]: from sknn.mlp import Regressor, Layer

        capasinicio = TodasEstaciones.loc['1/01/1989':'31/12/1989'].as_matrix()[:,[15,16]]
        capasalida = TodasEstaciones.loc['1/01/1989':'31/12/1989'].as_matrix()[:,0]
        neurones =  800
        tasaaprendizaje = 0.00001
        numiteraciones = 800

        #Definition of the training for the neural network
        redneural = Regressor(
            layers=[
                Layer("ExpLin", units=neurones),
                Layer("ExpLin", units=neurones), Layer("Linear")],
            learning_rate=tasaaprendizaje,
            n_iter=numiteraciones)
        redneural.fit(capasinicio, capasalida)
        valortest = ([])

        for i in range(capasinicio.shape[0]):
            prediccion = redneural.predict(np.array([capasinicio[i,:].tolist()]))
            valortest.append(prediccion[0][0])
```

Figura 36 Red Neuronal - Caso de Estudio

Fuente: Elaboración Propia

Se crea un array con el nombre de valortest y se rellena con los datos predichos y se grafican los resultados ver (Figura 37).

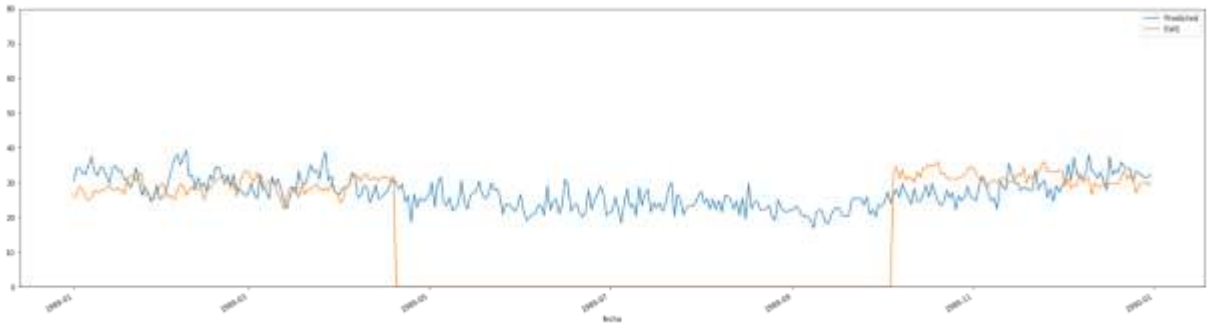


Figura 37 Resultado Red Neuronal - Caso de Estudio

Fuente: Elaboración propia

Según W*eather Spark en* Cúcuta, los veranos son cortos, muy calientes y nublados; los inviernos son calientes y mayormente nublados y está opresivo durante todo el año. Durante el transcurso del año, la temperatura generalmente varía de 22 °C a 33 °C y rara vez baja a menos de 20 °C o sube a más de 35 °C, se tomaran estos valores como referencia para nuestro rango de uso (“El clima promedio en Cúcuta,” 2019).

En este caso se ve que no se tiene una convergencia tan clara como el ejemplo de la red neuronal, esto es porque no se hizo el ejercicio con condiciones óptimas, es decir empleando varias estaciones. Sin embargo, los valores no salen del rango permitido en dicha variable por tanto estos valores se pueden utilizar sin ningún problema.

B	C
1/01/1989	26,0211248
2/01/1989	29,3025982
3/01/1989	29,669591
4/01/1989	28,0995316
5/01/1989	27,9942926
6/01/1989	30,4530113
7/01/1989	32,6918976
8/01/1989	28,6859675
9/01/1989	27,5020244
10/01/1989	29,691222
11/01/1989	29,3997694
12/01/1989	27,314539
13/01/1989	25,4362832
14/01/1989	29,4738592
15/01/1989	30,0757814
16/01/1989	28,7884752
17/01/1989	29,0802031
18/01/1989	26,2546327
19/01/1989	26,2399975
20/01/1989	24,0806167
21/01/1989	25,630907
22/01/1989	29,8633342
23/01/1989	25,8421902

Figura 38 Tabla de Resultados

Fuente: Elaboración propia

Los datos predichos se pasan a un documento en Excel y se agregan los días del año a los cuales pertenecen. Luego se verifica la región de datos faltantes en el archivo del IDEAM y se pegan los datos de la red neuronal en las fechas correspondientes.

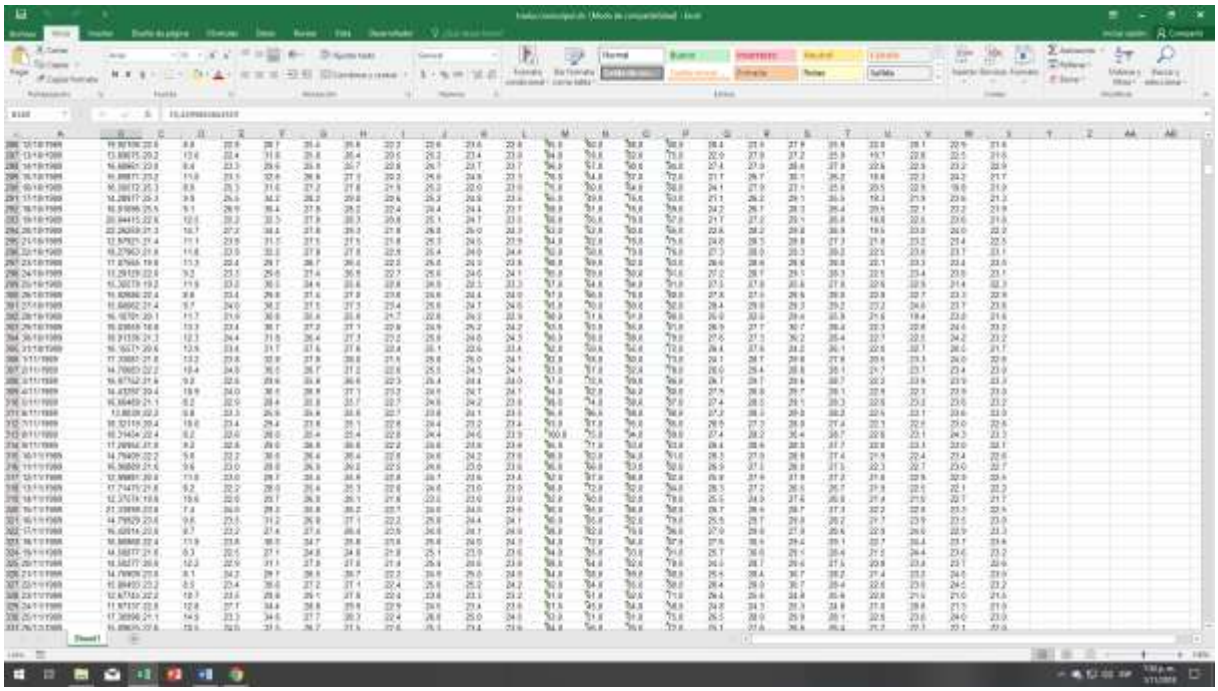


Figura 39 Resultados Rellenados

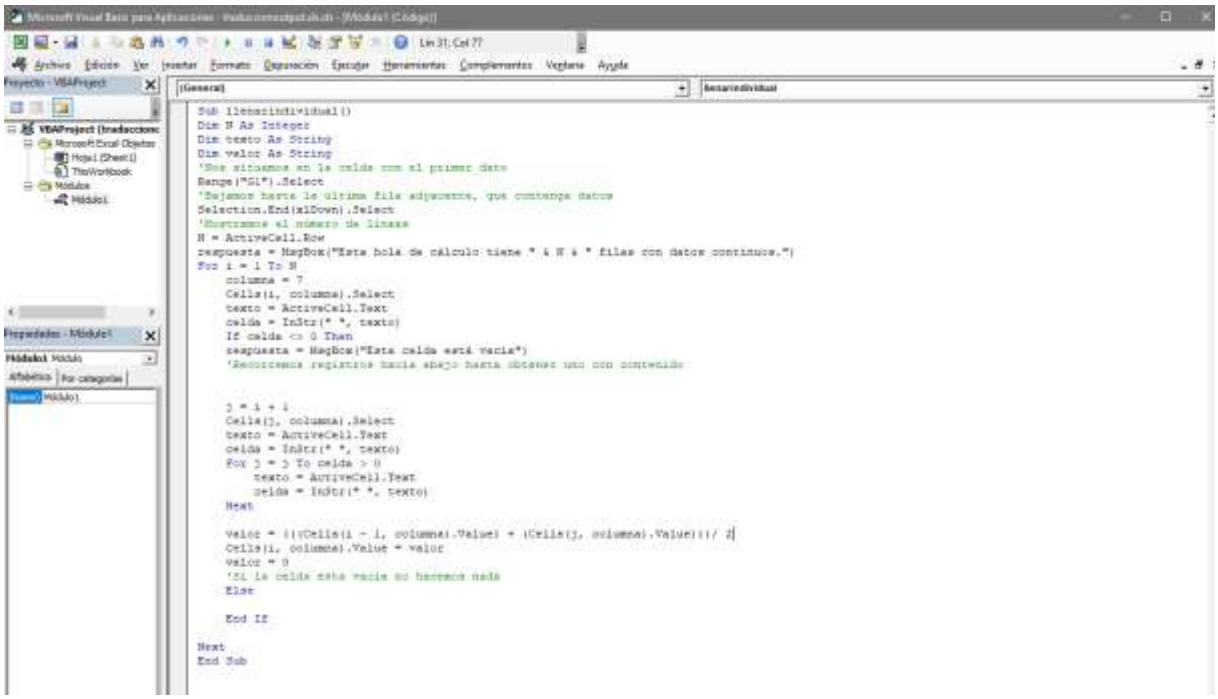
Fuente: Elaboración propia

3.2.2.3 Completación de Datos Meteorológicos individuales.

Además de los grandes grupos de datos faltantes que se consiguen en el documento a causa de mantenimiento o errores humanos, también se pueden conseguir datos faltantes en pequeñas cantidades o individuales estos ocurren debido a fallas en los sensores, sistema de monitoreo o fallas eléctricas.

Para llenar estos datos hay varias estrategias, pero una de las más sencillas y que da mejores resultados es la de promediar los valores cercanos antes y después de la perdida, ya sea horas antes y horas después como también valores a la misma hora, pero en días cercanos a la falla. Esto se debe a que los factores como lo son temperatura, precipitación o velocidad de vientos son progresivos y no tienen cambios bruscos en alguna hora específica del día.

Para el caso de la variable de estudio que es la temperatura se encuentran muchos registros individuales con valores nulos probablemente por las causas ya mencionadas, se elaboró un macro en visual basic que buscaba los registros nulos, y llenaba dichos registros con el promedio de los registros vecinos ver (Figura 40).



```
Sub llenarindividual()
Dim N As Integer
Dim texto As String
Dim valor As String
'Nos situamos en la celda con el primer dato
Range("A1").Select
'Bajamos hasta la ultima fila adyacente, que contenga datos
Selection.End(xlDown).Select
'Guardamos el número de linea
N = ActiveCell.Row
respuesta = MsgBox("Esta hoja de cálculo tiene " & N & " filas con datos continuos.")
For i = 1 To N
    columna = 7
    Cells(i, columna).Select
    texto = ActiveCell.Text
    celda = InStr(" ", texto)
    If celda <> 0 Then
    respuesta = MsgBox("Esta celda está vacía")
    'Recorremos registros hacia abajo hasta obtener uno con contenido


    j = i + 1
    Cells(j, columna).Select
    texto = ActiveCell.Text
    celda = InStr(" ", texto)
    For j = j To celda > 0
        texto = ActiveCell.Text
        celda = InStr(" ", texto)
    Next

    valor = ((Cells(i - 1, columna).Value) + (Cells(j, columna).Value))/ 2
    Cells(i, columna).Value = valor
    valor = 0
    'Si la celda esta vacia no hacemos nada
    Else

    End If

Next
End Sub
```

Figura 40 Macro Visual Basic Rellenar Datos Individuales

Fuente: Elaboración propia

Para rellenar los registros solo falta cambiar la variable columna por el número de la columna que se quiere completar y quedan completos todos los datos del archivo.

3.2.3 Fase de carga al almacén de datos.

Esta es la última fase del procedimiento, en ella se hará la migración de los datos a un entorno en donde se puedan consultarlos a una mayor velocidad gracias a que las herramientas utilizadas en esta fase están optimizadas para una gran cantidad de registros. Los datos serán almacenados en una base de datos NoSQL o en una Data Warehouse, el único concepto que comparten es que ambos se utilizan para analizar grandes cantidades de datos.

Las soluciones NoSQL generalmente administran esquemas relativamente limitados con una gran cardinalidad en pocas entidades, mientras que los almacenes de datos suelen tener muchos hechos y dimensiones (en un modelo dimensional) o muchas entidades en un modelo 3NF. Los sistemas DW generalmente administran varias líneas de negocios e intentan combinar esos datos.

Los sistemas DW normalmente tienen capacidades de generación de informes en SQL, lo que le permite acceder a todos los datos de forma estándar. Los sistemas NoSQL generalmente están más basados en código, por ejemplo, Map / Reduce.

Según la estructura de los datos y opiniones de la empresa se hará una elección entre NoSQL y Data Warehouse, pero para este procedimiento se explicará las dos opciones.

3.2.3.1 Carga hacia Base de datos Data WareHouse según Metodologia Ralph Kimball.

Modelo MER

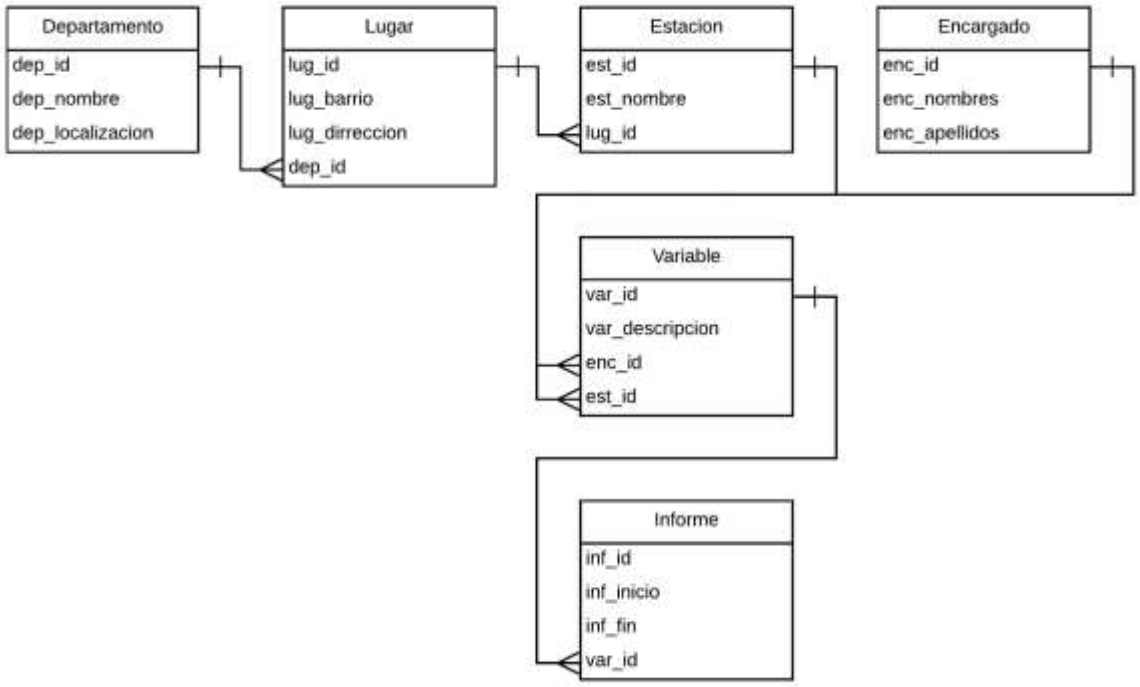


Figura 41 Ejemplo de Modelo MER - Caso de estudio

Fuente: Elaboración propia

Modelo Data Warehouse

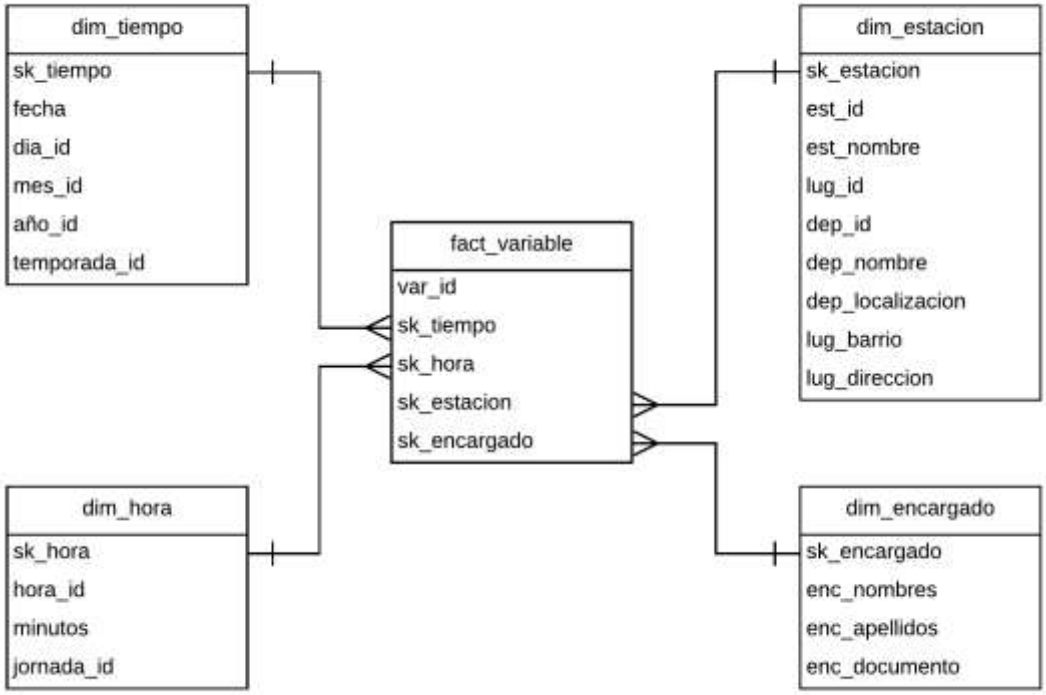


Figura 42 Ejemplo de Modelo DATA WAREHOUSE- Caso de estudio

Fuente: Elaboración propia

En este caso se va a cargar tabla por tabla desde el Pentaho, para esto se tiene que crear primero la base de datos y la estructura de ésta con el paso "Execute SQL script" ver (Figura 43).

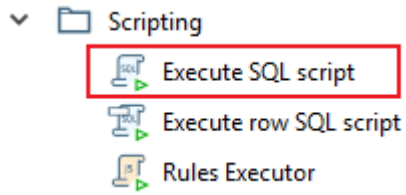


Figura 43 Paso para ejecutar SQL en Pentaho

Fuente: Elaboración propia

En este ejemplo se crea la tabla “dim_tiempo”, para esto se selecciona la opción anterior y se escribe el código como si se tratase de la consola en “SQL” ver (Figura 44).

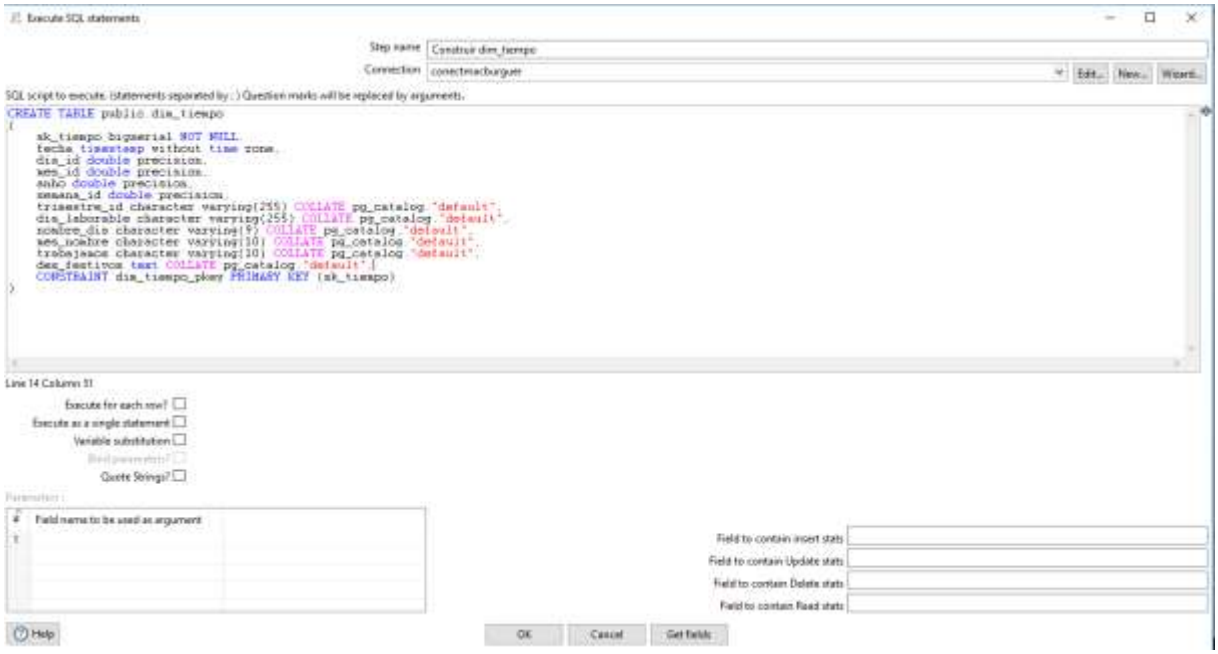


Figura 44 Ejemplo creacion tabla Pentaho

Fuente: Elaboración propia

Después de esto se le da en la opción “edit”, esta abre otra ventana con las opciones de conexión hacia el gestor de bases de datos, en este caso se usó “PostgreSQL” y se configuran las opciones como ver (Figura 45):

- Nombre de la base de datos
- Usuario
- Contraseña

- Puerto
- Hostname

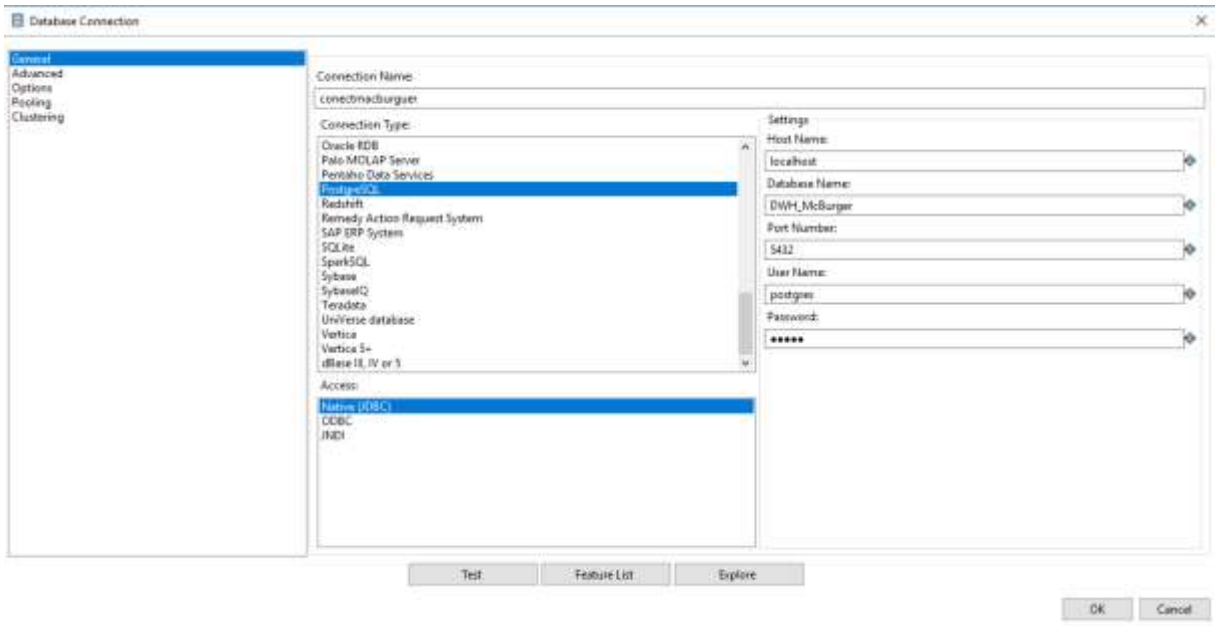


Figura 45 Configuración de Conexión

Fuente: Elaboración propia

Luego que ya este creada la tabla en el gestor de base de datos se procede a llenarla, esto se hace con el paso “table output” en la sección “output” ver (Figura 46).

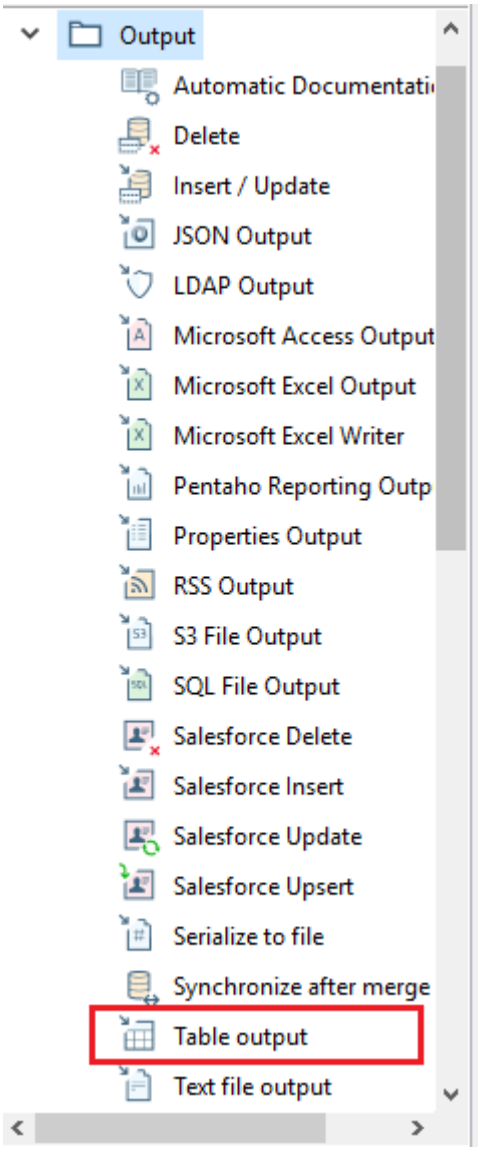


Figura 46 Opciones de salida Pentaho

Fuente: Elaboración propia

Se abrirá la siguiente ventana para poner las opciones de carga hacia la base de datos “SQL” ver (Figura 47)

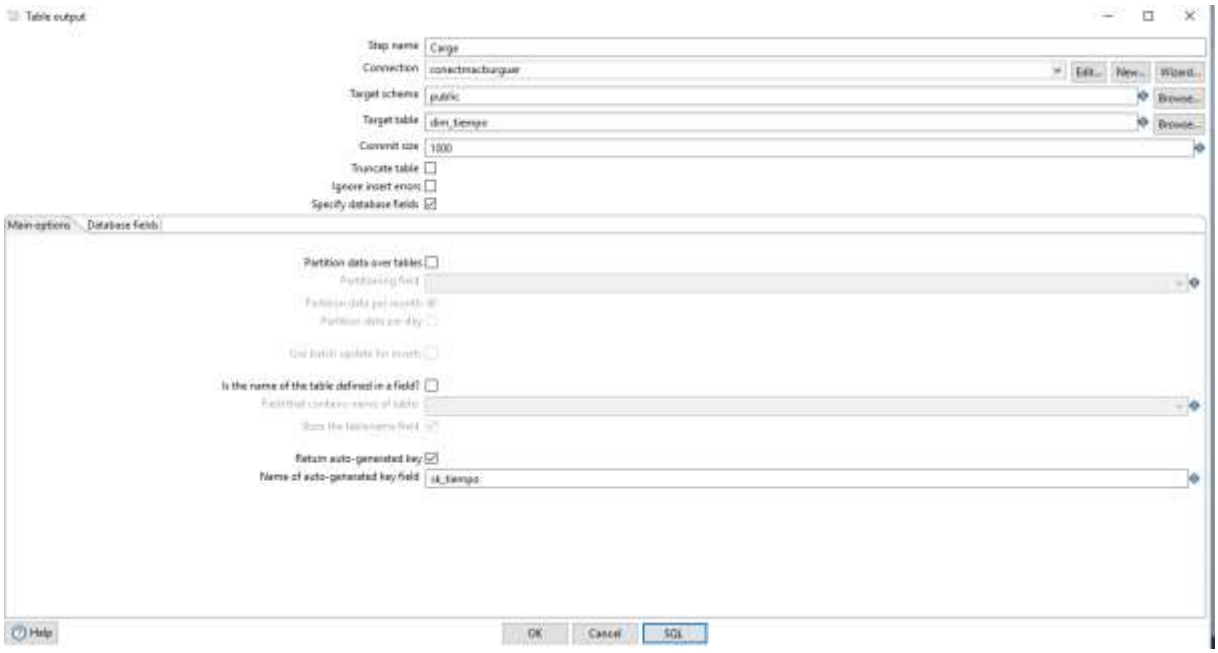


Figura 47 Ventana de configuración de carga SQL

Fuente: Elaboración propia

La conexión debería aparecer como ya hecha, si no, en el botón “Edit” aparecerán las opciones para volverla a configurar. Se selecciona un esquema que por defecto será “public” y la tabla objetivo para rellenar. Luego se tilda el cuadro “Return auto-generated key”, esta opción es para crear una llave subrogada, y se le da un nombre en el siguiente campo.

Luego de esta configuración básica, se pulsa la opción “Database fields” ver (Figura 48)

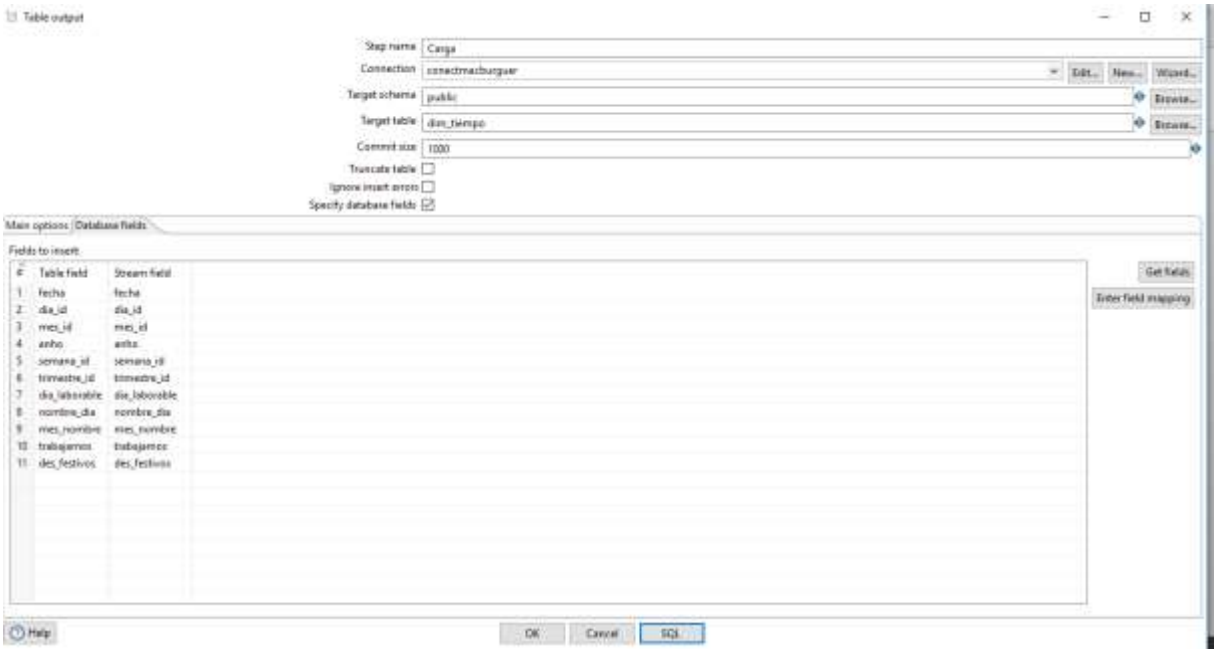


Figura 48 Carga de los campos a la tabla

Fuente: Elaboración propia

Se pulsa la opción "Get fields" y se le da el nombre que tiene en la base de datos anteriormente creada. Luego se pulsa "ok" para cargar los datos a la tabla en PostgreSQL.

3.2.3.2 Carga hacia Base de datos NoSQL.

Carga hacia MongoDb desde un archivo de Excel ver (Figura 49).

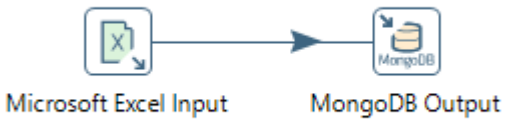


Figura 49 Carga desde Excel hacia Mongo

Fuente: Elaboración propia

Se extrae la salida de la fase anterior y se relaciona con el paso “MongoDB Output” ver (Figura 50).

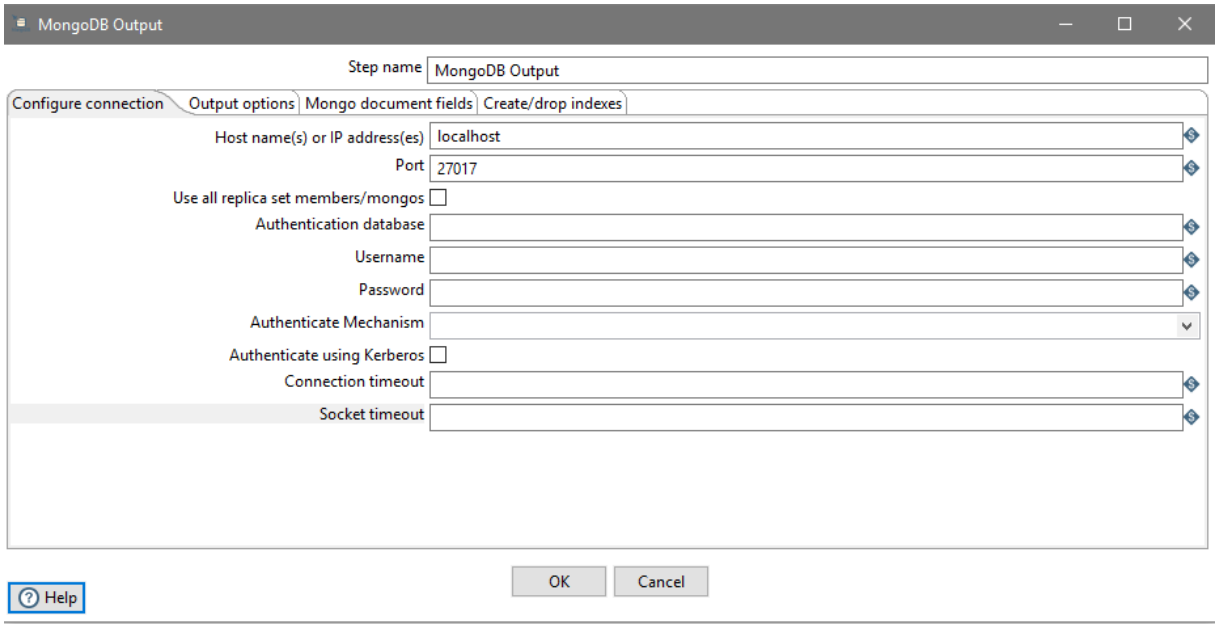


Figura 50 Configuración de conexión a mongo

Fuente: Elaboración propia

En la ventana de configurar conexión estará la conexión por defecto, así que no hay que mover nada, a menos de que se tenga un usuario y contraseña establecidos.

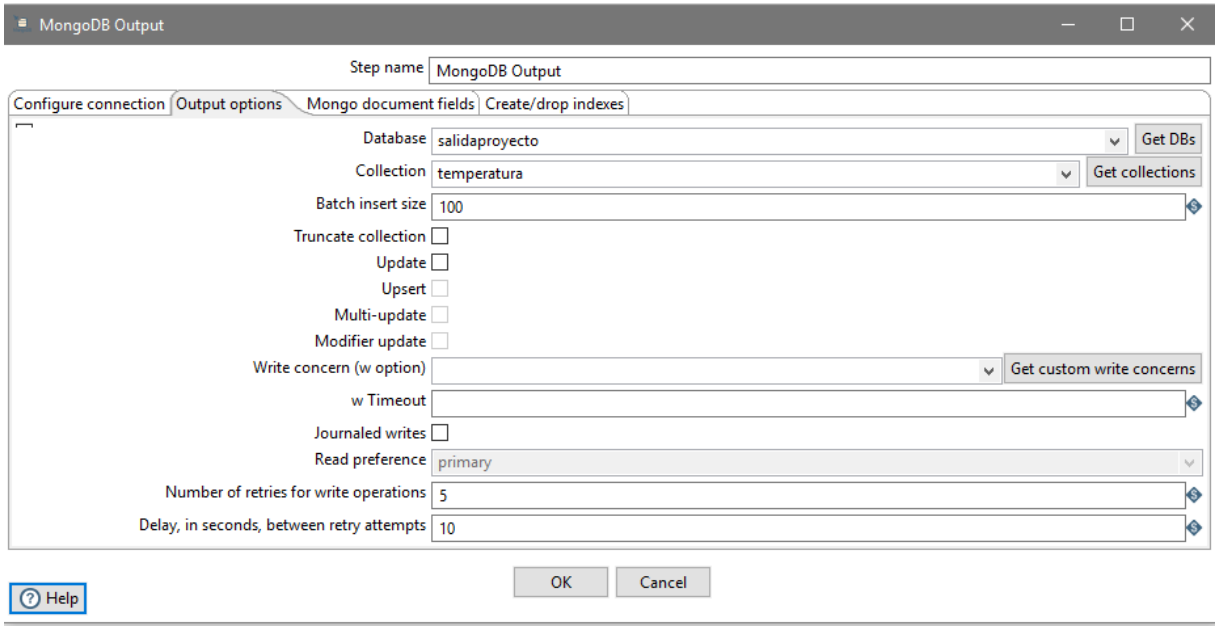


Figura 51 Configuración de documento de mongo

Fuente: Elaboración propia

En la ventana de opciones de salida aparecerán las opciones de la carga hacia la base de datos, cabe resaltar que, si no hay una base de datos creada con el nombre registrado, está se creará, igual si no existe una colección con el nombre registrado, esa colección se creará.

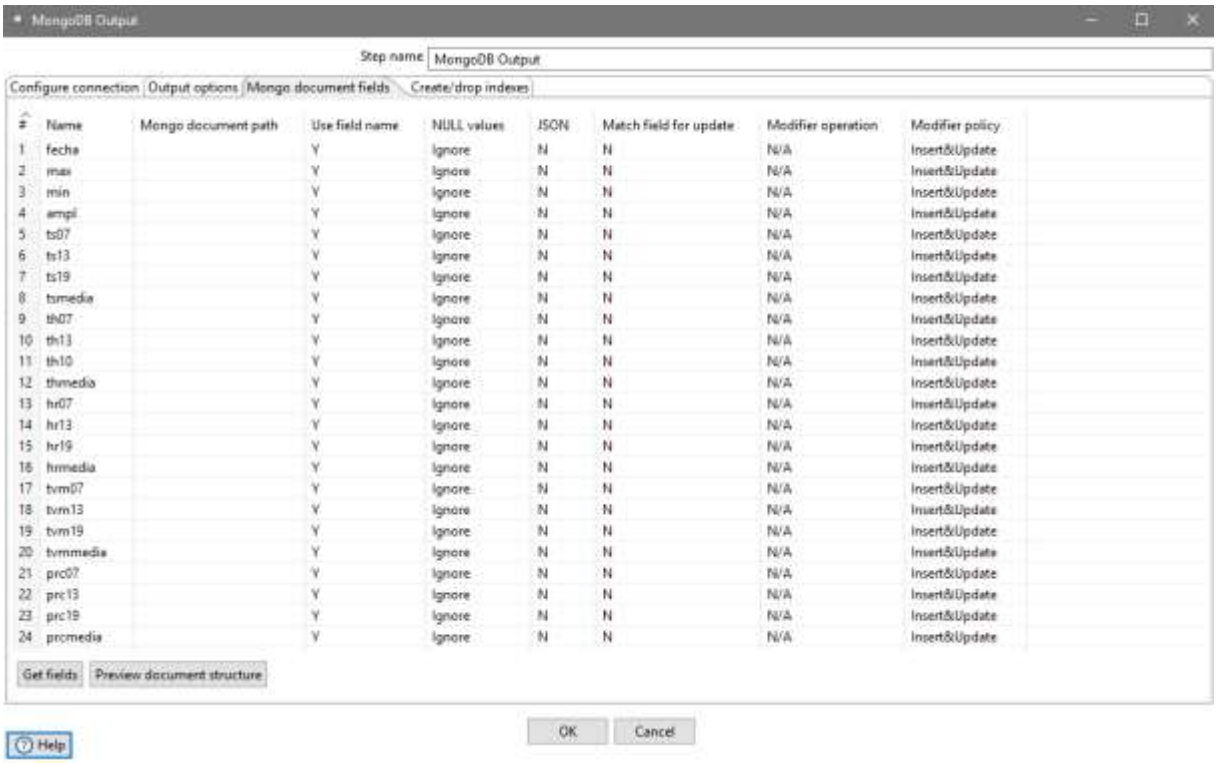


Figura 52 Carga de los encabezados a documento mongo

Fuente: Elaboración propia

En la pestaña “Mongo document fields” aparece la opción en el pie de la ventana “Get fields” está hará que se trasladen las cabeceras de la tabla para hacer los campos en el documento de mongo.

Por último, se mira la estructura del documento con la opción “Preview document structure”

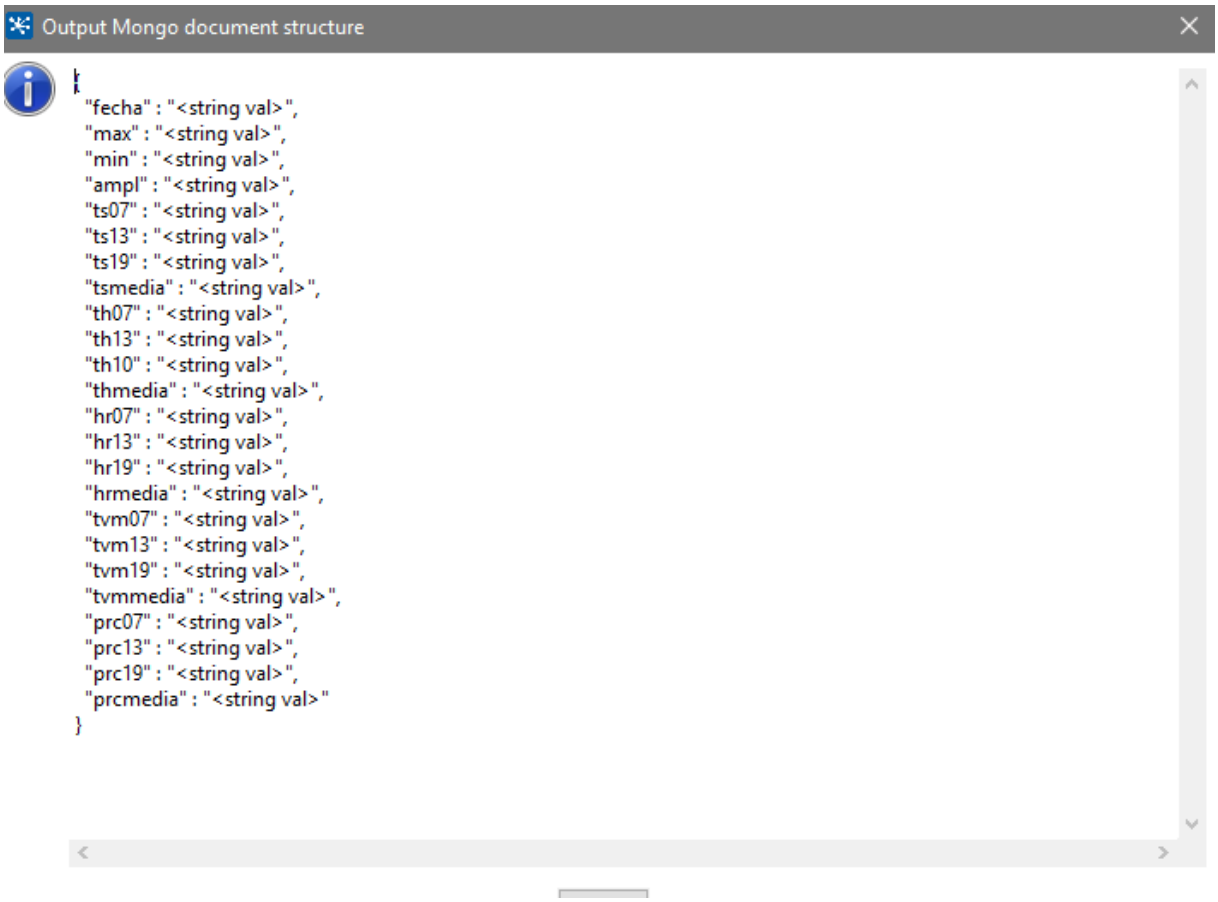


Figura 53 Estructura del documento mongo

Fuente: Elaboración propia

Cuando se corra la transformación se tiene que verificar que efectivamente los datos si se cargaron hacia mongo. Para esto se abre símbolo del sistema y se escribe “mongo”, luego se escribe “show dbs” para consultar las bases de datos.

Figura 54 Bases de datos mongo consultadas desde terminal

Fuente: Elaboración propia

Una vez comprobada que existe esta data, se escribe el comando "use xxxxxxx" reemplazando las "x" por el nombre de la base de datos, en este caso "salidaproyecto". Hay que tener en cuenta que el comando "use" puede crear una base de datos, así que, si no se escribe bien el nombre de la base de datos, se crea una data con el nombre erróneamente escrito. Luego, de esto se tiene que consultar las colecciones con el comando "show collections".

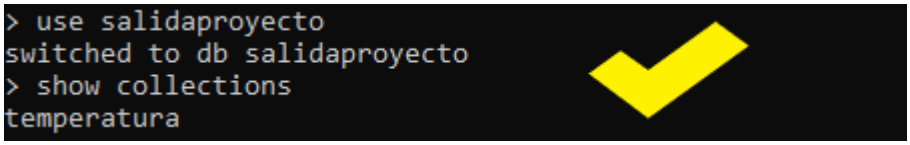


Figura 55 Colecciones consultadas desde terminal

Fuente: Elaboración propia

En este caso aparece la colección "temperatura" que es el nombre que se le dio en el entorno de pentaho, aquí ya se pueden consultar los datos que están en esa colección con el comando "db.xxxxxxx.find().pretty()" reemplazando las "x" por el nombre de la colección, el agregado "pretty()" es para que la información se vea de una manera más organizada en el símbolo del sistema.

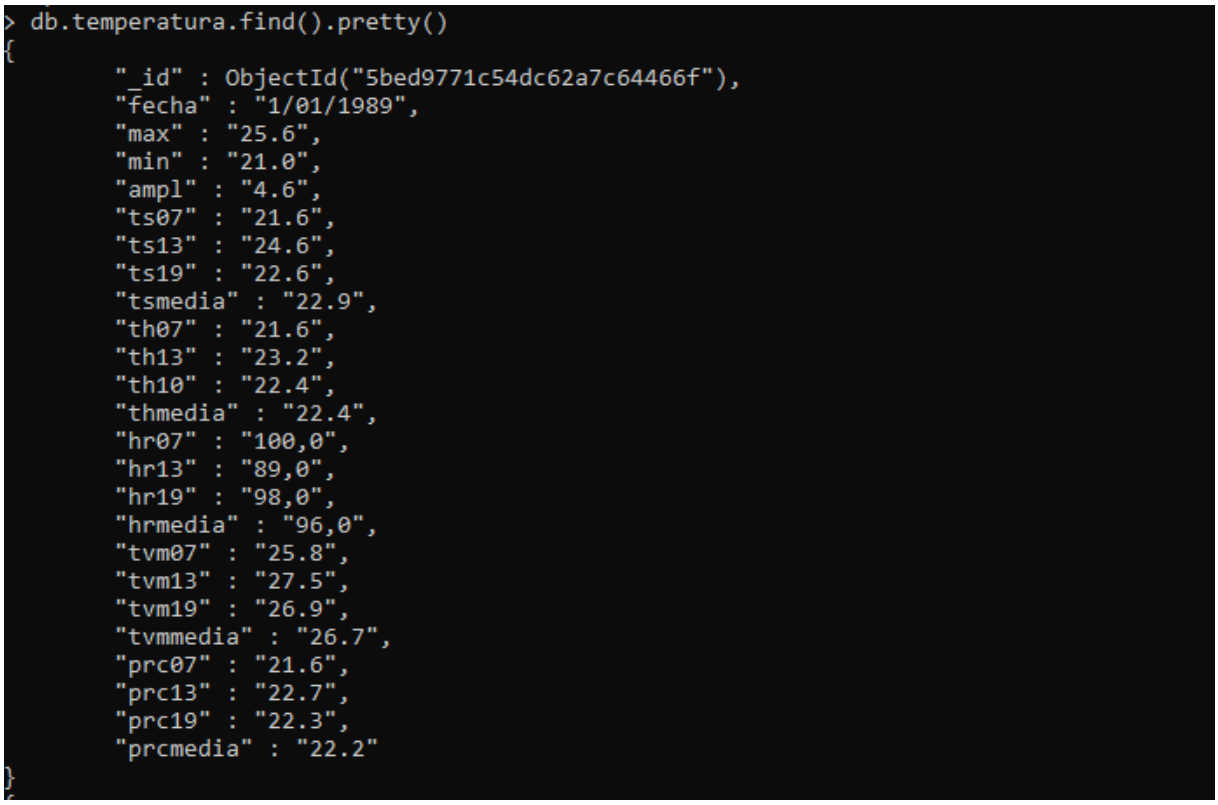


Figura 56 Muestra de la carga de datos hacia mongo

Fuente: Elaboración propia

Se comprueba que la información este bien estructurada, sólo se verán los primeros registros. Para ver más es necesario escribir en comando “it”.

4 Resultados

Cumpliendo con los objetivos específicos de la sección 1.3.2 el resultado principal es el documento del procedimiento que tiene la descripción de las actividades que deben seguirse para el tratamiento de datos climáticos plasmando una serie de métodos y estrategias Big Data para mejorar su procesamiento.

Con respecto a las fuentes de datos ambientales y a los datos resultantes del procedimiento se encuentra un avance muy importante, ya que los posteriores procesos de consulta y análisis en los datos resultantes van a ser más agiles, no se tendrán que hacer correcciones y la información va a estar completa con un alto grado de integridad y calidad.

Los archivos (.ktr) generados del Pentaho detallan las actividades hechas y reflejan el proceso de cada una de las fases, además también se anexan los archivos de entrada y salida desde las fuentes originales dadas por el IDEAM hasta la base de datos resultante del procedimiento. También se anexa el archivo (.ipynb) donde se trabajó el rellenado de datos con el uso de redes neuronales, este archivo se abre con *Jupyter Notebook* para visualizar bien cada una de las líneas de código y las imágenes que se muestran en los resultados, junto con éste los archivos de entrenamiento que necesita la red neuronal para ponerla en funcionamiento.

5 Recomendaciones y trabajos futuros

Para la fase de extracción se plantea trabajar en la aplicación de una arquitectura que comprenda la integración de herramientas para datos estructurados y no estructurados al tiempo. Además, se espera, poder incluir nuevas fuentes de datos para contar con mayor información del comportamiento de las variables y entregar otro tipo de transformaciones. Además, se propone investigar acerca de otras herramientas Big Data para esta fase del procedimiento.

Para la fase principal se plantea ampliar las técnicas para el tratamiento de los datos y también mejorar los métodos aplicados:

- Para la red neuronal se plantea revisar otros tipos de algoritmos, el usado en el caso de estudio fue Regresión Lineal pero la librería sknn cuenta con otros con los que se podrían hacer pruebas.
- Para el rellenado de los datos en general se recomienda evaluar otras estrategias y comparar los datos obtenidos con los resultantes de este proyecto, además verificar cual sería el método más idóneo para utilizar en problemas relacionados con Big Data.

Finalmente, en la fase de carga se recomienda cargar los datos modelando una base de datos relacional enfocada a la metodología Bill Inmon o bien trabajar con otra base de datos NoSQL diferente a la que se trabajó en el proyecto y destacando las diferencias entre su tipología y rendimiento.

6 Conclusiones

- El análisis de datos climatológicos llega a ser de manera extensa, debido a la cantidad inmensa que se recolectan sobre las variables que influyen en el clima; necesitando un almacenamiento de alta capacidad y un procesamiento veloz de los mismos; por lo cual Big Data coadyuva de gran manera para el análisis y procesamiento de estos datos, siendo una solución.

- El procedimiento que se formalizó en el proyecto demuestra que Big Data es un campo de estudio en donde prima el manejo de grandes cantidades de datos priorizando la integridad y la calidad de la información.

- Debido a los problemas generados en la meteorología en el último siglo por el crecimiento de datos queda claro que se necesitan mejorar las tecnologías de recolección y tratamiento de los datos para poder contribuir a que los estudios y predicciones ambientales sean más confiables.

- Como pudo observar el lector, herramientas como Python o Pentaho permiten manipular grandes cantidades de información, además como se mostró en el ejemplo de la red neuronal se puede mejorar la información para posteriormente generar conocimiento.

7 Bibliografía

- ¿Para qué sirve un sistema de Business Intelligence? (2017). Retrieved from http://www.preguntia.com/para-que-sirve-un-sistema-de-business-intelligence.html
- Aguilar, L. J. (2013). *Big Data - Analisis de grandes volumenes de datos en organizaciones*.
- Brathwaite, C., & Doug Moran. (2015). Pentaho Data Integration (Kettle) Tutorial. Retrieved from https://wiki.pentaho.com/display/EAI/Pentaho+Data+Integration+%28Kettle%29+Tutorial
- Briega, R. E. L. (2015). Machine Learning con Python. Retrieved from https://relopezbriega.github.io/blog/2015/10/10/machine-learning-con-python/
- Business Intelligence y Big Data. ¿Son lo mismo? (2018). Retrieved from http://www.conasa.es/blog/business-intelligence-y-big-data-son-lo-mismo/
- Bussiness Inteligence Facil. (2015). Claves Subrogadas. Retrieved from https://www.businessintelligence.info/serie-dwh/claves-subrogadas.html
- Corporación Colombia Digital. (2017). ¿Qué es un data warehouse y qué beneficios aporta a las organizaciones? *Colombiadigital.Net*. Retrieved from https://colombiadigital.net/actualidad/articulos-informativos/item/9814-que-es-un-data-warehouse-y-que-beneficios-aporta-a-las-organizaciones.html?fbclid=IwAR1qnWr0o7rfSLudtRyVQEPq8hn5c5yAn5RGy3szt0em2_auYdlU99HS_28
- DATA IS THE NEW OIL. (2016). *Spotlessdata*. Retrieved from https://spotlessdata.com/blog/data-new-oil
- Definición de Sistema transaccional (sistema de procesamiento de transacciones). (2018). Retrieved from http://www.alegsa.com.ar/Dic/sistema_transaccional.php
- Duque Méndez, N. D., Hernández Leal, E. J., Pérez Zapata, Á. M., Arroyave Tabares, A. F., & Espinosa Gómez, D. A. (2016). Modelo para el proceso de extracción, transformación y carga en bodegas de datos. Una aplicación con datos ambientales. *Ciencia e Ingeniería Neogranadina*, *26*(2), 95–109. https://doi.org/10.18359/rcin.1799
- El clima promedio en Cúcuta. (2019). Retrieved from https://es.weatherspark.com/y/25316/Clima-promedio-en-Cúcuta-Colombia-durante-todo-el-año
- Erika Díaz de Argandoña, A. S. T. (2016). Predicciones de Tecnología, Medios de Comunicación y Telecomunicaciones. Retrieved from https://www2.deloitte.com/content/dam/Deloitte/es/Documents/tecnologia-media-telecomunicaciones/Deloitte_ES_TMT_Predicciones-2016.pdf

- Espinosa, R. (2015). Construccion procesos ETL utilizando Kettle (Pentaho Data Integration). *El Rincon Del BI*. Retrieved from https://churriwifi.wordpress.com/2010/05/10/16-3-construccion-procesos-etl-utilizando-kettle-pentaho-data-integration/
- Feregrino, A. (2018). ¿Qué es MapReduce? Retrieved from https://thatcsharpguy.com/tv/mapreduce/
- Fernández, C. (2018, January 6). Cómo el Big Data puede ayudar a luchar contra el cambio climático. *Business Insider*. Retrieved from https://www.businessinsider.es/como-big-data-puede-ayudar-luchar-cambio-climatico-183600
- Fernando, R. (2016). BIG DATA EN EL COMPORTAMIENTO DE DATOS CLIMATOLÓGICOS Y ESTRATEGIAS INTERNACIONALES DE REDUCCIÓN DE DESASTRES PARA LA GESTION DE RIESGO AMBIENTAL. *UNIVERSIDAD MAYOR DE SAN ANDRÉS FACULTAD*.
- Fragoso, R. B. (2018). ¿Qué es Big Data? *IBM*. Retrieved from https://www.ibm.com/developerworks/ssa/local/im/que-es-big-data/index.html
- Fundamentos de Apache Hadoop y MapReduce. (2018). Retrieved from https://geekytheory.com/fundamentos-de-apache-hadoop-y-mapreduce
- gravitar. (2017). ¿Qué es pentaho? Retrieved from https://gravitar.biz/pentaho/
- GUILLEN BETANCOURT, A. (2014). Universidad de pamplona, (7), 5685305.
- Hernandez, Y. M. (2015). Pasos del Pentaho Data Integration en un contexto big data.
- Hung LeHong. (2012). Reporte de Gartner analiza "big data" alrededor de tecnología de datos. Retrieved from https://searchdatacenter.techtarget.com/es/noticias/2240171952/Reporte-de-Gartner-analiza-big-data-alrededor-de-tecnologia-de-datos
- James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and A. H. B. (2011). Big data: The next frontier for innovation, competition, and productivity. *McKinsey Global Institute*. Retrieved from https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/big-data-the-next-frontier-for-innovation
- Julián Pérez Porto, & Gardey, A. (2012). CONCEPTO DE INFORMACION. Retrieved from https://definicion.de/informacion/
- Laney, D. (2012). Deja VVVu: Others Claiming Gartner's Construct for Big Data, https://blogs.gartner.com. Retrieved from https://blogs.gartner.com/doug-laney/deja-vvvue-others-claiming-gartners-volume-velocity-variety-construct-for-big-data/
- Leninmhs. (2018). Manual para la Herramienta PDI – KETTLE de Pentaho. Retrieved from https://tubasededatoslibre.org.ve/manual-pdi-kettle-pentaho/
- Merv, A. (2011). Big Data. Retrieved from http://www.nxtbook.com/nxtbooks/mspcomm/teradata_2011q1/index.php?startid=8#/40
- Mesa, A. R. (2018). BIG DATA: La evolución de los datos. *Openwebinars.Net*. Retrieved from https://openwebinars.net/blog/big-data-la-evolucion-de-los-datos/?cat=big-data

- Nathan Sykes. (2018). Big Data y la lucha contra el cambio climático. Retrieved from https://www.bbvaopenmind.com/big-data-y-la-lucha-contra-el-cambio-climatico/
- Pinto, J. M. C. (2017). ¿Que es una ETL? Retrieved from https://www.linkedin.com/pulse/que-es-una-etl-juan-manuel-castillo-pinto
- Roberto Espinosa. (2010). Kimball vs Inmon. Ampliación de conceptos del Modelado Dimensional. Retrieved from https://churriwifi.wordpress.com/2010/04/19/15-2-ampliacion-conceptos-del-modelado-dimensional/?fbclid=IwAR0aiuPac6Z709UYa6YcJNDgsEHH-StiWs7zeBfrZH2mRATZgFaCb3QOFRc
- Rouse, M. (2016a). NoSQL (No Solo SQL). *Bases de Datos*. Retrieved from https://searchdatacenter.techtarget.com/es/definicion/NoSQL-No-Solo-SQL
- Rouse, M. (2016b). SQL o lenguaje de consultas estructuradas. *Bases de Datos*. Retrieved from https://searchdatacenter.techtarget.com/es/definicion/SQL-o-lenguaje-de-consultas-estructuradas
- Salas, R. (n.d.). Redes Neuronales Artificiales, 1–7. Retrieved from https://www.academia.edu/24633757/Redes_Neuronales_Artificiales
- SalesForce. (2018). ¿Qué es Cloud Computing? Retrieved from https://www.salesforce.com/mx/cloud-computing/
- Sánchez, M. A. P. (2017). MODELIZACION DE UNA DWH.
- TAMAYO NEYRA ANTONIO. (2017, March 17). Desafío "Data for Climate Action." *EL FINANCIERO*. Retrieved from http://www.elfinanciero.com.mx/monterrey/desafio-data-for-climate-action
- TECNÓSFERA. (2017, February 21). El Ideam le apuesta al "big data" para medir la deforestación. *EL TIEMPO*, p. 01. Retrieved from http://www.eltiempo.com/archivo/documento/CMS-16825328
- Vega, J. J. C., Ortega, J. F. C., & Aguilar, L. J. (2016). Knowing the Big Data. *Evistas.Uptc.Edu.Co*. Retrieved from https://revistas.uptc.edu.co/index.php/ingenieria/article/view/3159/4346
- Yllanes, D. E. (2012). Nuevas tendencias tecnológicas han entrado a AL, idclatin. Retrieved from http://mx.idclatin.com/releases/news.aspx?id=1433
- ZALDÍVAR, A. R. (2014). IMPLEMENTACIÓN DE UN DATA MART COMO SOLUCIÓN DE INTELIGENCIA DE NEGOCIOS, BAJO LA METODOLOGÍA DE RALPH KIMBALL PARA OPTIMIZAR LA TOMA DE DECISIONES EN EL DEPARTAMENTO DE FINANZAS DE LA CONTRALORÍA GENERAL DE LA REPÚBLICA.