Tecnológico Nacional de México
Instituto Tecnológico de Apizaco
Maestría en Sistemas Computacionales

"MODELADO E IMPLEMENTACIÓN DE UN DATA WAREHOUSE PARA UN MDMS UTILIZANDO ALGORITMOS ETL Y VEE".

Ing. Mario Alberto Méndez Carmona

Julio, 2018

Tecnológico Nacional de México
Instituto Tecnológico de Apizaco
Maestría en Sistemas Computacionales

"MODELADO E IMPLEMENTACIÓN DE UN DATA WAREHOUSE PARA UN MDMS UTILIZANDO ALGORITMOS ETL Y VEE".

Ing. Mario Alberto Méndez Carmona

Proyecto de grado presentado para optar al Grado Académico de *Maestro en Sistemas Computacionales*.

Director: MD. Higinio Nava Bautista

Codirector: M.C. José Juan Hernández Mora
Tutor: M.C. María Guadalupe Medina Barrera
Revisor: M.C. Juan Ramos Ramos

Julio, 2018

SEP

TECNOLÓGICO NACIONAL DE MEXICO

Instituto Tecnológico de Apizaco

Apizaco, Tlax., 25 de Junio de 2018

No. de Oficio: DEPI/234/18

ASUNTO: **Se Autoriza Impresión de Tesis de Grado.**

ING. MARIO ALBERTO MÉNDEZ CARMONA
CANDIDATO AL GRADO DE MAESTRO
EN SISTEMAS COMPUTACIONALES
No. de Control: **M11370834,**
P R E S E N T E.

Por este medio me permito informar a usted, que por aprobación de la Comisión Revisora asignada para valorar el trabajo, mediante la Opción: **I Tesis de Grado por Proyecto de Investigación**, de la **Maestría en Sistemas Computacionales**, que presenta con el tema: **"MODELADO E IMPLEMENTACIÓN DE UN DATA WAREHOUSE PARA UN MDMS UTILIZANDO ALGORITMOS ETL Y VEE"** y conforme a lo establecido en el Procedimiento para la Obtención del Grado de Maestría en el Instituto Tecnológico, la División de Estudios de Posgrado e Investigación a mi cargo le emite la:

AUTORIZACIÓN DE IMPRESIÓN

Debiendo entregar un ejemplar del mismo debidamente encuadernado y seis copias en CD en formato PDF, para presentar su Acto de Recepción Profesional a la brevedad.

Sin otro particular por el momento, le envío un cordial saludo.

A T E N T A M E N T E
EXCELENCIA EN EDUCACIÓN TECNOLÓGICA®
PENSAR PARA SERVIR, SERVIR PARA TRIUNFAR®

DR. JOSÉ FEDERICO CASCO VÁSQUEZ
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN.

ESTADOS UNIDOS MEXICANOS
SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO
DIVISIÓN DE ESTUDIO DE POSGRADO E INVESTIGACIÓN

JFCV/MHSH*mebr.

C.p. Expediente.

SEP

TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Apizaco

Apizaco, Tlax., 16 de Mayo de 2018

ASUNTO: **Aprobación del trabajo de Tesis de Maestría.**

DR. JOSÉ FEDERICO CASCO VÁSQUEZ
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN.
P R E S E N T E.

Por este medio se le informa a usted, que los integrantes de la **Comisión Revisora** para el trabajo de tesis de maestría que presenta el **ING. MARIO ALBERTO MÉNDEZ CARMONA**, con número de control **M11370834,** candidato al grado de **Maestro en Sistemas Computacionales** y egresado del **Instituto Tecnológico de Apizaco,** cuyo tema es **"MODELADO E IMPLEMENTACIÓN DE UN DATA WAREHOUSE PARA UN MDMS UTILIZANDO ALGORITMOS ETL Y VEE"**, fue:

A P R O B A D O

Lo anterior, al valorar el trabajo profesional presentado por el candidato y constatar que las observaciones que con anterioridad se le marcaron así como correcciones sugeridas para su mejora ya han sido realizadas.

Por lo que se avala se continúe con los trámites pertinentes para su titulación.

Sin otro particular por el momento, le envió un cordial saludo.

LA COMISIÓN REVISORA

M.D.S. HIGINIO NAVA BAUTISTA

M.C. JOSÉ JUAN HERNÁNDEZ MORA

M.C. MARÍA GUADALUPE MEDINA BARRERA

M.C. JUAN RAMOS RAMOS

C. p.- Interesado.

Dedicatoria

Dedico este trabajo con todo cariño:

A mi maravilloso hijo Mario Daniel, porque gracias a el pude desarrollar este proyecto tan importante, el es mi motivación en la vida, además quiero ponerle el ejemplo de que con dedicación, esfuerzos y sacrificios todo se puede lograr en esta vida.

A mi esposa Carolina, que con su amor, cariño y aliento ha estado a mi lado como compañera, por lo tanto, cada proyecto, meta u objetivo que realice, también le pertenece porque ella me brinda su apoyo incondicionalmente.

A mi padre Mario Méndez, que siempre estuvo ahí para mi, poniendo el ejemplo de siempre hacer lo correcto, de ser una persona honesta, trabajadora y siempre anteponiendo a la familia.

A mi madre Juana Carmona, por su amor, cariño, consejos, sabiduría y por haberme alentado a dar los primeros pasos en el ámbito académico, y que hoy con orgullo gracias a ella pude culminar esta etapa que me ayudará a defenderme en la vida.

A mis hermanas Magaly y Celeste, por velar siempre por mi y brindarme ese cariño tan especial.

A la Familia Méndez Mejía y Carmona Vázquez, por creer en mi, por creer que podía superarme y llegar lejos y gracias a sus buenos deseos pude lograrlo.

Agradecimientos

A Dios por brindarme la vida, darme salud y entendimiento ya que gracias a ello pude lograr las metas y desafíos que a lo largo de mi vida he superado con éxito.

A mis padres Mario Méndez Mejía y Juana Carmona Vázquez por su apoyo incondicional y por brindarme su apoyo en todo momento alentando a cada día ser mejor poniéndome el ejemplo.

A los catedráticos del Instituto Tecnológico de Apizaco que fungieron como guía en mi desarrollo académico y profesional compartiendo su conocimiento y experiencias permitiéndome adquirir esa enseñanza que me acompañara y me ayudará en mi desempeño laboral.

A la empresa EosTech S.A. de C.V. ubicada en el estado de Querétaro por darme la oportunidad de desarrollarme profesionalmente permitiéndome realizar mis estancias profesionales por un lapso de medio año, adquiriendo la experiencia y conocimientos que me permitieron desarrollar este trabajo de grado.

Al M.C. Brigido Rodríguez Cruz egresado de esta misma institución por su apoyo como líder de proyecto en el área de desarrollo de software dentro de la empresa EosTech mencionada anteriormente.

A la comisión revisora de este trabajo conformado por MD. Higinio Nava Bautista, M.C. José Juan Hernández Mora, M.C. María Guadalupe Medina Barrera y M.C. Juan Ramos Ramos que con sus aporte hicieron posible la realización y publicación de este trabajo de grado.

A mis compañeros de maestría con los cuales compartí momentos agradables e hicieron ameno el tiempo de estudio.

A una compañera y amiga muy especial que estuvo a mi lado en los mejores y peores momentos ayudándome a superar las metas que nos propusimos juntos, alentándome a ser cada día mejor y destacando siempre en nuestras actividades.

Resumen

La comunicación inalámbrica se ha hecho una realidad con el paso de los años, cada vez encontramos una gran variedad de dispositivos que pueden transmitir y recibir información con o sin la necesidad de un cable. Por supuesto, los medidores de energía eléctrica no están exentos, cada vez son más la empresas que pretenden desarrollar *Smart Devices* implementando diferentes tecnologías de transmisión de datos tales como: radiofrecuencia, wi-fi, bluetooth, bandas 3G, GSM, etc. Todo esto con el fin de administrar los medidores de manera remota y centralizada, optimizando así la efectividad y la rapidez con la que los datos son recolectados.
Tomado como referencia esto, esta tesis propone el modelado e implementación de la capa de datos así como el modelado e implementación de un *Data Werehouse* para un Sistema de Gestión de Datos de Medidores *(MDMS)* capaz de administrar la red eléctrica del país de manera más eficiente. El trabajo desrrollado presenta el estado del arte de las Redes Eléctricas Inteligentes *(REI)*, de igual manera una visión general de la Infraestructura Avanzada de Medición *(AMI)*.
Posteriormente, se propone el modelado de la capa de datos que permitirá al MDMS interactuar con los *Smart Devices*, de igual manera se presenta el modelado e implementan de un *Data Werehouse* encargado de almacenar los datos obtenidos por los medidores mediante archivos XML por supuesto para obtener dicha información debe existir todo un proceso de Extracción Transformación y Carga *(ETL)*.

El prototipo del MDMS esta desarrollado con diversas tecnologías, por ejemplo para la capa de datos se utilizo OracleDB® como gestor de base de datos, de esta manera el motor de base de datos almacena información esencial para el MDMS como: información de seguridad la cual permite el acceso a los usuarios encargados de administrar el sistemas; así como información que permite interactuar con los *Smart Devices*, de igual manera el *Data Werehouse* utiliza OracleDB® como mecanismo de almacenamiento de datos.

Para que sea posible la comunicación entre los diversos sistemas que conforman la infraestructura AMI se creo una Red Mesh encargada de conectar todos los *Smart Devices* mediante radiofrecuencia de 800Mhz, por consecuencia los *Smart Devices* se organizan en grupos de 12 o 24 dispositivos, cuando la información es recolectada en este grupo ya se encuentra lista para ser enviada, por otra parte existe otro sistema informático intermedio entre los *Smart Devices* y el MDMS llamado *Head-End System (HES)*, el cual es el encargado de realizar las peticiones para recolectar la información, así mismo decodificarla ya que los medidores almacenan y transmiten la información bajo la norma ANSI C12.18 [29] y ANSI C12.19 [30] para finalmente entregar la información al MDMS mediante archivos XML, así mismo el MDMS procesa y almacena la información, en este punto la información se encuentra lista para ser entregada a sistemas de terceros encargados de hacer inteligencia de negocios, facturación, estadísticas, etc.

Esta tesis describe detalladamente la estructura interna del MDMS así como las tecnologías implementadas y su justificación, también se describen algunos algoritmos que permiten realizar, optimizar y mejorar el rendimiento del MDMS. También, se describe el funcionamiento a grandes rasgos de los sistemas que interactúan con el MDMS y como se realiza la comunicación entre los sistemas.

La pruebas se realizaron como un sistema MDM prototipo utilizando una cantidad limitada de *Smart Devices*, de igual manera se utilizo un prototipo de un sistema HES con funcionalidades limitadas, lo cual permitió realizar una demostración de la funcionalidad y potencial de implementar un software especializado en las redes electricas del país, como consecuencia de dicho prototipo se realizaron exitosamente conexiones y desconexiones remotamente, lecturas de consumo remotas en tiempo real, así como obtener datos estadísticos de los medidores que permitió tener un historial de consumos y demandas máximas, por ultimo, también se pudo obtener una serie de eventos desencadenados en los medidores inteligentes que nos permitirán dar aviso de una violación de seguridad o una caída de energía, etc.

Palabras claves— Redes Eléctricas Inteligente, Infraestructura Avanzada de Medición, Sistema de Gestión de Datos de Medidores, Red Mesh, *Head-End System*, *Data Werehouse*.

Índice general

Índice de tablas

Índice de figuras

Capítulo 1

Introducción

1.1. Descripción del problema

Con el paso del tiempo surgen nuevas tecnologías tanto productos de software como de hardware las cuales proponen soluciones innovadoras a problemas comunes y no tan comunes encontrados en la vida real, de igual modo pueden aportar mejoras a las tecnologías que surgieron anteriormente añadiendo nuevas mejoras, nuevas características, una mayor eficiencia, una mayor rentabilidad, etc. A causa de lo anterior, los medidores de energía eléctrica se han vuelto obsoletos al menos en el país de México, por esa razón, implementando sistemas modernos tales como las redes *Advanced Metering Infrastructure*–(Infraestructura Avanzada de Medición) (AMI) se incrementaría la eficiencia y se reducirían los costos de operabilidad de las empresas proveedoras del servicio eléctrico, por lo tanto, existen empresas dedicadas a desarrollar *Smart Devices* para el sector energético, dichos dispositivos son desarrollados con la finalidad de administrar las redes eléctricas de una manera más eficiente, dichos dispositivos son capaces de almacenar información de diferente interés, en consecuencia surge como reto desarrollar sistemas informáticos capaces de recuperar dicha información, decodificarla, procesarla y almacenarla para posteriormente ser utilizada a conveniencia. Es por eso que se deben crear software que permita dar soporte a un gran número de dispositivos sin reducir el rendimiento, además, permitir almacenar una infinidad de registros, de igual manera dicho software debe permitir una eficiencia y disponibilidad del 99.9 %. Algunas de la características de dicho software deben ser:

- Conexión y desconexión de servicios brindados remotamente en tiempo real.
- Registros de alarmas y/o eventos desencadenados en un dispositivo.
- Lecturas de consumo del servicio en tiempo real.
- Gráficas estadísticas sobre consumos.
- Visualizar demandas máximas y mínimas de los servicios.
- Entre otras.

Los resultados solicitados al software deben ser en el menor tiempo posible y con una efectividad del 99.9 %.

1.2. Hipótesis

Utilizar un *Data Warehouse* estructurado en *Data Marts* en un *MDMS* y alimentándolo mediante procesos ETL y VEE se puede mejorar y automatizar la eficiencia de las redes eléctricas del país.

1.3. Objetivo general

Desarrollar un modelo e implementar un software *(MDMS)* que forme parte de la Infraestructura Avanzada de Medición *(AMI)*. Dicho software permitirá administrar dispositivos inteligentes de medición remotamente, con una alta eficiencia y con un alto grado de escalabilidad integrando una gran cantidad de dispositivos a la infraestructura, sin reducir el desempeño.

1.4. Objetivos específicos

- Diseñar e implementar la capa de datos del *Meter Data Management System*–(Sistema de Gestión de datos de medidores) (MDMS), esta fase incluirá una base de datos de seguridad, una base de datos operacional y un *Data Warehouse* estructurado en *Data Marts*.
- Diseñar e implementar algoritmos que permitan la depuración de las bases de datos y del *Data Warehouse* que conformarán el MDMS, dichos algoritmos buscarán información obsoleta, duplicada o innecesaria que consuman recursos y espacio de almacenamiento innecesario.
- Diseñar e implementar un canal de comunicación entre los Sistemas *Head-End System* (HES) y MDMS, el cual permitirá la transferencia bidireccional de información entre ambos sistemas.
- Diseñar e implementar un canal de comunicación entre los sistemas de la compañía proveedora del servicio eléctrico y el MDMS, con el fin de proporcionar la información recolectada y procesada por el MDMS de los dispositivos.
- Diseñar un motor de cálculo que permita resumir y realizar cálculos, estimaciones y estadísticas sobre los datos contenidos en el *Data Warehouse*, con el fin de mejorar el rendimiento y el tiempo de respuesta cuando se realice una petición, la cual involucre datos contenidos en el *Data Warehouse* del MDMS.

1.5. Alcance

El alcance de este trabajo esta limitado al estudio, modelado e implementación de la capa de datos que permitirá almacenar los datos de los *Smart Devices*, implementación y desarrollo de algoritmos que permitan la persistencia de datos haciendo uso de técnicas y procesos tales como *Extract Transform and Load*–(Extracción Transformación y Carga) (ETL) y *Validation Estimation and Edition*–(Validación Estimación y Edición) (VEE), de igual manera algoritmos que permitan depurar la base de datos cada X intervalo de tiempo. La capa de datos será desarrollada para soportar dispositivos de diferentes tipos, proveedores y fabricantes siempre y cuando dichos dispositivos cumplan con las normas que permitan estructurar la información adecuadamente dentro de los mecanismos de almacenamiento del MDMS.

1.6. Estructura de la tesis

El proyecto de grado se encuentra estructurado en seis capítulos: el primer capítulo describe la problemática, hipótesis, objetivos y alcances del proyecto. El capítulo dos muestra un panorama general sobre las *Redes Eléctricas Inteligentes*–(Smart Grid) (REI) y la Infraestructura Avanzada de Medición (AMI) como una mejora y alternativa a los dispositivos de medición actuales, así mismo, se citan algunos de los trabajos más sobresalientes relacionados con esta trabajo de grado, con el objetivo de familiarizar al lector con esta área. El capítulo tres describe la teoría que fundamenta el trabajo propuesto en Redes Eléctricas de Medición, Sistemas de Medición Inteligentes, Redes Mesh, Sistemas Head-End, Sistemas de Gestión de Datos de Medidores, *Data Warehouse* y *Data Marts*. El capítulo cuatro describe la metodología empleada para diseñar la capa de datos de un MDMS, la cual será la encargada de gestionar los datos provenientes de los dispositivos inteligentes, de igual modo definirá la estructura de un *Data Warehouse* estructurado en *Data Marts*, los cuales serán los encargados de almacenar y segmentar la información generada por los dispositivos de medición inteligentes, con el fin de obtener resultados más certeros y en el tiempo más corto posible. El capítulo cinco, evalúa los resultados obtenidos para determinar si el MDMS es capaz de agilizar la recolección de datos, con la finalidad de proveer una ayuda para la facturación, el análisis de datos estadísticos y estudios de mercados. Por ultimo, el capítulo seis presenta las conclusiones obtenidas en base al Sistema MDMS, como una alternativa a las técnicas actuales de recolección de datos de medidores de las redes de energía eléctrica.

Capítulo 2

Estado del arte

2.1. Introducción

Muchas compañías están enfocadas a desarrollar una Infraestructura Avanzada de Medición también conocida por sus siglas en inglés AMI, dicha infraestructura hace uso de múltiples tecnologías, las cuales trabajando en conjunto aportan un gran beneficio a los sistemas de medición, haciéndolos más efectivos, más rentables y mucho más rápidos; algunas de las ramas indispensables para que la infraestructura AMI sea una realidad son: la electrónica, la informática, las telecomunicaciones, entre otras que destacan entre las más importantes. La característica de los sistemas AMI es que permiten una comunicación bidireccional[22, Página 38], la cual permite enviar y recibir paquetes mediante un canal de comunicación, dichos paquetes contiene instrucciones que pueden ser procesadas y ejecutadas por los dispositivos inteligentes y/o los servidores centrales, los cuales son los encargados de almacenar y procesar la información proveniente de los sistemas de medición inteligentes. Si comparamos la tecnología AMI con su antecesora *Automated Meter Reading*–(Lectura de Medición Automática) (ARM), AMI es más sustentable y más rentable, ya que recaba más información de interés para los usuarios finales y para los administradores, los cuales son los encargados de monitorear dicha infraestructura, por el otro lado, la tecnología ARM realiza efectivamente su trabajo, el cual es la recolección de datos para facturación, pero con la desventaja de no tener la capacidad de recolectar información secundaria que permita realizar cálculos estadísticos o incluso estudios de mercado.

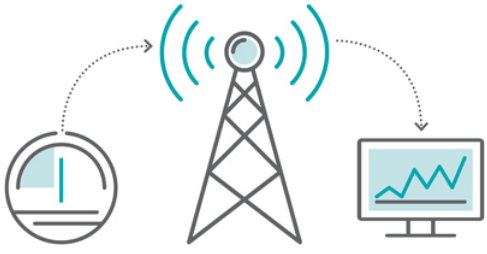


Figura 2.1.1: Comunicación unidireccional (Sara Samovalov, 2016).

2.2. Redes Eléctricas Inteligentes (REI)

Con cada día que transcurre vemos llegar el futuro, que hasta hace algunos años era parte de la ciencia ficción, una tecnología muy esperada y no muy alejada de la realidad es la revolución de la electricidad, imaginar un mundo donde seamos libres de decidir a quien comprar la energía eléctrica e incluso generar nuestra propia energía y poder venderla, utilizarla y administrarla de manera más inteligente y eficiente, la cual utilizamos diariamente, entre otras cosas, eso sin duda es el futuro.

Desde hace décadas la energía ha sido generada en grandes centrales termoeléctricas y/o hidroeléctricas, desafortunadamente las energías no renovables son las más utilizadas para proveer al mundo de energía. Una vez que las grandes centrales generan la electricidad esta es transmitida por grandes distancia y es adecuada para poder distribuirla entre la población, este proceso ha generado grandes perdidas de electricidad que involucran fallas técnicas o no técnicas *(robo de energía)*, a eso hay que sumarle el deficiente servicio ante una falla ya que el proveedor puede tardar tiempo en detectar, reportar y/o reparar las fallas, por ultimo, la utilización de energías no renovables ha contribuido a la generación de gases del efecto invernadero.

Intentando atacar esta problemática surgieron términos como REI también conocidas por su termino en inglés como *Smart Grid*, según la investigación realizada por [13], se hace referencia a un sistema integrado por diferentes elementos del sistema eléctrico tradicionales tales como: *generación, transmisión, distribución y comercialización de la energía eléctrica* más un sistema de comunicaciones.

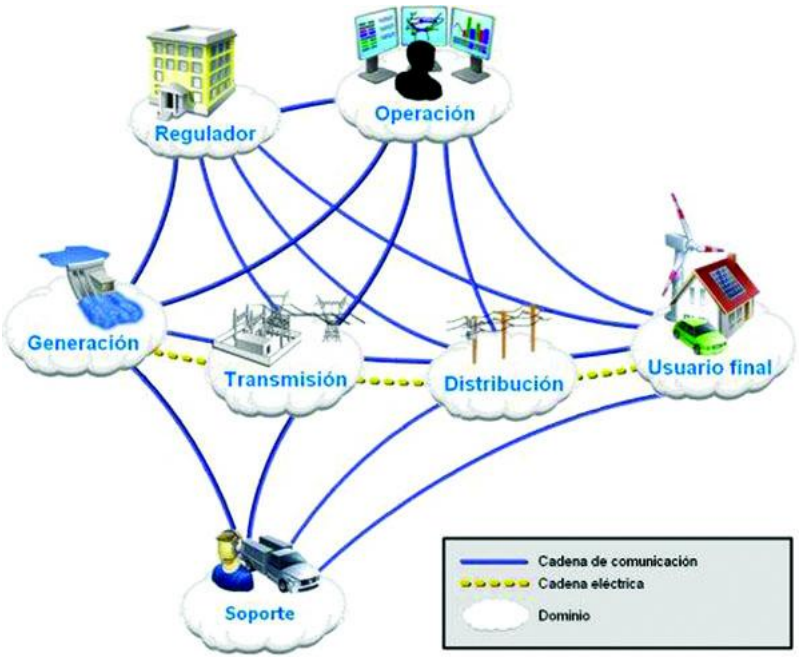


Figura 2.2.1: Redes Eléctricas Inteligentes (REI) (Ing. Juan C. Tripaldi, 2013).

2.2.1. Infraestructura de una REI

De acuerdo con el Programa de Redes Inteligentes de la Secretaría de Energía, emitido en 2016, *una REI debe mejorar el sistema eléctrico nacional a través de ser eficiente, seguro, flexible, de calidad, confiable y sustentable. Ante todo, debe ser capaz de reestructurarse y de recopilar información para conocer cuáles fueron las fallas que se dieron en el sistema y solucionarlas*.

La infraestructura de una REI debe estar integrada por los *clientes* y un *proveedor* de servicios, quien supervisará los productos ofrecidos por terceros, como portales web que ofertan la electricidad a los clientes, la instalación y el mantenimiento. Otros sistemas que son parte de las REI son la *operación*, la cual es la encargada de gestionar el flujo de electricidad de los distintos dominios de la red y el *mercado* enfocado a coordinar a los que participan en el comercio de servicios energéticos dentro de la REI. Estos componentes son adicionales *(clientes, proveedor, operación y mercado)*, por lo tanto, en conjunto con la generación, transmisión y distribución conforman las REI

Estos elementos en conjunto le otorgarían a las REI mayores ventajas en cuanto a su funcionamiento, costo y eficiencia. Además de que se espera que puedan agregar el uso de energías renovables *(como la solar, la eólica o la mareográfica)* a su desarrollo.

Para conformar y consolidar una REI no basta con la infraestructura y la participación de las personas involucradas, también es necesario de software especializados y sistemas informáticos incorporados a la red eléctrica tradicional. Por ejemplo, sistemas informáticos que almacenen y analicen información geográfica y estadística, así como el estado de la infraestructura y su monitorización entre otras tareas.

Es necesario crear e implementar infraestructura y tecnología nueva que permita operar a las REI, tales como, dispositivos de medición inteligentes que recopilen información de diferente interés, sistemas de comunicación bidireccional que permitan la comunicación entre los dispositivos y los servidores de la empresa, sistemas de administración de datos encargados de recopilar, decodificar, almacenar y procesar la información recolectada.

Además de software adicionales, que permitan operar de manera más eficiente este tipo de infraestructura como por ejemplo, diseñar e implementar un sistema que indique la ubicación de cada trabajador, para saber qué actividad realiza y si pueden acudir a una emergencia; un sistema que administre las fallas de la red eléctrica, como lugar y magnitud, así como la automatización de la distribución de la energía, entre otros.

Toda la información que se recopile sería enviada a un centro de control para ser analizada, buscar fallas y ofrecer mejoras para el sistema.

Una de las desventajas aparentes y quizás la que genere un alto impacto si llega a ocurrir, es que dichas redes al ser interconectadas pueden ser interceptadas, saboteadas y/o alteradas de forma maliciosa por hackers, como consecuencia causar un gran daño que va desde el robo de información hasta un apagón masivo.

Por el contrario, entre los principales beneficios que aportarían este tipo de REI es la integración de las energías renovables, disminuyendo gradualmente la contaminación del medio ambiente, lograr precios competitivos en materia de electricidad debido a la *oferta-demanda*, mejorar la calidad y confiabilidad del servicio, autoreparación de fallas del sistema eléctrico *(si le es posible)*, automatización de mantenimiento y reparación del mismo entre otros.

Una REI podría detectar y atender las fallas antes de que representen un problema para el usuario, es decir, después de segundos que empezó, realizar diagnósticos y mediciones del sistema. Esto se podría llevar a cabo mediante la tecnología implementada en este tipo de redes, ya que al ocurrir una falla el dispositivo podría mandar una alarma al centro de operación indicando el inconveniente, así mismo, el dispositivo trataría de autorepararse, este concepto es conocido como ***resiliencia***, es decir, que el sistema se repare solo, si no le es posible se tomarían medidas secundarias, tales como la visita de un técnico especializado para reparar la falla a la brevedad posible.

La capacidad de que el sistema se repare solo permitirá que el usuario no se percate de que hubo un problema en el transformador. En caso de que éste falle, la red tiene que estar configurada para que los usuarios que estaban conectados se puedan conectar a otro dispositivo, de manera instantánea.

2.3. Infraestructura Avanzada de Medición (AMI)

Los sistemas AMI están compuestos por cinco elementos básicos:

- Dispositivos inteligentes de medición.
- Concentrador de datos.
- Redes de comunicación.
- Sistema de gestión de información.
- Aplicaciones.

Las empresas proveedoras del servicio eléctrico han disminuido su efectividad en países en desarrollo, a causa de perdidas técnicas o no técnicas y/o fraudes eléctricos, ya que una gran parte de los usuarios comunes tiene al alcance dichos dispositivos de medición, por lo tanto, esto da lugar a que los usuarios hagan modificaciones mal intencionadas, con el fin de reducir el consumo eléctrico registrado por el dispositivo y así disminuir el costo en la facturación de su servicio, la infraestructura AMI es una herramienta preventiva contra el robo de energía, basados en un sistema avanzado antifraude, ya que al reducir el tamaño de los dispositivos y aumentando su capacidad para registrar eventos, estos dispositivos pueden ser concentrados en un gabinete de una configuración N*N haciéndolos más portables para poder ser instalados en postes, paredes y/o muretes, así mismo, el diseño de los dispositivos y del gabinete impide la alteración física o lógica de los dispositivos de medición, ya que cumple con estándares que permiten codificar y estructurar la información, atacando este tipo de problemas. Estos dispositivos tiene la capacidad de transmitir información[22, Página 46] hacia un concentrador mediante radiofrecuencia, Bluetooth, *Wireless Fidelity*–(Fidelidad Inalámbrica) (WI-FI), redes *Third Generation*–(Tercera Generación) (3G)-*Fourth Generation*–(Cuarta Generación) (4G), etc. *(esto depende de la tecnología que la empresa implemente)*, para que finalmente esta información sea transmitida a los servidores y sistemas que procesaran y almacenarán dicha información, estos sistemas permiten la recolección de datos, corte y reconexión de servicios remotamente, lo cual es traducido en una gran ventaja ya que reduce los costos de operación, trasporte y mantenimiento garantizando un mejor, más rápido y más eficiente servicio a los usuarios. Así mismo, los usuarios mediante un dispositivo compacto puede monitorear el consumo energético en tiempo real, otorgándoles una idea del consumo por el cual tendrá que pagar en su próxima facturación, este dispositivo normalmente utiliza una comunicación *Power Line Communications*–(Linea de Comunicación de Energía) (PLC), aunque puede ser implementado con otras tecnologías para la comunicación entre el dispositivo de medición y el dispositivo compacto de lectura. La instalación es fácil, flexible y de bajo costo permitiendo a la empresa proveedora del servicio eléctrico tener un mayor control sobre la energía entregada y recibida.

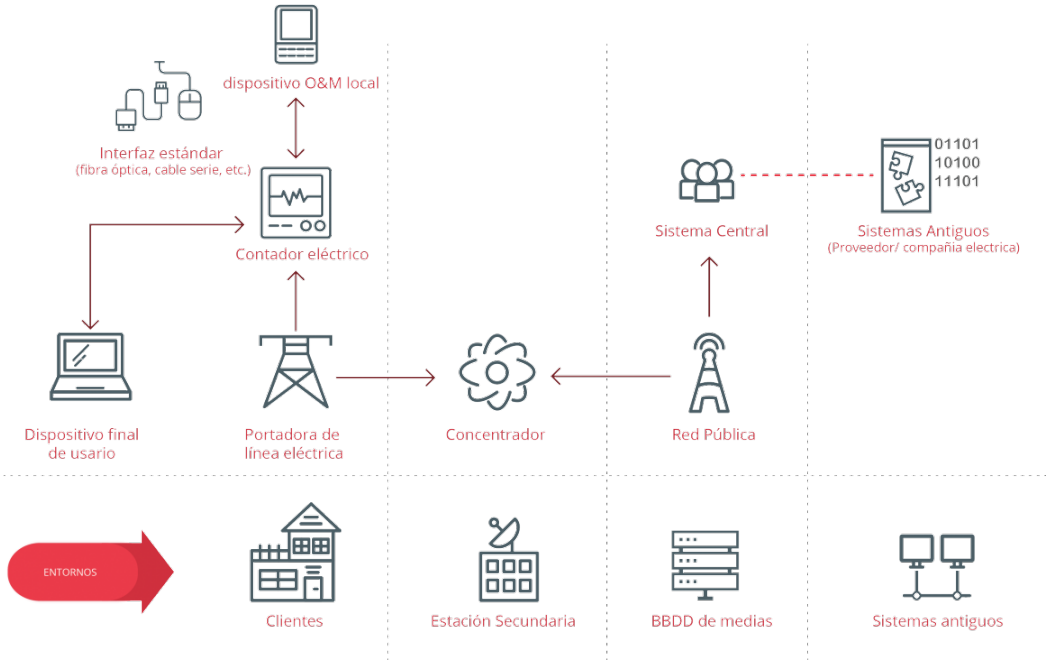


Figura 2.3.1: Componentes e interfaces de la Infraestructura AMI (INCIBE, 2017).

2.3.1. Dispositivos inteligentes de medición

Los dispositivos inteligentes son también conocidos como *Smart Devices* y son el primero de los cinco componentes fundamentales que forman la Infraestructura Avanzada de Medición (AMI), estos novedosos dispositivos que recopilan la información periódicamente sobre el consumo energético entregado a los usuarios, permitirá a los clientes monitorear el consumo en tiempo real de su servicio, así como datos más detallados sobre el consumo, permitiendo así hacer conciencia en los usuarios sobre el uso adecuado de la energía eléctrica. Comparado con los antiguos dispositivos de medición que la mayoría son análogos, los nuevos dispositivos llamados *Smart Devices* son dispositivos digitales, lo que permitirá transmitir la información hacia un concentrador mediante un canal de comunicación guiada o no guiada, permitiendo lecturas instantáneas, suspensión y reactivación del servicio en tiempo real y de una manera remota, por lo tanto, no será necesario de personal que ingrese al predio para realizar dichas operaciones, eliminando así las posibilidades de suplantación que utilizan algunos delincuentes. Dichos dispositivos garantizan transparencia, los medidores tienen la capacidad de ser minitoreados las 24 horas por un software, por lo cual, si es desencadenado un evento de violación de seguridad se enviara un inmediato aviso para dar a conocer dicho suceso.

Figura 2.3.2: *Smart Device* (Infoguerras, 2017).

2.3.2. Concentrador de datos

Un concentrador de datos es un dispositivo que permite centralizar las conexiones de distintos dispositivos *(Smart Devices)* para poder ampliarla o propagarla, esto quiere decir que recibe una o múltiples señales por un puerto especifico y repite o duplica dicha señal por un puerto diferente *(otro puerto diferente al de entrada)*. Esto es utilizado para interconectar múltiples dispositivos y recolectar o transmitir paquetes desde el destino hacia el dispositivo y/o del dispositivo al destino.

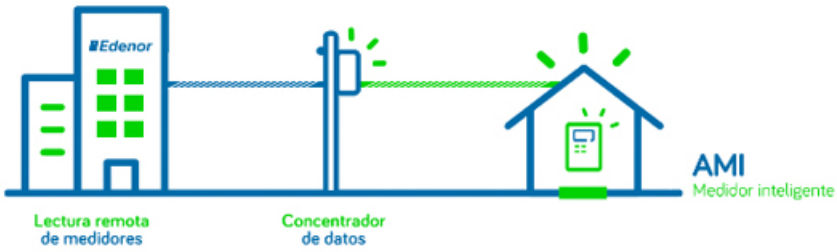


Figura 2.3.3: Concentrador de datos en la infraestructura AMI (EconoJournal, 2017).

2.3.3. Redes de comunicación

Una red de comunicaciones también conocida como *network*, es una conexión que permite conectar o interconenctar múltiples dispositivos (*Smart Devices*, concentradores de datos, etc.), además permitir que puedan comunicarse e intercambiar paquetes entre ellos, la comunicación puede realizarse de dos formas, usando sus propios recursos o usando recursos ajenos. Cuando los dispositivos se encuentran cercanos unos a otros estas redes suelen llamarse *Local Area Network*–(Red de Área Local) (LAN)[21, Página 51], por lo tanto, si se interconectan múltiples redes de área local separadas por una distancia considerable *(cientos o miles de kilómetros)*, estas redes suelen conocerse como *Wide Area Network*–(Red de Área Extensa) (WAN). Cada dispositivo suele conocerse como nodo, por lo tanto, una red de comunicaciones contiene una infinidad de nodos que permiten una comunicación entre ellos. Existen diversas topologías en redes[21, Página 55] de comunicaciones las más conocidas son:

- Topología en estrella.
- Topología mixta.
- Topología en anillo.
- Topología en Malla.
- Topología en árbol.
- Entre otras.

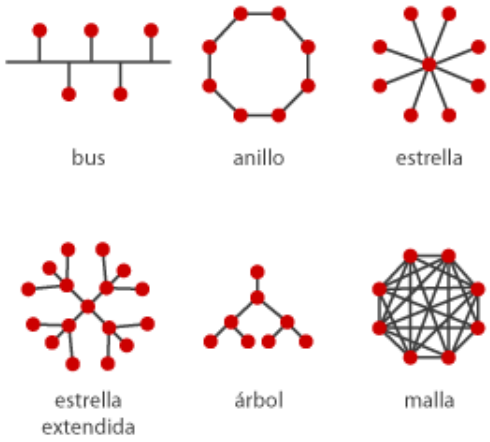


Figura 2.3.4: Topologías de redes de comunicación (CASIOPEA, 2011).

2.3.4. Sistemas de Gestión de Información

Un Sistema de Gestión de Información es un software que permite la administración de los dispositivos inteligentes, haciendo uso de las redes de comunicaciones y los concentradores de datos, proporcionando así la adquisición, recepción y transmisión de paquetes de información, dicho Sistema de Gestión de Información debe estar implementado dentro de la empresa que provee el servicio, formando así parte de la Infraestructura Avanzada de Medición (AMI). Este tipo de sistemas deben ser escalables, permitiendo agregar y administrar un gran número de dispositivos, por lo tanto, el sistema debe funcionar sin interrupciones y debe ser 99.9 % confiable, implementando mecanismos tales como: redundancia de información, suministro de energía interrumpido, respaldo de información, balanceo de carga, etc. para lograr dicho objetivo. Algunas características que automatizan y agilizan los Sistemas de Gestión de Información son los siguientes:

- Recopilación de datos mediante tareas programadas.
- Recopilación de datos en tiempo real.
- Automatización de mecanismos de recolección de datos.
- Evaluar la calidad de los datos generando estimaciones donde existan errores o lagunas de datos.
- Homologar y almacenar los datos provenientes de diversas fuentes.
- Realizar cálculos y estimaciones sobre los datos almacenados.
- Entre otros.

2.3.5. Aplicaciones

Cuando el Sistema de Gestión de Información ha recolectado y procesado la información satisfactoriamente deben existir aplicaciones que hagan uso de dicha información, esta información previamente procesada revelará datos importantes para los usuarios y para la empresa proveedora del servicio, por lo tanto, las aplicaciones deben permitir ***(tabla 2.3.1)***:

Tabla 2.3.1: Principales acciones de las aplicaciones AMI.

No.	Acción
1	Desconectar dispositivo.
2	Conectar dispositivo.
3	Alta de dispositivo.
4	Baja de dispositivo.
5	Modificación de dispositivo.
6	Toma de lecturas actual.
7	Visualizar estadísticas de consumos, demandas y lecturas.
8	Visualizar alarmas y eventos de los dispositivos.
9	Entre otros.

2.4. Aportaciones relevantes

Una de las investigaciones y aportaciones realizadas por ***Anner Jhoan Pellicer Navarro en 2016***[20], quien define los aspectos fundamentales de la infraestructura AMI aplicada al servicio de energía eléctrica, así como el desarrollo de un algoritmo que permite obtener formas de onda de tensión y de corriente aleatoria, permitiendo simular y representar el comportamiento de múltiples usuarios de consumo de energía eléctrica. Al poder simular el comportamiento de los usuarios, ***Pellicer***[20] desarrollo un prototipo de medidor que realiza funciones similares a los dispositivos de medición convencionales, además, agrego otras características que permiten mejorar las capacidades de dichos dispositivos de medición, tales como: lectura periódica, corte y reconexión, sistemas de cobro prepago y postpago, así mismo, desarrolló un algoritmo que toma las salidas del dispositivo prototipo de medición y las organiza en una base de datos.

Pellicer[20], enfocó su interés en desarrollar un modelo y prototipo de medidor computacional, para evaluar las funciones y resultados del dispositivo para posteriormente poder reproducirlo en masa. Su modelo contempla parámetros de entrada así como resultados de salida provenientes del medidor inteligente de medición, una vez que son proporcionados los parámetros de entrada, los valores de salida pueden ser analizados, ***Pellicer***[20] almaceno los datos en una base de datos en Microsoft Access®, y realizó simulaciones en una *Plataforma de Simulación Multidominio* (SIMULINK)® por lo cual, obtuvo resultados satisfactorios, dejando ver que la infraestructura AMI es viable y puede ser implementada en zonas especificas de su país como programa piloto.

Otra de las aportaciones importantes es la realizada por ***Ing. Abraham Valdiosera Marroquin en 2013***[17], quien observando la tendencia de la tecnología y el surgimiento de nuevos términos, tales como la infraestructura AMI, realizó la propuesta de un sistema de medición con comunicación bidireccional mediante un canal de comunicación no guiado *(comunicación inalámbrica)*, apegándose a los esquemas de la infraestructura AMI, proponiendo un sistema de medición que consta de un prototipo de medidor eléctrico inteligente, diversos sensores, un concentrador de datos y una interfaz web que permita la administración y gestión del sistema.

El prototipo de medidor eléctrico propuesto por ***Valdiosera***[17], utiliza un microcontrolador Freescale®, el cual, permite traducir las señales análogas a digitales, calcular algunas variables eléctricas y controlar todos los procesos que se ejecutan en el medidor, el dispositivo de medición también cuenta con un *Global Positioning System*–(Sistema de Posicionamiento Global) (GPS), integrado que le permite realizar mediciones fasoriales sincronizadas, y mediante un módulo de radiofrecuencia, le permite enlazarse a redes inalámbricas que operen bajo el protocolo ZigBee.

El concentrador de datos que forma parte de la infraestructura de ***Valdiosera***[17], es un Gateway ZigBee-Ethernet, este dispositivo es el encargado de crear y coordinar una red de radio local, formado por los prototipos de medidores inteligentes y otros sensores adicionales con tecnología ZigBee, que le permite estar comunicados inalámbricamente. Simultáneamente, el concentrador de datos enlaza dicha red local a internet a través de un puerto Ethernet de alta velocidad, gracias a esta característica, es posible realizar una comunicación bidireccional con los dispositivos inteligentes de medición, conectados a una red local por medio del concentrador de datos, la consulta se puede realizar mediante dispositivos conectados a internet *(SmartPhone, Computadora Personal, Servidor, etc.)*, haciendo uso de la interfaz web.

Valdiosera[17], describe detalladamente el diseño y codificación en el lenguaje de programación C, así como los algoritmos empleados en el medidor inteligente prototipo, así como cada uno de los componentes de hardware que lo componen, tales como el GPS y el módulo de radiofrecuencia que le permite una comunicación bidireccional, así mismo detalla la configuración y programación del concentrador de datos, el cual hace posible la creación de redes locales y la comunicación a internet, además detalla el funcionamiento de la interfaz web que permite la interacción con los medidores inteligentes y los usuarios finales.

Valdiosera[17], obtuvo resultados satisfactorios, al realizar pruebas conectando diversos prototipos de medidores inteligentes y los diversos dispositivos que implementó, tales como el concentrador de datos para el funcionamiento de su infraestructura, así mismo utilizó una computadora personal para recibir los datos provenientes de los medidores inteligentes, lo cual le permitió comparar los valores de las lecturas obtenidas mediante el concentrador y los reales registrados por el medidor directamente, arrojando una exactitud de +/-0.2 % de exactitud para valores de voltaje y corriente.

Julio Yalan Castillo y Luis Palomino Paniora[1], en uno de sus artículos titulado "*Implementación de un Datamart como una solución de Inteligencia de Negocios*", publicado en el año 2013, observando el crecimiento de la información dentro de una organización, presenciaron el incremento a gran escala y la capacidad de generar información y su almacenamiento, observaron que los volúmenes de información han crecido tanto que su análisis no puede ser realizado con los métodos tradicionales, ya que mientras mayor es la capacidad para almacenar información mayor es la incapacidad para extraer datos realmente útiles, ya que muchas veces la información importante para la toma de decisiones queda oculta o no es tomada en cuenta, al no poder ser extraída en el momento requerido, además, sumado a esto, los sistemas transaccionales que son comúnmente utilizados en las organizaciones no son los adecuados para realizar *Business Intelligence*–(Inteligencia de negocios) (BI).

Yalan y Palomino[1], orientan su trabajo a implementar un *Data Mart*, como una herramienta que permitirá desarrollar BI dentro de una organización, además, plantean la simplificación de procesos ETL, para obtener información y brindar soporte para la toma de decisiones. Para el desarrollo del DataMart como su solución de Inteligencia de Negocio, hacen uso de la guía *Business Intelligence RoadMap* elaborada por ***Larissa T. Moss*** y ***Shaku Atre***[19], y la guía *Metodologías de Soluciones Cognos y Cognos Business Intelligence Roadmap[3]*, ya que brinda los paso a seguir para el ciclo de vida de un proyecto de BI, los pasos que emplean descritos en la guía *Business Intelligence RoadMap*, pueden resumirse de la siguiente manera:

- Evaluación del negocio.
- Definición de requerimientos.
- Análisis de datos.
- Prototipo de aplicación.
- Diseño de la base de datos.
- Diseño del proceso ETL.
- Desarrollo del proceso ETL.
- Desarrollo de la aplicación.
- Certificación.
- Implementación.

Cada uno de los pasos anteriores esta resumido, por lo tanto, ***Yalan y Palomino***[1], siguieron una serie de pasos adicionales por cada uno de los pasos descritos en la guía que utilizaron como referencia, de tal manera que mediante procesos ETL aplicados a información histórica obtuvieron de forma automática un repositorio *(DataMart)*, que servirá como una herramienta que les permitirá la explotación de la información de una manera más eficiente y fiable. El DataMart sirvió como soporte en una área especifica de la organización para la toma de decisiones, a través de la entrega relevante y oportuna de la información.

Alexander Bustamante Martínez, Ernesto Amaru Galvis Lista y Luis Carlos Gómez Flórez[18], alumnos de la *Universidad Industrial de Santander*, publicaron un artículo en el año 2013, titulado "*ETL Processes modeling techniques: an alternatives review and its application in a BI solution development project*". En su artículo, describen que existe una serie de paso que un diseñador de procesos ETL debe seguir, entre los cuales se encuentran: analizar las fuentes de datos existentes para encontrar la semántica oculta en ella, diseñar un flujo de trabajo que extraiga los datos desde las fuentes, repare sus inconsistencias, los transforme en un formato deseado y útil y finalmente los almacene en un mecanismo especifico de almacenamiento.
Como se puede observar, diseñar e implementar un proceso *ETL* no es una tarea sencilla, con el propósito de facilitar dicha tarea se han desarrollado diferentes técnicas, de las cuales sobre sales dos las cuales son: las que son inspirada en los diagramas de flujo y de procesos, y las técnicas que están inspiradas en el paradigma de *Object-Oriented Programming*–(Programación Orientada a Objetos) (POO) y los diagramas *Unified Modeling Language*–(Lenguaje Unificado de Modelado) (UML).

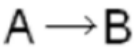
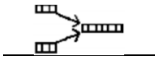
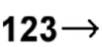
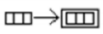
Bustamante, Galvis y Gómez[18], describen la técnica "*enfoque orientado al modelado de procesos*", los autores cubren dos tipos de modelados: *conceptual y el lógico*. Para el modelado conceptual, plantean identificar las fuentes, los atributos y las transformaciones a realizar, además, proponen una serie de iconos para modelar y representar este tipo de modelo. Respecto al modelado lógico y tomando en cuenta que este involucra el flujo de datos necesario para garantizar que el *Data Warehouse*–(Almacén de Datos) (DW) sea poblado con éxito, los autores plantean considerar otros elementos, tales como el conjunto de valores admitido por un atributo y los parámetros necesarios para que las funciones ejecuten su trabajo. Adicionalmente, en esta propuesta también se expone un método, para mapear el diseño conceptual al lógico, ilustrando como cada elemento del modelado conceptual puede representarse en elementos lógicos, minimizando la complejidad y la perdida de información.

Tabla 2.4.1: Enfoque orientado al modelado de procesos (Bustamante, Galvis y Gómez, 2013).

Elemento	Descripción
Concepto	Representa una entidad en la base de datos de origen, o en la bodega de datos.
Atributo	Es el nodo más granular de información.
Transformación	Son abstracciones que representan partes, rutinas completas ejecutan dos tareas esencialmente: (a) filtrar, y (b) transformar datos.
Restricción ETL	Es utilizada para indicar que los datos deben cumplir un conjunto de requerimientos.
Nota	Son usadas para capturas comentarios extras que los diseñadores quieren realizar durante la fase de diseño. Además de explicar la semántica de las funciones.
Relación-Parte	Un concepto está compuesto por un conjunto de atributos.
Proveedor 1:1 N:M	Mapea un conjunto de atributos de entrada, a un conjunto de atributo de salida a través de una transformación relevante.
Composición Serial	Se usa cuando necesitamos combinar varias relaciones en un único proveedor.
Candidatos	Captura el concepto de que un Concepto de la bodega de datos, puede ser poblado por más que un Concepto de la fuente de datos.

Los autores describen la segunda técnica "*enfoque orientado a objetos*", como una alternativa haciendo uso de diagramas de clases UML, los autores exponen que al hacer uso de diagramas de clases UML, no solo se logra cubrir la especificación de conceptos clases, atributos y funciones, sino que al ser UML reconocido como una buena practica para el desarrollo de software, logra acortar la curva de aprendizaje en la asimilación de una nueva tecnología. Esta técnica hace uso de estereotipos para cualificar el comportamiento general de la clase.

Tabla 2.4.2: Descripción de los estereotipos (Bustamante, Galvis y Gómez, 2013).

Estereotipo	descripción	Icono
Agregación	Realizar agregaciones, con los datos, basados en algún criterio.	
Conversión	Cambiar sea: el tipo de dato, el formato u obtener un nuevo dato, derivado de otro existente.	
Filtro	Filtrar los dados por algún criterio.	
Incorrecto	Marcar y re-direccionar datos incorrectos para la operación.	
Join	Unir dos conjuntos de datos, tomando como referencia algún(os) atributos.	
Cargador	Insertar los datos en la estructura de datos de destino.	
Log	Registrar información sobre el proceso.	
Combinación	Integrar dos o más conjuntos de datos, estos deben tener atributos compatibles.	
Delegados	Genera llaves delegadas únicas,	
Wrapper	Transforma una fuente de datos, en un conjunto de datos en memoria, similar a la fuente.	
Tabla	Representa una tabla de la base de datos, se de origen o de destino.	
Nota	Permite realizar anotaciones que ayudan a clarificar las operaciones realizadas.	

Bustamante, Galvis y Gómez[18], concluyen que no importa el producto de software que se construya, el diseño es una actividad esencial ya que el mismo reduce la incertidumbre inherente a la implementación y propicia su comprensión, extensibilidad y mantenibilidad. Por lo tanto, los autores concluyen que el diseño de los procesos ETL es crucial al desarrollar una solución de BI

Los procesos ETL contiene entradas, operaciones y salidas que consumen recursos, por lo tanto, realizar un diseño previo utilizando alguna de las técnicas mencionadas anteriormente para el modelado, puede considerarse adecuado, por lo cual, una notación útil para realizar el modelado de los procesos dejaría ver características mediante el diagrama de clases UML.

En un artículo[10] titulado "*Importancia de un Master Data Management impulsando tu Data Warehouse*", publicado por el sitio ***powerdata***, en el año 2017, en dicho artículo se hacen referencia a la importancia de los sistemas especializados en hacer BI, relatando que dichos sistemas son utilizados principalmente para mejorar la calidad en la toma de decisiones de una empresa. Sin embargo, mencionan que si los datos persistidos en el *Data Warehouse* no tiene calidad, los datos extraídos mediante herramientas para realizar BI solo perjudicaran a la empresa, ya que que si los informes y reportes extraídos del *Data Warehouse* no son inteligentes, el trabajo de todos los usuarios se vera relentizado y las decisiones pueden ser cuestionadas.

El autor [10] del artículo sugiere, "*para que un Data Warehouse puede asegurar su inteligencia, es necesario que reciba el apoyo de un Master Data Management*", de esta manera la empresa podrá desarrollar todo su potencial.

El autor [10] describe dos enfoques para la *Gestión de Datos Maestros* ***(MDM)***: *MDM operativo* y *MDM analítico*: El funcionamiento del *MDM operativo* es verificar la duplicidad de los datos en los diferentes sistemas operativos, mientras que el *MDM analítico* es asociado generalmente con el almacenamiento de los datos, y adaptándolo a las necesidades de la organización pretende mejorar la velocidad y la calidad de sus procesos de generación de informes de BI.

Existe una estrecha relación entre el MDM y el *Data Warehousing*, ya que las dimensiones de un *Data Warehouse* son esencialmente datos maestros. Pero a menudo estas dos áreas importantes tienden a ser tratadas como áreas separadas entre si.

El autor [10] nos muestra una idea de como combinar un MDM y un *Business Intellegence*, comienza con señalar que los datos maestros se representan como dimensiones en los sistemas BI, pero dichos datos maestros no están asociados con los hechos *(es decir, transacciones)* en los sistemas BI. La introducción de un sistema MDM dentro de una empresa debería tener un impacto positivo en los sistemas de BI. El autor [10], nos pone como ejemplo: "*es típico en un sistema MDM que los nombres de datos de atributos y las definiciones de datos utilizadas para describir entidades de datos maestros sean los nombres de datos estándar y las definiciones de datos para la empresa*". Por lo general, estas definiciones de datos maestros se denominan *Shared Business Vocabulary*–(Vocabulario Empresarial Compartido) (SBV) para la empresa, por lo tanto, el SBV es un metadata maestro.

Podemos aprovechar un SBV de datos maestros en un sistema de BI para reforzar la reutilización de las mismas definiciones de datos en todos los modelos dimensionales, cubos y vistas de negocio de herramienta de BI, con el fin de impulsar la consistencia entre datos dimensionales. De esta manera, la adopción de un SBV de datos maestros mejora la comprensión de los datos presentados en los informes del sistema BI, los análisis *On-Line Analytical Processing*–(Procesamiento Analítico en Linea) (OLAP), los paneles de control y los *scorecards*.

Además de los metadatos consistentes, la integración de un sistema MDM en la organización puede afectar a la integración de datos en el DW de un sistema de BI. Al no disponer de un sistema MDM, los sistemas de BI se basan en una arquitectura clásica de DW, en esta arquitectura clásica los datos se encuentran divididos entre múltiples almacenes de datos en diferentes líneas de sistemas operacionales de negocio. Por lo tanto, para crear datos dimensionales integrados en un sistema de BI, a menudo, se utilizan herramientas que permiten la integración de datos, integrando datos maestros dispersos mantenidos en diversos sistemas operacionales para construir dimensiones.

Por lo tanto, a menudo, es confundido el concepto de un centro de datos maestro y un DW cuando ambos están integrando datos, por lo tanto, surgen preguntas equivocadas como: ¿por qué necesitamos un sistema MDM cuando ya tenemos un DW? pero esta pregunta esta equivocada ya la pregunta adecuada debería ser: ¿por qué estamos haciendo integración de datos maestros en un sistema de BI? el autor [10], nos da la respuesta a esta pregunta: *los datos maestros deben de estar integrados y deben ser tratados como una fuente de datos por las herramientas de integración de datos utilizadas en un sistema BI*. Por lo tanto, esto resulta una mejor opción, porque los datos maestros tiene que ser suministrados a algo más que un sistema BI, estos deben suministrarse a los sistemas operacionales y de BI. El autor [10], indica tres maneras en que los datos maestros se pueden suministrar a las herramientas de integración de datos del sistema BI:

- ***Mediante el uso de servicios de integración y emparejamiento de un software de calidad de datos***: habilitado para SOA, para suministrar datos maestros directamente en un DW o para procesos de ETL que alimentan el DW.
- ***Mediante el uso de una solución MDM para crear una fuente de datos virtual de datos maestros***: a la que puede acceder una herramienta de integración de datos del sistema de BI.
- ***Utilizando un hub de datos MDM***: construido o comprado como una fuente de datos persistente para una herramienta de integración de datos del sistema de BI.

El autor [10], concluye que "*la gestión de datos maestros fortalece los sistemas DW/BI*" y esto se logra de la siguientes maneras:

- Proporcionando metadatos maestros para su uso en modelos de datos dimensionales y cubos.
- Proporcionando datos maestros de alta calidad, como fuente de datos de confianza para el procesamiento ETL.
- Proporcionando vistas federadas de datos maestros, a través de sistemas dispares para la generación de informes
- Seguimiento de versiones de jerarquías a través del tiempo.
- Automatizando la recreación de diferentes versiones de una dimensión, en un esquema de cubo o estrella para reflejar cambios en jerarquías.
- Proporcionando datos confiables para la elaboración de informes y análisis.

Capítulo 3

Marco teórico

3.1. Modelo Cliente-Servidor

El modelo Cliente-Servidor ***(figura 3.1.1)***, según la investigación realizada por [33], es un modelo de servicio distribuido conectado por una red de computadoras, de tal modo que, un cliente realiza una petición a un servidor y el servidor proporciona la información requerida por el cliente mediante un protocolo *Transmission Control Protocol*–(Protocolo de Control de Transmisión) (TCP)/*Internet Protocol*–(Protocolo de Internet) (IP)[21, Página 69], la ventaja de utilizar la arquitectura Cliente-Servidor[33], es que todas las operaciones solicitadas por el cliente se realizan en el servidor, reduciendo así drásticamente la carga de tareas de lado del cliente.

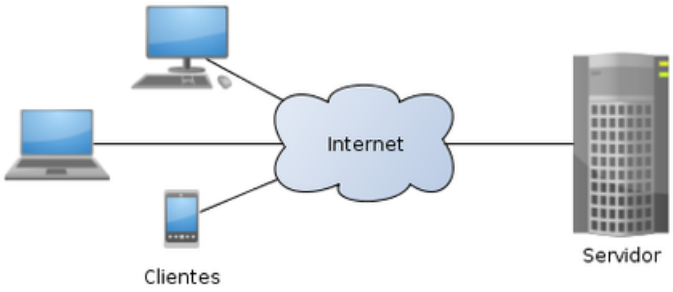


Figura 3.1.1: Arquitectura Cliente-Servidor (Arlen Alvarado, 2015).

3.2. Modelo Vista Controlador (MVC)

El *Model View Controller*–(Modelo Vista Controlador) (MVC), es un patrón de la arquitectura de software, según [25], este modelo busca facilitar el desarrollo de una aplicación mediante la separación de conceptos, la reutilización de código y por supuesto facilitar su mantenimiento, esto se logra separando los datos y la lógica de negocio en tres componentes esenciales,los cuales son: el *Modelo*, la *Vista* y el *Controlador*, estos componentes se encuentran divididos o fragmentados, en componentes que se encargarán de representar la información e interactuar con el usuario respectivamente.

Los componentes que forman parte de este patrón los podemos describir de una manera genérica.

- ***Modelo:*** El Modelo contiene la representación de la información con la cual el sistema operará, es decir, el modelo contendrá el código que involucre a los mecanismos de almacenamiento del sistema *(bases de datos)*, el modelo mantendrá encapsulada la complejidad de los mecanismos de almacenamiento, y simplemente existirán funciones que permitan manipular la información *(insertar, consultar, actualizar y/o borrar)*, otorgándole parámetros de entrada y la función retornará uno o más resultados de respuesta. Al mantener las operaciones de los mecanismos de almacenamiento mediante funciones, estas pueden ser invocadas desde otras partes del programa que necesiten del modelo, y este a su vez, se encargara de procesar dichas llamadas. Al construir las funciones del modelo hay que tener en cuenta el mecanismo de almacenamiento de destino, las entidades, los tipos de datos, relaciones, entre otras cosas, pero una vez que el modelo es construido, se pueden invocar dichas funciones sin preocuparse nuevamente de estos aspectos.

- ***Vista:*** La Vista mantiene la presentación final de una aplicación hacia el usuario, es decir, la capa vista contendrá el código de programación de cada una de las *Graphical User Interface*–(Interfaz Gráfica de Usuario) (GUI) tal cual como queremos que el usuario las visualice, usando uno o combinando distintos lenguajes de programación, que permita la codificación de la vista (*HyperText Markup Language*–(Lenguaje de Marcas de Hipertexto) (HTML), *Cascading Style Sheets*–(Hoja de Estilo en Cascada) (CSS), *JavaScript* (JS)), *JavaScript Object Notation*–(Notación de Objetos de JavaScript) (JSON), etc).
  Se debe tener en cuenta que en la practica, la capa vista de este modelo no solo involucra a las interfaces de una aplicación, sino que también la capa vista incluye la presentación de información en ficheros, tales como: *Portable Document Format*–(Formato de Documento Portable) (PDF), *eXtensible Markup Language*–(Lenguaje de Marcado Extensible) (XML), *Comma-Separated Values*–(Valores Separados por Comas) (CSV), etc.

- ***Controlador:*** El Controlador es una de las capas más importantes del MVC, ya que realiza el enlace entre el modelo y la vista, además de gestionar los recursos necesarios para procesar información en el servidor. En resumen, la capa del controlador es la encargada de la lógica de negocio de nuestra aplicación ayudándose de las capas vista y modelo para generar, procesar y representar información solicitada.

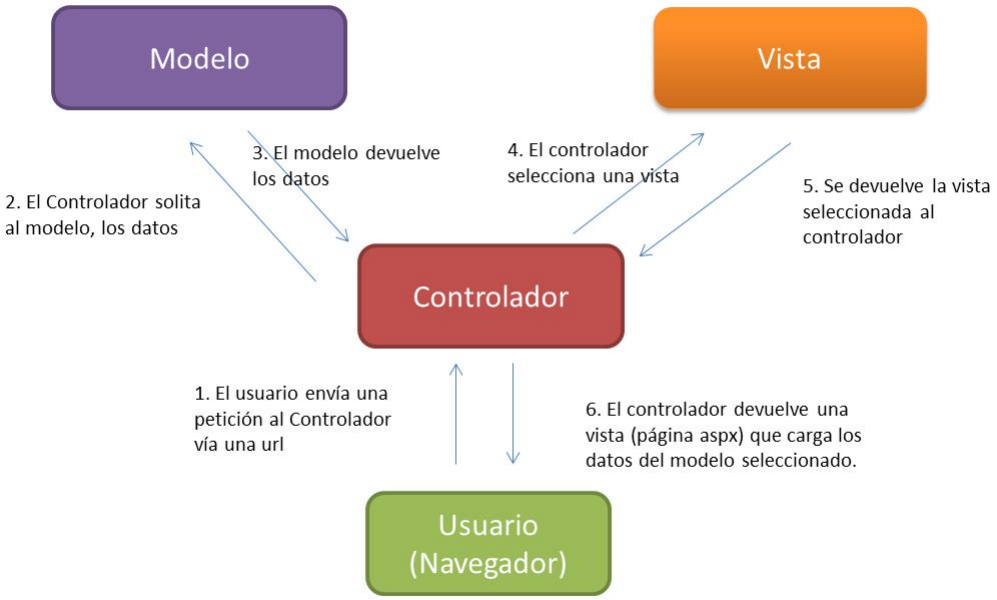


Figura 3.2.1: Arquitectura del Modelo Vista Controlador (QualityDev, 2015).

3.3. Base de datos

Una base de datos, a menudo también conocida como un banco de datos, no es más que una forma digital de almacenar información de diverso tipo de un mismo contexto; los servidores de base de datos trabajan bajo la arquitectura cliente-servidor[33][27, Página 445].

Un servidor de base de datos, es un software que da servicio de almacenamiento y gestión de base de datos a sus clientes, el software que permite la gestión de la base de datos es comúnmente conocidos como Motores de Base de Datos. Un servidor de base de datos, nos permite almacenar grandes cantidades de información, normalmente estructurada y organizada en tablas y registros para explotarla posteriormente.

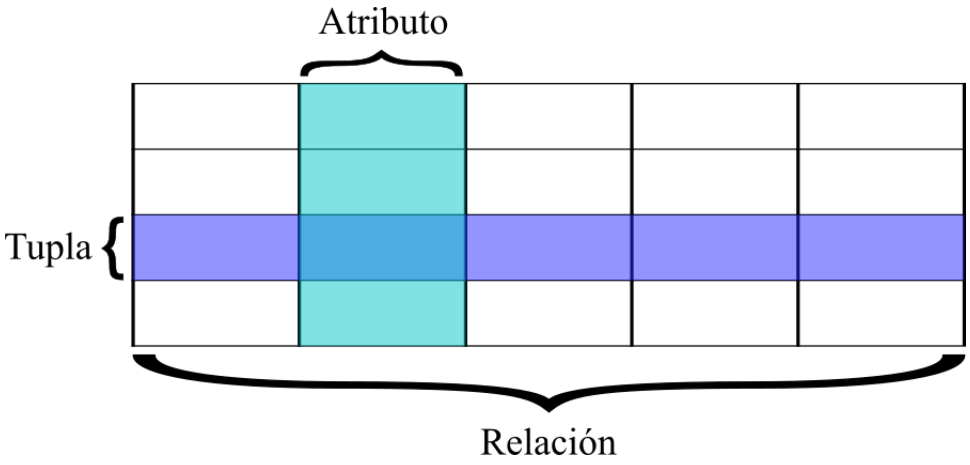


Figura 3.3.1: Representación del almacenamiento de información de una entidad en una DB (Leandro Alegsa, 2016).

3.3.1. Normalización de las bases de datos

Se ha definido una base de datos como un software que permite almacenar grandes cantidades de datos, pero como en todo debe de haber normas o estándares que rijan este tipo de almacenamiento todo esto con el fin de:

- Evitar la redundancia de datos.
- Disminuir el problema de extraer, actualizar, almacenar o eliminar información ya que al tener datos duplicados en la base de datos puede ocurrir este tipo de problemas.
- Proteger la integridad de los datos.

Existen conceptos o términos que debemos comprender ya que son parte fundamental de las bases de datos:

- ***Relación:*** También conocida como tabla y es un modelo donde se almacenan datos con una estructura y tipos de datos definidos.
- ***Registro:*** También conocido como tupla y contiene un objeto de datos único dentro de una tabla.
- ***Atributo:*** Columna o campo de una tabla y esta puede contener un solo valor de un registro.
- ***Clave:*** Llave o código de identificación son usadas para identificar un registro de otro.
- ***Clave candidata:*** Súper clave mínima por ejemplo puede ser un número consecutivo *(Auto_Increment)* de cada registro contenido en una tabla.
- ***Clave primaria:*** Clave candidata elegida, este es un identificador designado por el diseñador de la base de datos que le da un identificador único a cada registro contenido en una tabla de la base de datos por ejemplo (Registro Federal de Contribuyentes) (RFC), (Clave Única de Registro de Población) (CURP), (Número de Seguridad Social) (NSS), etc.
- ***Clave externa:*** Clave ajena o clave foránea, es un campo de la tabla que contiene un identificador, este identificador debe de ser una clave primaria de un registro dentro de otra tabla, una tabla puede contener llaves foráneas duplicada mientras que las llaves primarias son únicas y por lo tanto no está permitido que estén duplicadas dentro de una misma tabla.

- ***Clave alternativa:*** Es una clave secundaria, al igual que la llave primaria no está permitido que existan dos claves alternativas en la misma tabla por ejemplo una clave alternativa puede ser el nombre y apellidos de una persona.
- ***Clave compuesta:*** Es una clave formada por dos o más campos que combinados pueden formar una clave única por ejemplo supongamos que el nombre de una persona está descompuesto es tres partes dentro de una tabla nombres, apellido paterno y apellido materno, haciendo la combinación de estos tres campos podemos formar una clave compuesta única.
- ***Relational DataBase Management System–(Sistema de Gestión de Base de Datos Relacional) (RDBMS):*** Es el software que permite la manipulación de los datos contenidos en una base de datos.

Algunas normas o principios fundamentales son:

- Cada tabla de contener un nombre único, corto y significativo.
- No pueden existir filas iguales ya de ser así la información estaría duplicada y habría inconsistencia en la base de datos.
- Todos lo datos de una columna deben ser del mismo tipo.

Edgar Frank Codd fue un científico informático inglés conocido por crear el Modelo *Entity-Relationship*–(Entidad-Relación) (ER) el se dio cuenta de que en el mercado existían base de datos que afirmaban ser relacionales pero lo único que hacían era almacenar su información en tablas y de cierta manera dicha información estaba organizada pero no cumplía con los requisitos para ser llamadas bases de datos relacionales es por eso que Codd publico 12 reglas[11] en el año de 1985 que debe seguir un verdadero sistema relacional.

- ***Regla No. 1 –Regla de la información–:*** Deben existir metadatos dentro de la base de datos es decir la base de datos no solo debe contener información almacenada de los datos de los usuarios sino que también debe contener información de las tabla como nombre de tablas, nombre de vistas, nombre de columnas y los datos de las columnas deben estar almacenados en tablas de esta manera se tiene una especie de ”diccionario—catálogo” que puede ser explotado mediante el lenguaje *Structured Query Language*–(Lenguaje de Consulta Estructurado) (SQL).
- ***Regla No. 2 –Regla de acceso garantizado–:*** El uso de las llaves primarias y foráneas es obligatoria ya que debe existir una unión entre todas las tablas haciendo uso de el nombre de la tabla, una llave primaria, una llave foránea y el nombre del campo único de un registro que permita hacer el cruce de información para obtener uno y solamente un valor.
- ***Regla No. 3 –Tratamiento sistemático de los valores nulos–:*** Cuando ingresamos información a una base de datos se puede dar el caso de que cierta información sea desconocida o dicha información no aplica, esta regla define que dicha información faltante o desconocida puede ser representada a través de valores nulos ya que una RDBMS debe soportar el uso de operaciones con valores nulos y esto conlleva a realizar una tabla ***(tabla 3.3.1)*** de verdad que defina los resultados de estas operaciones:

Tabla 3.3.1: Tratamiento sistemático de valores nulos.

Valor	Operador	Valor	Resultado
null	AND	null	null
true	AND	null	null
false	AND	null	false
true	OR	null	true

- ***Regla No. 4 –Regla de la descripción de la base de datos–:*** Debe existir una descripción de la base de datos, tablas, vistas, permisos de usuarios autorizados, etc. esta información debe ser almacenada al igual que los demás datos ordinarios en tablas y columnas y solo debe ser estar disponible para los usuarios autorizados y debe ser consultada mediante consultas SQL todo esto con el fin de mantener una rápida y eficiente administración, mantenimiento y depuración de la base de datos.
- ***Regla No. 5 –Regla del sub-lenguaje integral–:*** Debe existir al menos un lenguaje con una sintaxis bien definida que permita administrar completamente la base de datos entre las acciones que debe facilitar el lenguaje deben existir: definir estructuras y datos de una tabla, manipulación de datos, definición de vistas, restricciones de integridad, control de autorizaciones, gestión de usuarios y transacciones.
- ***Regla No. 6 –Regla de la actualización de vistas–:*** Una vista es el resultado del cruce de información de dos o más tablas *(Join)*, dichas vistas se crearon con el fin de tener cierta información disponible y ahorrar tiempo y recursos del servidor. Esta regla define que si existe un cambio en las tablas utilizadas por la vista esta debe actualizarse automáticamente tomando los valores alterados en las tablas utilizadas por la vista.
- ***Regla No. 7 –Regla de insertar y actualizar–:*** Una base de datos debe contar con la capacidad de permitir *Create Read Update and Delete*–(Crear Leer Actualizar y Borrar) (CRUD), esta característica debe estar siempre disponible sin importar las relaciones y restricciones que haya en las tablas contenidas en la base de datos.
- ***Regla No. 8 –Regla de la independencia física–:*** Los datos contenidos en una base de datos deben permanecer íntegros aun cuando se realicen cambios a nivel físico como por ejemplo la migración de un sistema de base de datos a otro *(siempre y cuando sean compatibles)*, cambio de los ficheros dentro del almacenamiento a otro destino, etc.
- ***Regla No. 9 –Regla de la independencia lógica–:*** Cuando se realizan cambios en la base de datos a nivel lógico la información se debe de preservar y mantener integra es decir, no debe sufrir cambios, malformaciones o perdida de esta. Por ejemplo cuando se crea una nueva tabla en la base de datos, al agregar una columna a una tabla existente, al modificar el nombre de una columna, etc. la información debe permanecer integra aun cuando se realicen cambios en el esquema de la base de datos.
- ***Regla No. 10 –Regla de la independencia de la integridad–:*** Existen dos reglas de integridad:
  - Ninguna clave primaria dentro de una tabla debe estar en blanco nula.
  - Cada llave foránea definida en una tabla debe existir un valor de clave primaria concordante.
- ***Regla No. 11 –Regla de la distribución–:*** Debe existir un lenguaje que soporte la distribución de una base de datos en diversos servidores incluyendo servidores ejecutando diversos sistemas operativos ubicados en diversas zonas geográficas conectados por una red de computadoras permitiendo realizar operaciones sobre

los datos sin poner limitantes, ejecutándose como si se tratara de un servidor de base de datos corriendo en un equipo local.

- ***Regla No. 12 –Regla de la no-subversión–:*** Si el software que administra y almacena la base de datos tiene soporte o incluye lenguajes de bajo nivel de ninguna manera estos pueden romper las reglas violando la integridad de los datos tal y como están definido en los lenguajes de alto nivel como lo es SQL.

3.3.1.1. Objeto Mapeo Relacional (ORM)

También conocidos como mapeo de objetos relacionales[9] utilizado para convertir datos usados en los diferentes sistemas utilizando un lenguaje de programación orientada a objetos y utilizando una base de datos como motor de almacenamiento usualmente utilizando en bases de datos en lenguaje SQL ya que estas almacenan su información en tablas y hacen uso de enlaces o apuntadores para relacionar su información entre ellas. Se hace uso de las características de la programación orientada a objetos como la herencia y el polimorfismo para hacer relaciones entre las diversas clases.

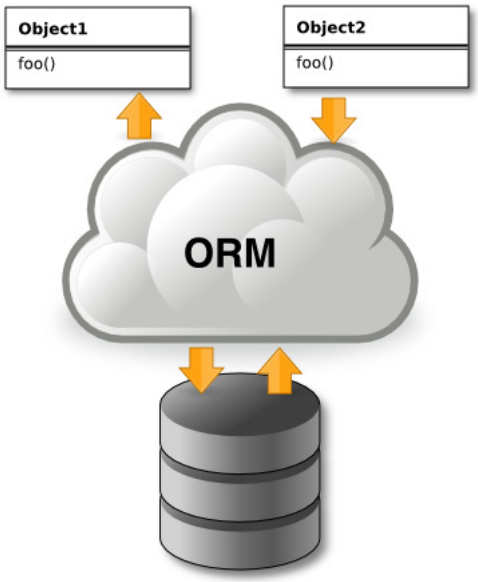


Figura 3.3.2: Conversión de datos a objetos mediante ORM (Alberto Fernández, 2013).

3.3.1.2. JPA y clases POJO

Java Persistence *Application Programming Interface*–(Interfaz de Programación de Aplicaciones) (API) también conocido más comúnmente como *Java Persistence API*–(API de Persistencia de Java) (JPA)[16][27, Página 207] es un *framework* del lenguaje de programación Java que maneja datos relacionales y además proporciona un estándar para gestionar dichos datos relacionales en las plataformas de Java *Standard Edition*–(Edición Estándar) (SE) y Java *Enterprise Edition*–(Edición Empresarial) (EE) permitiendo simplificar la persistencia en las bases de datos. Este *framework* hace uso de las clases *Plain Old Java Object*–(Objeto Java Plano Antiguo) (POJO)[15, Capítulo 4] y estas son instancias de una clase que no extienden ni implementan nada en especial en otras palabras son clases simples y no dependen de un *framework* en especial tiene la característica de tener todos sus atributos como privados y además de contener métodos públicos setters y gettes para poder asignar u obtener sus atributos. Hace uso de las siguientes anotaciones principales para poder mapear las entidades[15, Capítulo 8]:

- @Entity
- @Table
- @Column
- @Id
- @OneToOne
- @OneToMany
- @ManyToOne
- @ManyToMany

Combinando JPA[16][27, Página 207] con las clases POJO[15, Capítulo 4] mediante anotaciones se puede crear un tipo de estándar permitiendo utilizar diversos tipos de motores de base datos para almacenar, extraer, actualizar y/o eliminar información sin hacer uso del lenguaje de SQL simplemente se tiene que configurar para indicar que tipo de base de datos se desea utilizar y el *framework* hará el resto.

3.3.1.3. Hibernate®

Es un *framework* para Java de software libre usado para ORM[9] lanzado oficialmente en el año 2001. Su característica es que utiliza lenguajes de alto nivel *(Java)* para realizar el mapeo y la persistencia de datos en una base de datos relacional. No solo se limita a la inserción de datos sino que también se pueden realizar consultas ya que proporciona su propio lenguaje SQL llamado *Hibernate Query Languaje*–(Lenguaje de Consulta de Hibernate) (HQL)[15, Capítulo 16], actualizaciones y/o borrado de registro de la base de datos. Este mapeo se puede realizar de varias forma una de ellas es haciendo uso de las clases POJO[15, Capítulo 4] y el uso de anotaciones para hacer uso del mapeo relacional mediante un lenguaje de programación *(Java)*, dentro de estas clases POJO[15, Capítulo 4] se pueden realizar las entidades, columnas mediante el uso de anotaciones con JPA[16] o también puede ser mediante un archivo de configuración en formato XML[15, Página 46].

3.3.2. Modelos de base de datos

Teniendo en cuenta los conceptos anteriores existen dos tipos de bases de datos y diversos modelos tales como:

- ***Bases de datos estáticas:*** Este tipo de base de datos son principalmente de solo lectura no se puede interactuar directamente con ellas para almacenar, eliminar y/o modificar información, solamente se pueden hacer consultas y pueden contener o almacenar datos históricos o registros y estos pueden ser utilizados posteriormente para procesarlos y obtener datos estadísticos o informativos. Dependiendo del enfoque que se de a este tipo de base de datos también puede ser una buena alternativa para tener un respaldo de la base de datos original.
- ***Bases de datos dinámicas:*** Estas bases de datos permiten la interacción y manipulación de los datos que almacenan en su interior realizando operaciones como inserción, edición, eliminación y por supuesto consultas de los registros que contiene permitiendo un flujo de información tanto de entrada como de salida.
- ***Modelo de bases de datos relacionales (RDBMS):*** Este tipo de base de datos es la más usada y la más común encontrar en diversas organizaciones ya que la información esta organizada en tablas o tuplas separadas por categorías predefinidas, dichas tuplas contiene registros de información. Las bases de datos relacionales[27,

Capítulo 7] permite la extracción, modificación e inserción de información mediante sentencias SQL *(Lenguaje estructurado de consultas)*.

- ***Modelo de bases de datos no relacionales (Non SQL–(No sólo SQL) (noSQL)):*** Este tipo de base de datos no almacena su información en tablas o tuplas y normalmente no soportan operación Join y tampoco manejan *Atomicity Consistency Isolation and Durability*–(Atomicidad Consistencia Aislamiento Durabilidad) (ACID) una de las ventajas de este tipo de base de datos es que son altamente escalables de forma vertical, este tipo de base de datos ganaron terreno con forme fue creciendo Internet ya que se tienen grandes cantidades de información normalmente estructurada con una clave-valor, el reto hoy en día es crear sistemas rápidos y eficientes que puedan procesar y analizar toda esta información con el fin de obtener resultados y poder usarlos a favor de una empresa u organización para fines de lucro o estadísticos.
- ***Bases de datos transaccionales:*** Este tipo de base datos son dinámicas con la característica de que manejan grandes volúmenes de información tanto de entrada como de salida a grandes velocidades, estas bases de datos tiene un alto índice de seguridad además de que manejan ACID ya que cada transacción debe ser atómica para evitar perdida de información son muy usadas por los bancos y empresas industriales.

Estos son algunos modelos tradicionales de las bases de datos aunque existe más como por ejemplo bases de datos de red, bases de datos jerárquicas, base de datos documentales, bases de datos deductivas, etc. cada una con ciertas características pero todas con la finalidad de almacenar información y ponerla a disposición de los usuarios.

3.3.2.1. Bases de datos distribuidas

Una *Distributed Database*–(Base de Datos Distribuida) (BDD) es un banco de datos distribuido en diversas partes geográficas interconectadas por una red de computadoras formado nodos o estaciones, también este tipo de base de datos pueden estar distribuidas lógicamente en un mismo servidor con diferentes particiones y/o corriendo maquinas virtuales, este banco de datos que se encuentra distribuido están relacionado lógicamente y a estos nodos pueden conectarse diversos nodos que actúen como clientes y realizar operaciones locales o distribuidas y tener la ilusión de estar realizando operaciones localmente.

Este tipo de base de datos deben ser atómicas es decir deben de contar con la capacidad de realizar transacciones[27, Capítilo 15] ya que como son sistemas distribuidos deben de haber consistencia en los datos es por eso que hay que tener en cuenta la concurrencia ya que de lo contrario la base de datos ya no estaría cumpliendo con la característica de ACID por lo tanto hay que tener en cuenta el bloqueo:

- Nivel de tabla.
- Nivel de registro.
- Nivel de campo.

Existen diversos tipos de bases distribuidas y cada una tiene un enfoque de trabajo especifico:

- ***Centralizada:*** Este tipo de base de datos almacena toda su información en un solo sitio hablando geográficamente o lógicamente(un servidor) y los usuarios son los únicos que se encuentran distribuidos realizando peticiones al servidor bajo el modelo cliente-servidor[33][27, Página 445] de este modo el servidor realiza el procesamiento distribuido de las diferentes peticiones de los usuarios y entrega los resultados solicitados. Una desventaja de este tipo de base de datos es que no tiene disponibilidad, fiabilidad y dependiendo de las características del servidor tiene un procesamiento limitado ya que solo se cuenta con

un servidor que da respuesta a las peticiones de los usuarios y no múltiples servidores como pasa en otros modelos.

- ***Replicada:*** Si se busca un modelo de base de datos que permita la fiabilidad de los datos, la disponibilidad y la alta concurrencia este tipo de base de datos es el ideal ya que en caso de algún accidente controlado o no controlado ejemplo un accidente no controlado puede ser un desastre natural, una falla en la infraestructura, etc.) el sistema seguirá funcionando claro para esto hay que implementar otros métodos que permitan este funcionamiento (ejemplo balanceadores de carga[28] que permitan redireccionar las peticiones a otro servidor disponible), normalmente este tipo de bases de datos tiene una copia exactamente igual y se están actualizando periódicamente cuando existe algún cambio en alguna de las bases de datos distribuidas esto por lo tanto genera un gran flujo de información en la red que interconecta dichos nodos por lo tanto los costos de implementación de este modelo se incrementan ya que también hay que tener en cuenta la infraestructura que implica implementar un modelo de este tipo pero por otro lado se puede tener acceso a la misma información más rápidamente y soportando un mayor número de peticiones simultáneamente por lo que si se requiere es un modelo seguro y eficiente este modelo es óptimo.
- ***Particionada:*** En este tipo de modelo de base de datos no existen redundancia o replicación de la información esto quiere decir que cada nodo contiene un fragmento de una base de datos y todos los nodos en conjunto conectados por una red de computadoras conforman una base de datos completa y concisa hay que tener en cuenta que en este modelo los nodos pueden o no estar distribuidos geográficamente, el problema de este tipo de base de datos es que no existe disponibilidad ni redundancia por lo que si un nodo llega a fallar los usuarios que intente hacer peticiones a ese fragmento de la base de datos fallaran, por otro lado los costes de infraestructura se reducen drásticamente ya que no hay que crear servidores que estén replicando la información, también se puede trabajar de forma paralela ya que se pueden separar las consultas a diferentes servidores y obtener datos diferentes simultáneamente.
- ***Híbrida:*** Este modelo de base de datos combina dos modelos el distribuido y el particionado, aquí los datos se encuentran distribuidos en diferentes nodos y cada uno de los nodos es replicado todo esto con el fin de seguridad y disponibilidad permitiendo que gracias al modelo particionado se pueda trabajar paralelamente con las bases de datos y con el modelo distribuido la información este siempre disponible y segura gracias a la replicación que se realiza periódicamente cada que se realiza algún cambio en las bases de datos particionadas.

3.3.3. Transacciones

Una transacción[27, Capítulo 15] es una serie de instrucciones que se ejecutan de forma secuencial y que son atómicas o indivisibles y si por alguna razón una de las instrucciones falla o es interrumpida se deben deshacer todas las operaciones realizadas anteriormente hasta dejar la base de datos como se encontraba originalmente antes de realizar la transacción.

Una base de datos es transaccional cuando cumple con ACID:

- ***Atomicidad:*** En una transacción se ejecutan una serie de sentencias y todas deben completarse satisfactoriamente o ninguna de ellas si el sistema falla o se cancela la transacción en un termino medio, las transacciones[27, Capítulo 15] deben de ser completas.
- ***Consistencia:*** Esta característica también conocida como integridad marca que los datos deben ser exactos y que por ninguna razón vas a ser alterados y/o modificados esto quiere decir que la base de datos va a estar intacta.

- ***Aislamiento:*** Asegura que más de un usuario no puedan manipular o acceder a la misma información es decir si un usuario esta realizando operaciones sobre un registro el aislamiento asegura que ese registro este bloqueado hasta que se deje de trabajar sobre el y finalmente sea liberado para que otro usuario pueda acceder a el, de otro modo habría inconsistencia y perdida de información.
- ***Durabilidad:*** También conocida como persistencia, una vez que una transacción es completada se debe de tener la certeza de que los cambios no pueden ser deshechos aunque los sistemas fallen.

Si se cumple con estas características se puede decir que un *DataBase Management System*–(Sistema Gestor de Base de Datos) (SGBD)[23, Página 22] es transaccional y para implementar esto SQL provee las herramientas necesarias para crear este tipo de sistemas con mecanismos tales como:

- ***BEGIN TRAN:*** Específica que va a empezar una transacción.
- ***COMMIT TRAN:*** Le indica al motor que puede considerar la transacción completada con éxito.
- ***ROLLBACK TRAN:*** Indica que se ha alcanzado un fallo y que debe restablecer la base al punto de integridad.

3.3.4. Procedimientos almacenados

Un procedimiento almacenado también conocido como triggers o disparadores son programas o sentencias de instrucciones dentro del motor de la base de datos la ventaja de dichos disparadores es que los usuarios hacen peticiones a la base de datos y esta procesa la información y devuelve el resultado deseado reduciendo así la carga de procesamiento del lado del cliente esto es muy eficiente cuando se tienen múltiples clientes simultáneamente que solicitan información y realizan operaciones al la base de datos contenida en el servidor, normalmente al usar los procedimientos almacenado se separa la base de datos en un servidor diferente.

Los procedimientos almacenados normalmente actúan cuando hay un cambio en el alguna tabla o registro de la base de datos como un *Update*, *Delete* y/o *Insert* permitiendo así tener mayor control sobre lo que pasa en la base de datos y también organizando de mejor manera los programas externos que interactúan con la base de datos permitiendo la simplificación, el mantenimiento y escalabilidad del programa ya que parte de las operaciones están contenidas en el motor de la base de datos y el programa solo espera parámetros que la base de datos ya ha procesado previamente.

Una de las ventajas de los procedimientos almacenados es la rapidez con la que se ejecutan las operaciones, también se evita que los datos sean alterados o corrompidos por los programas externos defectuosos o mal codificados de los distintos usuarios que interactúan con el servidor de base de datos. Hay quienes piensa que la base de datos se debe usar solo para almacenar datos y estos deben ser procesador por una aplicación externa pero depende del enfoque que se le vaya a dar al sistema si es un sistema que maneja una gran cantidad de operaciones y lo que se requiere es que se realice en el menor tiempo posible y además es importante la integridad de los datos se puede optar por la opción de crear procesos almacenados para solucionar este tipo de problemas.

3.3.5. Replica de las bases de datos

Una replica es una imagen o una copia exacta ya sea de una base de datos, una partición, un disco, una carpeta o todo un sistema. esta copia se actualiza si surge un cambio en la información original *(o en el resto de las replicas dependiendo de la arquitectura implementada)* todo esto con el fin de mantener la integridad de los datos ante algún fallo de software o de hardware, también las replicas son utilizadas para brindar eficiencia y disponibilidad a los

usuarios ya que permiten seguir operando si surge algún problema con algún servidor, también permite que más usuarios accedan a la información simultáneamente una desventaja de este tipo de sistemas es la infraestructura ya que mientras más replicas de la base de datos se tengan se tienen que implementar más servidores y con ello una red de computadoras que permita conectar estos nodos, es recomendable que los nodos implementados donde se replicara la información se encuentren distribuidos[27, Capítulo 19] en alguna otra ubicación geográfica todo esto con el fin de mantener los dispositivos siempre disponibles para los usuarios ya que de lo contrario si se implementa un sistema centralizado pueden ocurrir desastres no controlados que puedan afectar la disponibilidad de los sistemas por ejemplo un fallo en la energía eléctrica, fallo en el servicio de internet, un desastre natural, etc.).

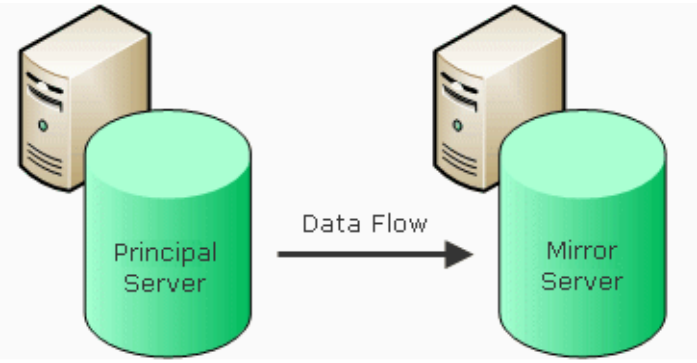


Figura 3.3.3: Descripción gráfica de una replica de DB (Imma, 2010).

Existen diversas arquitecturas de replicas de bases de datos:

- ***Maestro-Maestro:*** Cuando tenemos un gran número de usuarios escribiendo en la base de datos simultáneamente llegara un momento en que el servidor no dará abasto para dar respuestas a todas las solicitudes es por eso que en este tipo de arquitectura master-master se tienen dos servidores en los que se puede escribir información para ello deben de estar sincronizados perfectamente para que no existan errores de escritura y por lo tanto se sigue el esquema:

  - Server1 es el Master del Server2 (por lo tanto el Server2 es el Slave del Server1).
  - Server2 es el Master del Server1 (por lo tanto el Server1 es el Slave del Server2).

  Con esta arquitectura se tiene la certeza de que todo lo que se escribe en el Server1 se replicara en el Server2 y por lo tanto todo lo que se escriba en el Server2 se replicara en el Sever1. En este tipo de arquitectura solo pueden existir dos nodos actuando como master ya que de lo contrario formaría otro tipo de arquitectura *(circular)*, también se puede combinar la arquitectura Master-Slave para replicar la información y proporcionar un sistema más eficiente y disponible.

- ***Maestro-Esclavo:*** En este tipo de arquitectura la regla es: un Master puede tener múltiples Slave y un Slave solo puede tener un Master. Este tipo de arquitectura es eficiente cuando se tiene una alta demanda de consultas en la base de datos y cuando es importante la integridad de los datos ya que los Slaves sirven como *backup*, hay que tener en cuenta que los Slave solo sirven como banco de datos para consultas es decir no se puede hacer escritura en ellas solo están permitidos los SELECT, por lo contrario el Master tiene la capacidad de escritura en todos sus registros y cuando esto sucede automáticamente se actualizan los cambios en los diversos Slaves teniendo así replicas exactas de la base de datos maestra.

- ***Circular:*** Cuando se necesita escribir en más de un servidor de base de datos y la arquitectura Master-Master no es suficiente este es una alternativa pero no es recomendable ya que los nodos están conectados en forma lineal y dependen el uno del otro por lo tanto si uno o más nodos intermedios llegan a fallar la integridad de la base de datos se vera en riesgo además entre más nodos se conecte con este tipo de arquitectura mayor será el caos de la administración. Este tipo de arquitectura permite escribir y/o consultar en cualquiera de los nodos disponibles replicando la información de uno a otro ya que el Server1 se conecta con el Server2, el Sever2 con el Server3 y así sucesivamente hasta llegar al ultimo nodo que se conectara al Server1 formando así una arquitectura de anillo o también conocida como circular.

3.4. Servidores de base de datos

Existen diversas formas de implementar un servidor de base de datos, para implementar un servidor es esencial tomar en cuenta aspectos muy importantes tales como:

- ***Disponibilidad:*** Para que un sistema cuente con disponibilidad debe de estar activo las 24 horas del día los 7 días de la semana sin interrupciones para los usuarios y para lograr esto se tienen que implementar mecanismos que permitan esto, mecanismos de infraestructura, hardware y software tales como:
  - Se debe contar con 2 líneas de corriente eléctrica diferentes para evitar cualquier apagón de los servidores. Los servidores también pueden incorporar dos fuentes de poder para evitar que el servidor se quede sin energía eléctrica por un desperfecto o un desgaste en el hardware.
  - Contar con *Uninterruptible Power Supply*–(Sistema de Alimentación Interrumpida) (UPS) que permitan operar a los servidores por un lapso de tiempo relativo ya sea para seguir brindado servicio a los usuarios o para apagar de manera satisfactoria los sistemas y evitar daño en ellos.
  - Del mismo modo hay que contar con dos *Internet Service Provider*–(Proveedor de Servicio de Internet) (ISP) diferentes para evitar fallos de comunicación en la red entre el servidor y los diferentes usuarios, hay que contar mínimo con 2 tarjetas de red que permitan disminuir la incertidumbre de falla en el hardware, también se puede optar por realizar un bonding para aumentar la velocidad de la red y aprovechar los dos servicios brindados por el ISP.
  - Replicar la información en tiempo real de la base de datos y de los sistemas en diferentes servidores ubicados en diferentes zonas geográficas de preferencia e implementar balanceadores de carga[28] que permitan redireccionar el trafico aumentando la capacidad de usuarios que se pueda dar respuesta y evitando fallas en el servicio, por lo que si un servidor falla por X razón el balanceador se encargara[28] de enrutar la petición hacia otro servidor que se encuentre disponible.

  Estos son algunos mecanismos que se pueden implementar para que un sistema este siempre disponible y atendiendo las peticiones de los usuarios, hay que tener en cuenta que para que un sistema sea de calidad, fiable y usable debe de estar siempre disponible y debe entregar los resultados deseados

- ***Eficiencia:*** Se debe de analizar el rendimiento del servidor que se va a implementar, como por ejemplo la *Random Access Memory*–(Memoria de Acceso Aleatorio) (RAM) para evitar paginación y evitar que los procesos tarden más de lo normal, se tiene que tener una memoria RAM que permita gestionar todas las operaciones solicitadas a la base de datos y gestionar el sistema operativo sin ningún problema, hay que tener en cuenta los núcleos del procesador para que nos permitan trabajar de forma paralela con la base de datos, asegurarnos que el sistema operativo sea el óptimo además teniendo en cuenta su arquitectura, tener en cuenta el ancho de banda para evitar los cuellos de botella.
- ***Fiabilidad:*** Todos los sistemas deben ser seguros y confiables hablando de un servidor de base de datos los datos almacenados deben de tener un nivel de seguridad óptimo es decir que no cualquier persona pueda acceder a ellos es por ello que se implementan mecanismos de seguridad que permiten resguardar dicha información tales mecanismos incluyen *firewalls* a nivel de software o hardware, bloqueo de puertos, encriptación de información, etc. también para que una base de datos sea confiable debe de realizar y almacenar los datos correctamente es decir no debe haber deformación o manipulación errónea de la información o peor aun que la información desaparezca parcial o definitivamente, para evitar este tipo de circunstancias muchos motores de base de datos implementan las transacciones[27, Capítulo 15] y dichas transacciones tienen la característica de ACID que implementa mecanismos de fiabilidad de la información.
- ***Almacenamiento:*** Hay que tener en cuenta el espacio y la capacidad de almacenamiento no volátil del servidor tanto para las actividades actuales como a futuro ya que si son bases de datos muy concurrentes donde múltiples usuarios agregan información estas pueden ir creciendo exponencialmente en un periodo reducido y pueden llegar a un punto donde se alcance la capacidad máxima de almacenamiento es por eso que hay que realizar cálculos que nos permitan estimar el crecimiento de las bases de datos, para evitar este tipo de problemas se pueden implementar mecanismos como bases de datos distribuidas[27, Capítulo 19] o *Redundant Array of Independent Disks*–(Arreglo Redundante de Discos Independientes) (RAID)'s[27, Página 255] que permitan aumentar la capacidad del almacenamiento no volátil ya sea por software o por hardware.

3.4.1. Servidores dedicados

Un servidor dedicado solo contiene un servicio corriendo en este caso el servidor de base de datos *(en lugar de alojar otros servicios como por ejemplo el servidor web, servidor de archivos)*, normalmente tiene capacidades de procesamiento y almacenamiento superiores con la ventaja de una mejor administración, en dicho servidor se pueden visualizar mejor los procesos ejecutándose y puede haber una mayor portabilidad y escalabilidad ya que todo esto se puede hacer separado de los demás servicios que se encuentre trabajando en conjunto con la base de datos para formar un sistemas complejo y robusto.

Creando servidores dedicados se tiene un mayor desempeño de los procesos ejecutándose y también se evita que si un servidor falla ya sea por una falla mecánica o una falla lógica los otros sistemas seguirán disponibles ya que se encuentran aislados del servidor defectuoso. Estos nodos se conectan por una red de computadoras que permitan unificar todos los servicios para trabajar en conjunto como uno solo.

Una desventaja de los servidores dedicados es el coste de la infraestructura ya que requiere un host por cada servicio que vaya a ser usado o implementado en dicho servidor además de los costes de los servicios que necesita ese servidor para operar como energía, internet, mantenimiento, etc.

3.4.2. Servidores no dedicados

Son aquellos que ejecutan más de un servicio a la vez, comparten sus tareas *(procesador y memoria RAM)* y almacenamiento *(discos duros)* con otras aplicaciones, cabe implementar un servidor de este tipo si el procesamiento y el número de clientes que acceden paralelamente es reducido de este modo se ahorran costes, pero el mantenimiento, escalabilidad y portabilidad es más limitado ya que todos los servicios están corriendo en el mismo servidor y si ocurre un falla mecánica o lógica todos los sistemas dejaran de funcionar.

3.4.3. Clúster de servidores

También conocido como granja de servidores es un conjunto de ordenadores conectados entre si mediante una red de alta velocidad y de forma lógica se comporta como si fuera una sola computadora. Estos son empleados para mejorar el rendimiento y la disponibilidad ya que tener una granja de servidores es más económico que tener varios equipos con características de alto rendimiento y disponibilidad. Un clúster proporciona servicios como:

- Alto rendimiento.
- Alta disponibilidad.
- Alta eficiencia.
- Balanceo de carga.
- Escalabilidad.

Existen tres tipos de clúster, existe una gran flexibilidad al crear un clúster una de las ventajas es que estos ordenadores conectados entre si mediante una red LAN[21, Página 51] de alta velocidad los nodos no tienen que estar distribuidos geográficamente sino que es un sistema centralizado por lo tanto no se tienen que estar interconectando ni implementando protocolos de comunicación a gran distancia además la administración es centralizada y por lo tanto los costes de mantenimiento disminuyen:

- ***Clúster homogéneo:*** Aquí los clúster conectados entre si cuentan con el mismo hardware y el mismo software.
- ***Clúster semihomogéneo:*** Aquí los ordenadores tiene un diferente rendimiento uno de otro pero cuentan con un sistema operativo y arquitecturas similares.
- ***Clúster heterogéneo:*** Aquí los ordenadores que conforman el clúster hardware y sistema operativo diferente este tipo de clúster es más económico construirlo y ya combinando cierto número de ordenadores aumenta el rendimiento y la disponibilidad.

Para implementar un clúster no basta con conectar los nodos entre si, existen sistemas operativos y software que permiten la administración de los recursos de un clúster como por ejemplo Red-Hat, Plan9, etc.

Estos son los cuatro servidores de base de datos más usados:

- ***Oracle Database®:*** Es un sistema de gestión de base de datos[27, Capítulo 25] de tipo *Object-Relational DataBase Management System*–(Sistema de Gestión de Base de Datos Objeto-Relacional) (ORDBMS) desarrollado por Oracle Corporation®. Se consideraba uno de los sistemas de base de datos que acaparaba el mercado en cuestión de servidores empresariales pero eso disminuyo con la llegada de sistemas como Microsoft SQL Server®[27, Capítulo 27], PostgreSQL®[12], MySQL®[26] entre otros. Oracle Database® es considerado como uno de los sistemas más completos ya que soporta transacciones[27, Capítulo 15], procedimientos almacenados, estabilidad, escalabilidad y soporte multiplataforma, puede ejecutarse en una computadora personal hasta un súper ordenador. Puede ser administrado por línea de comandos o desde una interfaz gráfica muy intuitiva y cómoda de utilizar que Oracle® desarrollo para una mayor facilidad en el manejo y gestión de las bases de datos. Una desventaja es que este software es propietario por lo tanto el costo de las licencias es elevado además si la instalación del software es inadecuado puede causar problemas con el sistema operativo volviéndolo lento.
- ***Microsoft SQL Server®:*** Es un sistema de base de datos[27, Capítulo 27] de modelo relacional fue desarrollado por Microsoft por lo tanto no es multiplataforma y solo puede ser usado bajo sistemas operativos Windows. Este sistema soporta transacciones[27, Capítulo 15], procedimientos almacenados, trabaja bajo el modelo cliente-servidor[33][27, Página 445], contiene un entorno gráfico que permite la administración y el uso de comandos aunque también cuenta con línea de comandos llamada SQLCMD o PowerSell. Una de sus principales características es la comprensión de datos ya que permite que estos se almacenen de una manera más eficiente y reduzca los requerimientos de almacenamiento, además gracias a la comprensión de datos ofrece mejoras significativas en el rendimiento para grandes cargas de trabajo, también permite agregar otros servidores de SQL Server e interactuar con ellos. Las desventajas de este software es que utiliza en grandes cantidades la memoria RAM en la instalación como en la ejecución del software, es un software privativo así que restringe ciertas funcionalidades y tiene un pésima implementación en los tipos de datos variables.
- ***dBASE II (DB2)®:*** Es una marca comercial propiedad de *International Business Machines*–(Maquina de Negocios Internacionales) (IBM)® bajo la cual se comercializa un sistema de gestión de base de datos. DB2®[27, Capítulo 26] es un motor de base de datos que permite gestionar datos relacionales convencionales como datos¸ XML de forma nativa es decir no tiene que realizar conversiones de archivos para poder procesar y buscar información esta característica es única en el mercado, también tiene la capacidad de ejecutar procedimientos almacenados. DB2® posee un motor gráfico el cual permite ver el tiempo de ejecución de una sentencia SQL y corregir detalles para aumentar el rendimiento. Posee tablas de resumen, tablas replicadas, uniones hash y utiliza una combinación de seguridad externa y control interno de acceso a proteger datos. Una de las desventajas de DB2® es la lentitud al crear y ejecutar consultas, utiliza de manera excesiva la memoria RAM para la instalación del software como para la ejecución el software y posee un alto costo en licencias personales. Existe otra versión de DB2® llamada DB2® Express C que ofrece a los desarrolladores un conjunto de herramientas para crear aplicaciones de base de datos para el escritorio, entornos cliente-servidor[33][27, Página 445] o para la web.
- ***MySQL®:*** Es un sistema de gestión de base de datos[26] relacional por Oracle® distribuido bajo licencia dual y es considerado como uno de los SGBD[23, Página 22] más populares a nivel mundial sobre todo para entornos de desarrollo web. Es un sistema multiplataforma y es una base de datos muy rápida en modo lectura aunque con problemas en ambientes concurridos en situaciones de transacciones y modificación de información. Una de las características es que usa multihilos mediante hilos del kernel aunque Oracle® provee el código fuente de MySQL®[26] para diferentes tipos de sistemas operativos el rendimiento aumenta

considerablemente en sistemas operativos *GNU's Not Unix*–(GNU No es Unix) (GNU)/Linux. Cuenta con características como transacciones[27, Capítulo 15], claves primarias y foráneas, conectividad segura mediante validación basada en el host y el trafico de contraseña esta cifrado al conectarse a un servidor, replicación, búsqueda e indexación de campos de texto, contiene un sólido y amplio subconjunto del lenguaje SQL e integración perfecta con PHP. Una de las ventajas de utilizar este sistema es que es Open Source, ofrece velocidad al realizar las operaciones lo que lo hace uno de los gestores con mayor rendimiento en el mercado, bajo costo en requerimientos para la implementación de base de datos, ya que gracias a su bajo consumo puede ser ejecutado en un equipo de escasos recursos sin ningún problema, posee facilidad de configuración e instalación. Una de las principales desventajas es que es muy limitado y no soporta integridad relacional ni transacciones en aplicaciones web complejas que requieren múltiples usuarios.

3.5. Clúster de alta eficiencia, alta disponibilidad y alto rendimiento

Cada sistema de software que se desarrolla tiene características y requerimientos diferentes es importante tener en cuenta cada uno de ellos, por eso es importante evaluar detalladamente los requerimientos del sistema, no solo los requerimientos de software sino también hay que evaluar los requerimientos de hardware ya que nos permitirán implementar estos sistemas adecuadamente y podremos aprovechar de mejor manera tanto el software como el hardware que compondrán el sistema informático.

3.5.1. Clúster de alta eficiencia

Son equipos conectados entre si formando clúster conectados entre si mediante una red de computadoras, su objetivo es ejecutar la mayor cantidad de tareas en el menor tiempo posible. Estos pueden clasificarse en:

- ***Clúster comerciales:*** De alta disponibilidad y de alta eficiencia.
- ***Clúster científicos:*** de alto rendimiento.

3.5.2. Clúster de alta disponibilidad

La alta disponibilidad[24, Página 41] en términos informáticos se refiere a un conjunto de dos o más equipos de cómputo que contienen servicios compartidos *(sitio web, base de datos, servidor de archivos, etc.)*, estos servicios son monitoreados constantemente por algún mecanismo de software que permite determinar cuando un equipo de cómputo ha fallado ya sea un fallo mecánico o un fallo lógico. La alta disponibilidad la podemos clasificar el dos tipos de clases.

- ***Alta disponibilidad de infraestructura:*** Cuando un equipo informático llega a fallar por algún desperfecto o por desgaste en alguno de sus componentes el software que realiza el monitoreo debe ser capaz de detectar este fallo y determinar que el equipo que contenía dicho servicio a fallado de esta manera debe ser capaz de iniciar los servicios de otro equipo de cómputo *(failover)*, cuando el equipo original es restaurado los servicios deben ser migrados a al equipo de cómputo original *(failback)* y seguir operando como si nada hubiese pasado, de esta manera se tiene un servicio de alta disponibilidad ya que en un fallo de hardware los usuarios no deben percibir que el equipo ha fallado.

- ***Alta disponibilidad de aplicación:*** Aquí también debe de existir un software que monitoré los servicios prestados por un equipo de cómputo ya que de haber un fallo en las aplicaciones el software debe de iniciar algún otro equipo de cómputo que este disponible para que pueda seguir brindando el servicio, una vez que el equipo original es restaurado se deben de migrar los datos al equipo original asegurando así la integridad de los datos y reduciendo la percepción hacia los usuarios de que algo ha fallado.

3.5.3. Clúster de alto rendimiento

Es la capacidad de realizar grandes operaciones de cálculo ya sea por un solo ordenador o por un conjunto de ellos *(clúster de alto rendimiento)*[7][24, Página 42]. Los clúster de alto rendimiento es un conjunto de computadoras que trabajan en conjunto conectadas por una red de computadoras de alta velocidad que permiten realizar una gran cantidad de cálculos de manera más rápida, estos usualmente son construidos y utilizados por las universidades ya que en muchas circunstancias el tamaño de problema que necesitan resolver requiere de muchos cálculos matemáticos y además es más económico construir un clúster de alto rendimiento que adquirir una súper computadora que permita realizar las mismas operaciones de cálculos requeridas ya que en el clúster pueden ser utilizadas las computadoras personales o computadoras que tal vez se hayan dado por obsoletas.

Para poder hacer uso de estos clúster hay que dividir el problema en problemas más pequeños y los pequeños problemas deben ser paralelizados en los diferentes equipos que construyen el clúster para agilizar el procesamiento de cálculo ya que de otra manera no tendría mucho sentido la construcción del clúster. También hay que tener en cuenta que para que un clúster funcione hay que utilizar bibliotecas que nos permitan interconectar y procesar tareas de manera paralela en los diversos equipos que conforman el clúster, bibliotecas como *Parallel Virtual Machine*–(Maquina Virtual Paralela) (PVM) que permiten conectar nodos heterogéneos, *Message Passing Interface*–(Interfaz de Paso de Mensajes) (MPI) que permite conectar nodos homogéneos, también hay que hacer uso de software que permita el procesamiento distribuido como OSCAR (Open Source Clúster Aplication Resources: software libre que trabaja en sistemas Linux), *Windows Computer Cluster*–(Cluster de Cómputo en Windows) (WCC) 2003 para hacer clúster con sistemas operativos Windows.

3.6. *Big Data*

Es importante no confundir el *Big Data* con el *Data Warehouse* ya que estos dos términos a menudo se confunden, el Big Data[32, Página 5] es una tendencia que engloba a toda aquella información que es generada por múltiples fuentes, por lo tanto, no tiene una estructura definida ni tampoco puede ser procesada o analizada utilizando procesos o herramientas tradicionales, a diferencia de los *Data Warehouse* que es una arquitectura donde es cierto que existe un gran volumen de información pero no es comparable con el Big Data, además la información almacenada en el *Data Warehouse* tiene una estructura definida y esta relacionada entre si, por lo tanto, existen herramientas que pueden analizar y procesar dicha información.

El *Big Data* en un conjunto de datos de gran volumen, ocupan Petabytes, Exabytes incluso Zettabytes[32, Página 5] de información, todos estos datos exceden a la tecnología y software actual para poder almacenarlos, administrarlos, procesarlos y analizarlos *(encontrar patrones repetitivos, hacer predicciones, toma de decisiones, etc.)* en un tiempo razonable, todos estos datos siguen creciendo de manera exponencial conforme pasan los años *(se estima que cada 40 meses la información se duplica.)*. Todos estos datos son recolectados de múltiples fuentes que son generados directa e indirectamente[32, Página 12]:

- ***Generados por las personas:*** Cada vez son más las personas que hoy en día se encuentra conectada a internet y genera cantidades masivas de información desde enviar un correo, subir un video en YouTube, publicar una imagen en Facebook, mandar un mensaje por WhatsApp, navegar por internet, etc. la mayor parte de estos datos generados por los usuarios es generada desestructuradamente por lo que es difícil procesar este tipo de información.
- ***Transacciones de datos:*** Gracias a la revolución de internet cada vez son más los negocios que venden en línea sus productos y/o servicios, la integración de los bancos con sus plataformas en línea o aplicaciones permiten consultar, hacer transferencias bancarias, pago de servicios, entre otras cosas, hacen que se genere una gran cantidad de información acerca de las transacciones que se llevan a cabo diariamente entre diversos clientes, transformada y analizada dicha información puede ser importante para diversas organizaciones lucrativas.
- ***Marketing y la web:*** Antes de que llegara la web 2.0 las páginas web eran estáticas poco atractivas comparándolas con las existentes hoy en día incluso podríamos llamarlas un poco torpes ya que solo mostraban el contenido que tenían que mostrar y nada mas; hoy en día la web 2.0 rompió esa barrera haciendo más atractivas e intuitivas las páginas que visitamos y por supuesto permitiendo hacer uso de múltiples herramientas y tecnologías que permitan capturar y guardar los sitios que son visitados en internet tanto así que incluso los movimientos del cursor son capturados en mapas para saber que páginas y sitios web visitamos todo esto con el fin de hacer marketing, vendernos productos y/o servicios o simplemente hacernos sugerencias o tener un historial de las páginas que visitamos.
- ***Machine to Machine–(Maquina a Maquina) (M2M):*** Existen múltiples sensores distribuidos por todo el mundo y cada uno de ellos tiene la tarea de recolectar información, estos sensores existen desde hace décadas pero con la llegada de las nuevas tecnologías *(principalmente las inalámbricas)* como el Bluetooth, WI-FI, *Radio Frequency Identification*–(Identificación por Radiofrecuencia) (RFID), infrarrojos, *Near Field Communication*–(Comunicación de Campo Cercano) (NFC), etc. estos se han incrementado ya que los costos de infraestructura se han visto reducido. Estos sensores como por ejemplo sensores de presión, movimiento, luz, temperatura, sonido, etc. tienen la tarea de convertir la información recolectada en datos para poder ser almacenada en una computadora y posteriormente poder analizarla y realizar modelos, simulaciones y o predicciones, pero todo esto es un reto ya que existen montañas de información y para poder llegar a conclusiones o tomar decisiones es necesario realizar millones de cálculos.
- ***Biometría:*** Normalmente estos datos son propiedades del ejército, departamentos de defensa e inteligencia ya que los datos almacenados provienen de estas instituciones y son datos generados por sistema de seguridad, sistemas como biométricos tales como escáneres de retina, lector de huellas digitales incluso lectores de *Deoxyribonucleic Acid*–(Ácido Desoxirribonucleico) (ADN). Todos estos datos son almacenados, procesados y analizados con el fin de crear sistemas más seguros y fiables.

Existen múltiples herramientas para procesar[32, Página 21] la información de los *Big Data* alguna de ellas son: noSQL —Herramienta que no usa el lenguaje SQL como lenguaje principal y los datos almacenados no requieren estructuras—, Hadoop —Es un *Framework* que soporta aplicaciones distribuidas y puede trabajar con miles de nodos y PetaBytes de información—, Cassandra –Es una base de datos noSQL distribuida y permite grandes volúmenes de información de manera distribuida—, MapReduce — Modelo de programación para dar soporte a la programación paralela sobre grandes colecciones de datos— entre otras, estas herramientas pueden procesar datos de alguno de los tres tipos de Big Data:

- ***Datos estructurados:*** Estos *Big Data* tienen bien definida su longitud y su formato por ejemplo las fecha, las cadenas o número de caracteres que contienen, etc. Se almacén en tablas, filas y/o tuplas y pueden ser base de datos relacionales o hojas de cálculo.
- ***Datos no estructurados:*** Son datos originales es decir están contenidos tal y como fueron recolectados normalmente estos datos no se pueden descomponer en tipos de datos básicos o específicos para almacenarlos en tablas, además no tienen un formato especifico es decir pueden ser recolectados en distintos formatos como: PDF, CSV, *Document*–(Documento) (DOC), emails, documentos en texto plano, etc.
- ***Datos semiestructurados:*** contienen diversos tipos de datos con la excepción de que contienen marcadores, etiquetas y/o tags que delimitan la información, incluso estos datos contienen metadatos que describen los objetos y las relaciones entre ellos algunos ejemplos de ellos son XML, HTML, JSON, etc.

Una vez que se han localizado las fuentes de datos que se requieren para realzar un cierto análisis es probable que dispongamos de montañas de información que puede o no estar relacionada entre si, el siguiente paso es almacenarla y darle un mismo formato para que posteriormente se pueda analizar y tomar una decisión, aquí es donde entra la Extracción, Transformación y la Carga (ETL), el objetivo es extraer los datos de las diversas fuentes aplicarles filtros para obtener solo la información necesaria, convertirla a un solo formato y guardarla en una base de datos llamada *Data Warehouse*.

3.7. *Data Warehousing*

Para que el BI sea una realidad deben existir mecanismos y herramientas encargadas de gestionar la información, dicha información esta contenida en diversos formatos, tipos, fuentes y en un volumen considerable que las herramientas tradicionales no serían capaces de procesar. Estas herramientas deben ser capaces de depurar e integrar dichos datos para finalmente almacenarlos en un solo lugar *(Data Warehouse)*, para posteriormente analizarla, explotarla y darle un uso práctico a dicha información. Este proceso se denomina *Data Warehousing*[5, Capítulo 2] o por sus siglas en inglés *DWH*.

Según la metodología de Hefesto[6] el *Data Warehousing* (DWH) es el encargado de extraer, transformar, consolidar, integrar y centralizar los datos generados por toda la estructura de una organización en sus actividades diarias, así mismo la información externa relacionada con la organización. Permitiendo así tomar decisiones gracias a la información que el proceso del DWH ha recopilado y procesado.

El DWH además de permitir la integración de la información también realiza la homogeneización de los datos con el fin de ir estandarizando los datos que serán guardados en el DW, la información que ha sido transformada y sumarizada proveerá ayuda para la toma de decisiones estratégicas y tácticas ya que el DWH convertirá los datos de la organización en una herramienta que pondrá a disposición los datos de manera integrada y estandarizada para cuando estos sean requeridos.

Pare que el DWH cumpla con su objetivo exitosamente debe de almacenar la información procesada en una base de datos centralizada con una estructura multidimensional denominada *Data Warehouse* (DW).

Cabe recalcar que el DWH no se compone solo de datos o de un deposito de datos aislado. El DWH es un conjunto de herramientas que permiten consultar, analizar y presentar información mediante herramientas de extracción y explotación de los datos con el fin de realizar análisis y reportes, mediante la transformación de dichos datos en información valiosa para la organización.

El DWH hace uso de múltiples tecnologías que permiten realizar su proceso:

- Arquitectura cliente-servidor[33].
- Técnicas avanzadas para replicar, refrescar y actualizar datos.
- Software *front-end*, para acceso y análisis de datos.
- Herramientas para Extraer, Transformar y Cargar datos Procesos ETL en el DW, desde múltiples fuentes muy heterogéneas.
- Sistema de Gestión de Base de Datos (SGBD[23, Página 22]).
- Las técnicas utilizadas y mencionadas anteriormente, son imposibles de implementar en un único ambiente operacional, por lo tanto, esta es la razón de ser del *Data Warehousing*.

3.7.1. Ventajas

[5, Página 14]

- Realiza una conversión de los datos orientado a las aplicaciones en información orientada a la toma de decisiones.
- Integra y consolida la información proveniente de diversas fuentes *(externas y/o internas)* de los diferentes bloques que componen la estructura organizacional de la empresa que anteriormente se encontraba aislada, concentrándola en un único almacén de datos sólido, estandarizado y centralizado.
- Permite analizar y explotar la información de una o más áreas especificas de la organización y de una manera inmediata.
- Permite reaccionar rápidamente a los cambios de mercado, por lo tanto, aumenta la competitividad.
- Aumenta la eficiencia del personal encargado de tomar decisiones al tener centralizada y estandarizada la información.
- La calidad de la información aumenta, ya que se encuentra más completa, estandarizada, consistente, oportuna, accesible y en formatos apropiados.
- Elimina el procesamiento y almacenamiento de información innecesario por software de terceros, ya que el DWH integra y procesa dicho información de manera centralizada.
- La información se encuentra almacenada de manera lógica en servidores de almacenamiento por lo tanto aumenta la seguridad de dicha información, al almacenar la información lógicamente los diferentes usuarios pueden acceder a dicha información en línea paralelamente desde una estación de trabajo, así mismo, hacer uso de herramientas para evaluar y manipular la información tales como: hojas de cálculo, procesadores de texto, herramientas de análisis de datos, herramientas de análisis estadísticos, reportes, tableros, etc.
- Aprovechar el valor potencial de sus recursos de información transformándolo en valor verdadero, optimizar el tiempo al obtener resultados evitando la información incorrecta, inconsistente y/o inexistente permitiendo así adquirir una mayor confianza en los resultados arrojados y obtener así una menor incertidumbre en la toma de decisiones.
- Permite la toma de decisiones estratégicas y tácticas.

3.7.2. Desventajas

[5, Página 15]

- Requiere de una inversión elevada ya que su construcción es una tarea compleja, por lo tanto, es necesario personal especializado en el área, requiere de un análisis exhaustivo, muchos recurso y la implementación implica la adquisición de herramientas de consulta y análisis y por su puesto la capacitación del personal que administrara y hará uso de la herramienta. Por lo general las empresas desarrolladas son las que implementan este tipo de mecanismos ya que las medianas y pequeñas empresas no pueden solventar estos gastos y optan por utilizar las herramientas tradicionales.
- Existe una resistencia al cambio por parte de los usuarios ya que muchas veces suelen subestimarse las capacidades y resultados que puede arrojar el correcto uso de las herramientas como el DWH que forman parte de BI en general.
- Los beneficios de los almacenes de datos no son apreciados de inmediato por los usuarios es decir, los beneficios son apreciados a mediano y largo plazo al comprobar la efectividad de dicha herramienta, cabe mencionar que su correcto funcionamiento surge mediante la practica y la experiencia al utilizar estos almacenes de datos.
- Si las fuentes que son utilizadas para alimentar estos almacenes de datos contiene información como clientes, proveedores, empleados, etc, se estará atentando contra la privacidad de dichos datos ya que el personal que tenga acceso a estos almacenes de datos tendrá acceso a dicha información confidencial.

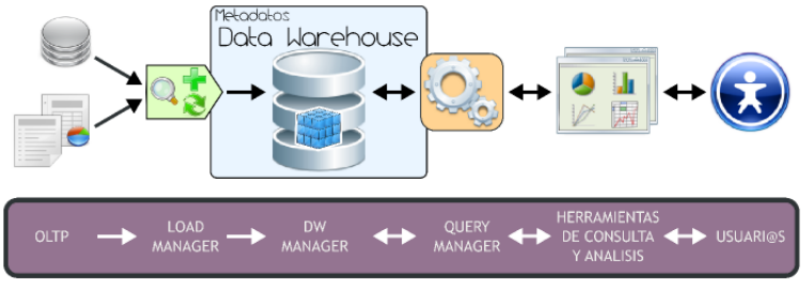


Figura 3.7.1: Arquitectura del DWH (Ing. Bernabeu R. Dario, 2009).

El esquema ***(figura 3.7.1)*** muestra la arquitectura del DWH y deja ver que esta compuesto por múltiples componentes que interactúan entre si para llegar a un objetivo especifico dentro de un sistema. A continuación se describe cada uno de los componentes que conforman este esquema:

- La primera fase consiste en extraer la información desde diversas fuentes tales como: bases de datos, aplicaciones, archivos etc. esta información esta contenida en diversos sistemas, orígenes y arquitecturas por lo general dicha información se encuentra aislada, semi-estructurada y/o desestructurada y en formatos muy variados.
- La segunda fase consiste en integrar, transformar y limpiar los datos para luego ser almacenados en el DW. Para este proceso se utilizan múltiples algoritmos de uso comercial o algoritmos propios que permitan realizar dicho proceso.

- La información en el DW esta almacenada baja un esquema especifico, una de las estructuras más utilizadas son los cubos multidimensionales ya que estos pre-procesan la información para responder consultas dinámicas con un alto desempeño.
- Por ultimo los usuarios acceden a los cubos multidimensionales para obtener información valiosa mediante herramientas de consulta, análisis, reportes, explotación de los datos, etc.

A continuación se describirá de manera más detallada los componentes que conforman parte de la arquitectura del DWH según[5]:

3.7.3. OLTP *(On Line Transaction Processing)*

Como ya se menciono anteriormente *OnLine Transaction Processing*–(Procesamiento de Transacciones en Linea) (OLTP)[5, Página 20] representa la información que es generada por las diversas áreas que conforman la organización o empresa en su operación diaria *(ordenes de compra, reportes, facturas, vales de almacén, etc.)*, esto engloba información de fuentes internas como información de fuentes externas.

La fuentes de información por lo general no están relacionadas entre si, es decir, se encuentran aisladas una de otra, por lo tanto la información se encuentra en formatos y extensiones muy variantes.

Entre los OLTP más comunes que se pueden encontrar en una organización son los siguientes:

- Archivos de textos.
- Hipertextos.
- Hojas de cálculos.
- Informes semanales, mensuales, anuales, etc.
- Bases de datos transaccionales.
- Archivos PDF.
- Entre otros.

3.7.4. *Load Manager*

[5, Página 21] Para poder extraer los datos desde los OLTP para posteriormente manipularlos, integrarlos, transformarlos y finalmente persistirlos en un DW debe de existir un procesos que se encargue de realizar dichas tareas.

La integración será el proceso que se encargara de gestionar los datos esto incluye desde tomar los datos desde los OLTP hasta cargar los datos procesados en DW. El proceso ETL *(Extracción, Transformación y carga)* ***(figura 3.7.2)*** es una de las principales y más utilizadas técnicas para la integración de los datos aunque existen más que permiten realizar este proceso. Este proceso consiste en tres fases, en la fase de *Extracción* existirá un grupo de técnicas enfocadas a tomar los datos indicados de los OLTP y almacenarlos en algún mecanismo intermedio *(por ejemplo la memoria RAM)*, el proceso de *Transformación* consiste en tomar los datos almacenados en los mecanismos intermedios y mediante un grupo de técnicas se analizaran para verificar que sean correctos y validos, por ultimo en la fase de *Carga* se utilizan técnicas para persistir y/o actualizar los datos en el DW.

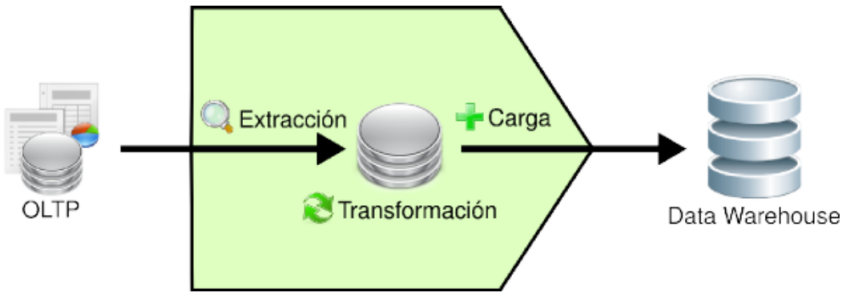


Figura 3.7.2: Proceso ETL (Ing. Bernabeu R. Dario, 2009).

A continuación se detalla cada una de las fases que compone el proceso ETL[5, Página 26]:

3.7.4.1. Extracción

En este proceso se explora y se obtienen los datos que son relevantes de acuerdo a las necesidades y requisitos de los usuarios de los diversos OLTP que se tienen a disposición.

La extracción puede ser tan sencilla como extraer información de bases de datos relacionales mediante consultas SQL o rutinas programadas, en cambio si los OLTP se encuentra en fuentes externas y/o internas no tradicionales tales como archivos de texto, hojas de cálculo, archivos PDF, etc. la extracción se complica ya que la información se encuentra en diversos formatos, por lo tanto, se deben emplear herramientas para volcar la información y poder extraerla.

Una vez que los datos son seleccionados y extraídos de las diversas fuentes se almacenan en mecanismos intermedios tales como: la memoria RAM, el disco duro mediante una base de datos temporal que contenga tablas auxiliares y/o tablas temporales *(para su posterior transformación)* lo cual permitirá:

- Seguir manipulando los datos sin interrumpir ni paralizar los OLTP o el DW.
- No depender de la disponibilidad de los OLTP.
- Almacenar y gestionar los metadatos que se generarán en los procesos ETL.
- Facilitar la integración de las diversas fuentes, internas y externas.

3.7.4.2. Transformación

Esta proceso toma los datos contenido en las fuentes intermedias, los analiza y convierte o transforma aquellos datos incompatible e incongruentes en un conjunto de datos compatibles y congruentes antes de que sean cargados al DW. Este proceso se lleva a cabo por la simple razón de que al tener múltiples fuentes de datos estos pueden ser extraídos en formatos diferentes, la tarea del proceso de Transformación es estandarizar los datos, es decir, conciliar un formato y forma única antes de ser persistidos en el DW. Cabe mencionar que este proceso también es el encargado de realizar la limpieza de los datos *(Data Cleansing)* y validar su calidad.

Algunos de los casos más comunes donde se debe de integrar la información son los siguientes:

- Codificación.
- Medida de atributos.
- Convenciones de nombramiento.
- Fuentes múltiples.

3.7.4.2.1. Codificación Uno inconsistencia muy típica cuando se obtienen datos de múltiples fuentes y estos se intentan integrar es contar con más de una manera de representar los datos de un atributo *(Masculino, Hombre, H, Male, M, etc.)*. La solución a esta inconsistencia es definir un único valor para representar el dato del atributo de esta manera cuando la información llegue al DW estará integrada de manera uniforme.

3.7.4.2.2. Medida de atributos Las unidades de medida *(litros, mililitros, yardas, metros, kilómetros, etc.)* para representar los atributos de una entidad son muy variados y aun más cuando la información es extraída de múltiples fuentes a través de los diferentes OLTP. Se deben de usar mecanismos que estandaricen las medidas de dichos atributos inconsistentes para que todas las fuentes de datos expresen sus valores de igual medida. A menudo suele pasar que los algoritmos implementados para resolver este tipo de inconsistencias suele ser los más complejos.

3.7.4.2.3. Convenciones de nombramiento A menudo un atributo suele llamarse de diferentes maneras y tener el mismo significado a través de las diversas fuente de de datos *(nombre del proveedor, razón social, proveedor, etc)*, se debe realizar una conversión de nombramiento la cual sea la más significativa para los usuarios.

3.7.4.2.4. Fuentes múltiples Las diversas fuentes de datos pueden compartir uno o más atributos en común, en este caso se debe elegir la fuentes que se considere más fiable y apropiada para no tener inconsistencias o perdida de información

3.7.4.2.5. Limpieza de datos El objetivo de la limpieza de datos es aplicar distintos tipos de acciones, filtros, estadísticas, algoritmos, etc. contra el mayor número de datos erróneos, inconsistentes o irrelevantes todo esto con el fin de recuperar o ignorar datos perdidos y/o inconsistentes.

Las acciones más comunes que se pueden realizar al encontrar valores inconsistentes *(Outliers)* son:

- Ignorarlos.
- Eliminar o ignorar la columna.
- Filtrar la columna.
- Filtrar la fila errónea, ya que a veces su origen, se debe a casos especiales.
- Reemplazar el valor.
- Discretizar los valores de las columnas.

Las acciones que suele aplicarse al identificar valores faltantes *(Missing Values)* son:

- Ignorarlos.
- Eliminar o ignorar la columna.
- Filtrar la columna.
- Filtrar la fila errónea, ya que a veces su origen, se debe a casos especiales.
- Reemplazar el valor.
- Esperar hasta que los datos faltantes estén disponibles.

Es muy importante que al identificar valores inconsistente o faltantes se rastree la anomalía de inmediato para identificar el porque del problema y así evitar las inconsistencias o valores faltantes de la información cuando la acción se repita de esta manera se agrega un valor mayor a los datos de la organización.

3.7.4.3. Carga

Este proceso esta dividido en dos sub-procesos con tareas especificas cada uno.

El primer sub-proceso es el encargado de realizar tareas tales como:

- ***Carga inicial (Initial Load):*** como su nombre lo indica este sub-proceso se encarga de alimentar el DW por primera vez, por lo general esta tarea consume mucho tiempo y recursos de cómputo ya que se deben de insertar un número considerable de registros, a menudo se cargan los registros contenidos en las fuentes de datos con una fecha no mayor a cinco años aunque esto puede variar dependiendo de los requerimientos del cliente.
- ***Actualización y mantenimiento periódico:*** aquí se mueven pequeñas cantidades de datos periódicamente, el intervalo de tiempo se encuentra establecido por los requerimientos del usuario o por el crecimiento de las fuentes de datos. El objetivo es ir agregando los nuevos datos que se han generado en las fuentes de datos desde la ultima actualización *(refresh)*.

  Es importante que antes de realizar una actualización en los registros del DW es necesario identificar si las fuentes de datos originales no han sufrido cambios y/o modificaciones desde la fecha del ultimo mantenimiento, con el fin de no atentar contra la consistencia de los registros del DW. Para llevar a cabo esta operación se pueden realizar las siguientes acciones:

  - Cotejar las instancias de los OLTP involucrados.
  - Utilizar disparadores en los OLTP.
  - Recurrir a Marcas de Tiempo *(Time Stamp)*, en los registros de los OLTP.
  - Comparar los datos existentes en los dos ambientes (OLTP y DW).
  - Hacer uso de técnicas mixtas.

  Si este proceso consumo mucho tiempo, demasiados recursos de hardware para procesar la información o simplemente no puede realizarse la acción por algún motivo cabe la posibilidad de eliminar toda la información del DW y volver a realizar la carga desde cero, este proceso se denomina Carga Total *(Full Load)*.

Los datos que ingresen al DW para su carga y/o actualización deben ser:

- Aquellos datos que han sido transformados y que residen en el almacenamiento intermedio.
- Aquellos datos de los OLTP que tienen correspondencia directa con el depósito de datos.

Se debe tomar en cuenta que los datos que serán movidos al DW deben ser analizados y procesados y por ninguna circunstancia se debe evitar este paso ya que este mismo asegura la calidad e integridad de los datos.

El segundo sub-proceso tiene como objetivo de mantener la estructura interna del DW y tratar temas relacionados tales como:

- Relaciones muchos a muchos.
- Claves Subrogadas.
- Dimensiones Lentamente Cambiantes.
- Dimensiones Degeneradas.

3.7.5. *Data Warehouse Manager*

[5, Página 27] También conocidos como almacén de datos, son una colección de datos de una empresa, una organización, una institución, etc. que sirve como historial de la misma, esta información debe perdurar es decir debe ser no volátil deben existir mecanismos que permitan almacenarla por un tiempo indefinido, esta puede ser variable con el tiempo. Esta información por lo regular no es usadas para operaciones o transacciones de una empresa u instituciones sino que más bien es procesada para la toma de decisiones.

Es importante recalcar que estos almacenes de datos deben contener información histórica por lo tanto no deben ser procesados con información actual ya de lo contrario no dará los resultados esperados. Estos almacenes de datos son por lo regular de solo lectura y sirven para realizar ofertas, estudios de mercado, promociones, comportamiento, etc. Por lo tanto deben existir herramientas que permitan extraer, transformar y cargar datos, herramientas para el análisis y herramientas para gestionar y recuperar los datos.

En un *Data Warehouse* existen información tanto útil como inútil para los usuarios es por eso que se deben de crear o utilizar herramientas y/o mecanismos que permitan analizar dicha información y entregarla al usuario como información útil que le permita tomar decisiones. Hay que tener dos conceptos importantes cuando se decida crear un almacén de datos:

- ***Integración:*** Se debe de integrar los datos obtenidos por las diferentes fuentes normalmente las bases de datos y sistemas distribuidas de cierta organización, esta información proveniente de diversas fuentes tendrán estructuras variables, esta información debe almacenase en el banco de datos para su explotación posteriormente.
- ***Separación:*** Ambas bases de datos deben estar separadas una de la otra es decir la base de datos que es ocupada por la organización para realizar operaciones y transacciones debe ser independiente a la base de datos que es utilizada para almacenar el banco de datos ya que de no ser así el rendimiento tanto de una o de otra se vería reducido.

Una de las partes importantes y fundamentales en los almacenes de datos son los metadatos *(datos acerca de los datos)* aquí se describe la estructura de los datos que se van a almacenar, la relación que existe entre ellos, la estructura de las tablas que almacenan los datos, tablas que contiene el almacén de datos, los tipos de datos de las tablas, etc. Estos metadatos son importantes para los usuarios o las aplicaciones que analizaran la información para tener una referencia de las relaciones, tablas, e jerarquía de los datos contenidos en el almacén de datos.

Los procesos ETL *(Extracción, Transformación y Carga)* son conceptos importantes ya que son los mecanismos que un almacén de datos utiliza para guardar y procesar información.

- ***Extracción:*** Es el mecanismo de obtener la información necesaria a partir de los datos almacenados o de los diferentes sistemas que contienen información de una empresa o corporación para posteriormente transformarlos y almacenarlos en un *Data Warehouse*.
- ***Transformación:*** Los datos que vas a ser resguardados en el un almacén de datos pueden venir de diversas fuentes y con ello pueden venir estructurados de diversa manera es por eso que antes de ser almacenados en la bases de datos deberán ser transformados a un formato especifico.
- ***Carga:*** Una vez que los datos son extraídos y transformados ahora si ya pueden ser almacenados en el *Data Warehouse* para posteriormente poder utilizarlos para la toma de decisiones.

3.7.5.1. Tabla de hechos

Normalmente una tabla de hechos[5, Página 30] guarda información numérica de eventos o sucesos, estos datos numéricos pueden ser claves foráneas que hacen referencia a las tablas de dimensiones que guardan la descripción de dichos eventos. Las tablas de hechos generalmente contienen una *Surrogate key* (clave sustituta) para identificar cada una de las filas que componen la tabla de hechos de manera irrevocable, también las tablas de hechos guardan información a bajo nivel y estas se definen en tres tipos:

- ***Tabla de hechos transaccionales:*** registran hechos relativos a eventos específicos.
- ***Tablas de hechos Snapshot:*** registran hechos en un punto dado en el tiempo con un intervalo de tiempo específico para realizar la medición sobre la entidad de un objeto en observación.
- ***Tablas Snapshot acumulativas:*** registran hechos de un objeto en específico acumulados en un punto dado en el tiempo.

3.7.5.1.1. Tablas de hechos agregadas y pre-agregadas La *tablas de hechos agregadas y pre-agregadas*[5, Página 32] son utilizadas para almacenar información resumida *(anual, mensual, quincenal, etc.)*, es decir a un nivel superior de granularidad de la que los datos originales en el DW fueron almacenados. Para poder almacenar estos datos se deben de establecer los criterios necesarios para realizar los resúmenes y así alimentar las tablas de hechos.

Cuando se requiere que los datos de una consulta sean representados en un nivel superior de granularidad de lo que se encuentran almacenados en el DW, se debe realizar un proceso para cumplir con ciertos criterios, este procesos se denomina *agregación*.

Las *tablas de hechos pre-agregadas y agregadas* tienen un objetivo similar, pero su forma de operar es diferente:

- ***Tablas de hechos agregadas:*** Esta tabla se genera al procesar una consulta con ciertos criterios establecidos aplicados a una tabla de hechos que se desea resumir. Por lo general estas tablas se generan utilizando consultas en lenguaje SQL que resumen o sumarizan información proveniente de una o más tablas de hechos de acuerdo a criterios establecidos en dicha consulta.
- ***Tablas de hechos pre-agregadas:*** Esta tabla se genera antes de que se procese una consulta sobre una tabla de hechos que será resumida, de esta manera cuando se ejecute una consulta de sumarización esta se realizara sobre una tabla que ya ha sido resumida previamente. Por lo general las sumarizaciones previas se realizan mediante procesos ETL, las tablas pre-agregadas aportan ciertos beneficios entre los cuales se encuentran:
  - Se reduce el uso de recursos de hardware que son utilizados para realizar cálculos de sumarizaciones.
  - El número de registros que serán analizados se reduce gracias a las tablas pre-agregadas que han utilizado sumarizaciones previas para su construcción.
  - Gracias a que los datos se encuentra resumidos previamente por un proceso, al generar consultas sobre estas tablas el tiempo de respuesta tiene un mayor rendimiento.

  Las *tablas de hechos pre-agregadas* son muy útiles en los siguientes casos:
  - Cuando los datos a nivel detalle (menor nivel granular) son innecesarios y/o no son requeridos.
  - Cuando una consulta sumarizada a determinado nivel de granularidad es solicitado con mucha frecuencia.
  - Cuando los datos son muy abundantes, y las consultas demoran en ser procesadas demasiado tiempo.

  Una de las desventajas de este tipo de tablas son las siguientes:
  - Requieren que se mantengan y gestionen nuevos procesos ETL para alimentar estas tablas.
  - Demandan espacio de almacenamiento extra en el depósito de datos *(hoy en día el almacenamiento ya no se toma en cuenta como una desventaja debido a los medias de almacenamiento masivo de gran capacidad que encontramos en el mercado)*.

3.7.5.2. Tabla de dimensiones

Las tablas de dimensiones[5, Página 28] normalmente contienen un bajo número de registros ya que estas tablas no almacenan eventos sino estas almacenan la descripción de dichos eventos que son registrados en la tabla de hechos, estas tablas no aumentan su tamaño rápidamente en comparación con las tablas de hechos y aunque pueden contener un número bajo de registros cada uno de los registros puede contener múltiples atributos que sirven para describir a los datos contenidos en la tabla de hechos. Algunas tablas de dimensiones muy comunes son:

- ***Tablas de tiempo:*** describen el tiempo al más pequeño nivel de granularidad para el cual los eventos se registran en la tabla de hechos.
- ***Tablas de dimensiones geográficas:*** describen datos de localización.
- ***Tablas de dimensiones de productos:*** describen un amplio catálogo de productos.
- ***Tabla de dimensiones de empleados:*** describen detalladamente información de los empleados en una sucursal, empresa, organización, etc.

- ***Tabla de dimensiones de rangos:*** describen rangos de tiempo, valores de dólar, u otras cualidades medibles para simplificar los reportes.

3.7.5.2.1. Tabla de dimensión de tiempo La *tabla de dimensión de tiempo* es obligatoria en la construcción de un DW ya que esta contendrá una especie de sello de tiempo que determinara la ocurrencia de un hecho representando cambios, actualizaciones y/o carga de información en diferentes versiones. La *tabla de dimensión de tiempo* no solo contiene una secuencia cronológica de fechas numéricas, sino que también mantiene los niveles jerárquicos especiales que inciden en las actividades de la organización.

Aunque existen diversas maneras de crear esta tabla no es una tarea fácil ya que se debe evaluar detalladamente la información y tomar en cuenta varios factores como: la temporalidad de los datos, la forma de operar de la organización, los resultados esperados del almacén de datos relacionados con las unidades de tiempo y la flexibilidad que se desea obtener de dicha tabla, así mismos si se requiere de una análisis de datos por fecha y por hora, es recomendable construir dos *tablas de dimensiones de tiempo* una que contenga solamente fechas *(año, mes y día)* y otra que almacene las horas *(horas, minutos y segundos)* con el fin de tener una mayor organización y una mayor flexibilidad de la información.

Es cierto que el lenguaje SQL permite realizar operaciones con fechas *(tipo DATE)*, pero en la *tabla de dimensión de tiempo* se modelan y representan datos temporales los cuales no se pueden calcular mediante en lenguaje SQL o los cálculos se vuelven más complejos de lo común, es por eso que esta tabla de dimensión aporta una ventaja significativa.

Es conveniente mantener en la *tabla de dimensión de tiempo* un campo que se refiera al día Juliano. El día juliano se representa a través de un número secuencial e identifica unívocamente cada día. Mantener este campo permitirá la posibilidad de realizar consultas que involucren condiciones de filtrado de fechas desde-hasta, mayor que, menor que, etc, de manera sencilla e intuitiva.

3.7.5.3. *Data Marts*

Los *Data Marts*[5, Página 73] también conocidos como *mercados de datos* pueden ser construidos como un subconjunto de datos contenidos en un *Data Warehouse* o construirlos independientemente con el fin de fraccionar la información en áreas especificas del negocio o giro que se desee analizar, el *Data Warehouse* puede ser separado en bloques que pueden ser procesado individualmente y obtener así resultados con una mayor eficiencia y una mayor rapidez y esto mismo sucede con los *Data Marts* independientes, estos engloban una o más áreas de interés del negocio ya que por lo regular tienen un propósito especifico y la información restante que forma parte del negocio o del DW no es de interés o relevante para el área que se esta analizando. Un *Data Mart* cumple los mismos principios que el *Data Warehouse*, construir un repositorio de datos único, consistente, fiable y de fácil acceso.

Los *Data Marts* que son construidos individualmente son una buena opción para empresas pequeñas o medianas que no pueden solventar la construcción de una *Data Warehouse* con la ventaja de ir construyendo más *Data Marts* e ir haciendo escalable el almacén de datos hasta construir un *Data Warehouse*.

Existen dos tipos de Dara Marts:

- ***Data Marts dependientes (figura 3.7.3):*** Los datos utilizados para alimentar este *Data Mart* proviene del *Data Warehouse*. Esta estrategia es apropiada o una buena opción cuando el *Data Warehouse* crece a pasos

agigantados y áreas especificas del negocio requieren realizar análisis sobre una pequeña porción de los datos contenidos en el DW.

- ***Data Marts independientes:*** Los datos utilizados para alimentar este *Data Mart* provienen de las fuentes externas y los sistemas que operan en el negocio. Esta estrategia es una buena opción para pequeñas y medianas empresas.

  Una de las desventajas aparentes al construir *Data Marts* independientes es obtener un resultado no deseado de integración ya que uno o múltiples *Data Mart* puedes no estar integrados, por lo tanto, la administración y mantenimiento de los *Data Marts* puede ser un problema.

  Otro inconveniente que puede surgir es que los *Data Marts* compartan los mismos datos para alimentarse y dar respuesta a determinadas preguntas, por lo tanto, habría información duplicada ya que cada *Data Mart* implementaría su propio proceso de Extracción, Transformación y Carga (ETL).

Algunos de los inconvenientes de los *Data Mart* y los *Data Warehouse* es el mantenimiento ya que los datos no son estáticos, por ende los volúmenes de información pueden ir creciendo, por lo tanto, el almacenamiento debe ser mayor con forme transcurra el tiempo, otro inconveniente es la información esta puede quedar obsoleta debido a la tendencia al cambio en una línea de tiempo reducida, entre otros.

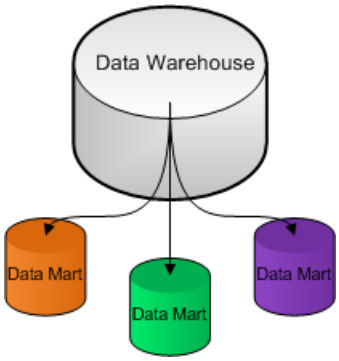


Figura 3.7.3: *Data Marts* dentro de un *Data Warehouse* (Ana Buigues, 2010).

3.7.5.4. Tipos de modelado de un *Data Warehouse*

3.7.5.4.1. Esquema en estrella Este esquema ***(figura 3.7.4)***[5, Página 37] es usado en el *Data Warehouse* aunque también es común ver este tipo de esquemas en las bases de datos relacionales que almacenan información de una empresa u organización para su operabilidad. Este esquema contiene una tabla central llamada "*tabla de hechos o tabla fact*" esta tabla a su vez contiene tablas más pequeñas que la rodean llamadas "*tablas de dimensiones*" asemejando esta estructura a una estrella. Una tabla de hechos contiende datos medibles y cuantitativos mientras que una tabla de dimensiones contiene atributos que describen la información contenida en la tabla de hechos.

En este tipo de esquemas se hace uso de las llaves primarias *(Primary keys)*[4, Página 14] y llaves foráneas *(Foreign Keys)*[4, Página 14] para relacionar la información de las diversas tablas que conforman el *Data Warehouse*, una tabla de dimensiones siempre contendrá una llave primaria y un gran número de atributos mientras que una tabla de hechos contendrá múltiples llaves foráneas de los datos que componen las múltiples tablas de dimensiones, de esta manera la tabla de hechos puede relacionarse con las múltiples tablas de dimensiones que la rodean y de la misma

manera una tabla de dimensiones puede relacionarse con la tabla de estados y con otras tablas de dimensiones haciendo uso de la tabla de hechos como conexión o puente entre ambas.

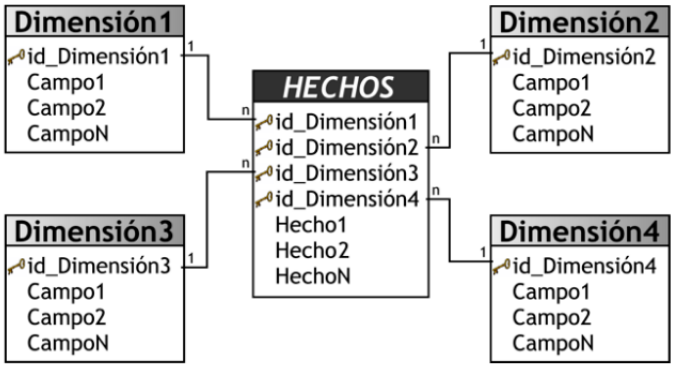


Figura 3.7.4: Modelado de un DW bajo un esquema en estrella (Ing. Bernabeu R. Dario, 2009).

Este esquema es fácil de implementar, además de que el rendimiento y la velocidad de respuesta es mejor en comparación con otros esquemas, es ideal para realizar análisis de grandes cantidades de registros ya que las consultas no son complicadas, ya que las condiciones y las uniones *(join)* entre las diversas tablas solo involucran la tabla de hechos y las tablas de dimensiones permitiendo trabajar en dos niveles disminuyendo así el nivel de complejidad.

3.7.5.4.2. Esquema copo de nieve Este esquema ***(figura 3.7.5)***[5, Página 39] está enfocado en la normalización de las tablas para reducir el espacio de almacenamiento y no tener redundancia de datos, este esquema tiene mayor complejidad que el esquema de estrella, en este esquema existe una tabla central de hechos donde se almacenan los eventos o los sucesos que se desean analizar y las tablas de dimensiones que contienen la descripción de los eventos almacenados en la tabla de hechos pero con el único detalle de que las tablas de dimensiones tiene tablas ligadas *(visto de otra manera una tabla de dimensiones puede tener una o más tablas de dimensiones)*, esto se hace con el fin de descomponer la información *(granularidad)* y almacenarla estructuradamente de mejor manera evitando la redundancia de datos, la desventaja de este esquema es que los niveles de información se vuelven más complejos y para realizar consultas complejas se tiene que hacer uso de múltiples niveles de tablas afectado el rendimiento tardando más la ejecución y la respuesta de las consultas.

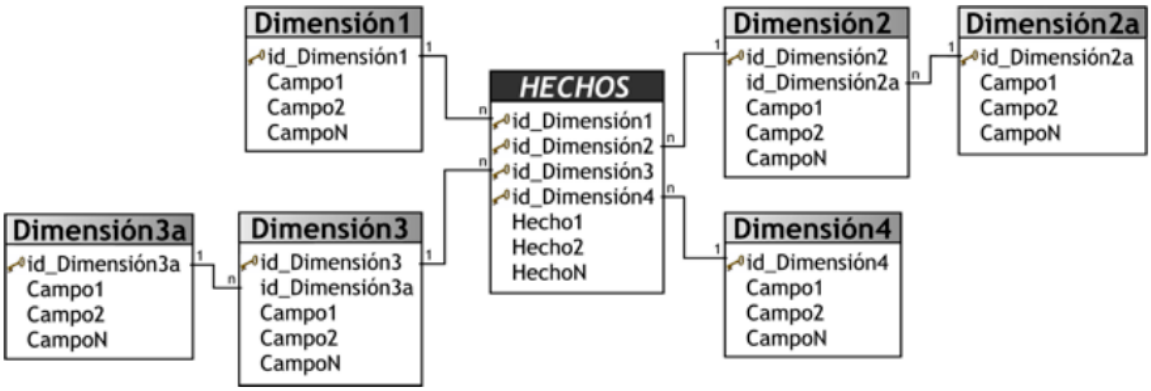


Figura 3.7.5: Modelado de un DW bajo un esquema copo de nieve (Ing. Bernabeu R. Dario, 2009).

Hoy en día el almacenamiento ya no es un problema pero el rendimiento sí, hay que tener en cuenta este esquema ya que la ventaja de este es la normalización de las tablas y con ello reducir el almacenamiento de la información, implementar un esquema en copo de nieve implica realizar líneas de código más complejas para realizar consultas y estas pueden llevarse más tiempo en obtener la información y mostrar el resultado deseado ya que se tendrán que hacer más uniones *(joins)* entre los diversos niveles que conforman la base de datos.

Se puede implementar este esquema en un *Data Warehouse* grande y complejo donde el tiempo de respuesta no sea crucial; pero implementar un esquema de este tipo donde el tiempo de respuesta sea crucial será deficiente por lo tanto es mejor implementar un esquema de estrella, de esta manera el *Data Warehouse* entregara resultados más rápidamente, aunque el espacio de almacenamiento será mayor.

3.7.5.4.3. Esquema constelación Este tipo de esquema ***(figura 3.7.6)***[5, Página 40][5, Página 39] es una combinación entre los esquemas en estrella y los esquemas en copo de nieve, por lo tanto, en este tipo de esquema existe una tabla de hechos principal y se puede hacer uso de tablas de hechos auxiliares de modo que las tablas de hechos pueden contener sus propias tablas de dimensiones e incluso pueden compartir dichas tablas de dimensiones entre las tablas de hechos. Las tablas compuestas por la tabla de hechos principal y las tablas de hechos auxiliares se encuentran en el centro del modelo y las tablas de dimensiones se propagan a cualquier dirección sobre el modelo formando así un esquema en constelación.

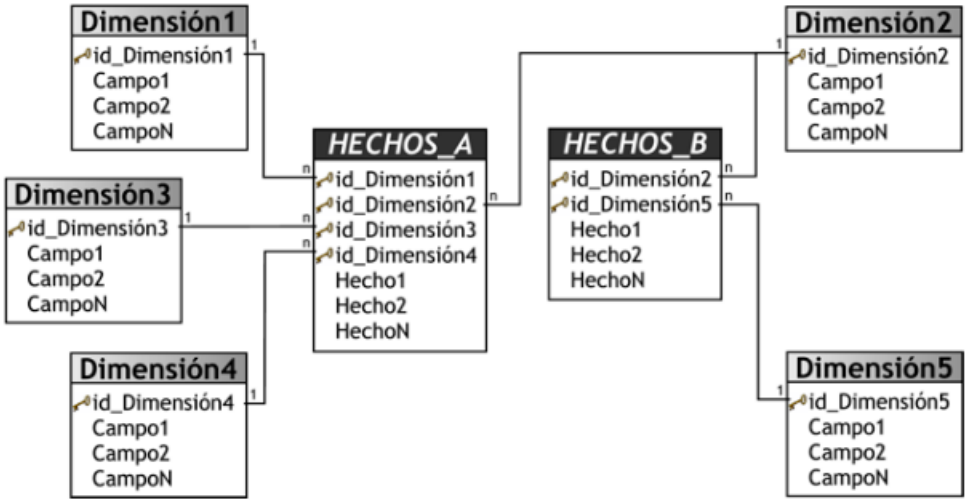


Figura 3.7.6: Modelado de un DW bajo un esquema constelación (Ing. Bernabeu R. Dario, 2009).

El diseño es muy similar al esquema en estrella aunque con algunas cualidades y desventajas extras entre las cuales se encuentran:

- Al tener más de una tabla de hechos se obtiene la ventaja de poder analizar más datos del negocio sin realizar cambios complejos al diseño del modelo del *Data Warehouse*.
- Se optimiza la reutilización de las tablas de dimensiones, ya que pueden ser reutilizadas por las diversas tablas de hechos que componen el modelo en constelación.
- Una de las desventajas más evidentes son el uso de herramientas para explotar la información ya que no todas soportan este tipo de modelo, por lo tanto, se deben de emplear diferentes mecanismos que permitan realizar dichas actividades.

3.7.5.5. Implementación de los esquemas

Los esquemas mencionados anteriormente (esquema estrella, copo de nieve y constelación) pueden ser implementados de diversas maneras, que independientemente del tipo de arquitectura se necesita que la información este desnormalizada o simi-desnormalizada todo esto con el fin de evitar realizar complejos cruces de información *(joins)* entre las diversas tablas de hechos y dimensiones que componen el DW para poder accesar a la información y agilizar la ejecución de las consultas.

3.7.5.5.1. Relacional (ROLAP) Este tipo de organización lógico de información se implementa sobre la tecnología relacional con el plus de integrar algunas facilidades para optimizar el rendimiento.

Relational Online Analytical Processing–(Procesamiento Analítico en Linea Relacional) (ROLAP) cuenta con todos los beneficios de un Sistema Gestor de Base de Datos (SGBD[23, Página 22]) relacional el cual provee herramientas para poder utilizarlo como un Sistema Gestor de un DW.

Este tipo de organización facilita la generación dinámica de los cubos multidimensionales al realizar consultas para la extracción de la información, haciendo el manejo de los cubos multidimensionales transparente para el usuario*(s)*. Este proceso se puede resumir a través de los siguientes pasos:

- El primer paso consiste en seleccionar los indicadores, atributos, jerarquías, etc, que compondrán el cubo multidimensional.
- El segundo paso consiste en ejecutar las consultas sobre los atributos, indicadores, etc, mencionados anteriormente, por lo tanto, de manera transparente para los usuarios se crea y se calcula dinámicamente el cubo multidimensional correspondiente, el cual dará respuesta a las consultas que se ejecuten.

Una de las ventajas evidentes de este tipo de organización es la flexibilidad que brinda al evitar que los usuarios intervengan en la creación y mantenimiento explícito de los cubos ya que estos son generados dinámicamente cuando se establecen los criterios para la extracción de la información y son ejecutadas dichas consultas. Cabe destacar que al almacenar y gestionar los datos contenidos en un DW mediante un SGBD[23, Página 22] no se necesitan de herramientas, programas o software extras para administrar y gestionar los datos de manera multidimensional.

Una de las desventajas de ROLAP es la perdida de eficiencia de rapidez al entregar resultados, esta razón tiene que ver con la creación de los cubos multidimensionales ya que como se menciono anteriormente estos son creados dinámicamente cada que es ejecutada una consulta, por lo tanto, se tienen que calcular dichos datos cada vez que es ejecutada una consulta de acuerdo a las criterios pre-establecidos y esto da lugar a requerir un mayor tiempo de respuesta para aplicar las operaciones correspondientes.

Para disminuir el impacto de esta desventaja en algunos casos se pueden crear mecanismos que almacenen los resultados de los cubos multidimensionales en la memoria caché, optimizando el tiempo de respuesta cuando en un futuro sea ejecutada la misma consulta.

Entre las características más importantes de ROLAP, se encuentran las siguientes:

- Almacena la información en una base de datos relacional.
- Utiliza índices de mapas de bits.
- Utiliza índices de Join.
- Posee optimizadores de consultas.
- Cuenta con extensiones de SQL *(drill-up, drill-down, etc)*.

3.7.5.5.2. Multidimensional (MOLAP) El objetivo de las estructuras *Multidimensional Online Analytical Processing*–(Procesamiento Analítico en Linea Multidimensional) (MOLAP) es almacenar los datos lógicamente en estructuras especificas multidimensionales como *Arrays*, así mismo utilizar técnicas de comparación de datos que mejoren considerablemente el rendimiento de la extracción de la información de estas estructuras multidimensionales.

MOLAP requiere de instancias previas para generar y pre-calcular cubos multidimensionales para que sobre ellos se puedan realizar cálculos. Este proceso se puede resumir a través de los siguientes pasos:

- Se seleccionan los indicadores, atributos, jerarquías, etc., que compondrán el cubo multidimensional.
- Se pre-calculan los datos del cubo.
- Se ejecutan las consultas sobre los datos pre-calculados del cubo.

Una de las principales ventajas al utilizar este tipo de estructuras es el tiempo de respuesta ya que comparado con la estructura anterior (ROLAP) los datos en MOLAP se encuentran pre-calculados por lo tanto se requiere de un menor tiempo de respuesta para obtener resultados cuando se realizan consultas sobre los cubos con resultados pre-calculados, cabe mencionar que existen una serie de desventajas que están directamente relacionadas con la ventaja de realizar pre-cálculos en los cubos multidimensionales las cuales son:

- Cuando un cubo requiere o necesita un cambio, se tiene que re-calcular el cubo totalmente, para que se reflejen las modificaciones llevadas a cabo, provocando una disminución importante en cuanto a flexibilidad.
- Se requiere de más espacio físico para almacenar dichos datos *(hoy en día esta ya no es una desventaja tomando en cuenta los medios de almacenamiento con una alta capacidad)*.
- Se requiere de un software que administre y gestione la extracción de los datos contenidos en los cubos multidimensionales.

3.7.5.5.3. Híbrido (HOLAP) Las estructuras *Hybrid Online Analytical Processing*–(Procesamiento Analítico en Linea Híbrido) (HOLAP) combina las dos estructuras anteriores ROLAP y MOLAP para almacenar datos en un motor relacional y otros en una base de datos multidimensional.

Los datos agregados y pre-calculados se almacenan en estructuras multidimensionales (MOLAP) y los datos de menor nivel de detalle en estructuras relacionales (ROLAP). Es decir, se utilizará ROLAP para navegar y explorar los datos, y se empleará MOLAP para la realización de tableros.

Uno de los desafíos de esta estructura es realizar un buen análisis para identificar los diferentes tipos de datos.

3.7.5.6. Cubo multidimensional

Esta estructura de datos es una de las más utilizadas para representar los datos contenidos en un *Data Warehouse* aunque es una de las más complejas de entender.

Los cubos multidimensionales o hipercubos ***(figura 3.7.7)***, representan y convierten los datos planos es decir los datos que se encuentran en filas y columnas en un nuevo tipo de estructura tal como una matriz de NxN dimensiones.

Los objetos más importantes que se pueden incluir en un cubo multidimensional, son los siguientes:

- ***Indicadores:*** sumarizaciones que se efectúan sobre algún hecho o expresiones basadas en sumarizaciones, pertenecientes a una tabla de hechos ***(figura 3.7.8)***.
- ***Atributos:*** campos o criterios de análisis, pertenecientes a tablas de dimensiones ***(figura 3.7.8)***.
- ***Jerarquías:*** representa una relación lógica entre dos o más atributos ***(figura 3.7.8)***.

De esta manera al transformar los datos en un nuevo tipo de estructura con el fin de extraer resultados de estos hipercubos, los atributos existen a través de varios ejes o dimensiones, y la intersección de las mismas representa el valor que tomará el indicador que se está tomando.

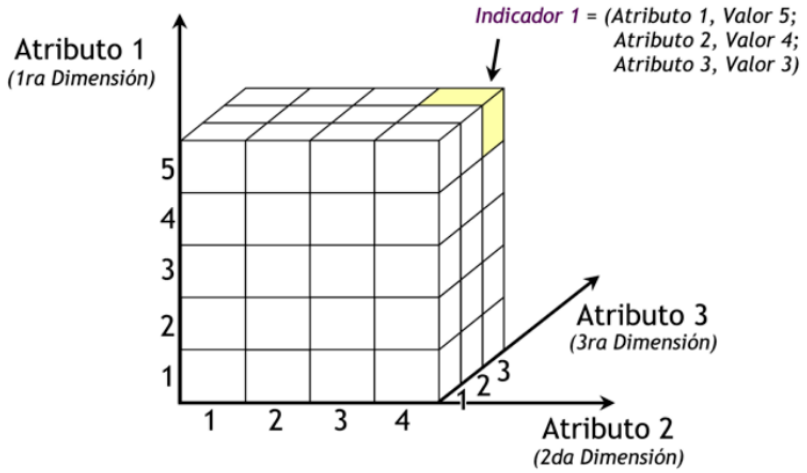


Figura 3.7.7: Hipercubo que representa una estructura de datos (Ing. Bernabeu R. Dario, 2009).

El resultados del análisis esta dado por los cruces matriciales de acuerdo a los valores de las dimensiones seleccionadas. En otras palabras para obtener los datos del DW primero se deben convertir los datos planos *(filas y columnas)* de las tablas de hechos y dimensiones a una nueva estructura de datos en este caso un hipercubo *(matriz NxN)*, ahora sobre los datos que componen el hipercubo se deben ejecutar consultas u operaciones para extraer los resultados requeridos. Ahora dicho cubo debe cumplir con ciertas condiciones para que la información contenida en los cubos multidimensionales pueda ser extraída de manera integra y exitosamente entre los cuales se deben incluir indicadores, atributos, jerarquías, etc. basados en los campos de las tablas de hechos y dimensiones que se deseen analizar. De esta manera las consultas y operaciones que son solicitadas a los cubos multidimensionales son respondidas con un alto rendimiento y en un tiempo menor que si fueran ejecutadas sobre la base de datos transaccional.

3.7.5.6.1. Indicadores Los indicadores ***(figura 3.7.8)*** son sumarizaciones efectuadas sobre un hecho o expresiones, estas sumarizaciones serán incluidas en los cubos multidimensionales con el fin de analizar de una manera más eficiente los datos contenidos en el DW. El valor que estos adopten estarán condicionados por los atributos y jerarquías que han sido utilizadas para analizar dichos datos.

Los indicadores pueden estar compuestos por hechos o indicadores pero no por ambos, en otras palabras se pueden crear operaciones o funciones aritméticas *(suma, resta, promedio, conteo, etc.)*, matemáticas, estadísticas y operadores lógicos.

3.7.5.6.2. Atributos Los atributos ***(figura 3.7.8)*** construyen los criterios de análisis que se utilizaran para analizar los indicadores dentro de un hipercubo, dichos atributos están basados en los campos que componen a las tablas de estados y hechos aunque pueden existir otros atributos que no estén relacionados con las tablas que conforman en DW.

3.7.5.6.3. Jerarquías Un jerarquía ***(figura 3.7.8)*** representa una relación lógica entre dos o más atributos dentro de un hipercubo, siempre y cuando posean su correspondiente relación ”padre-hijo” entre los propios atributos del cubo.

Las jerarquías poseen las siguientes características:

- Pueden existir varias jerarquías en un mismo cubo multidimensional.
- Están compuestas por dos o más niveles de jerarquización.
- Se tiene una relación ”1-n” o ”padre-hijo” entre atributos consecutivos de un nivel superior y uno inferior.

Una de las ventajas que se obtiene al utilizar jerarquías es analizar de manera más eficiente la información ya que esta se descompone de lo más general a lo más detallado y viceversa al desplazarse por los diferentes niveles de jerarquización.

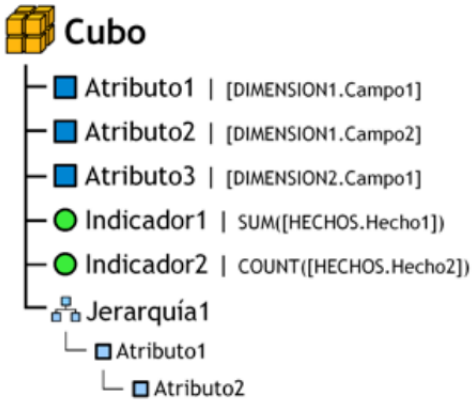


Figura 3.7.8: Estructura interna de un Hipercubo (Ing. Bernabeu R. Dario, 2009).

3.7.5.6.4. Relaciones Las relaciones básicamente representan la forma en que dos atributos interactúan dentro de una jerarquía. Estas pueden identificarse por sus dos tipos de relaciones:

- ***Explicitas:*** Este tipo de relaciones son las más comunes y pueden ser modeladas a partir de atributos directos ya que están en línea continua de una jerarquía. Aquí las relaciones pueden ser *uno a muchos* o *muchos a uno*.
- ***Implícitas:*** Estas relaciones ocurren en la vida real, pero su relación no es de vista directa. Aquí las relaciones pueden ser *muchos a muchos*.

3.7.5.6.5. Granularidad El termino granularidad se refiere al nivel de detalle con el que la información es almacenada, por lo tanto, si la granularidad es fina existe una mayor posibilidad de analizar, resumir y sumarizar los datos con un alto grado de detalle en cambio si la granularidad es media o gruesa dicha información puede ser analizada, resumida y hasta sumarizada pero no a un nivel detallado como sucede con la granularidad alta.

Los datos que son almacenados con una granularidad fina necesitan de un espacio mayor de almacenamiento ya que dichos datos contiene información con un alto grado de detalle y como resultado se obtiene un mayor número de registros para almacenar y procesar, en cambio los datos almacenados con una granularidad media o gruesa necesitan una menor cantidad de espacio de almacenamiento, por lo tanto, se debe de analizar la importancia y los detalles que pueden proporcionar los datos que serán almacenados para definir el tipo de granularidad que se aplicara a los datos de acuerdo al negocio que se desee analizar.

Capítulo 4

Metodología

Este capítulo describe los requisitos funcionales que deben ser soportados por el Sistema *Meter Data Management*–(Gestión de Datos de Medidores) (MDM). El propósito del capítulo es proporcionar información técnica sobre el desarrollo del sistema buscando cumplir los requerimientos internos y las normas correspondientes. Además se identifican los elementos necesarios para la transferencia de datos de entrada y salida del MDMS incluidos los Sistemas HES y el operador de la red de distribución eléctrica.

El Sistema MDMS proporcionará una infraestructura capaz de enviar y recibir datos sobre los dispositivos de un Sistema Head-End implementado dentro de una compañía proveedora de servicios de energía eléctrica, calculará la energía eléctrica consumida por los usuarios *(es decir recolectará y proporcionará dichos datos a los sistemas de la compañía para realizar los cálculos correspondientes)*, gestionara y almacenará los datos y los proporcionará a todas las partes interesadas de la infraestructura AMI.

El alcance del Sistema MDMS estará en el rango de 100,000 1,000,000 de dispositivos, por lo tanto, el Sistema MDMS debe ser lo bastantemente flexible para facilitar una alta escalabilidad y una facilidad de actualización.

El MDMS debe ser implementado utilizando la infraestructura de telecomunicaciones y computadoras en la empresa corporativa, de tal manera que el funcionamiento del sistema trabaje sin interrupciones, sea confiable y siempre se encuentre activo implementando mecanismos tales como: redundancia, suministro interrumpido, respaldo de datos, balanceadores[28], espejos, etc.

Se espera que el Sistemas HES recopile los datos de los *Smart Devices* de acuerdo a las tareas *(jobs)* o programas *(schedules)* y los transfiera al Sistema MDMS mediante archivos con una estructura definida, según la investigación y propuesta realizada por [14] y [8] dichos archivos se pueden modelar bajo el estándar *Common Information Model*–(Modelo de Información Común) (CIM) y con extensión XML.

El Sistema MDMS proporcionará un repositorio seguro de base de datos y una lógica de negocio para:

- Automatizar y agilizar el complejo proceso de recolección de datos de dispositivos a partir de tecnologías de recolección de datos de múltiples dispositivos.
- Evaluar la calidad de esos datos y generar estimaciones donde existen errores y lagunas.
- Entregar esos datos en el formato apropiado a los sistemas de facturación y a otros sistemas de información de la compañía de servicios de energía eléctrica.

4.1. Funciones del Sistema MDM

Las principales funcionalidades que debe integrar el Sistema MDM para realizar diversas actividades son las siguientes:

- ***Transferencia de datos:*** Los datos transferidos y obtenidos por el Sistema MDM debe ser consistentes tanto para las partes involucradas de la Infraestructura Avanzada de Medición (AMI) así como los subsistemas de información dentro de la empresa proveedora del servicio eléctrico y otras partes.
  Los datos involucrados en la transferencia incluyen:
  - Datos e informes provenientes del Sistema Head-End.
  - *Customer Information System*–(Sistema de Información del Cliente) (CIS).
  - Datos transferidos desde los sistema de facturación de la empresa proveedora del servicio de energía eléctrica.
  - Datos del Sistema MDM requeridos por otros subsistemas de información dentro de la compañía de electricidad y otras partes interesadas.
  - Informes y confirmaciones.
- ***Entrada de datos manualmente:*** El Sistema MDM debe proporcionar la posibilidad de introducir datos manualmente mediante una interfaz gráfica o al proporcionarle una ruta o path que haga referencia a un Script que contenga información de diferente interés para automatizar el proceso de carga, la información que se proporcionará manualmente puede ser la administración de dispositivos inteligentes, administración de grupos de dispositivos, programación de tareas, etc. Los datos que sean proporcionados manualmente o mediante el Script deben estar estructurados con mismo formato que los datos que se ingresan automáticamente en el Sistema MDM, por lo que, se realizara la misma validación de los datos para evaluar su estructura.
- ***Confirmaciones:*** Como ya se menciono anteriormente la infraestructura AMI esta compuesta por varios componentes, dichos componentes deben contar con un canal de comunicación para transferencia y/o recepción de información entre sistemas, por lo tanto, debe existir una *confirmación* que indique que la información a sido recibida y procesada con éxito o en su defecto informar que la información se ha corrompido, no es posible su recepción o que los comandos solicitados no se ejecutaron con éxito. El objetivo de dichas confirmaciones es hacer al sistema más confiable y tolerante a fallos ya que si por alguna circunstancia la información no puede ser transferida y/o recibida se pueden tomar medidas para corregir e identificar dichos problemas.
- ***Envío y recepción de informes por el Sistema MDM al Sistema HEAD-END:*** El Sistema Head-End y MDM deben generar y presentar varios reportes entre sistemas, estos informes están definidos de acuerdo a las especificaciones funcionales de cada sistema, los informes provenientes del Sistema Head-End proporcionan información sobre acciones realizadas físicas o lógicamente sobre los dispositivos de medición y/o respuestas a información solicitada por el Sistema MDM si los informes presentados contiene cambios en la información almacenada por ambos sistemas esta debe ser actualizada inmediatamente en ambos sistemas, mientras que el Sistema MDM genera informes para hacer peticiones al Sistema Head-End, así mismo el Sistema MDM debe archivar dichos informes e indexarlos para permitir la búsqueda y visualización de una manera ágil por el operador del MDM, para agilizar dicho proceso los informes deben ser indexados al menos por fecha y tipo de informe.

El Sistema MDM debe ser capaz de recibir todos los informes indicados en la especificación del Sistema Head-End y lo mismo para el Sistema Head-End, estos informes especificados en los requisitos funcionales de cada sistema son determinados de acuerdo con los requisitos del negocio de la empresa eléctrica.

Basándose en los informes presentados por el Sistema Head-End, no es necesario que el Sistema MDM procese los datos contenidos en dichos informes, ya que el Sistema Head-End es el encargado de procesar los datos que proviene de los dispositivos de medición inteligentes ya que estos se encuentran en formato hexadecimal y son almacenados y transmitidos bajos las normas *American National Standards Institute*–(Instituto Nacional Estadounidense de Estándares) (ANSI) C12.18 [29] y ANSI C12.19 [30], por lo tanto, el Sistema Head-End es el encargado de realizar la conversión de dichos datos y proporcionarlos al Sistema MDM evitando así la sobre carga de trabajo al sistema.

- ***Procesos de Extracción, Transformación y Carga (ETL):*** El MDM debe contar con un proceso ETL basado en los metadatos del Sistema Head-End para realizar la extracción y transformación de la información mejorando la calidad y homologando dichos datos antes de ser persistidos en el almacén de datos *(Data Warehouse)*.

  El Sistema MDM antes de pasar al proceso de carga de datos en el *Data Warehouse* realizara varias transformaciones *(semánticas y no semánticas)* sobre los datos y trabajará en conjunto con el procesos de Validación, Estimación y Edición (VEE) especialmente sobre los datos de contabilidad con el fin de mejorar significativamente la calidad de los datos.

  Cuando los datos son transmitidos desde los dispositivos inteligentes de medición y son recibidos por el Sistema MDM proporcionados mediante el sistema intermediario Head-End, el sistema debe validad la sintaxis y semántica de los datos y debe establecer que la estructura de dichos datos se encuentra dentro de los estándares implementados.

  Cada mensaje que es transmitido contiene una suma de comprobación *(también conocida como Checksum)*, tanto el Sistema MDM como del Sistema Head-End generan esta suma de comprobación antes de transmitir un mensaje, por lo tanto, cuando el MDM recibe información del Head-End debe ejecutar el cálculo de comprobación por cada mensaje recibido y compararlo con el valor original generado por el sistema transmisor, si los valores son idénticos se concluye que el mensaje esta integro y se procederá con la siguiente fase de procesamiento de lo contrario se concluye que el mensaje es corrupto y debe solicitarse la transmisión del paquete nuevamente para realizar el proceso anterior.

  El Sistema MDM debe cargar los datos que provienen de los paquetes enviados por el Sistema Head-End en la base de datos operacional o en el *Data Warehouse* según sea el caso y además dichos paquetes *(por lo regular archivos XML)* deben ser archivados en un directorio perteneciente al Sistema MDM para tener un respaldo, evitar perdida de datos o realizar cálculos complejos o no integrados en el Sistema MDM.

  El Sistema MDM debe realizar, sin restricciones, las siguientes validaciones de datos cargadas en el Sistema MDM:

  - Durante cada transferencia de datos, comprueba si la combinación Identificación de Punto de Entrega (*Pay On Delivery*–(Identificación de Punto de Entrega) (POD))/*Identifier*–(Identificador) (ID) de medidor es válida y si son concurrentes con los datos del subsistema MDM.
  - Valida si la hora exacta se distribuye en todos los medidores avanzados, así como otros dispositivos dentro del Sistema Head-End.
  - Valida cambios anormales en los datos recibidos de los dispositivos.

Los datos de los dispositivos cargados diariamente de manera autónoma en el *Data Warehouse* del Sistema MDM son por lo general datos de lecturas de consumos registrados por los dispositivos del día anterior, es muy probable que el sistema cargue datos de lecturas de dos o más días anteriores a la fecha actual, esto sucede porque algunos dispositivos de medición pierden comunicación con la red *(red mesh)* o con el servidor encargado de gestionarlos *(Head-End)*, por esta razón se realizan transferencias de datos simultaneas que contiene datos de lecturas de más de un día una vez que el dispositivo a recuperado la comunicación con el sistema. Los datos con lecturas más antiguas que no han sido persistidos en el *Data Warehouse* tiene una mayor prioridad para ser transferidos, procesados y almacenados con el fin de eliminar las lagunas de información lo antes posible.

Si los datos que son utilizados por los sistemas de facturación de la empresa proveedora del servicio eléctrico son requeridos y estos no se encuentran en su totalidad completados a la fecha de la petición se realizan una estimación de los valores en base a los datos históricos almacenados con anterioridad para que sean proporcionados y dar respuesta a la petición.

- ***Procesos de Validación, Estimación y Edición (VEE):*** Todos los datos de los dispositivos de medición inteligente recibidos por el Sistema MDM deben estar sujetos al análisis VEE dicho proceso debe ser autónomo y debe realizarse dentro del Sistema MDM.

  El proceso de análisis VEE debe realizarse sobre los datos actuales del dispositivo con el propósito de encontrar posibles anomalías, en un escenario donde se descubren anomalías en los datos extraídos de los dispositivos, se debe generar un informe de error, así como la solicitud de corrección de datos dentro del Sistema MDM con el valor estimado. Dichos datos estimados deben ser etiquetados con el fin de poder identificar aquellos datos que proviene de los dispositivos de medición y los que han sido estimados y/o modificados para ofrecer información de su autenticidad.

  El Sistema MDM debe permitir el acceso de manera transparente a los datos generales y contables de los dispositivos de medición, cuando estos estén disponibles después de un análisis (VEE), proporcionándolos a otros subsistemas de información del cliente.

  El Sistema MDM debe tener un sistema de reemplazo de datos implementado *(implementando mecanismos de confirmación para saber el status de la solicitud)*, en caso de que otros subsistemas de información del cliente envíen correcciones de datos no válidos dentro del Sistema MDM.

  Con el fin de proporcionar datos para el cálculo a otros Sistemas de Información del Cliente (CIS), el Sistema MDM debe ser capaz de determinar la prioridad para el análisis VEE de datos de acuerdo a los puntos de medición teniendo en cuenta los requisitos específicos para el cálculo y facturación.

  El proceso VEE debe proporcionar una o múltiples herramientas al operador con la capacidad de crear una serie de algoritmos estándar mediante una interfaz gráfica para agilizar el proceso o de una manera más avanzada mediante Scrips que puedan ser cargados al sistema ejecutando una serie de instrucciones, dichos algoritmos deben estar basado en reglas y parámetros predefinidos por el operador, con plena transparencia y presentación de informes sobre el desarrollo de esos algoritmos.

Las siguientes son funciones fundamentales de *Validación y Estimación* que el Sistema MDM debe permitir:

- Estimar el intervalo de datos basado en las lecturas del dispositivo.
- Sustituir todos los valores por una constante.
- Multiplicar o dividir por una constante.
- Sumar o restar una constante.
- Deslizar una gama de datos de intervalos adelante o atrás en el tiempo.
- Realizar interpolación lineal.
- Dividir o combinar intervalos.
- Restaurar una versión anterior.

Además, el operador debe ser capaz de *Editar* los valores utilizando una serie de funciones de edición estándar:

- Agregar o reemplazar valores manualmente.
- Modificar estado de lectura.
- Mostrar o editar varias lecturas.
- Copiar o cortar/pegar una cadena de valores de un dispositivo a otro.
- Copiar o cortar/pegar valores de una hoja de cálculo.

Las interfaces MDM estándar hacen que los cálculos nuevos y editados estén inmediatamente disponibles para todos los servicios de información de la compañía eléctrica. Esto soluciona tres problemas que se han tenido durante años:

- Sustituye los procesos manuales de importación y exportación de datos por procesos automatizados seguros y auditables.
- Proporciona un cálculo único para su uso por todos los sistemas de servicios públicos. Los mismos datos de carga ya no generan valores ligeramente diferentes dependiendo de qué hoja de cálculo de utilidad se utilizó.
- Proporciona la planificación de escenarios para ejecutar escenarios de "*qué pasa si*" para diferentes tipos de tasas a través de diferentes clases de clientes u otros estratos.

El Sistema MDM debe contemplar el almacenamiento de versiones de datos, esto con el fin de mantener una control de los datos cargados en los mecanismos de almacenamiento del MDM de tal manera que se tenga una fotografía de cada lectura de los dispositivos asociados y gestionados con una referencia de tiempo, esto facilitara la resolución de problemas, discusiones de facturación y liquidación. Es importante recalcar que los datos que son estimados y cargados por el proceso VEE que incorpora el MDM son "*los mejores disponibles*" en un punto de tiempo definido y, por lo tanto, esta es una capacidad critica del Sistema MDM.

Al almacenar datos por versiones estos se vuelven críticos al mantener la integridad de los datos, ya que dichos datos se comparten a través de múltiples sistemas de información del cliente *(respuesta a la demanda, investigación de carga, pronóstico, análisis de activos de distribución, etc.)*, el Sistema MDM se convierte en un recurso central ya que se convierte en una fuente única que proporciona datos a múltiples sistemas de los demás servicios, por lo tanto, debe tener la capacidad de reproducir un conjunto de datos al menos por fecha y hora concreta ya que este proceso resulta vital para el funcionamiento de los diversos sistemas del cliente.

El Sistema MDM debe proporcionar acceso a los datos de los dispositivos de medición inteligente utilizando la versión de datos correspondiente. Las versiones de datos que son almacenadas en el sistema representan conceptos de calidad en los datos del negocio, por lo tanto, cada que se alteren los datos de un dispositivo el Sistema MDM debe tomar medidas para actualizar únicamente los datos que han sido modificados en el dispositivo correspondiente tomando como referencia la fecha y hora de modificación, el Sistema MDM debe monitorear como, cuando y por que se han hecho cambios en los datos de los dispositivos e identificar que usuario o proceso VEE ha realizado dicho cambio, con el fin de poder rastrear las modificaciones y/o alteraciones realizadas en los dispositivos ya sea por un usuario en especifico o por el mismo sistema.

Un ejemplo de las versiones de datos almacenadas por el Sistema MDM es, si una lectura cambia cinco veces, el Sistema MDM debe crear cinco versiones de esa lectura, cada una de esas versiones tiene un periodo de referencia, indicando la versión actual y las versiones que le anteceden.

El Sistema MDM debe proporcionar mecanismos que permitan identificar datos de dispositivos modificados o estimados, permitiendo al operador identificar dichos datos rápidamente incluso dándole la posibilidad de ingresar un comentario que describa más detalladamente la alteración realizada.

Durante una revisión de algún dispositivo de medición el Sistemas MDM debe proporcionar los datos reales obtenidos del dispositivo de medición si estos están disponibles, en lugar de proporcionar los datos estimados o reemplazados por los mecanismos VEE utilizados para generar informes contables. Todos los datos provenientes de los dispositivos de medición serán procesados a través del análisis VEE para mejores resultados.

- ***Motor de cálculo ejecutándose en segundo plano:*** El Sistema MDM debe proporcionar un motor de cálculo integrado y trabajando en segundo plano cada X intervalo de tiempo *(este intervalo de tiempo debe ser definido por el operador de acuerdo a sus necesidades)* que le permita al operador producir determinantes de facturación y otro tipo de reportes lo más rápido posible. Una forma de optimizar la respuesta a las peticiones realizadas por el(los) operador(es) es usar tablas de hechos agregadas o pre-agregadas dentro del *Data Warehouse* perteneciente al Sistema MDM para almacenar la información y resultados que el motor de cálculo procesa, de esta manera se reduce el procesamiento en tiempo real cada que una petición hecha con anterioridad vuelve a ser solicitada.

  El motor de cálculo debe soportar todos los operadores y funciones matemáticas comunes, así como funciones condicionales y lógicas, además debe proporcionar una interfaz gráfica de hoja de cálculo simple e intuitiva para realizar la extracción de información de acuerdo a las necesidades del operador. Las funciones que debe incorpora el motor de cálculo son las siguientes:

- Operadores comunes: +, -, x, /, raíz cuadrada, cuadrado, seno, coseno, etc.
- Condición/funciones lógicas: si, y, o, no,¿, =, ¡, etc.
- Funciones de hora y fecha: max, min, avg, total, etc.
- Conversión de unidades: (Kilovatio Hora) (KWh)/(Kilo Voltio-Amperio Reactivo Hora) (KVARh) a (Kilo Voltio-Amperio Hora) (KVAh), factor de potencia, etc.

La funcionalidad del motor de cálculo permite a los operadores calcular diferentes resultados tales como: carga, perdida o agregación compleja para aplicaciones de facturación, así como cálculos para otros procesos de utilidad, incluyendo el proceso de estimación VEE. En el contexto de una solución de gestión de datos de dispositivos inteligentes, los determinantes de facturación deben calcularse y entregarse de manera autónoma sin intervenciones manuales a petición del sistema de facturación de la compañía proveedora del servicio eléctrico. El motor de cálculo simplifica la actualización y el mantenimiento de los cálculos y el control de versiones controla los cambios en la información de los dispositivos.

- ***Almacenamiento seguro de los datos:*** El Sistema MDM garantiza el almacenamiento seguro de la información dentro de sus mecanismos implementados de almacenamiento.
- ***Mecanismos de redundancia y recuperación autónomos:*** El Sistema MDM contempla mecanismos de redundancia y recuperación de desastres.
- ***Almacenamiento:*** El Sistema MDM tiene la capacidad de brindar almacenamiento de datos sin procesar y/o procesados provenientes de los dispositivos de medición inteligentes en un formato definido.
- ***Ejecución automática de tareas (JOBS):*** El sistema debe realizar la ejecución de tareas periódicas *(jobs)* de manera autónoma, así mismo debe ejecutar tareas de programas *(schedule)* que contengan instrucciones tales como la adquisición de datos de los dispositivos inteligentes, envío de instrucciones a un dispositivo en especifico, a uno o más grupo(s) de dispositivos o realizar un broadcast de instrucciones a los dispositivos gestionados por el sistema MDM. Los *jobs* y *schedule* deben ser ejecutados con periodicidad y con una jerarquía definidas de acuerdo a los criterios del operador.
- ***Adquisición de datos y envío de comandos:*** El Sistema MDM debe realizar la adquisición de datos o envío de instrucciones en tiempo real a petición del operador de un dispositivo de medición en especifico, uno o más grupo(s) de dispositivos o de la totalidad de los dispositivos de medición. Esta característica nos permitirá obtener datos de diversos interese o ejecutar instrucciones en uno o más dispositivos tales como conexión, desconexión, diagnostico, etc.
- ***Consulta de datos específicos:*** El Sistema MDM debe permitir la consulta de datos de uno o más dispositivos de medición a petición del operador mediante un filtrado seleccionable.
- ***Uso de prioridades:*** El Sistema MDM debe hacer uso de prioridades para el manejo de adquisición de datos y envío de instrucciones que involucren a los dispositivos inteligentes de medición.
- ***Recepción de forma autónoma de alarmas:*** El Sistema MDM debe recibir y registrar de manera automática las alarmas *(eventos desencadenados en un dispositivo, el operador toma criterios para definir un evento como una alarma)* de un dispositivo de medición o de uno o más grupos de dispositivos o de la totalidad de los dispositivos de medición gestionados por el Sistema MDM.
- ***Generación de informes:*** El Sistema MDM debe generar informes de manera automática o a petición del operador tales como:

- Visualización de la estructura de red con regiones y concentradores.
- Filtro de búsqueda para acceso rápido a todos los nodos de comunicación y dispositivos de medición.
- Visualización de parámetros, atributos, estadísticas, estado, etc. de todos los nodos de comunicación y dispositivos de medición.
- Informes de resumen de nodos de comunicación y dispositivos de medición por su estado como: *nuevo, activo, fallido, perdido, etc.*
- Informes de resumen de información estadística, estado de los nodos de comunicación y/o dispositivos de medición.
- Búsqueda de nodos de comunicación y dispositivos de medición con diferentes criterios como por su estado: *nuevo, fallido, perdido, etc.*

- ***Interfaz de importación de información de dispositivos:*** Dicha interfaz debe permitir agilizar el proceso de carga y asignación de gestión de dispositivos de medición al sistema MDM, la información del dispositivo de medición puede incluir: *números de identificación (número de serie del fabricante/cliente, configuración, etc.), información del dispositivo (tipo de dispositivo, configuración, tipo de antena, etc.), información de ubicación (nombre de la ubicación, dirección, coordenadas, etc.)*. Esta información forma parte esencial para el sistema central y sirve para identificar y gestionar de forma eficaz cada uno de los dispositivos de medición que conforman la infraestructura AMI.

  La importación puede ser realizada autónomamente por una tarea programada o ejecutada manualmente dependiendo de los criterios establecidos por del operador.

4.1.1. Parámetros configurables del Sistema MDM

El Sistema MDM debe contener interfaces o medios de acceso para el operador u otro componente del Sistema AMI, para acceder a la configuración de los siguientes parámetros:

- Programación de tareas por excepción o periódicas (*jobs*).
- Gestión del intervalo de tiempo de las tareas periódicas.
- Selección de los parámetros de las tareas que son ejecutadas autónomamente sobre los dispositivo de medición, los cuales pueden ser:
  - Lecturas de registros.
  - Demandas.
  - Alarmas.
  - Información de los eventos desencadenados en los dispositivos de medición.
  - Perfil de carga.
  - Tarifas horarias.
- Selección de las instrucciones pertenecientes a las tareas que son enviadas de manera autónoma a los dispositivo de medición, las cuales pueden ser:

  - Sincronización de tiempo.
  - Configuración de eventos para conectar/desconectar dispositivos de medición.
  - Configuración de parámetros en los dispositivos de medición.
  - Descarga de tareas *(jobs)* en los nodos de comunicación y/o dispositivos de medición.
  - Descarga de programas de tareas *(schedule)* en los nodos de comunicación dispositivos de medición.
  - Actualizaciones de firmware en los dispositivos de medición.
- Programación de programas de tareas *(schedule)*.
- Creación de grupos de dispositivos de medición para asociar una o más tareas y/o a programas de tareas.
- Asignación de un dispositivo de medición o un grupo de dispositivos a tareas y/o a programas de tareas.
- Configuración de roles y contraseñas de los operadores que ingresan al Sistema MDM.
- Programación de informes con elementos seleccionables por el operador.

4.2. Requerimientos de seguridad

El Sistema MDM debe contener mecanismos de seguridad para la protección al acceso de la información del sistema con autenticación de usuario y control de acceso basado en roles, para asegurar que la información del cliente no esté comprometida contra accesos de dispositivos o usuarios no autorizados. Así mismo deben existir mecanismos que permitan cifrar y descifrar la información *(implementando algoritmos propios o haciendo uso de los ya existentes)* punto a punto, es decir, cuando esta es transmitida o recibida por los diversos sistemas que conforman la infraestructura AMI y los subsistemas que trabajan en la compañía proveedora del servicio eléctrico.

4.3. Requerimientos de integración

Para el intercambio de información e instrucciones con otros componentes del sistema de gestión AMI, el Sistema MDM debe contar con formatos de interfaz estándar basados en arquitectura Web-Service y alguno de los siguientes protocolos de comunicación y transferencia de archivos como: *File Transfer Protocol*–(Protocolo de Transferencia de Archivos) (FTP)[21, Página 209], *SSH File Transfer Protocol*–(Protocolo de Transferencia de Archivos Seguro) (SFTP), *Secure SHell*–(Interprete de Ordenes Seguro) (SSH) y/o *Telecommunication Network*–(Red de Telecomunicación) (Telnet)[21, Página 193].

Los modelos de intercambio de información deben cumplir los lineamientos establecidos por la norma Common Information Model (CIM) Base, IEC 61970-501: Common Information Model (CIM) XML Codification for Programmable Reference and Model Data Exchange e interactuar con sistemas del tipo Head-End, CIS, WFM, OMS, etc.

Capítulo 5

Resultados

5.1. Ubicación del MDMS en la Infraestructura Avanzada de Medición

El Sistema MDM es un sistema que se especializa en la administración de datos generados por los dispositivos de medición inteligentes haciendo uso de diversos sistemas y tecnologías implementadas antes y después de la arquitectura del MDM.

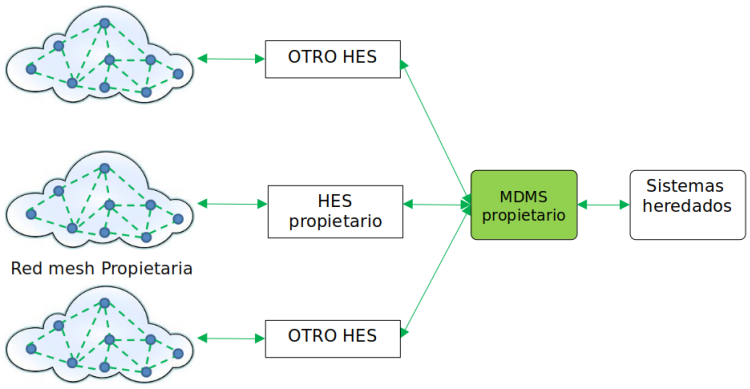


Figura 5.1.1: Representación gráfica del MDMS en la infraestructura AMI (EosTech S.A de C.V, 2017).

5.2. Arquitectura del MDMS

La arquitectura del Sistema MDM se encuentra estructurado en tres capas: *Vista, Controlador y Modelo*, esta estructura fue implementada gracias a que este modelo desacopla los tres componentes permitiendo la reutilización de código más eficiente, el desarrollo en paralelo, una mejor estructura del proyecto y un mantenimiento mucho más flexible.

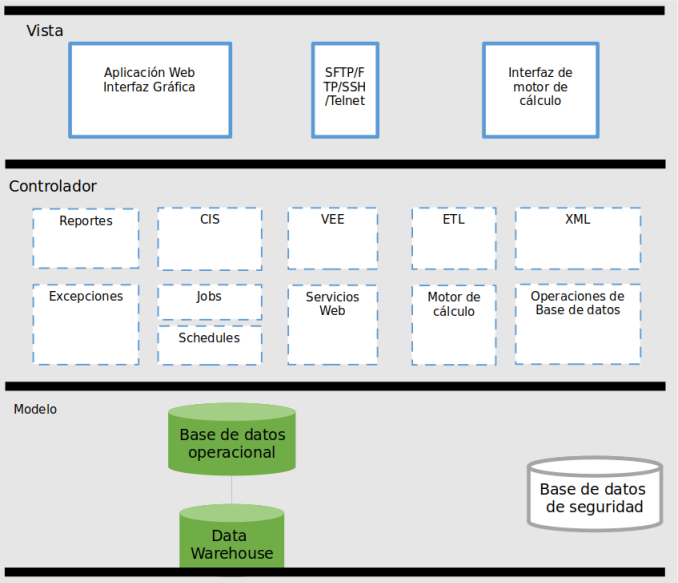


Figura 5.2.1: Modelo Vista Controlador del Sistema MDM (EosTech S.A de C.V, 2017).

5.2.1. Interacción entre los Sistemas MDMS, HES y otros sistemas

El Sistema MDM interactúa con el Sistema Head-End (HES) un sistema intermediario encargado de gestionar y decodificar los datos de los dispositivos inteligentes, esta comunicación se realiza mediante la transferencia de ficheros XML *(los ficheros XML contienen un formato definido por una norma o por el proveedor de energía eléctrica)*, así mismo el Sistema MDM interactúa con otros sistemas principalmente sistemas de terceros que pertenecen a la compañía proveedora del servicio eléctrico utilizando el mismo mecanismo de archivos XML.

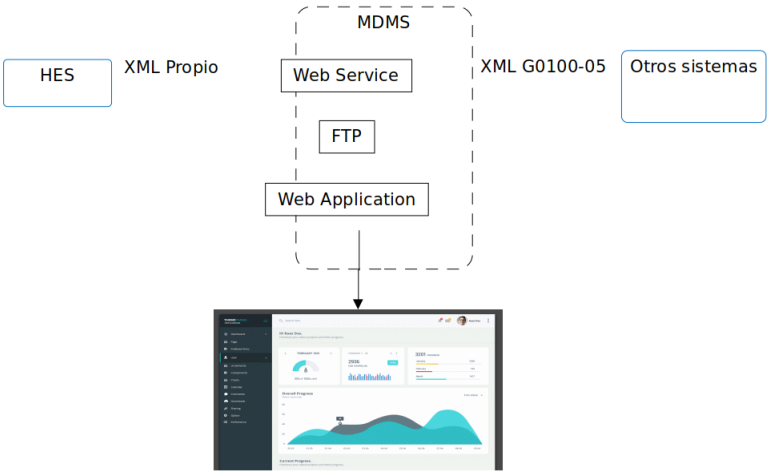


Figura 5.2.2: Interacción del MDMS con otros sistemas (EosTech S.A de C.V, 2017).

5.2.2. Capa vista

5.2.2.1. Aplicación web

La capa de presentación es la encargada de visualizar y realizar acciones para el usuario final, estos datos pueden presentarse de diversas maneras tales como: *mostrados en pantalla, impresos, formato PDF, CSV, XML, entre otros*. Así mismo los servicios ofrecidos por la aplicación web que incorpora el Sistema MDM son considerados de presentación ya que funcionan y pueden ser operados mediante una interfaz gráfica entre los diversos sistemas que interactúan con el Sistema MDM.

5.2.2.1.1. Generalidades

- El lenguaje de programación de servidor donde se encontrara alojado el Sistema MDM debe ser Java con HTML implementado en un servidor TomCat, puede contener contenidos basados en JavaScript, *Asynchronous JavaScript And XML*–(JavaScript Asíncrono y XML) (AJAX), JSON, entre otras tecnologías.
- La codificación debe realizarse sobre los estándares definidos por la empresa desarrolladora del sistema.
- El proceso de desarrollo debe estar basado en el alto desempeño y calidad del producto final.

5.2.2.1.2. Administración de usuarios

- Nuevo usuario.
- Login.
- Edición.
- Eliminación.
- Roles.
- Permisos.

5.2.2.1.3. Dashboard *(casos de uso)*

- Administración de dispositivos.
- Administración de *jobs*.
- Administración de *schedules*.
- Administración de reportes.
- Administración de alarmas.
- Administración de carga de datos.
- Administración de estimación de datos.
- Administración de base de datos.
- Administración de excepciones.

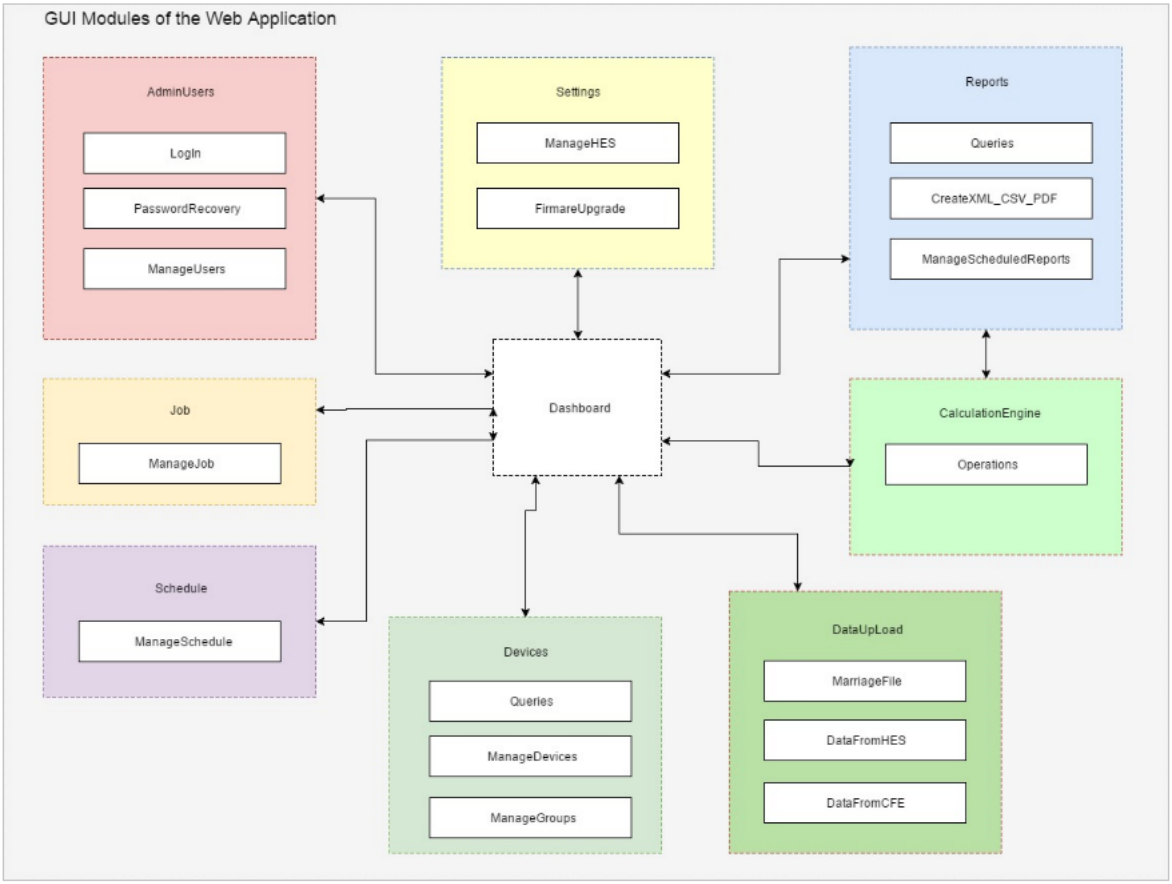


Figura 5.2.3: Módulos de la interfaz gráfica de la aplicación web (EosTech S.A de C.V, 2017).

5.2.2.2. Transferencia de ficheros

5.2.2.2.1. Servicio SFTP/FTP/SSH/Telnet El Sistema MDM debe implementar un servicio que permita la transferencia de ficheros entre sistemas para poder comunicarse e intercambiar paquetes, los servicios que se pueden implementar son: SFTP/FTP[21, Página 209]/SSH y/o Telnet[21, Página 193] entre otros dependiendo de los criterios que el desarrollador del sistema y el cliente crean pertinentes. El Sistema MDM implementa dos servidores para la transferencia de archivos, un servidor FTP y un servidor SSH.

El Protocolo de transferencia de archivos (FTP)[21, Página 209] es un protocolo estándar de red utilizado para la transferencia de archivos entre sistemas utilizando el modelo cliente-servidor[33]. La transferencia de archivos entre el cliente y el servidor FTP que incorpora el Sistema MDM es bidireccional, es decir, los clientes puede enviar o solicitar un fichero al servidor del Sistema MDM.

Para acceder a los ficheros remotos que almacena el Sistema MDM, los usuarios deben identificarse con el servidor, el servidor es responsable de autentificar a los clientes antes de permitir la transferencia de ficheros, el enlace está orientado a la conexión, es decir, es necesario que ambos hosts estén activos y ejecutando TCP/IP[21, Página 69] para establecer una transferencias de ficheros.

El servidor FTP usa TCP/IP[21, Página 69] como protocolo de transporte para proporcionar conexiones fiables entre los sistema, el servidor FTP emplea dos conexiones: la primera es utilizada para realizar un *login* en el servidor y la segunda es para gestionar la transferencia de archivos. Al realizar un *login* en el servidor FTP, los clientes deben contar con un usuario y un *password* para acceder exitosamente al servidor y así tener acceso a los ficheros y directorios. El usuario que inicia la conexión asume la función de cliente, mientras que el Sistema MDM adopta la función de servidor.

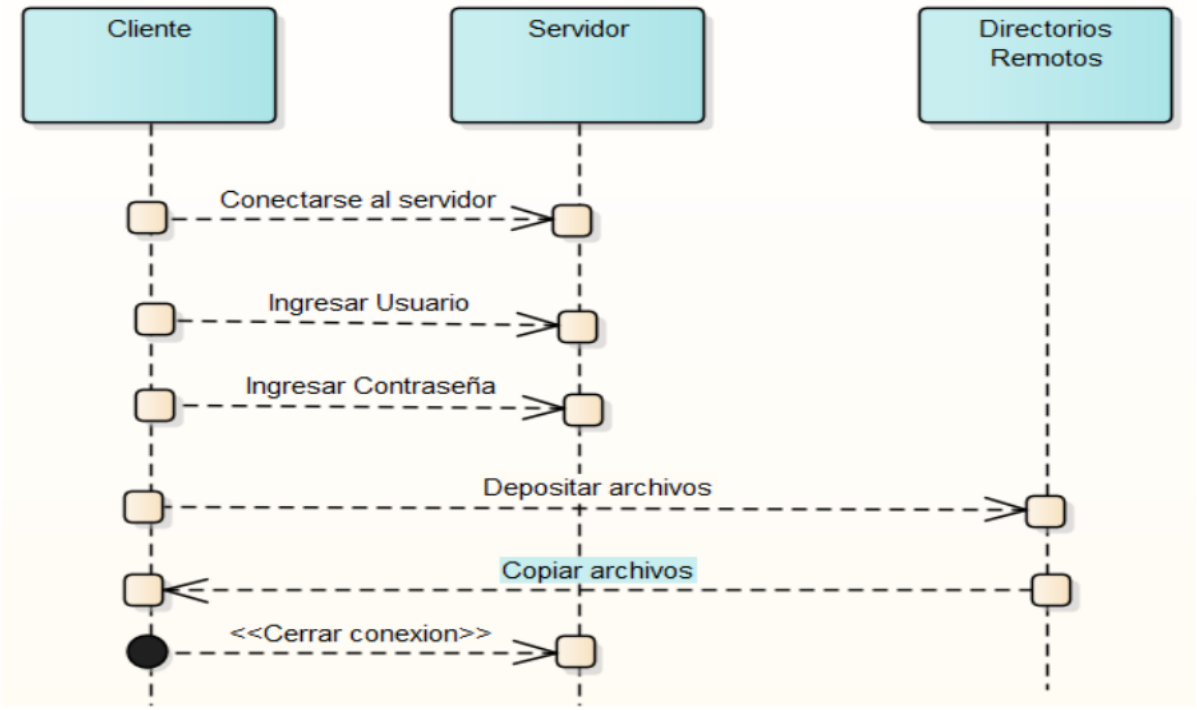


Figura 5.2.4: Diagrama de secuencia del servidor FTP.

El protocolo de transferencia de archivos mediante SSH (SFTP–Secure File Transfer Protocol–) es incorporado al sistema MDM, es un protocolo de red TCP/IP[21, Página 69] que proporciona acceso, transferencia y la gestión de archivos remotamente mediante una conexión punto a punto cifrada, permite conectarse de manera más segura y fiable a equipos remotos. El protocolo SFTP proporciona una amplia gama de utilerías sobre los archivos disponibles en el servidor, entre las utilerías se encuentra: *reanudación de transferencias de archivos interrumpidas, listado de directorios, eliminación y modificación de archivos remotamente*.

El protocolo SFTP es multiplataforma, por lo tanto, puede ser implementado en casi cualquier plataforma, es importante no confundir el protocolo FTP[21, Página 209] con el protocolo SFTP ya que el protocolo SFTP es un protocolo más seguro al cifra la conexión punto a punto lo que no sucede con el protocolo FTP.

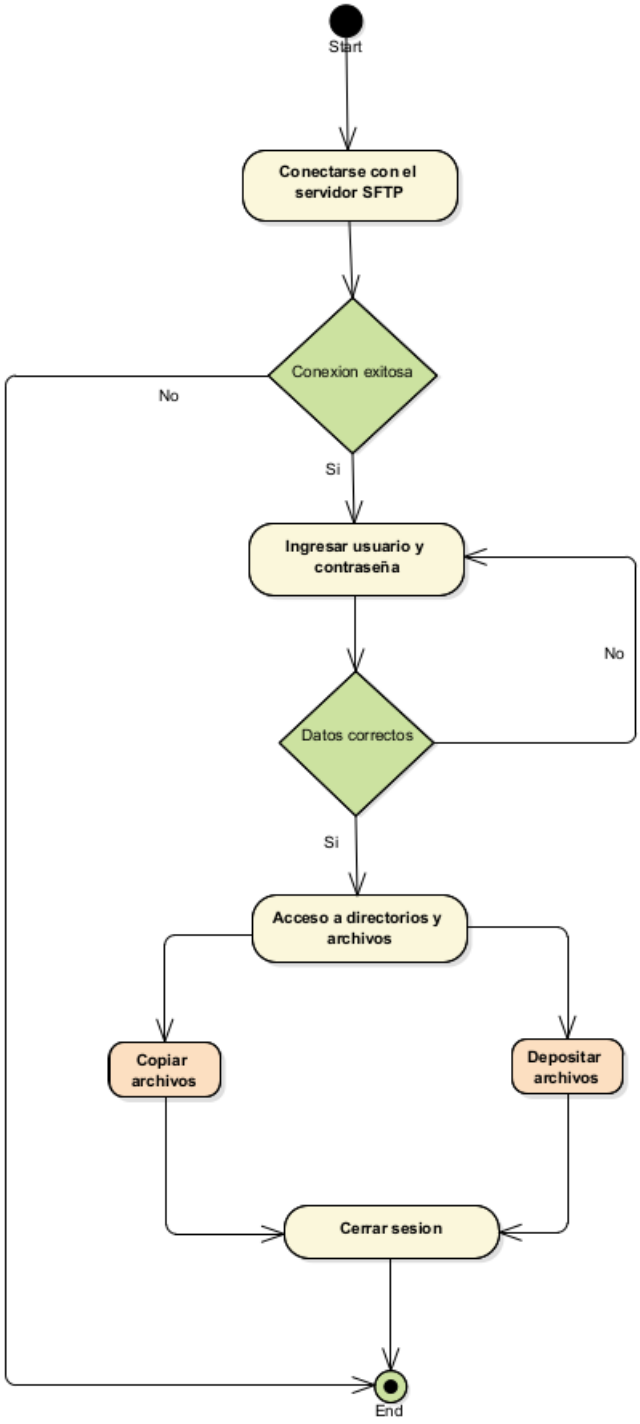


Figura 5.2.5: Diagrama de flujo del servidor SFTP.

Debe existir una estructura interna definida en los directorios del Sistema MDM que permitan depositar, tomar y respaldar ficheros del cliente y del HES como la que se muestra a continuación:

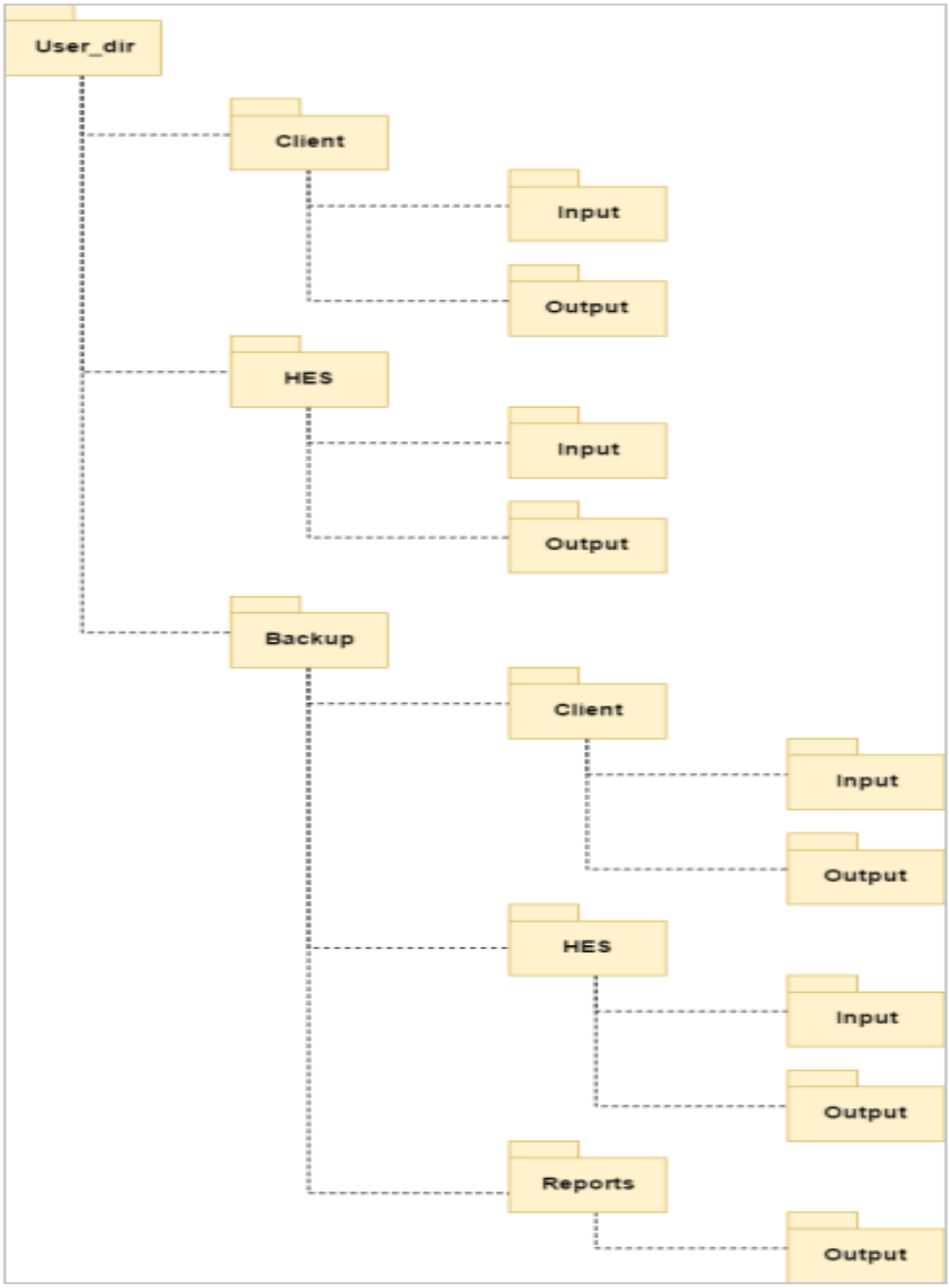


Figura 5.2.6: Estructura interna de los directorios del MDMS (EosTech S.A de C.V, 2017).

El Sistema MDM debe contar con un directorio local de almacenamiento y de respaldo de los ficheros que son creados y recibidos por el sistema bajo la siguiente estructura. Estos archivos deben contener un identificador o folio único para poder rastrearlos y gestionarlos.

Tabla 5.2.1: Estructura del almacenamiento de los ficheros generados y recibidos por el Sistema MDM.

Número	Carpeta	Descripción
1	Archivos y peticiones para el HES.	El Sistema MDM escribe en el directorio ***HES/Input*** los archivos XML que deben ser enviados a los HES correspondientes con la siguiente nomenclatura:*rqhsXXXXXAAAAMMDD_HHmmss_SS.xml*
2	Archivos y respuestas generados por el HES.	El Sistema HES escribe las respuestas a las peticiones solicitadas en el directorio ***HES/Output*** que pertenece al Sistema MDM.
3	Archivos para la compañía eléctrica interesada.	El Sistema MDM escribe en el directorio ***Client/Output*** los archivos XML que deben ser enviados al cliente *(proveedor del servicio eléctrico)* con la siguiente nomenclatura: *rpcltXXXXXAAAAMMDD_HHmmss_SS.xml*
4	Archivos generados por la compañía eléctrica.	Los archivos y peticiones generados por el cliente *(compañía proveedora del servicio eléctrico)* son escritos en el directorio ***Client/Input*** que pertenece al Sistema MDM generalmente estos archivos son generados con extensión XML.
5	Archivos de respaldo.	Cada que un fichero es generado y/o recibido por el Sistema MDM se realiza una copia de seguridad contenido en el directorio ***Backup*** que pertenece al Sistema MDM seguido del subdirectorio ***HES/Client*** dependiendo del origen o destino del fichero. Esta copia de respaldo es almacenada con el fin de realizar aclaraciones o detectar errores en el funcionamiento de alguno de los sistemas que interviene en el proceso.
6	Reportes.	Los reportes generados por el Sistema MDM son escritos en el directorio ***Backup/Reports*** con la siguiente nomenclatura: *repXXXXXAAAAMMDD_HHmmss_SS.xml*

5.2.3. Capa controlador

La capa del controlador del sistema también conocida como capa lógica o *Business Layer* es la encargada de procesar la información dicho proceso involucra procesos tales como: extraer, transformar, validar, cargar, recuperar la información dentro del sistema, así mismo dicha capa es la encargada de tomar decisiones tomando como criterios los resultados arrojados por la información procesada por la capa lógica.

5.2.3.1. Servicios web

5.2.3.1.1. Generalidades

- El lenguaje de programación utilizado para codificar la capa del controlador debe ser implementado en Java utilizando la metodología de POO.
- El lenguaje del servidor debe ser Java.
- Los estándares de codificación de los módulos que componen la capa del controlador deben estar basados y definidos por la empresa de desarrollo de dicho software.
- El proceso de desarrollo esta basado en estándares y calidad del producto.

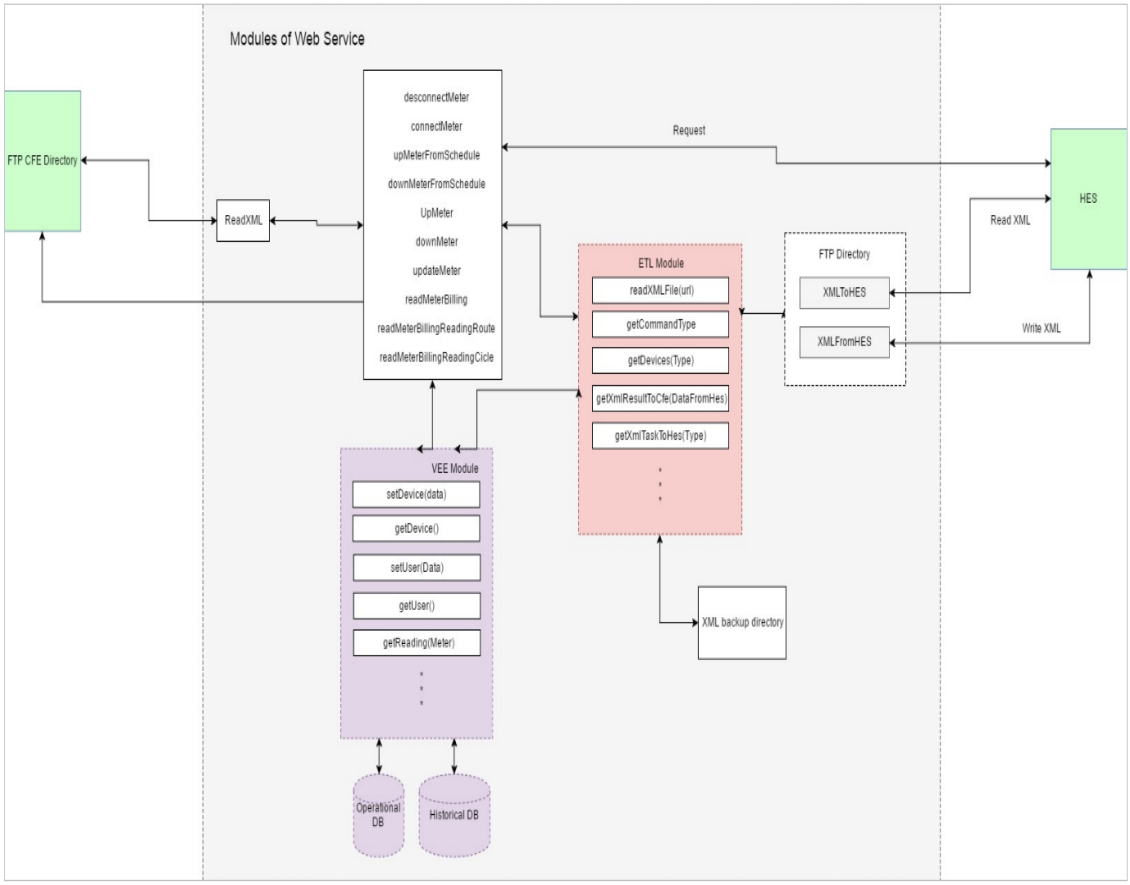


Figura 5.2.7: Módulos que componen el servicio web del Sistema MDM (EosTech S.A de C.V, 2017).

5.2.3.1.2. Peticiones a sistemas externos El Sistema MDM debe realizar peticiones a sistemas externos como por ejemplo al Sistema HES para realizar una o más tareas en especifico, dicha petición debe ser solicitada a través del servicio web pasando archivos XML en un estándar definido por una norma *(G0100-05)*[2], dichas peticiones deben estar basadas en un catálogo previamente definido por el cliente y el desarrollador del sistema e implementada por los sistemas involucrados (MDMS y HES). Todas las peticiones y respuestas deben contener un identificador único que permita diferenciarlas y rastrearlas para aclaraciones futuras. Las peticiones que pueden ser solicitadas mediante el servicio web se listan a continuación:

Tabla 5.2.2: Peticiones que realiza el Sistema MDM mediante su servicio web.

Número	Petición	Descripción
1	DesconnectMeter	Desconectar medidor.
2	ConnectMeter	Conectar medidor.
3	Meter	Alta, baja o modificación de un medidor.
4	UpMeterToSchedule	Alta de un medidor a un calendario de lectura.
5	DownMeterToSchedule	Baja de un medidor de un calendario de lectura.
6	Reading	Toma de lectura actual.
7	BillingRoute	Tomar lectura para facturación por ruta.
8	BillingCicle	Tomar lectura para facturación por ciclo.
9	LoadProfile	Recuperar perfil de carga completo.
10	LoadProfileInterval	Recuperar perfil de carga en un intervalo de tiempo.
11	Events	Recuperar todos los eventos.
12	EventsInterval	Recuperar los intervalos en un intervalo de tiempo.
13	Register	Recuperar los datos de registros completos.
14	RegisterInterval	Recuperar los datos de registros en un intervalo de tiempo.
15	FirmwareUpgrade	Actualizar firmware.
16	Ping[21, Página 103]	Ping[21, Página 103] para verificar si existe comunicación con el dispositivo.
17	FullRead	Lectura completa de una tabla de un dispositivo.
18	FullWrite	Escritura completa de unja tabla en un dispositivo.

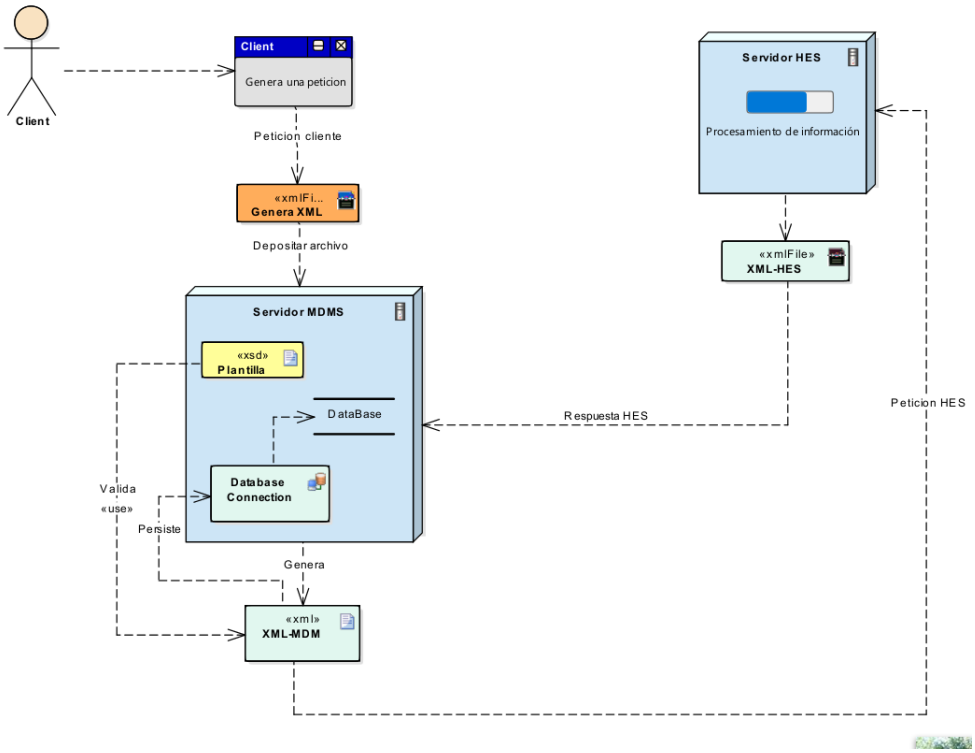


Figura 5.2.8: Diagrama de flujo de petición y respuesta entre sistemas.

5.2.3.1.3. Servicios a otros sistemas El Sistema MDM ofrece servicios a sistemas de terceros mediante peticiones, normalmente los sistemas de terceros son aquellos que se encuentra operando en la central del proveedor del servicio de energía eléctrica y requiere datos gestionados por el Sistema MDM para realizar diversas operaciones. Las peticiones de los sistemas del cliente deben estar basados en normas *(G0100-05)*[2] y en un token que permite identificar y autentificar al sistemas fuente y destino. Las peticiones que pueden ser solicitadas por los sistemas del cliente son los siguientes:

Tabla 5.2.3: Peticiones con destino al Sistema MDM.

Número	Servicio	Descripción	Respuesta*(return)*
1	UpRequest	Desconectar dispositivo.	XML con el resultado de la tarea según la norma *(G0100-05)*[2]. XML con lecturas en caso de éxito.
		Conectar dispositivo.	XML con el resultado de la tarea según la norma *(G0100-05)*[2]. XML con datos del medidor en caso de éxito.
		Alta de dispositivo.	XML con el resultado de la tarea según la norma *(G0100-05)*[2]. XML con datos del medidor en caso de éxito.
		Baja de dispositivo.	XML con el resultado de la tarea según la norma *(G0100-05)*[2]. XML con datos del medidor en caso de éxito.
		Modificación de dispositivo.	XML con el resultado de la tarea según la norma *(G0100-05)*[2]. XML con datos del medidor en caso de éxito.
		Alta de un medidor a un calendario de lectura.	XML con el resultado de la tarea según la norma *(G0100-05)*[2]. XML con datos del medidor en caso de éxito.
		Baja de un medidor a un calendario de lectura.	XML con el resultado de la tarea según la norma *(G0100-05)*[2]. XML con datos del medidor en caso de éxito.
		Toma de lectura actual.	XML con el resultado de la tarea según la norma *(G0100-05)*[2]. XML con lecturas en caso de éxito.
		Toma de lectura para facturación por ruta.	XML con el resultado de la tarea según la norma *(G0100-05)*[2]. XML con lecturas en caso de éxito.
		Toma de lectura para facturación por ciclo.	XML con el resultado de la tarea según la norma *(G0100-05)*[2]. XML con lecturas en caso de éxito.
2	Login	Autenticación de usuarios.	Privilegios.
Continua...			

Tabla 5.2.3: Peticiones con destino al Sistema MDM *(continuación)*.

Número	Servicio	Descripción	Respuesta*(return)*
3	UpResponse	Los Sistemas HES informan de una respuesta. Antes de hacer un UpResponse se debe colocar primero el archivo XML en la carpeta local del Sistema MDM mediante la comunicación SSH/FTP[21, Página 209]/SFTP/Telnet[21, Página 193] según implementada.	
Fin de la tabla.			

Los servicios que ofrece el Sistema MDM deben incluir un código de respuesta único que permita identificar las peticiones.

5.2.3.2. Módulo de usuarios

5.2.3.2.1. Generalidades El módulo que compone la administración de usuarios, roles y permisos se realiza mediante un *Framework* llamada Spring Security. Este *Framework* ofrece una serie de servicios de seguridad aplicables para sistemas basados en la arquitectura *Java Enterprise Edition*–(Edición de Java Empresarial) (J2EE).

5.2.3.2.2. Administración de usuarios El módulo de administración de usuarios debe permitir a los operadores las operaciones fundamentales tales como: alta, baja, y actualización de usuarios.

5.2.3.2.3. Administración de roles El módulo de administración de roles debe permitir a los operadores acciones como: alta, baja y actualización de roles. El catálogo inicial de roles se específica en la siguiente tabla.

Tabla 5.2.4: Roles del Sistema MDM.

Número	Rol	Descripción
1	Administrador inicial *(Superusuario/root)*.	Tiene todos los permisos, puede realizar todas las tareas.
2	Administrador de aplicaciones.	Puede realizar todas las tareas, excepto crear y editar usuarios.
3	Administrador de medidor.	Gestión de medidor y rendimiento del medidor con otros roles.
4	Administrador de usuarios.	Solo puede agregar y editar usuarios; normalmente usado en conjunto con otros roles.
5	Operador de facturación.	Se encarga de las tareas relacionadas directamente con la generación de facturas (menos permisos que los administradores, más permisos que el operador de cliente).
6	Operador de cliente.	Se encarga de las consultas de los cliente (ve datos de cuenta del medidor y los informes de lectura y brecha del medidor; ve y edita eventos, y realiza lecturas y pings[21, Página 103] on-demand).
7	Operador de conexión remota.	Permite conectar o reconectar dispositivos y tareas de limitación de demanda.
7	Operador de desconexión remota.	Puede realizar la mayoría de las funciones que el Administrador de conexión remota, excepto ESCC y tareas de limitación de demanda.
8	Acceso a datos de exportación.	Se utiliza para funciones de gestión de datos del medidor.
9	Técnico de medidor.	Permite acceso de solo lectura limitado y permiso para realizar lecturas on-demand para medidores.
10	Soporte técnico.	Reinicializar contraseña y habilitar/deshabilitar cuentas.
11	Invitado plus.	Permiso de invitado, además puede realizar lecturas y pings[21, Página 103] on-demand y guardar búsquedas.
12	Invitado.	Acceso solo de vista a dispositivos, grupos, cronogramas e informes básicos.

5.2.3.2.4. Administración de permisos Los permisos del Sistema MDM se encuentran definidos en la siguiente tabla:

Tabla 5.2.5: Permisos del Sistema MDM.

Número	Permiso	Descripción
1	P0001	Alta de usuarios.
2	P0002	Actualización de usuarios.
3	P0003	Eliminación de usuarios.
4	P0004	Alta de dispositivos.
5	P0005	Actualización de dispositivos.
6	P0006	Eliminación de dispositivos.
7	P0007	Alta de tareas programadas.
8	P0008	Actualización de tareas programadas.
9	P0009	Eliminación de tareas programadas.
10	P00010	Alta de grupos de dispositivos.
11	P00011	Actualización de grupos de dispositivos.
12	P00012	Eliminación de grupos de dispositivos.
13	P00013	Alta de *Jobs*.
14	P00014	Exportar datos de dispositivos.
15	P00015	Exportar reportes.
16	P00016	Importar datos.
17	P00017	Conectar servicios.
18	P00018	Desconectar servicios.
19	P00019	Estadísticas.

5.2.3.3. Módulo de reportes

5.2.3.3.1. Generación de reportes El Sistema MDM mediante la capa del controlador debe permitir la generación de reportes de acuerdo al rol y permisos de cada usuario, los reportes deben de se proporcionados bajo los siguientes formatos:

- PDF.
- XML.
- CSV.
- Formato en pantalla.
- Formato de impresión.

5.2.3.3.2. Respaldo de reportes Todos los reportes deben ser respaldados en los directorios locales del Sistema MDM descritos anteriormente para aclaraciones futuras. Independientemente de que los reportes sean generados y tal vez respaldados por sistemas de terceros o por el mismo operador, el Sistema MDM debe implementar mecanismos en la capa del controlador que permitan respaldar dichos reportes autónomamente.

5.2.3.3.3. Notificación de reportes El Sistema MDM debe implementar mecanismos que permitan programar la generación de reportes autónomamente, así mismo cuando dicho reporte se haya generado debe existir una notificación por parte del Sistema MDM que indique que el reporte se ha generado e incluso dicho reporte puede ser enviado al usuario u operador por medio de correo electrónico, una aplicación móvil o de escritorio u otro medio de comunicación y transmisión.

5.2.3.4. Módulo ETL

5.2.3.4.1. Extracción

- El controlador del Sistema MDM debe soportar la extracción información de archivos con extensión XML ya que el(los) Sistemas HES son los encargados de gestionar la información directamente con la red de telecomunicaciones y los dispositivos de medición inteligentes, por lo tanto, el Sistema HES entrega la información en archivos XML con un formato definidor por una norma *(G0100-05)*[2].
- El controlador debe implementar mecanismos que permitan validar el esquema de los archivos XML para evitar inconsistencia de la información o fallos del sistema.
- EL MDM esta diseñado para ser un sistema escalable, por lo tanto, el sistema debe soportar información de diferentes Sistemas HES, los múltiples Sistemas HES interconectados mediante el Sistema MDM pueden administrar dispositivos de diferentes proveedores esto implica que la información de los dispositivos de medición puede estar contenida y estructurada de diferente manera, de tal manera, que el Sistema MDM debe implementar mecanismo que permitan identificar, filtrar, almacenar y procesar los archivos por HES:

5.2.3.4.2. Transformación

- La transformación de eventos debe estar basada en un catálogo predefinido por el cliente y este debe ser soportado por los dispositivos inteligentes de medición y estar previamente registrados en la base de datos del sistema MDM. Cabe mencionar que los dispositivos almacenan los eventos mediante códigos, el Sistema HES al obtener dicha información de los dispositivos de medición y entregarla al Sistema MDM no realiza la conversión de dichos códigos, por lo tanto, el Sistema MDM debe contener un catálogo interno almacenado en una base de datos que permita comparar los códigos recibidos contra los almacenados para realizar la traducción correspondiente.
- El perfil de carga y los registros de los dispositivos de medición debe ser almacenado en unidades de medición de energía en KWh. Si la información entregada por el Sistema HES sobre los dispositivos se encuentra cuantificada en unidades de energía diferentes a los KWh el Sistema MDM debe transformar dichas unidades a KWh antes de ser almacenada.
- Los datos de facturación deben ser almacenados por tarifa y con una unidad monetaria en pesos. El sistema debe transformar la información contenida en los dispositivos de medición antes de ser almacenada para posteriormente ser almacenada y realizar la facturación de los servicio
- Otras transformaciones necesarias y requeridas para el funcionamiento de los sistemas según se defina por el cliente o los desarrolladores.

5.2.3.4.3. Carga

- Los datos deben ser almacenados en la base de datos operacional o en el *Data Warehouse* según sea el caso. Si la información involucra datos generales de los dispositivos, grupos de dispositivos, *Jobs*, *Schedule*, entre otras esta debe de ser actualizada/agregada y sincronizada a la brevedad en la base de datos operacional ya que estos datos son cruciales para el funcionamiento e integridad entre sistemas. Si los datos proviene del almacenamiento de los dispositivos como alarmas, eventos, perfiles, registros, etc. estos deben de almacenarse en el *Data Warehouse* una vez que han pasado por el proceso de transformación para ser utilizados a conveniencia en un futuro.
- Ningún registro puede ser eliminado físicamente de la base de datos de seguridad, base de datos operacional o del *Data Warehouse*, esta restricción tiene como objetivo mantener la integridad de los datos y mantener la información almacenada para aclaraciones futuras, sin embargo el operador puede eliminar registros de los mecanismos de almacenamiento pero dicho borrado es a un nivel lógico, es decir, tanto la base de datos de seguridad, base de datos operacional y *Data Warehouse* cuenta con una columna que permite identificar si la información se encuentra activa o eliminada este verificación la realiza el Sistema MDM internamente, por lo tanto, si el sistema encuentra registros eliminados dichos registros no serán visualizados en la interfaz del sistema sin embargo dichos registros seguirán existiendo.

  Existe una forma de eliminar estos registros para evitar el almacenamiento y/o procesamiento innecesario, el Sistema MDM mediante algoritmos codificados mediante el lenguaje de programación Java y usando el propio motor de base de datos permiten depurar los mecanismos de almacenamiento buscando patrones en la información que permita eliminar registros sin efectos secundarios.

5.2.3.5. Módulo VEE

Antes de que los datos sean cargados por el proceso *Load* del ETL son evaluados por otra serie de procesos llamados VEE (Validación, Estimación y Edición) con el fin de mantener la integridad de los mecanismos de almacenamiento *(principalmente el Data Warehouse)* del Sistema MDM. El proceso VEE consta de tres subprocesos descritos a continuación:

5.2.3.5.1. Validación

- Verificar la fecha y hora de la creación de la información.
- La hora de generación de la información no debe ser mayor a dos horas en comparación con la fecha y hora actual del Sistema MDM.
- La información no debe ser redundante en los mecanismos de almacenamiento, es decir, la información que fue proporcionada por otros Sistemas (Sistema HES y/o sistemas del cliente) no debe encontrarse persistida en la base de datos y/o en el *Data Warehouse* del Sistema MDM.
- La información proporcionada por los otros sistemas debe encontrase completa en su totalidad.
- La información proporcionada debe ser por dispositivos de medición registrados previamente en la base de datos operacional del Sistema MDM para poder asociarla.
- Si el Sistema MDM realiza una petición al Sistema HES la respuesta que proporcione el Sistema HES a dicha petición debe ser validada por el Sistema MDM para verificar que los datos coincidan con la petición solicitada.

5.2.3.5.2. Edición

- Facilitar las operaciones sobre los datos.
- Almacenar metadatos para facilitar operaciones futuras.

5.2.3.5.3. Estimación El controlador debe implementar un módulo de estimación y este debe trabajar bajo dos funciones fundamentales para hacer estimación de datos:

- Debe existir un módulo de funciones para estimar datos faltantes mediante:
  - Regresión lineal.
  - Mínimos cuadrados.
  - Aproximación polinomial.
  - Entropía.
- Debe existir un módulo de funciones para pronosticar datos:
  - Clasificación.
  - Regresión.
  - Segmentación.
  - Análisis de secuencia.

5.2.3.6. Motor de cálculo

El motor de cálculo que incorpora el Sistema MDM en la capa del controlador cumple con dos objetivos fundamentales: *realizar cálculos de información en tiempo real y cálculos de información en segundo plano* cada uno tiene objetivos específicos y proporciona un mayor desempeño al Sistema MDM.

La información que será calculada en tiempo real por el motor de cálculo, por lo general, es información simple y/o las operaciones no son complejas, por ejemplo: la conversión de unidades de medida, operaciones aritméticas simples a un conjunto de datos limitado, etc. por lo tanto, el sistema no necesita de un alto procesamiento o de un tiempo excesivo para entregar resultados, por esta razón el sistema puede realizar estas operaciones en tiempo real mediante una interfaz gráfica que será incorporada en la capa de la vista que puede ser manipulada por el operador del sistema para obtener los resultados requeridos.

Por el contrario la información que será calculada en segundo plano por el motor de cálculo involucrará un alto procesamiento y un tiempo mayor para ser procesada debido a la gran cantidad de registros y/o la complejidad de las operaciones, por esta razón, el motor de cálculo trabajará en segundo plano realizando dichos cálculos, los resultados obtenidos serán almacenados en tablas de hechos agregadas o pre-agregadas dentro del *Data Warehouse* para que el operador pueda acceder a dichos resultados posteriormente.

El operador tomara criterios para decidir que información será procesada en tiempo real y cual será procesada en segundo plano, además al tomar la decisión de que uno o más cálculos serán procesados en segundo plano el operador decidirá la repetición, fecha y hora de dicho cálculo, por lo general la información calculada en segundo plano involucra información registrada por los dispositivos de medición almacenada en el *Data Warehouse*.

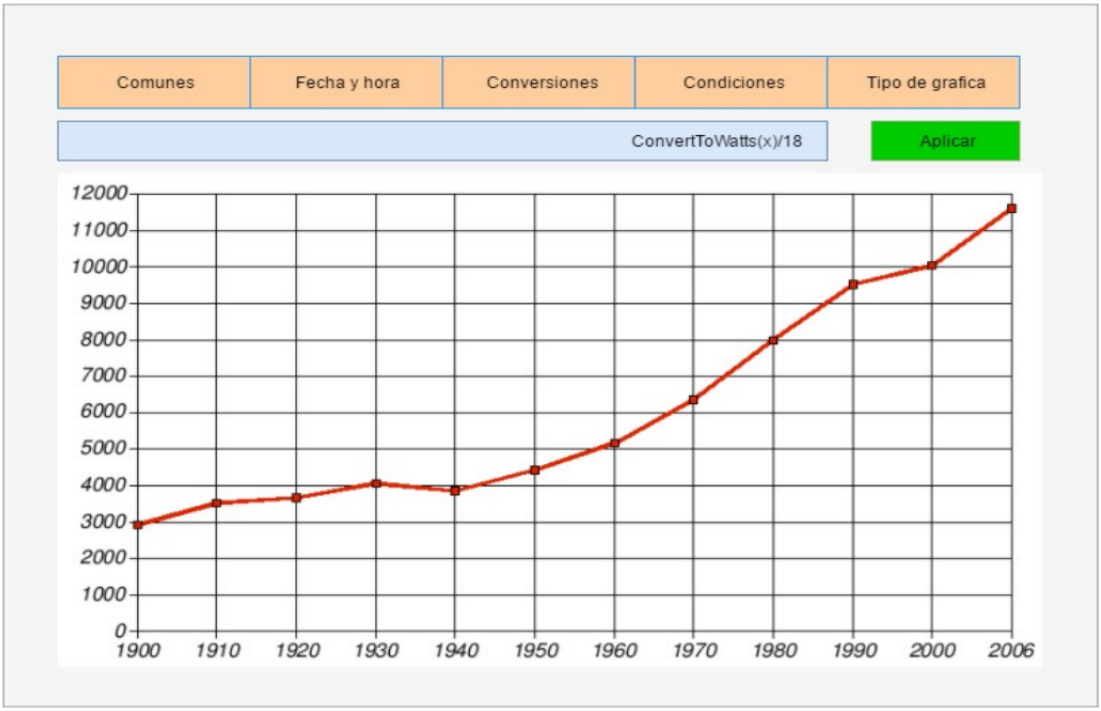


Figura 5.2.9: Prototipo de una GUI para el motor de cálculo en tiempo real (EosTech S.A de C.V, 2017).

5.2.3.6.1. Definiciones generales

- Los datos devueltos por el motor de cálculo son el resultado de una o más operaciones de cálculo en tiempo real o en segundo plano.
- El motor de cálculo utilizara una interfaz gráfica que permita realizar cálculos de información en tiempo real.
- El motor de cálculo incorporara una interfaz y línea de comandos que permita la administración de los procesos en segundo plano tales como: *agregar cálculos, modificar cálculos, eliminar cálculos, definir intervalos de tiempo, etc.*
- El motor de cálculo debe incluir un archivo *.log* que almacene todos los eventos y operaciones realizadas por dicho motor para aclaraciones futuras.
- El motor de cálculo puede hacer uso del motor de base de datos, utilizar el lenguaje de programación y/o de manera híbrida utilizando el motor de base de datos y el lenguaje de programación del sistema para realizar procesamiento de información y obtener resultados.

5.2.3.6.2. Operaciones comunes Las operaciones comunes que debe implementar el Sistema MDM dentro de la capa del controlador tanto para el motor de cálculo como para las operaciones en tiempo real deben estar basadas sobre la clase *Java.Math* utilizando su abreviatura convencional en inglés o según la operación aritmética.

Tabla 5.2.6: Funciones matemáticas y aritméticas de la clase *Java.Math*.

Número	Operador	Descripción
1	+	Suma aritmética.
2	-	Resta aritmética.
3	*	Multiplicación aritmética.
4	/	División aritmética.
5	abs	Devuelve el valor absoluto de un número.
6	acos	Devuelve el arco coseno.
7	asin	Devuelve el arco seno.
8	atan	Devuelve el arco tangente.
9	cos	Devuelve el coseno.
10	sin	Devuelve el seno.
11	tan	Devuelve la tangente.
12	log	Devuelve logaritmo natural en base e de un número.
13	max	Devuelve el valor máximo de un par de valores.
14	min	Devuelve el valor mínimo de un par de valores.
15	sqirt	Devuelve la raíz cuadrada de un número.
16	pow	Devuelve un número elevado a un exponente.

5.2.3.6.3. Condicionales/funciones lógicas Las operaciones de condición y funciones lógicas que debe implementar el Sistema MDM dentro de la capa de programación del controlador para realizar comparaciones, bucles, discriminación de información entre otras se listan a continuación:

Tabla 5.2.7: Operaciones condicionales, estructuras condicionales y funciones lógicas de la capa del controlador del Sistema MDM.

Número	Operador	Descripción
1	()	Operador de agrupación.
2	<	Operador de condicional ***menor que***.
3	>	Operador de condicional ***mayor que***.
4	==	Operador de condicional ***igual que***.
5	!=	Operador de condicional ***diferente de***.
6	<=	Operador de condicional ***menor o igual que***.
7	>=	Operador de condicional ***mayor o igual que***.
8	if	Estructura de control ***si***.
9	while	Estructura de control ***mientras***.
10	for	Estructura de control ***desde-hasta***.
11	case	Estructura de control ***caso n***.
12	and	Operador lógico ***conjunción***.
13	or	Operador lógico ***disyunción***.
14	not	Operador lógico ***negación***.

5.2.3.6.4. Funciones con fecha y hora El Sistema MDM debe entregar resultados precisos y concretos para tales situaciones el sistema debe tomar como base la fecha y hora donde están implicados reportes, resultados, peticiones, etc. que entraron o salieron del Sistema MDM. El Sistema MDM dentro de la capa de controlador debe implementar operaciones con fecha y hora para poder deducir ciertos resultados, las operaciones con fechas debe realizarse mediante la librería propia de Java *Java.util.Calendar y Java.util.Date* según la siguiente tabla:

Tabla 5.2.8: Operaciones con fecha y hora que el Sistema MDM debe integrar.

Número	Operador	Descripción
1	max	Devuelve el máximo de un rango de fechas incluyendo el tiempo.
2	min	Devuelve el mínimo de un rango de fechas incluyendo el tiempo.
3	avg	Devuelve la fecha media de un rango.
4	total	Devuelve la diferencia entre dos fechas incluyendo tiempo.
5	sum	Sumar tiempo a una fecha *(semanas, días, hrs, min y seg)*.
6	res	Resta tiempo a una fecha *(semanas, días, hrs, min y seg)*.

5.2.3.6.5. Conversión de unidades La conversión de unidades debe implementarse como una forma de visualizar los valores obtenidos y almacenados en el Sistema MDM para los diferentes operadores ya que en ocasiones es necesario unidades de medición de energía eléctrica especificas, por lo tanto el controlador incluye mecanismo de conversión de unidades que se listan a continuación.

La conversión de unidades de medida de *corriente* están basadas en las siguientes unidades:

Tabla 5.2.9: Conversión de unidades de energía en Amperios.

Número	Unidad	Descripción
1	nA	NanoAmperios 1000000000.
2	μA	MicroAmperios 1000000.
3	mA	MiliAmperios 1000.
4	A	Amperios 1.
5	kA	KiloAmperios 10^{-3}.
6	MA	Megamperios 10^{-6}.
7	GA	Gigamperios 10^{-9}.

La conversión de unidades de medida de *voltaje* están basadas en las siguientes unidades:

Tabla 5.2.10: Conversión de unidades de energía en Voltios.

Número	Unidad	Descripción
1	nV	NanoVoltios 1000000000.
2	μV	MicroVoltios 1000000.
3	mV	MiliVoltios 1000.
4	V	Voltios 1.
5	kV	KiloVoltios 10^{-3}.
6	MV	MegaVoltios 10^{-6}.
7	GV	GigaVoltios 10^{-9}.

La conversión de unidades de medida de *potencia* están basadas en las siguientes unidades:

Tabla 5.2.11: Conversión de unidades de energía en Watts.

Número	Unidad	Descripción
1	nW	NanoWatts 1000000000.
2	μW	MicroWatts 1000000.
3	mW	MiliWatts 1000.
4	W	Watts 1.
5	kW	KiloWatts 10^{-3}.
6	MW	MegaWatts 10^{-6}.
7	GW	GigaWatts 10^{-9}.

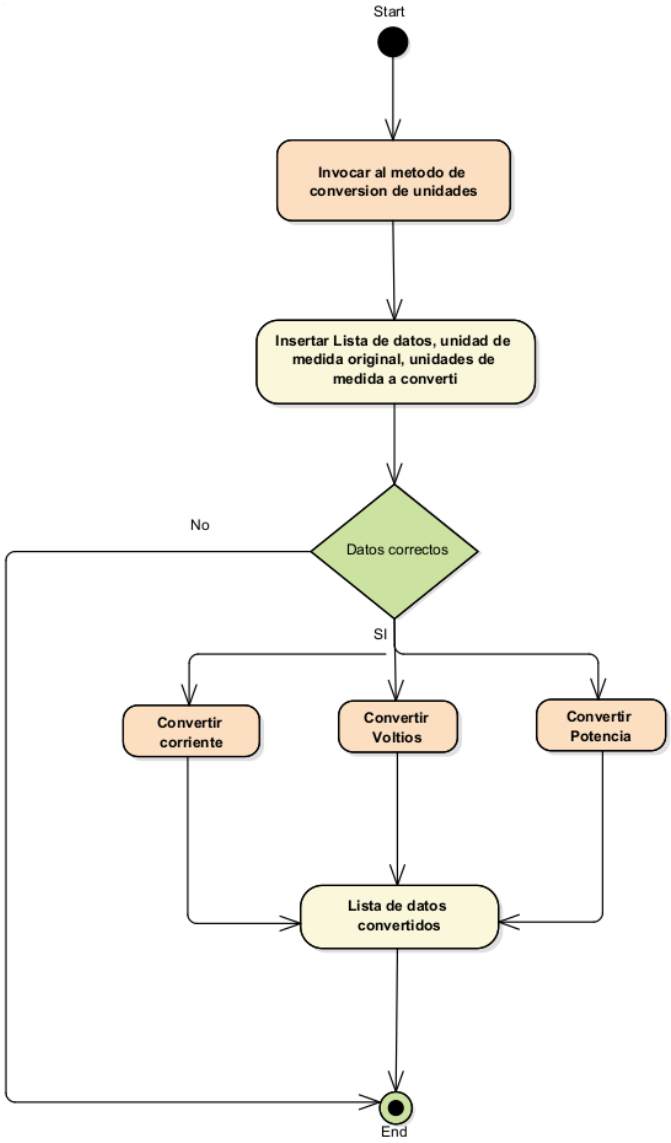


Figura 5.2.10: Diagrama de flujo del motor de cálculo para conversión de unidades de energía.

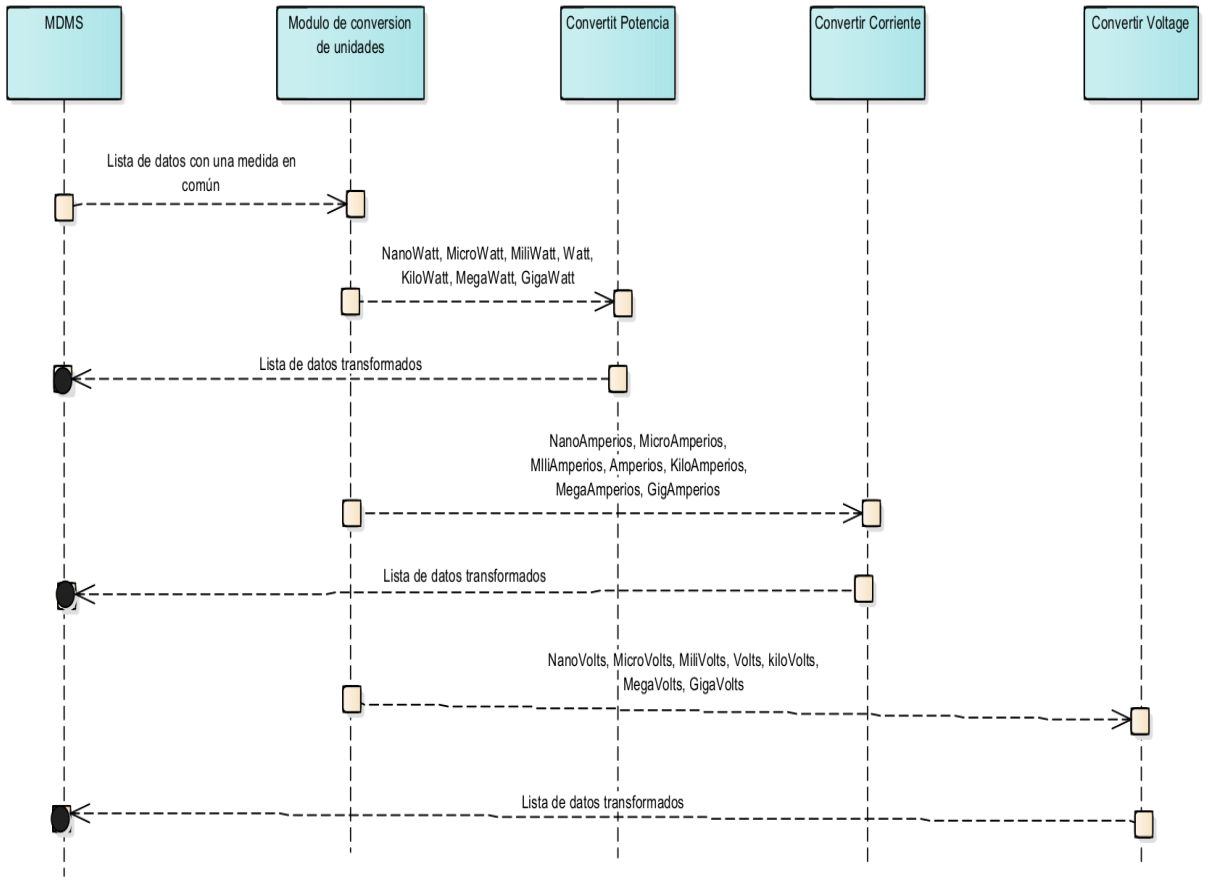


Figura 5.2.11: Diagrama de secuencia del motor de cálculo para conversión de unidades de energía.

5.2.3.7. Módulo de tareas

El módulo de tareas contenido en la capa del controlador del Sistema MDM es el encargado de realizar peticiones al Sistema HES. El módulo esta compuesto por dos sub-módulos fundamentales: *Jobs* y *Schedules*.

El sub-módulo de *Jobs* es el encargado de gestionar en tiempo real una petición al Sistema HES, el proceso involucra desde la validación, construcción y transferencia de la petición, posteriormente el Sistema HES debe procesar la información solicitada, recolectarla y devolverla al Sistema MDM, este proceso debe ser lo más rápido posible según la disponibilidad del Sistema HES y de los dispositivos de medición.

El sub-módulo de *Schedules* incorpora mecanismos que permiten programar la ejecución de instrucciones repetitivas, la información que es proporcionada por el operador *(instrucción(es), fecha, hora, destino, etc.)* es fragmentada y almacenada en la base de datos operacional del Sistema MDM para asegurar que las instrucciones sean ejecutadas según los criterios establecidos por el operador, al ejecutar un *Schedule* este es validado, construido y transferido al Sistema HES para dar respuesta a dicha petición, los resultados devueltos por el Sistema HES son procesados y persistidos en los mecanismos correspondientes de almacenamiento del Sistema MDM según sea el caso.

5.2.3.7.1. *Jobs*

- Gestiona autónomamente un número determinado de repeticiones en caso de que la petición no sea enviada al Sistema HES o que el Sistema HES no de respuesta a la petición en un intervalo de tiempo predefinido.
- El Sistema MDM debe generar un archivo *.log* que registra los eventos y operaciones sobre una petición para aclaraciones futuras.
- El Sistema MDM debe proporcionar una interfaz gráfica de usuario y una línea de comandos que permita gestionar los *Jobs*.
- El Sistema MDM debe proporcionar los resultados en diferentes formatos tales como: PDF, XML, pantalla, Excel, entre otros.

5.2.3.7.2. *Schedules*

- Gestiona autónomamente un número determinado de repeticiones en caso de que la petición no sea enviada al Sistema HES o que el Sistema HES no de respuesta a la petición en un intervalo de tiempo predefinido.
- El Sistema MDM debe generar un archivo *.log* que registra los eventos y operaciones sobre una petición para aclaraciones futuras.
- El Sistema MDM debe incorporar una interfaz gráfica de usuario que permita gestionar los *Schedules (crear, modificar, eliminar).*
- El Sistema MDM debe soportar la gestión de *Schedules* autónomamente mediante la lectura de archivos XML proporcionados por el operador del sistema.
- El Sistema MDM debe informar mediante algún medio de comunicación tales como: correo electrónico, *Short Message Service*–(Servicio de Mensajes Cortos) (SMS), etc. el estado de ejecución de una tarea *(completada, en proceso, fallida, etc.)*.
- El Sistema MDM debe proporcionar los resultados en diferentes formatos tales como: PDF, XML, pantalla, Excel, entre otros.
- El Sistema MDM debe mantener una relación lógica interna entre las peticiones y los resultados de una petición para poder rastrear alguna incongruencia o fallo de información.

5.2.3.8. Módulo de excepciones

El módulo de excepciones que incorpora el Sistema MDM en la capa del controlador cumple con la función de monitorear los *Jobs* y *Schedules* en tiempo de ejecución ya que los Scrip que son creados interna y autónomamente por el sistema para realizar dichas peticiones a menudo suelen ser complejos e involucran una gran cantidad de dispositivos de medición que en ocasiones los dispositivos y/o los Sistema (HES y MDM) no pueden dar respuesta a cierta petición por circunstancias como: información incompleta, incongruente, no asociada, etc. por lo tanto, al ser procesada el Sistema MDM genera excepciones y esto puede causar errores de procesamiento e incluso la falla total del sistema, el módulo de excepciones trata de corregir estas situaciones, al ser detectada una o más excepciones el sistema mediante su módulo de excepciones trata de auto-recuperarse corrigiendo las excepciones y/o errores detectados, si el sistema no es capaz de resolver dichos problemas debe existir un mecanismo de notificación que de aviso al operador para que intervenga y de solución a los errores a la brevedad.

5.2.3.8.1. Catálogo de excepciones El Sistema MDM debe implementar un catálogo de excepciones, es decir, debe definir un código por cada posible excepción que pueda ser generada y soportada por el sistema cuando se ejecuten los *Jobs* o *Schedules*, estos códigos deben ser persistidos por algún mecanismo de almacenamiento estructurado o semi-estructurado por ejemplo: una entidad dentro de una base de datos, un archivo XML, un archivo JSON, etc. al generarse una excepción se debe compara el código de la excepción generado contra los códigos almacenados en el sistema una vez detectado el código correspondiente se debe generar un informe autónomamente de la excepción generada con su respectivo código de excepción, descripción, fecha y hora de generación, así mismo, si la excepción es desconocida y no se encuentra almacenada en el sistema esta debe ser documentada en el mismo informe como desconocida. Los informes puede ser almacenados en un archivo *.log* y notificados al operador mediante alertas o avisos en la capa de la vista del Sistema MDM.

5.2.3.9. Módulo de depuración

El Sistema MDM incorpora un módulo de depuración de información para evitar el almacenamiento y procesamiento excesivo e innecesario en los mecanismos de almacenamiento del sistema, una de las especificaciones del cliente involucra la información almacenada ya que esta es útil solo dos años atrás tomando la fecha actual, por lo cual nos permite tomar como referencia la fecha en la que fue persistida la información que tenga una antigüedad de dos o más años atrás para ser eliminada.

El sistema cuenta con una interfaz en la capa de la vista que permite ejecutar de manera manual el módulo de depuración de registros, la interfaz es muy simple y solo necesita como entrada un número de días para poder ejecutarse, los días indican la fecha de antigüedad de los registros que desean ser mantenidos en los mecanismos de almacenamiento, una vez que el algoritmo es ejecutado manualmente y finalizado la interfaz muestra un resumen de las acciones ejecutadas mostrando el número de registros eliminados y los errores y/o excepciones encontrados. Así mismo se puede automatizar la ejecución de dicho módulo programandolo para que se ejecute periódicamente *(mensualmente, semanalmente, quincenalmente, etc)* a una fecha y hora determinada, los resultados y acciones ejecutadas son almacenadas en un archivo *.log* que permite aclaraciones futuras. Por seguridad e integridad de la información persistida, el Sistema MDM no permite la depuración de registros de dos años atrás *(730 días)* ya que los mecanismos de depuración también incluyen ser ejecutados en el DW, por lo tanto, los algoritmos de depuración cuentan con una restricción al ejecutarse en el DW ya que este debe contener información histórica para poder realizar análisis de datos e inteligencia de negocios.

Otra restricción implícita en dichos algoritmos involucra a la base de datos operacional ya que los algoritmos operaran sobre las entidades de *jobs, schedules, groups* buscando patrones que permitan depurar dichas entidades, el algoritmo además de evaluar la fecha verifica que dichos registros no estén ligados con otros evitando así ser eliminados perdiendo la integridad la información, generar excepciones y/o errores.

El módulo de depuración de información hace uso del motor de base de datos para ejecutar los algoritmos que buscarán patrones que permitan eliminar registros ya que al usar Hibernate®[15, Capítulo 1] como *Object-Relational Mapping*–(Mapeo Objeto Relacional) (ORM) para depurar la información el proceso requiere de un mayor tiempo, esto se debe a que Hibernate®[15, Capítulo 1] es una herramienta muy potente para persistir información pero una de sus desventajas es que es muy lento para consultar grandes volúmenes de información, el motor de base de datos a diferencia de Hibernate®[15, Capítulo 1] esta diseñado para operar con grandes cantidades de registros y entregar resultados en un tiempo relativamente corto, por esta razón el motor de base de datos es una alternativa para ejecutar los algoritmos de depuración.

La mayor parte de los algoritmos de depuración están programados bajo el lenguaje SQL, de igual modo se usa el lenguaje de programación Java que permite realizar algunas comparaciones y/o discriminación de información simple. Una ventaja de usar el lenguaje SQL en este tipo de algoritmos es su rapidez ya que los algoritmos son ejecutados casi al instante, una de las desventajas surge al migrar el sistema a otro Sistema Gestor de Base de Datos diferente al implementado actualmente ya que al incrustar código SQL nativo[15, Capítulo 18] puede generar problemas y/o excepciones de sintaxis al ejecutar los algoritmos de depuración en otro Sistema Gestor de Base de Datos, por lo tanto, los algoritmos pueden ser deficientes y/o no entregar los resultados esperados.

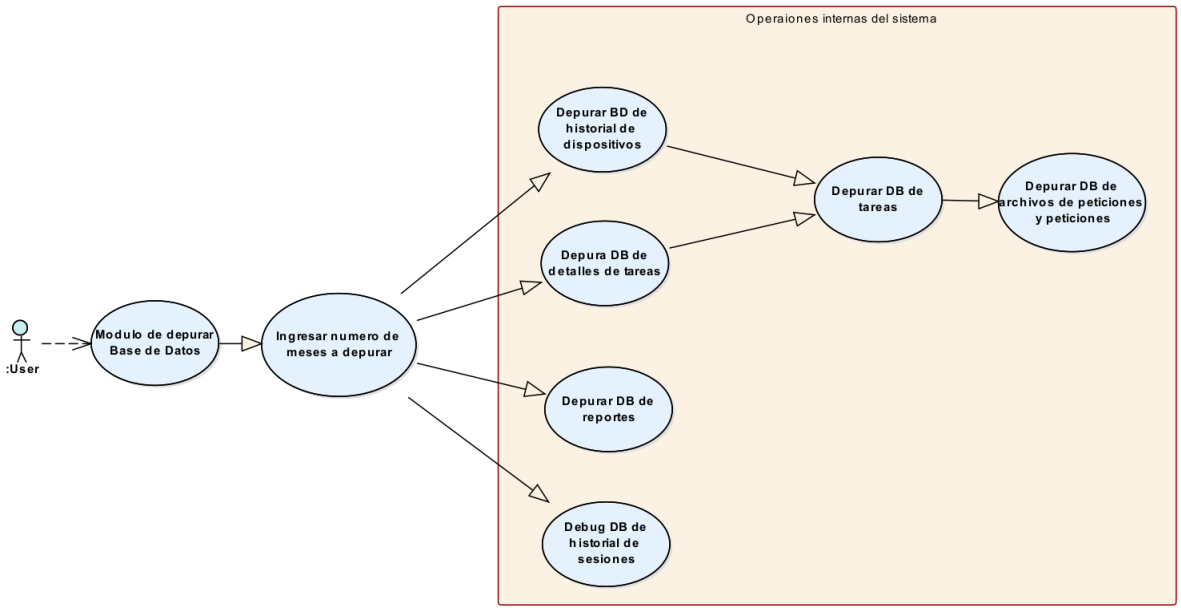


Figura 5.2.12: Diagrama de caso de uso del módulo de depuración de registros.

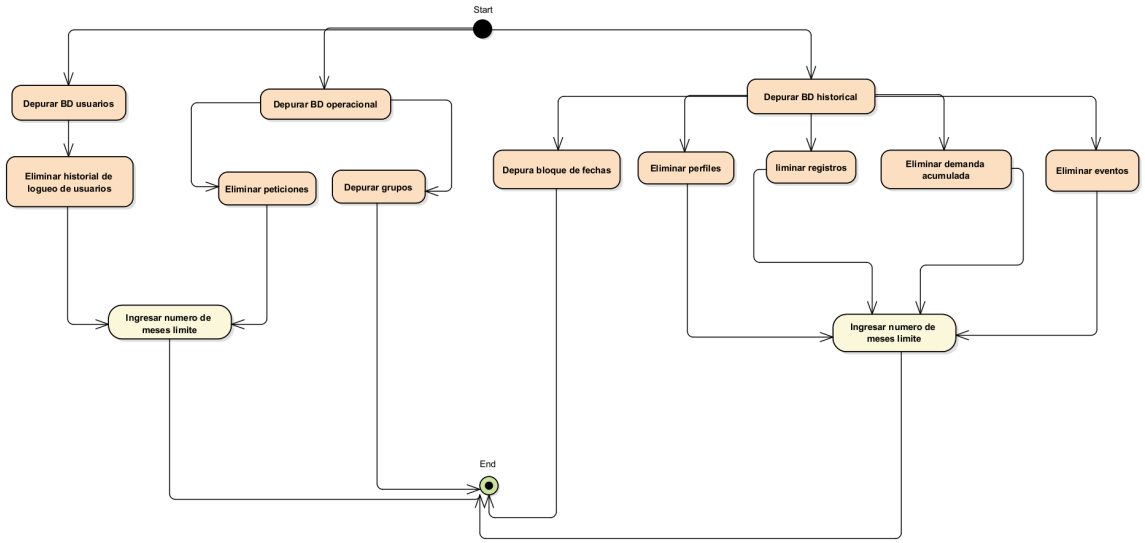


Figura 5.2.13: Diagrama de estados del módulo de depuración de registros.

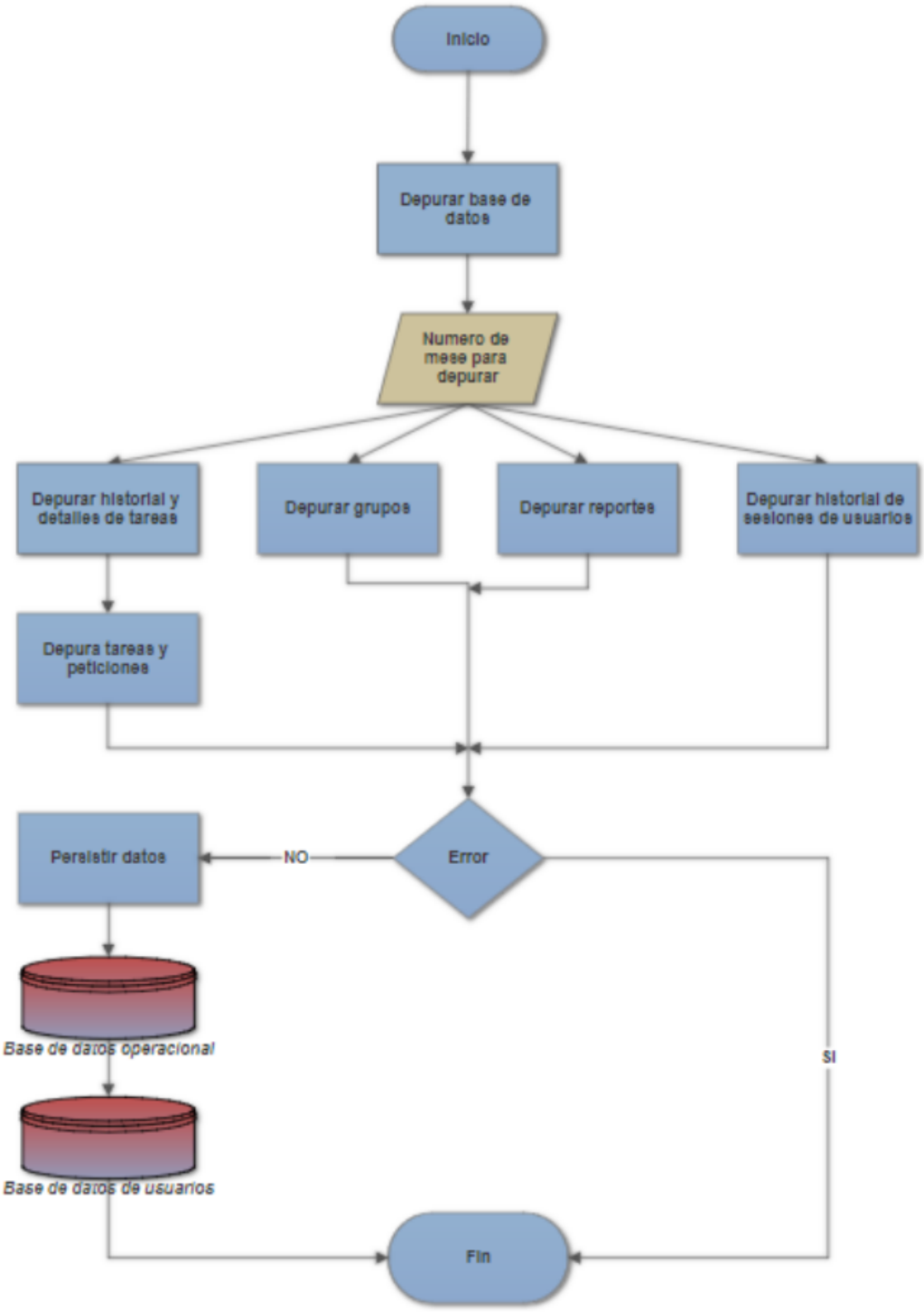


Figura 5.2.14: Diagrama de flujo del módulo de depuración de registros.

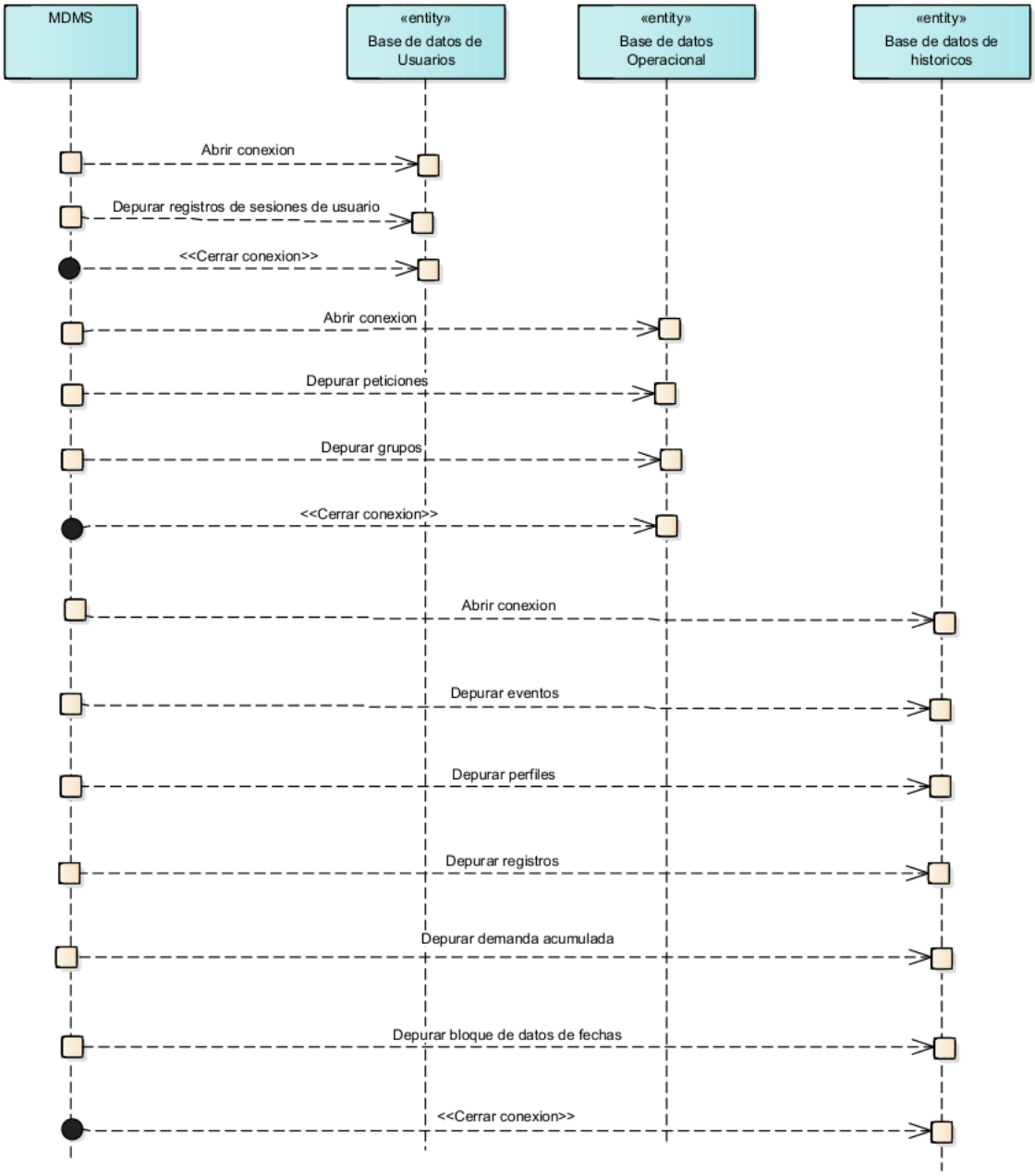


Figura 5.2.15: Diagrama de secuencia del módulo de depuración de registros.

5.2.4. Capa modelo

5.2.4.1. Generalidades:

La capa modelo que implementa el Sistema MDM es la encargada de mantener la integridad y salvaguardar los datos procesados y sin procesar, así mismo implementa mecanismos que permiten respaldar, ofrecer soporte y dar mantenimiento a dichos datos evitando así procesamiento y almacenamiento innecesario.

5.2.4.1.1. ORM Un ORM *(Mapeo Objeto Relacional)* permite agilizar la relación y gestión de los datos entre la aplicación y las bases de datos del Sistema MDM facilitando el manejo de las entidades mediante objetos del propio lenguaje de programación, sustituyendo casi al 100 % el uso del lenguaje propio del motor gestor de base de datos (SQL) para gestionar los datos. Además permite migrar, reestructurar y/o gestionar las entidades de la base de datos más fácilmente al compararlo con la integración de las sentencias SQL incorporadas en el lenguaje de programación del sistema.

Hibernate®[15, Capítulo 1] es una herramienta *(fremework)* de mapeo Objeto-Relacional para la plataforma Java® y .NET® que facilita el mapeo de atributos de una base de datos relacional tradicional y el modelo de objetos de una aplicación mediante anotaciones en clases POJO[15, Capítulo 4] *(Plain Old Java Object)* o mediante archivos declarativos *(XML)* de las entidades que permiten establecer estas relaciones.

5.2.4.1.2. Definición de almacenamiento

- Los mecanismos de almacenamiento implementados por el Sistema MDM pueden estar basados en los servidores de bases de datos relacionales tales como: Mysql®[26], MariaDB®, Oracle®, etc. se recomienda usar Oracle® ya que es un servidor comercial y con un amplio soporte aunque esta decisión será propuesta por los desarrolladores y el cliente tomara la decisión final ya que algunos SGBD[23, Página 22] son libres y otros de licencia.
- Los datos de los dispositivos de medición e información involucrada serán almacenados en el Sistema MDM mediante una base de datos relacional.
- Los datos crudos, es decir, los datos sin procesar serán almacenados en un directorio perteneciente al Sistema MDM con la extensión original de la fuente (XML) manteniéndolos fuera del alcance de los usuarios.

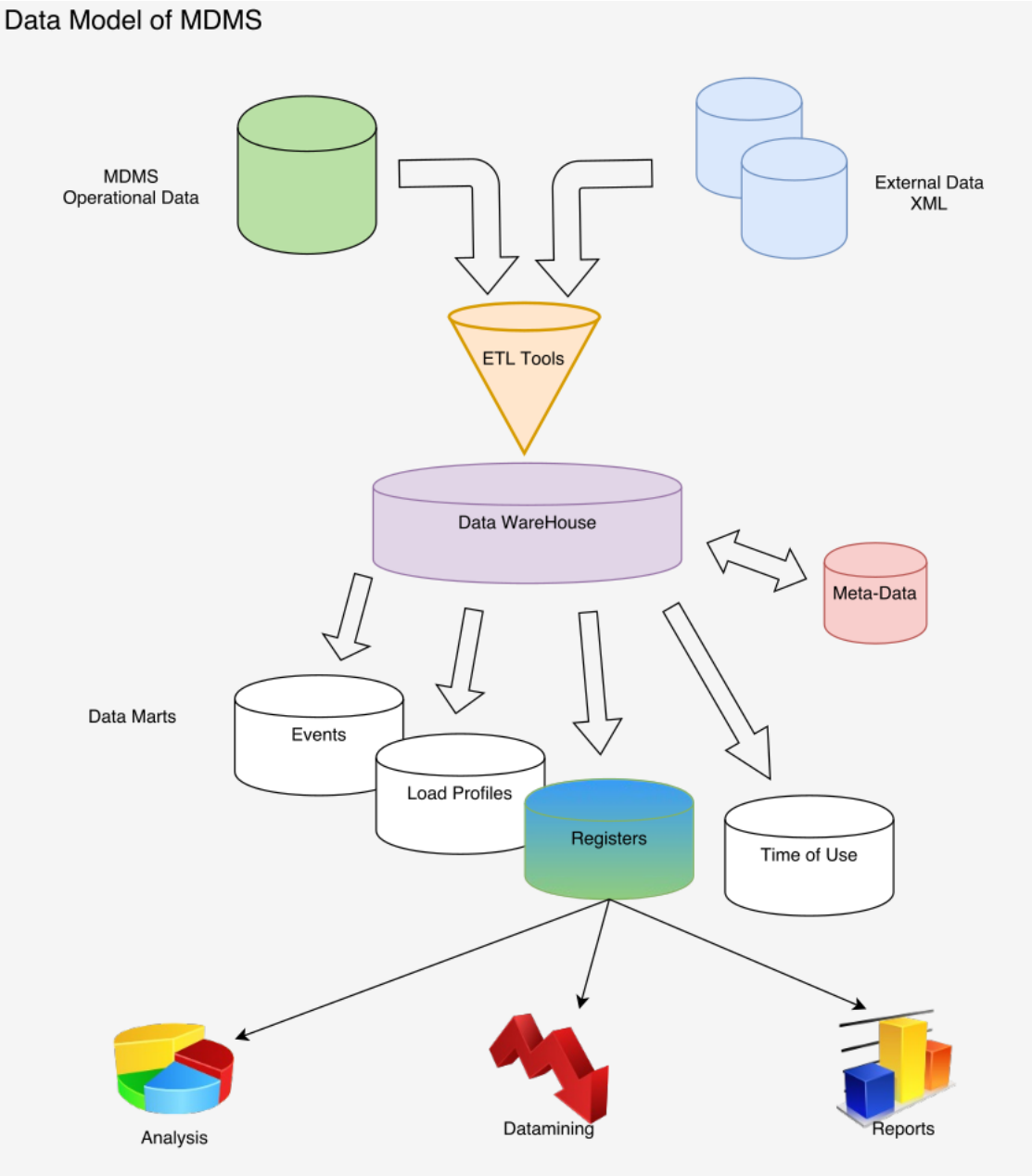


Figura 5.2.16: Modelo de datos del Sistema MDM (EosTech S.A de C.V, 2017).

5.2.4.2. Datos de usuario

Esta base de datos relacional almacena datos de los usuarios y operadores que pueden ingresar al Sistema MDM con sus respectivos roles y permisos con la finalidad de asegurar la integridad y seguridad de los datos y de las operaciones.

5.2.4.2.1. Entidades La base de datos relacional de usuarios debe estar basada en las siguientes entidades fundamentales:

Tabla 5.2.12: Entidades fundamentales para la seguridad del Sistema MDM.

Número	Entidad	Descripción
1	Usuarios	Esta entidad almacena datos generales de los usuarios tales como: *nombre de usuario, contraseña, fecha de registro en el sistema entre otros.*
2	Roles	Esta entidad almacena sobrenombres asignados a los usuarios tales como: *Root, Administrador, invitado, etc.* cada sobrenombre tiene permisos específicos que permiten acceder a módulos del sistema, para asignar los permisos a cada sobrenombre debe existir una entidad puente entre **Roles** y **Permisos** que relacione ambas entidades.
3	Permisos	Esta entidad almacena datos de permisos sobre los módulos del Sistema MDM tales como: *Lectura, Escritura, Ejecución, Lectura-Escritura, etc.* estos permisos serán asignados mediante entidades puente a cada usuarios según sea el caso para determinar las tareas y módulos que pueden visualizar y ejecutar los usuarios.
4	Historial de sesiones	Esta entidad almacena fechas por cada usuario, las fechas almacenadas indican el inicio y finalización de sesiones de los usuarios.

Diagrama Entidad-Relación

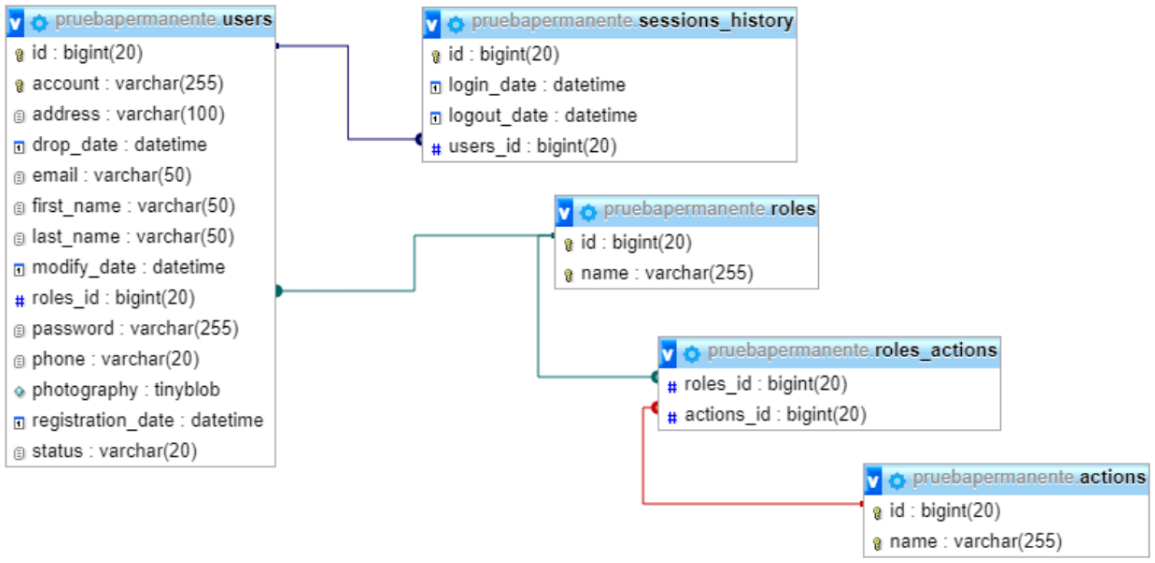


Figura 5.2.17: Diagrama ER de la base de datos de seguridad del sistema MDM.

5.2.4.3. Datos operacionales

Los datos operacionales están almacenados en una base de datos relacional dentro del Sistema MDM, existen datos dentro de este mecanismo de almacenamiento que deben estar sincronizados entre los sistemas del cliente, el Sistema HES y el Sistema MDM debido a que las peticiones suele involucrar dispositivos de medición, por lo tanto, deben estar almacenados y sincronizados por los sistemas involucrados en el proceso, al sincronizar los datos se

asegura la funcionalidad de las peticiones y la integridad de los datos. Así mismo el Sistema MDM almacena más información de interés en esta base de datos operacional que permite el funcionamiento adecuado y autónomo de las operaciones del sistema tales como: asociación de dispositivos de medición para automatizar procesos de lectura, conexión, des-conexión, entre otros, almacenar los *Jobs* y *Schedules*, localización geográfica de dispositivos de medición, etc.

5.2.4.3.1. Entidades La base de datos operacional debe contener las siguientes entidades fundamentales para el funcionamiento adecuado de los sistemas y procesos involucrados:

Tabla 5.2.13: Entidades fundamentales de la base de datos operacional del Sistema MDM.

Número	Entidad	Descripción
1	Catalogo de dispositivos	Almacena tipos de dispositivos de medición registrados y soportados por el MDM.
2	Dispositivos	Almacena características de los dispositivos de medición permitiendo gestionar su comunicación.
3	Grupos de dispositivos	Asocia dispositivos de medición para automatizar los *Schedules* o *Jobs* a un número determinado de dispositivos.
4	Servicios	Almacena las asociaciones de los dispositivos de medición con un número de servicio único ((Registro Patronal Único) (RPU)).
5	Localización	Almacena la localización geográfica de los dispositivos de medición.
6	Sistemas HES	Almacena la ruta de los Sistemas HES soportados e interconectados al Sistema MDM así como un TOKEN que permite la autenticación e interacción entre sistemas.
7	Catalogo de tareas	Almacena una lista de funciones soportadas por el Sistema MDM que el operador puede ejecutar o programar.
8	Tareas	Almacena las tareas por ejecutar así como la asociación de los dispositivos de medición de destino y datos que permiten ejecutar la tarea repetidamente y autónomamente.
9	Catalogo de eventos y/o alarmas	Almacena los eventos que un dispositivo de medición puede soportar.
10	Peticiones	Almacena las peticiones realizadas al Sistema MDM indicando fecha y archivo XML con las peticiones correspondientes y los destinos de dichas peticiones que serán gestionadas a través del Sistema MDM.
11	Historial de tareas	Almacena un registro de las acciones aplicadas a los dispositivos de medición para llevar un control e historial para aclaraciones futuras.

Diagrama Entidad-Relación:

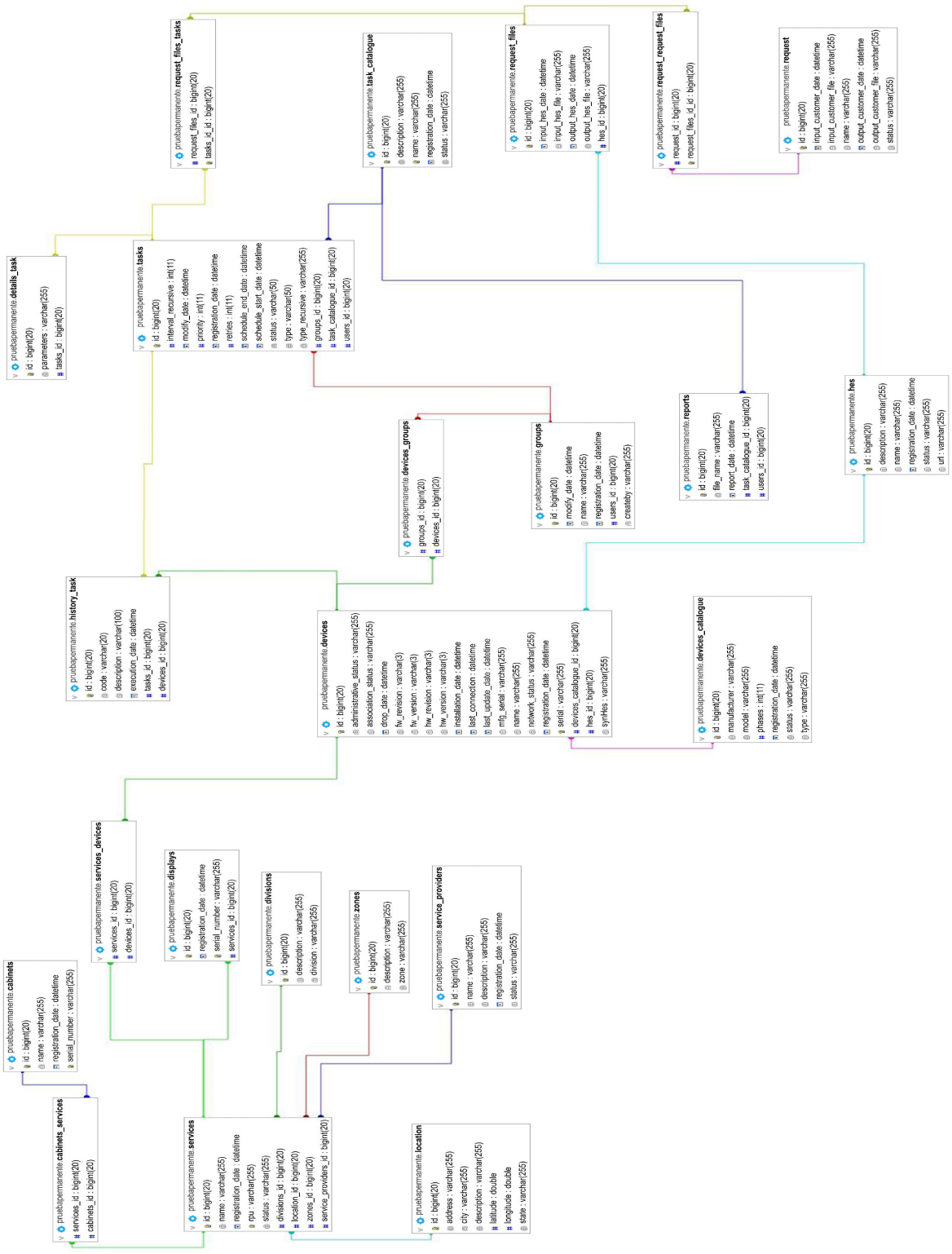


Figura 5.2.18: Diagrama ER de la base de datos operacional del Sistema MDM.

5.2.4.4. Datos históricos

Un mecanismo de almacenamiento y una alternativa para la persistencia masiva de información proveniente de los dispositivos inteligentes de medición es la construcción de un *Data Warehouse* dividido en *Data Marts*, ya que al almacenar la información en este tipo de mecanismos de almacenamiento permitirá realizar análisis y minería de datos para dar solución a los problemas del cliente.

5.2.4.4.1. Definiciones Un *Data Warehouse* es una base de datos corporativa caracterizada por integrar y depurar grandes volúmenes de información proveniente de una o más fuentes, para luego ser procesada permitiendo su análisis desde una infinidad de perspectivas y con una alta velocidad de respuesta. La creación de un *Data Warehouse* implanta una solución completa y fiable de inteligencia de negocios (BI).

Los datos históricos que serán almacenados en el *Data Warehouse* deben cumplir con los siguientes enfoques:

- ***Integrado:*** Estructurado, consistente, deben existir distintos niveles de detalle para adecuarse a las distintas necesidades de los usuarios.
- ***Temático:*** Sólo los datos necesarios deben ser almacenados en el *Data Warehouse*, organizados por temas *(Data Marts)*, agrupados por objetivo para una mejor explotación, organización, mantenimiento y permitiendo hacer escalable el *Data Warehouse* agregando *Data Marts* con el paso del tiempo y de acuerdo a las necesidades del cliente.
- ***Histórico:*** El tiempo es parte implícita de la información. Enfoque de análisis de tendencias y comparaciones.
- ***No volátil:*** El almacén de información existe para ser leído, pero no modificado. La información es por tanto permanente. La actualización está enfocado a agregar los valores que han tomado las variables desde la última actualización.

5.2.4.4.2. Entidades El *Data Warehouse* del Sistema MDM debe estar basado en las siguientes entidades fundamentales para el almacenamiento de los datos históricos de los dispositivos inteligentes de medición:

Tabla 5.2.14: Entidades fundamentales del DW del Sistema MDM.

Número	Entidad	Descripción
1	Eventos	Eventos registrados en los dispositivos de medición incluyendo datos como fecha de registro en el dispositivo y en el sistema.
2	Perfiles	Datos históricos de consumo registrados en los dispositivos de medición incluyendo fechas.
3	Demanda	Datos históricos de consumo registrados en los dispositivos de medición.
4	Calendarios	Datos históricos de calendarios.

Diagrama Entidad-Relación:

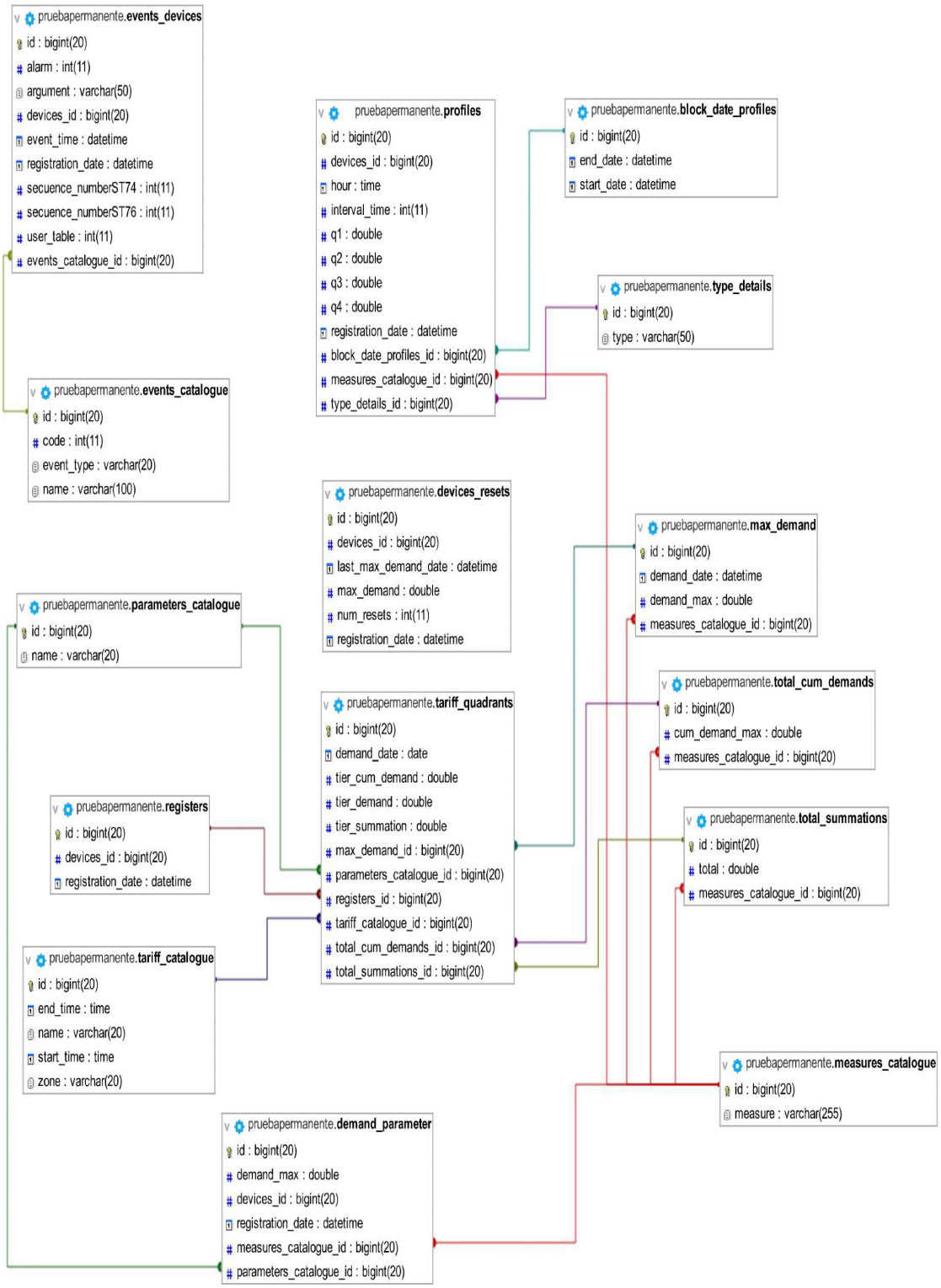


Figura 5.2.19: Diagrama ER del DW del Sistema MDM.

5.3. Plan de pruebas

A continuación se muestra un plan de pruebas elaborado con el fin de entender los objetivos de calidad sobre el desarrollo del software MDM, en el cual se definirán aspectos tales como: los módulos funcionales sujetos a una verificación, tipos de pruebas y entornos. El plan de pruebas tiene como finalidad afinar el proyecto en un futuro mediante la mejora continua.

5.3.1. Análisis de los requerimientos de desarrollo de software

Antes de elaborar un plan de pruebas de software lo primero que debemos hacer es entender los requerimientos del usuario que componen la iteración o el proyecto, ya que estos serán el sujeto de la verificación de la calidad que se va a realizar.

Para realizar esto se analizó toda la información existente tal como: ingeniería de requisitos, especificaciones y diseño funcional, requisitos no funcionales, casos de uso, historias de usuarios, entrevistas con el equipo de ingeniería para aclarar ciertas dudas y así ampliar la información que sea necesaria, así mismo, se hizo uso de algunas preguntas especificas de cada requisito para obtener una visión general del proyecto. Toda esta información se puede resumir en la matriz de trazabilidad (ver tabla 5.3.1) permitiendo ver los módulos más relevantes del sistema.

Al analizar la matriz de trazabilidad se pueden testear los módulos desarrollados permitiendo identificar posibles errores de codificación o errores de operación, de esta manera se asegura el correcto funcionamiento del sistema. Al identificar un error de programación en algún módulo este debe ser corregido de inmediato, las pruebas deben ser ejecutadas tantas veces como sea necesario, es decir, hasta que los defectos de un módulo sean nulos. De igual manera si se encuentran errores de operación estos deben ser documentados, incluyendo las posibles fallas y la acción correctiva que se puede tomar para solucionar dicho error de operación.

Tabla 5.3.1: Matriz de trazabilidad del Sistema MDM (EosTech S.A de C.V 2017).

Número	Requisitos	¿Requisito?	¿Completo?	¿Consistente?	¿No? ambiguo	¿Implementable?	¿Verificable?	¿Único?
1	Transferencia de datos	Si	Si	Si	Si	Si	Si	Si
2	Entrada de datos manual	Si	Si	Si	Si	Si	Si	Si
3	Confirmación	Si	Si	Si	Si	Si	Si	Si
4	Envió de informes al sistema MDM por el sistema Head-End	Si	Si	Si	Si	Si	Si	Si
5	Extracción, Transformación y Carga (ETL)	Si	Si	Si	Si	Si	Si	Si
6	Validación, Edición y Estimación (VEE)	Si	Si	Si	Si	Si	Si	Si
7	Motor de cálculo	Si	Si	Si	Si	Si	Si	Si
8	Almacenamiento seguro	Si	Si	Si	Si	Si	Si	Si
9	Mecanismos de redundancia y recuperación	Si	Si	Si	Si	Si	Si	Si
10	Almacenamiento	Si	Si	Si	Si	Si	Si	Si
11	Ejecución automática de tareas *(Jobs)*	Si	Si	Si	Si	Si	Si	Si
12	Adquisición de datos y envió de comando	Si	Si	Si	Si	Si	Si	Si
13	Consulta de datos seleccionable	Si	Si	Si	Si	Si	Si	Si
14	Uso de prioridades	Si	Si	Si	Si	Si	Si	Si
15	Recepción automática de alarmas	Si	Si	Si	Si	Si	Si	Si
16	Registro del estado e historial de la comunicación con los dispositivos	Si	Si	Si	Si	Si	Si	Si
17	Generación de informes	Si	Si	Si	Si	Si	Si	Si
18	Interfaz de importación de informacion de dispositivos	Si	Si	Si	Si	Si	Si	Si
19	Programación de tareas por excepción o periódicas *(Jobs)*	Si	Si	Si	Si	Si	Si	Si
20	Selección del periodo de las tareas periódicas	Si	Si	Si	Si	Si	Si	Si
21	Selección de los parámetros de las tareas a ser adquiridos de manera automática del dispositivo	Si	Si	Si	Si	Si	Si	Si
22	Selección de las instrucciones de las tareas a ser enviadas de manera automática al dispositivo	Si	Si	Si	Si	Si	Si	Si
23	Programación de programas de tareas *(Schedule)*	Si	Si	Si	Si	Si	Si	Si
24	Creación de grupos de dispositivos para asociarse a tareas y/o a programas de tareas	Si	Si	Si	Si	Si	Si	Si
25	Asignación de un dispositivo o un grupo de dispositivos a tareas y/o a programas de tareas	Si	Si	Si	Si	Si	Si	Si
26	Configuración de roles y contraseñas	Si	Si	Si	Si	Si	Si	Si
27	Programación de informes con elementos seleccionables por el operador	Si	Si	Si	Si	Si	Si	Si
28	Seguridad de acceso a la información	Si	Si	Si	Si	Si	Si	Si
29	Interfaces con el sistema de gestión AMI	Si	Si	Si	Si	Si	Si	Si

5.3.1.1. Identificación de riesgos y definición de planes

A continuación se muestran algunas de las pruebas realizadas a los módulos más relevantes del sistema tomando en cuenta la matriz de trazabilidad, se debe tener presente que los errores de codificación fueron resueltos ejecutando diversas pruebas, al utilizar un paradigma de codificación en bloques también conocido como *bulding blocks* esta tarea se facilito, ya que se tuvo la facilidad de analizar módulo por módulo a detalle y así encontrar posibles fallas y corregirlas tempranamente, si desea ver alguno de los documentos que avalan estas pruebas puede verificar los anexos en la sección A (no se incluyen todos los documentos de pruebas de banco del sistema MDM, solo se incluyen unos cuantos que comprueban la ejecución de dichas pruebas), dichos anexos se encuentran al final de este trabajo de grado, ahí se detallan las pruebas realizadas a los módulos de seguridad y modelo de datos operacional del sistema MDM durante su desarrollo, los documentos de los módulos restantes contiene la misma estructura pero detallando los errores y mejoras encontradas en los módulos restantes.

Por lo tanto, la figura 5.3.1 y 5.3.2 muestra algunos de los errores de operación que pueden dispararse al utilizar un módulo que componen al sistema MDM, de igual manera la tabla contiene la solución a dicho error permitiendo erradicar y prevenir dicho problema.

Para realizar este trabajo se utilizo *Failure Mode Effect Analysis*–(Análisis de Modo y Efecto de la Falla) (AMEF) el cual es un grupo sistematizado de actividades el cual permite:

- Reconocer y evaluar fallas potenciales y sus efectos.
- Identificar acciones que reduzcan y/o eliminen las probabilidades de falla.
- Documentar los hallazgos del análisis.

Así mismo AMEF proporciona un procedimiento disciplinado para identificar las formas en que un producto y/o proceso puede fallar, y planear y/o corregir la prevención de tales fallas.

Existen tres tipos de AMEF los cuales se describen a continuación, las figura 5.3.1 y 5.3.2 esta basada en el primer tipo de AMEF.

- ***AMEF de diseño:*** Se usa para analizar componentes de diseños. Se enfoca hacia los Modos de Falla asociados con la funcionalidad de un componente, causados por el diseño.
- ***AMEF de proceso:*** Se usa para analizar los procesos de manufactura y ensamble. Se enfoca a la incapacidad para producir el requerimiento que se pretende, un defecto. Los Modos de Falla pueden derivar de Causas identificadas en el AMEF de Diseño.
- ***Otros:*** Seguridad, servicio, ensamble.

El AMEF evalúa tres niveles de efecto del modo de falla:

- Efectos locales
  - Efectos en el área local
  - Impactos inmediatos
- Efectos mayores subsecuentes
  - Entre efectos locales y usuario final
- Efectos finales
  - Efecto en el usuario final del producto

La severidad es el valor asociado con el más serio efecto para un modo de falla dado. La severidad es de un rango relativo dentro del alcance del AMEF individual.
La figura 5.3.1 y 5.3.2 muestra una tabla, en la columna *efecto potencial de fallo* describe los efectos de modo de falla en: forma local, el mayor subsecuente y de usuario final (descritos anteriormente), la columna que le precede se observa el valor de la severidad asignado a cada uno de los *efectos potenciales de falla*, este valor debe ser comparado con los valores de la tabla 5.3.2 y así tener una idea del impacto que causaría la aparición de dicho fallo en una etapa/proceso/productos del sistema MDM.

Tabla 5.3.2: Rangos de severidad (AMEF).

Número	Efecto	Rango	Criterio
1	No	1	Sin efecto.
2	Muy poco	2	Cliente no molesto. Poco efecto en el desempeño del artículo o sistema.
3	Poco	3	Cliente algo molesto. Poco efecto en el desempeño del artículo o sistema.
4	Menor	4	El cliente se siente un poco fastidiado. Efecto menor en el desempeño del artículo o sistema.
5	Moderado	5	El cliente se siente algo insatisfecho. Efecto moderado en el desempeño del artículo o sistema.
6	Significativo	6	El cliente se siente algo inconforme. El desempeño del artículo se ve afectado, pero es operable y está a salvo. Falla parcial, pero operable.
7	Mayor	7	El cliente está insatisfecho. El desempeño del artículo se ve seriamente afectado, pero es funcional y está a salvo. Sistema afectado.
8	Extremo	8	El cliente muy insatisfecho. Artículo inoperable, pero a salvo. Sistema inoperable.
9	Serio	9	Efecto de peligro potencial. Capaz de descontinuar el uso sin perder tiempo, dependiendo de la falla. Se cumple con el reglamento del gobierno en materia de riesgo.
10	Peligro	10	Efecto peligroso. Seguridad relacionada - falla repentina. Incumplimiento con reglamento del gobierno.

Los criterio de la tabla 5.3.3 se basa en la probabilidad de que la causa/mecanismo ocurrirá. Se puede basar en el desempeño de un diseño similar en una aplicación similar.
La figura 5.3.1 y 5.3.2 muestra una tabla, en la columna *causa potencial* se deben identificar causas de diseño y mecanismos de falla que pueden ser señalados para los modos de falla, estas causas deben estar regidas por un valor que indicara el nivel de ocurrencia, este valor puede ser consultado en la tabla 5.3.3.

Tabla 5.3.3: Rangos de ocurrencia (AMEF).

Número	Ocurrencia	Criterios	Rengo	Probabilidad de falla
1	Remota	Falla improbable. No existen fallas asociadas con este producto o con un producto casi idéntico.	1	$<$ 1 en 1,500,000
2	Muy poca	Sólo fallas aisladas asociadas con este producto o con un producto casi idéntico.	2	1 en 150,000
3	Poca	Fallas aisladas asociadas con productos similares.	3	1 en 30,000
4	Moderada	Este producto o uno similar ha tenido fallas ocasionales.	4	1 en 4,500
			5	1 en 800
			6	1 en 150
5	Alta	Este producto o uno similar han fallado a menudo.	7	1 en 50
			8	1 en 15
6	Muy alta	La falla es casi inevitable.	9	1 en 6
			10	$>$ 1 en 3

La tabla 5.3.4 muestra un rango de probabilidad de detección basado en la efectividad del sistema de control actual; basado en el cumplimiento oportuno con el plazo fijado.
La figura 5.3.1 y 5.3.2 muestra una tabla, en la columna *situación actual* muestra cual es método de control actual que usa ingeniería para prevenir y detectar el modo de falla, esta columna esta regida por un valor que puede ser contrastado con la tabla 5.3.4 la cual indicara el tiempo de detección de una falla o ¿cuál es la probabilidad de detectar la causa?.

Tabla 5.3.4: Rangos de detección (AMEF).

Número	Rango	Criterio
1	1	Detectado antes de la ingeniería prototipo.
2	2-3	Detectado antes de entregar el diseño.
3	4-5	Detectado antes de producción masiva.
4	6-7	Detectado antes del embarque.
5	8	Detectado después del embarque pero antes de que el cliente lo reciba.
6	9	Detectado en campo, pero antes de que ocurra la falla.
7	10	No detectable hasta que ocurra la falla en campo.

El *Risk Priority Number*–(Número de Prioridad de Riesgo) (RPN) tiene que ser aplicado por y para todo el proceso/diseño. Una vez hecho esto, es fácil determinar las áreas de mayor preocupación. Los modos de fallo que tienen el mayor RPN se debe dar la máxima prioridad para la acción correctiva. Esto significa que no siempre los modos de fallo con los números de mayor gravedad debe ser tratada primero. No puede haber fallas menos graves, pero si fallas que ocurren con más frecuencia y otras que son menos detectables.

La formula para calcular RPN esta dada por:

$$RPN = Severidad * Ocurrencia * Deteccion. \tag{5.1}$$

Donde recordemos:

- La *severidad* representa la gravedad de la falla para el cliente o para una operación posterior, una ves que esta falla ha ocurrido. Se evalúa en escala del 1 al 10.
- La *ocurrencia* es una estimación de la frecuencia con la que se espera que ocurra la falla debido a las causas potenciales.
- La *detección* es una estimación de la probabilidad de detectar, suponiendo que ha ocurrido la falla.

Después de obtener los valores RPN se asignan a la tabla que documenta los diseños/procesos, y se agregan acciones recomendadas atacando o reduciendo las fallas maliciosas en el sistema. Una vez que las acciones han sido implementadas en el diseño/proceso, el RPN nuevo debe ser recalculado usando la formula anterior (5.1), para confirmar las mejoras.

Como ya se menciono anteriormente se deben analizar detalladamente y planear acciones recomendadas para cada uno de los *Critical To Quality*–(Características Criticas de Calidad) (CTQs) (si es posible), con el fin de reducir el riesgo general del diseño, por ejemplo:

- Se deben listar todas las acciones sugeridas por cada uno de los CTQs si es que las hay, además, incluir al responsable y fecha del objetivo de terminación atacando dicho CTQs.
- Se puede incluir información adicional como: la descripción de la acción adoptada y sus resultados.
- Al ejecutar las acciones recomendadas se debe recalcular el RPN para visualizar las mejoras.

La figura 5.3.1 y 5.3.2 muestra el cálculo de los RPN por cada uno de los CTQs que se detectaron durante el desarrollo del sistema MDM, afortunadamente los RPN obtuvieron valores relativamente bajos, incluso algunos con valores de cero, pero recordemos, ninguna falla debe ser omitida o debe ser pasada por desapercibida por más bajo que parezca el valor obtenido por el RPN, por esta razón, algunos de los CTQs a los cuales se les puede aplicar una acción recomendada se lista en la tabla de la figura.

Etapa/Proceso/ Producto	Entrada Clave Proceso	Requerimiento	Modo de fallo potencial	Efecto potencial de fallo	SEV	Causa potencial	OCC	SITUACIÓN ACTUAL		DET	RPN	Acciones recomendadas
								Controles actuales preventivos	Controles actuales de detección			
Módulo de XML	Documento XML	Leer documento XML	No encuentra el documento XML	No se realizan las tareas solicitadas	4	Nombre del documento incorrecto	1	Ninguno	Validar que existe archivo	2	8	Ninguno
		Generar documento XML	Se generan dos o mas documentos con el mismo nombre	Se sobreescribe el documento generando perdida de información	4	Error en la asignación de nombres	1	Generar nombres en base al tiempo del sistema usando una nomenclatura definida.	Validar que aun no existe un archivo con el nombre	2	8	Ninguno
		Validar formato de documento XML	Datos incompletos en el archivo	No se realizan las tareas solicitadas	4	Formato incorrecto	1	Ninguno	Validación de formatos mediante esquema XSD usando la API StAX	2	8	Ninguno
		Generar datos de acuerdo al formato de inserción en la base de datos	La capa de modelo de datos no acepta los parámetros	No se persiste en la base de datos		Formato incorrecto o incompleto	1	Ninguno	Validar la completitud e integridad de los datos		0	Ninguno
Servicio Web	Intercambio de informacion	Descriptor de los servicios que provee el sitio	No genera el wadl	No se puede consumir el servicio por desconocimiento de los datos de entrada y salida.	5	El archivo web.xml no se encuentra correctamente configurado	2	Configuración del web.xml	Ninguno	3	30	Ninguno
		Ninguno	no se encuentra el servicio web	No se puede conectar el cliente para consumir el servicio	5	El servidor web no esta inicializado o no esta instalado.	2	Instalar e iniciar apache	Ninguno	3	30	Ninguno
		Parámetros de entrada con formato especifico	No puede procesar la solicitud	No se realizan las tareas solicitadas	5	Formato incorrecto o datos incompletos	1	Recepción de parámetros a través de JSON	Ninguno	3	15	Ninguno
Servicio SFTP	Session en le servidor SFTP	Establecer conexión	No se puede establecer conexión	No se puede iniciar sesión	4	Servidor caído, error de configuración.	1	Respaldo del archivo de configuración.	Ninguno	3	12	Instalación de un servidor alterno enlazado con un balanceador de carga.
		Ninguno	No se puede establecer conexión	No se puede iniciar sesión	4	IP incorrecto de servidor	1	Tener una IP estática o un dominio propio.	IP estática interna	2	8	Ninguno
		Ninguno	No se puede iniciar sesión	No se puede acceder al directorio	4	Datos de autenticación incorrectos	1	Asignación de cuentas y contraseñas	Ninguno	3	12	Ninguno
		Transferir archivos	No genera error al instante. El error se genera en el momento de procesar el archivo.	Los archivos son transferidos a un directorio incorrecto	4	Directorio incorrecto	1	Ninguno	Definir un árbol de directorio definido	3	12	Ninguno
		Ninguno	No se transfieren los archivos	No se transfieren los archivos	4	Sin permisos en los directorios	2	Asignar permisos a los usuarios según sea necesario.	Ninguno	3	24	Ninguno
		Almacenar datos de administración de usuarios	Inconsistencia en la información	La recuperación de datos de la base de datos es incompleta	5	Servidor caído, error de configuración.	1	Ninguno	Ninguno	2	10	Monitor de servicio
		Ninguno	No se registra un usuario	No se guardan los datos	4	Tipos de datos incompatible	2	Validación de datos desde el módulo de VEE	Validación de datos desde GUI	2	16	Ninguno

Figura 5.3.1: Análisis de Modo Efecto y Falla del Sistema MDM (Parte I).

Módulo de Modelo de datos	Persistencia de datos	Ninguno	No se registra un usuario	No se guardan los datos	2	Esquema de datos incorrecto	2	Generación de tablas desde las clases POJO con un ORM	Mapeo de clases	3	12	Ninguno
		Almacenar datos de operacionales	Se desconoce quien efectuó una operación	No se puede realizar la trazabilidad de las operaciones	2	Las operaciones no se encuentran ligadas con los usuarios.	1	Las tablas operacionales tienen una llave foránea de usuarios.	Ninguno	2	4	Ninguno
		Ninguno	No se inserta una operación	No se puede solicitar una tarea al HES	3	No se encuentra una llave foránea en la base de datos.	2	Manejo de llaves foráneas a través de ORM.	Ninguno	2	12	Ninguno
		Ninguno	No se inserta una operación	Algunos dispositivos no están registrados	2	La base de datos no esta sincronizada.	2	Sincronizar con el HES los dispositivos nuevos	Ninguno	3	12	Ninguno
		Almacenar datos históricos	Datos incompletos en un rango de fechas	No se pueden realizar estadísticas al 100%	4	El HES no proporciona los datos.	2	Algoritmo de estimación de datos faltantes	Ninguno	3	24	Ninguno
		Ninguno	Exceso de tiempo de respuesta	Respuestas lentas	4	Datos excedentes en la base de datos	2	Algoritmo de depuración de datos en base a una fecha manteniendo consistencia en los datos	Ninguno	3	24	Ninguno
Módulo de interfaz gráfica de usuario	Interfaz web	Administración de usuarios	Usuarios acceden a administración de usuarios sin privilegios	Cambios no permitidos	5	Control de privilegios	1	Asignación de roles	Uso del framework Spring Security	3	15	Ninguno
		Administración de dispositivos	No se puede agregar un dispositivo a una tarea	No se puede obtener lecturas de un dispositivo	4	Dispositivo desconocido o eliminado	2	Validar que los dispositivos estén activos en el sistema	Ninguno	3	24	Consultar solo dispositivos con estatus Activado.
		Administración de *Job_Schedule*	No se puede realizar un *Job* o un *Schedule* desde la GUI	No se puede puede adquirir lectura de los datos desde GUI	3	El servicio web del MDM rechaza la petición, por datos incompletos proporcionados desde la GUI	2	Validar que el operador proporcione todos los datos necesarios desde la interfaz gráfica	Ninguno	2	12	Ninguno
		Administración de reportes	No se descarga el reporte	Los usuarios no pueden obtener su reporte en formato de descarga.	4	El reporte no existe porque se reemplazo con otro.	1	Nombrar los reportes en base a la fecha del sistema para evitar perdida de información	Ninguno	2	8	Ninguno

Figura 5.3.2: Análisis de Modo Efecto y Falla del Sistema MDM (Parte II).

Capítulo 6

Conclusiones

El trabajo descrito en esta tesis busca mejorar y automatizar los sistemas de medición de energía eléctrica implementados actualmente en el país, para lograr esto se necesito seguir el modelo de la infraestructura AMI la cual indica que esta conformada por cinco elementos básicos, los primeros tres elementos de la infraestructura AMI *(dispositivos inteligentes de medición, concentrador de datos y redes de comunicación)* fueron implementados por equipos de desarrollo ajenos a este trabajo de tesis involucrando áreas como: sistemas computacionales, electrónica, mecatrónica, entre otras, el cuarto elemento denominado *sistemas de gestión de información* fue dividido en dos subsistemas y trabajando en conjunto conforman el cuarto elemento de la infraestructura AMI, estos dos subsistemas reciben el nombre de HES y MDMS respectivamente. El objetivo de dividir el cuarto elemento en dos susbsistemas dependientes es mejorar el rendimiento y tiempo de respuesta de este elemento dividiendo tareas especificas entre subsistemas, el HES es el encargado de solicitar, decodificar y entrega la información al Sistema MDM proveniente de los dispositivos inteligentes de medición mientras que el Sistema MDM es el responsable de analizar, procesar y persistir la información que provee el sistema HES en sus bases de datos y en un DW para estar disponible cuando se necesite por los operadores y/o sistemas de terceros.

Cabe mencionar que aunque este trabajo de tesis se enfocó a desarrollar el cuarto componente de la infraestructura AMI, no hace ninguna aportación al subsistema HES ya que las aportaciones, diseño e implementaciones estuvieron a cargo de otro grupo de desarrolladores independientes a este trabajo de tesis. Este trabajo realiza aportaciones especificas al subsistema MDM, la aportación más relevante fue el diseño e implementación de un *Data Warehouse* encargado de almacenar información relevante proveniente de los dispositivos inteligentes de medición, de igual manera, este trabajo realiza aportaciones tales como: diseño e implementación de los mecanismos secundarios y de respaldo de almacenamiento, diseño e implementación de los procesos ETL y VEE, diseño e implementación de algoritmos de depuración de información aplicados a los mecanismos de almacenamiento del Sistema MDM, diseño e implementación de un canal de comunicación para interactuar entre el Sistema MDM y el Sistema HES. Dicho canal de comunicación también fue implementado para realizar una comunicación bidireccional entre los sistemas de la compañía y el Sistema MDM, como ultima aportación, el diseño de un motor de cálculo responsable de realizar operaciones aritméticas sencillas y complejas en tiempo real así como en segundo plano con el fin de aumentar el rendimiento del sistema.

Para lograrlo se realizó un análisis previo de las funciones que el Sistema MDM debería brindar, así como la interacción entre el Sistema MDM y los demás sistemas involucrados en la infraestructura AMI, identificando las posibles peticiones y respuestas que el Sistema MDM debería proporcionar, de igual manera se analizaron los datos proporcionados por los sistemas que interaccionan con el Sistema MDM para su almacenamiento y su usabilidad.

Una vez que se analizaron todos los posibles datos de entrada y salida se formo una idea cimentada del producto de software a construir, para lograrlo, se utilizaron ciertas metodologías, procesos y modelos; entre las usadas se encuentra el proceso Scrum para el desarrollo de software, una metodología ágil que se acoplo al desarrollo del sistema permitiendo realizar módulos funcionales del sistema, además, dicho proceso fue combinado con un termino llamado *bulding-blocks*, básicamente este concepto hace referencia a construir un producto de software por bloques o módulos funcionales hasta lograr una construcción completa, concisa y funcional de un producto de software, identificando así las necesidades, requerimientos y posible fallos entre los módulos construidos.

Para los mecanismos de almacenamiento se utilizo la metodología de Hefesto[6] para la construcción del DW, ya que es una metodología muy bien descrita y ofrece los paso y definiciones necesarias para la construcción de un DW, así mismo, se empleo la ayuda de los diagramas o modelos ER creado por Dr. Peter Pin-Shan Chen para visualizar las entidades y atributos para el almacenamiento de datos, por ultimo, los algoritmos que trabajan en conjunto con los mecanismos de almacenamiento y algunas otras fuentes que permiten realizar tareas especificas y definidas dentro del Sistema MDM se diseñaron bajo el paradigma de POO, con el fin de reutilizar el código, hacer escalable el sistema en un futuro y además se facilite el mantenimiento y depuración del mismo, para el modelado de dichos algoritmos se utilizaron diagramas de flujo, diagramas de estados, diagramas de secuencia y/o diagramas de casos de uso.

Tomando como referencia el trabajo desarrollado en esta tesis, se puede concluir que al implementar un DW estructurado en *Data Marts* y alimentándolos mediante procesos ETL y VEE se puede asegurar la integridad y la calidad de los datos, haciendo uso de mecanismos de almacenamiento de apoyo como: una base de datos operacional y una base de datos de seguridad dentro de un Sistema MDM para la gestión de las redes eléctricas del país, se puede mejorar significativamente la calidad del servicio, además de automatizar y agilizar ciertos procesos que hasta el día de hoy son realizados de forma manual mediante personal capacitado.

Las bases que nos permiten afirmar esta hipótesis son las siguientes:

- Los dispositivos de medición de energía eléctrica que hasta hoy en día se encuentran instalados en la mayoría de los hogares y negocios de los usuarios finales, solo almacenan información de consumo de energía por un periodo definido *(al menos en nuestro país)*, esto no quiere decir que sea una mala practica, de hecho, dichos dispositivos realizan su trabajo eficazmente, pero de acuerdo a las tendencias y necesidades tanto de las compañías proveedoras del servicio eléctrico así como los consumidores, el servicio se ha vuelto obsoleto y deficiente. Con la llegada de nueva tecnología y nuevos conceptos tales como la implementación de un Sistema MDM, perteneciente al cuarto elemento de la infraestructura AMI e implementado mecanismos de almacenamiento como un DW y teniendo en cuenta que los nuevos dispositivos de medición son inteligentes, por lo tanto, registran información y eventos adicionales comparándolos con los dispositivos de medición anteriores, gracias a la capacidad de los nuevos dispositivos de medición inteligentes y a las características que brinda la implementación de un DW, se puede almacenar información histórica que permita utilizar los datos a conveniencia, incluso realizar BI que permita mejorar el servicio y la infraestructura.
- El Sistema MDM además de almacenar información histórica de consumos y otra información de interés registrada por los dispositivos de medición inteligentes, también es capaz de almacenar e identificar alarmas. Las alarmas son eventos maliciosos registrados por un dispositivo de medición y/o gabinete, el Sistema MDM emitirá una alerta a un operador para que la alarma sea atendida de inmediato, de esta manera se reducirán las perdidas de energía tanto técnicas como no técnicas que hasta el día de hoy se hace caso omiso o son difíciles de identificar.

- Gracias a la característica de comunicación remota que brindan en conjunto los dispositivos inteligentes de medición, el concentradores de datos, las redes de comunicación y el sistema HES, se puede tener control sobre los dispositivos de medición inteligentes mediante el Sistema MDM, de tal manera que se pueden obtener datos en tiempo real o periódicos, corte y reconexión de los dispositivos inteligentes de medición, entre otras funciones adicionales, esto es posible gracias a los mecanismos de almacenamiento de apoyo que incorpora el Sistema MDM, ya que en la base de datos operacional se incluye la información (*Media Access Control*–(Control de Acceso al Medio) (MAC)[21, Página 65] dispositivo, número de servicio, localización, etc) necesaria para poder gestionar dichas acciones. Así mismo, el DW registra todos los eventos desencadenados por los dispositivos de medición inteligentes para aclaraciones futuras.
- Al contar con mecanismos de almacenamiento tales como un DW incorporado en el Sistema MDM, la implementación de un motor de cálculo mejora significativamente el análisis de los datos, ya que el motor de cálculo incorpora funciones predefinidas que pueden ser aplicadas a un conjunto de datos arrojando así resultados deseados, además, el motor de cálculo tiene la capacidad de realizar cálculos en segundo plano y almacenarlos en entidades dentro del DW denominadas tablas de hechos agregadas o pre-agregadas, almacenando resultados pre-calculados optimizando así el tiempo de respuesta a una petición.
- Una de las bases más significativas que aporta este trabajo es la automatización de los procesos, ya que al contar con un canal de comunicación bidireccional incorporado en el Sistema MDM se pueden solicitar peticiones o dar respuesta a peticiones realizadas por sistemas o subsistemas interconectados al Sistema MDM. Algunas de las peticiones más significativas almacenadas en un catalogo de tareas dentro de la base de datos operacional del sistema son: corte y reconexión remota, lecturas de consumo en tiempo real, lecturas de demanda en tiempo real, entre otras. Como podemos observar, con la implementación de un Sistema MDM se puede reducir la intervención humana automatizando así procesos que hasta el día de hoy son realizados manualmente.

6.1. Trabajos futuros

En futuras investigaciones los temas podrían ser:

- Implementar un módulo que permita operar con dispositivos inteligentes de medición de agua y gas. Esta investigación involucra el diseño e implementación de dispositivos inteligentes de medición adaptados a estos servicios bajo estándares (ANSI C12.18 [29] y ANSI C12.19 [30]), que permitan integrar dichos dispositivos de medición al Sistema MDM utilizados para el servicio de energía eléctrica, haciendo así al Sistema MDM más robusto e integrando y unificando los servicios básicos en un solo sistema que permita su administración. Si son utilizados estándares diferentes para el almacenamiento y transmisión de la información de los dispositivos de medición inteligentes, adaptados a los servicios de agua y gas se debe crear o adaptar el Sistema HES para que además de gestionar la comunicación y decodificar los datos provenientes de los dispositivos estructure la información antes de ser entregada al Sistema MDM, para que este la pueda procesar y persistir en sus mecanismos de almacenamiento sin generar errores y/o excepciones.

- Crear una aplicación para dispositivos móviles o un portal web, el cual permita visualizar a los usuarios finales el consumo actual de energía, historial de consumos, demandas, entre otros datos, ya que hasta ahora los fabricantes de los dispositivos inteligentes de medición de energía eléctrica tiene pensado realizar dicha consulta de datos mediante dispositivos denominado (Módulo Remoto de Energía) (MRE) utilizando PLC para la transmisión de los datos, dichos aparatos son proporcionados a cada uno de los titulares del servicio, por lo tanto, resultaría más práctico consultar la información mediante una aplicación móvil o un portal web, además de que dicha información se encuentra disponible en los repositorios del Sistema MDM bastaría con crear una aplicación y conectarse a los servidores de la compañía para realizar las consultas correspondientes.
- Hacer el sistema más escalable, actualmente el sistema esta diseñado para soportar aproximadamente 100,000 dispositivos, es decir, el Sistema MDM puede gestionar 100,000 dispositivos, para aumentar la cifra de gestión de medidores por el sistema se puede crear un repositorio de base de datos dedicado e independiente de donde se encuentra corriendo el Sistema MDM, de esta manera se puede utilizar más de un servidor ejecutando el Sistema MDM conectando cada uno con el repositorio de datos y mediante balanceadores de carga[28] se puede distribuir la carga entre los N servidores implementados, de esta manera el Sistema MDM podría administrar un mayor número de dispositivos de acuerdo a las necesidades de crecimiento.
- Crear un mecanismo que permita cuantificar la energía entregada contra la recibida en un área especifica y sea reportada al sistema MDM, con el fin de identificar las perdidas de energía técnicas o no técnicas. Los dispositivos inteligentes de medición de energía eléctrica solo están enfocados a cuantificar la energía utilizada por usuario, por lo tanto, identificar perdida de energía técnicas o no técnicas seria una tarea complicada. Por ejemplo, implementar un chip inteligente el los transformadores de cada área que permita cuantificar la energía entregada a los usuarios conectados a dicho transformador y compararla con la suma total de la energía utilizada por los usuarios, de esta manera el Sistema MDM podría identificar la perdida de energía y los operadores podrían tomar acciones necesarias para resolver dicha inconsistencia.

Bibliografía

[1] J. Y. Castillo and L. P. Paniora. Implementation of a Datamart as Solution Business Intelligence for T-Impulso Logistics Area. Technical Report 1, Universidad Nacional Mayor de San Marcos, Perú, 2012. 13, 14

[2] Comisión Federal de Electricidad. *Especificación CFE G0100-05: Sistema de Infraestructura Avanzada de Medición (AMI)*. CFE, México, 2015. 86, 87, 88, 92

[3] I. Corporation. *IBM Cognos Business Intelligence v10*. Pearson plc and IBM Press, 10th edition, 2013. 14

[4] D. C. Costa. *Base de datos: El modelo relacional y el álgebra relacional*, pages 9–31. GNU Free Documentation, Barcelona, España, 1 edition, 2005. 58

[5] B. R. Dario. *Data Warehousing: Investigación y Sistematización de Conceptos*, pages 1–80. Free Software Foundation, Córdoba, Argentina, 2.0 edition, 2010. 42, 44, 45, 47, 48, 53, 54, 55, 56, 57, 58, 60

[6] B. R. Dario. *HEFESTO: Metodología para la Construcción de un Data Warehouse*, pages 81–116. Free Software Foundation, Córdoba, Argentina, 2.0 edition, 2010. 42, 127

[7] J. de Jesús Rocha Quezada, S. B. Rionda, J. M. V. Félix, and I. A. M. Torres. Diseño e implementación de un clúster de cómputo de alto rendimiento. *Universidad de Guanajuato*, (3):24–33, 2011. Guanajuato, México. 39

[8] G. P. Delgado. El modelo de información común para un sistema de distribución inteligente. Master's thesis, Universidad Nacional Autónoma de México, México, 2015. 66

[9] O. Y. Enriquez and H. G. del Busto. Mapeo Objeto/Relacional (ORM). *TELEM@TICA*, 10(3):1–7, 2011. ISSN 1729-3804. 25, 26

[10] P. Especialista en Gestión de Datos. Importancia de un Master Data Management impulsando tu Data Warehouse. 2017. España. 17, 18

[11] I. N. V. Finol. Reglas de codd del modelo relacional. Artículo no publicado. 23

[12] M. G. Ginestà and O. P. Mora. *Base de datos: Bases de datos en PostgreSQL*, pages 7–8. GNU Free Documentation, Barcelona, España, 1 edition, 2005. 36

[13] M. V. C. Gutiérrez. Estudio para la implementación del sistema de Infraestructura de Medición Avanzada (AMI) en una empresa eléctrica regional. Master's thesis, Universidad Politécnica Salesiana, Ecuador, 2011. 5

[14] S. B. R. Jacobo, R. M. J. Antonio, and P. D. L. F. Eduardo. Introducción al modelo CIM de los sistemas de energía eléctrica. *Universidad Pontificia de Comillas*, LXXXVI(5):31–38, 2009. ISSN 0003-2506. Madrid, España. 66

[15] G. King, C. Bauer, M. R. Andersen, E. Bernard, S. Ebersole, and H. Ferentschik. *Hibernate Reference Documentation*. 3.6.10.final edition. 25, 26, 102, 103, 104, 108

[16] O. Linda DeMichiel. *Java Persistence API*. Oracle America, Inc., USA, 2.1 edition, 2013. 25, 26

[17] A. V. Marroquin. Diseño de medidor inteligente e implementación de sistema de comunicación bidireccional. Master's thesis, Instituto Politécnico Nacional, México, D.F, 2013. 13

[18] A. B. Martínez, E. A. G. Lista, and L. C. G. Flórez. ETL processes modeling techniques: an alternatives review and its application in a BI solution development project. Technical Report 1, Universidad Tecnológica de Pereira, Colombia, 2013. 14, 15, 16

[19] L. T. Moss and S. Atree. *Business Intelligence Roadmap: The Complete Project Lifecicle for Desicion-Support Applications*. Addison-Wesley, 9th edition, 2008. 14

[20] A. J. P. Navarro. Aspectos fundamentales de una Infraestructura Avanzada de Medición de energía eléctrica. Master's thesis, Universidad Simón Bolívar, Venezuela, 2016. 12

[21] J. M. B. Ordinas, J. Íñigo Griera, R. M. Escalé, E. P. Olivé, and X. P. Tornil. *Redes de computadores*. GNU Free Documentation, Barcelona, España, 1 edition, 2004. 9, 19, 35, 76, 80, 81, 86, 88, 90, 128

[22] E. M. I. Ortega. Redes de comunicación en Smart Grid. *INGENIUS*, (7):36–55, 2012. ISSN: 1390-650X. Ecuador. 4, 7

[23] R. C. Paré. *Base de datos: Introducción a las bases de datos*, pages 7–27. GNU Free Documentation, Barcelona, España, 1 edition, 2005. 30, 36, 44, 61, 108

[24] O. M. G. Prieto. Cluster de Alta Disponibilidad y Alto Desempeño para Servidores Web (ADAD-SW). *Universidad Nacional del Este-Facultad Politécnica*, (4):35–43, 2008. 37, 39

[25] Y. F. Romero and Y. D. González. Patrón Modelo-Vista-Controlador. *Revista Telem@tica*, 11(1):47–57, 2012. ISSN 1729-3804. La Habana, Cuba. 19

[26] L. A. C. Santillán, M. G. Ginestà, and Óscar Pérez Mora. *Base de datos: Bases de datos en MySQL*, pages 7–19. GNU Free Documentation, Barcelona, España, 1 edition, 2005. 36, 37, 108

[27] A. Silberschatz, H. F. Korth, and S. Sudarshan. *Fundamentos de bases de datos*. McGraw-Hill Inc, 4th edition, 2002. 21, 25, 26, 27, 28, 29, 31, 34, 36, 37

[28] M. M. Sinisterra, T. M. D. Henao, and E. G. R. López. Clúster de balanceo de carga y alta disponibilidad para servicios web y mail. *Informador Técnico*, (76):93–102, 2012. Colombia. 29, 33, 66, 129

[29] A. N. Standard. *ANSI C12.18-2006: protocol Specification for ANSI Type 2 Optical Port*. National Electrical Manufacturers Association, USA, 2006. iv, 68, 128

[30] A. N. Standard. *ANSI C12.19-2008: For Utility Industry End Device Data Tables*. National Electrical Manufacturers Association, USA, 2009. iv, 68, 128

[31] A. N. Standard. *ANSI C12.22-2012: Protocol Specification for Interfacing to Data Communication Networks*. National Electrical Manufacturers Association, USA, 2015.

[32] A. G. Suriol. *La creación de valor en las empresas a través del Big Data*. Universidad de Barcelona, Barcelona, España, 2014. 39, 40

[33] J. G. Valle and J. G. Gutierrez. Definición arquitectura cliente-servidor. Artículo no publicado, 2005. 19, 21, 28, 36, 44, 80

Apéndice A

Documentos de pruebas de banco

Información de número de parte/Descripción

Producto/Descripción	Módulo de encriptacion AES128
Producto	PR001339
Producto/Versión	1.0
Número de procedimiento de prueba	1
Módulo o Funcionalidad	Módulo de encriptación AES128
Fecha de la prueba	19-Marzo-2017
Responsable de la prueba	Ing. Mario Alberto Mendez Carmona
Resultado	Exitoso
Revisado por	M.C. Brigido Rodriguez Cruz

Lista de Equipo

#	Nombre del equipo	Descripción	Fecha de entrega de la calibración.
1	Lenovo G450	Intel Pentium Dual-core 2.20Ghz 4Gb memoria RAM Sistema operativo Linux Debian 8 Jessie	
2			
3			
4			
5			

Configuración de Procedimiento de Prueba

Describa los pasos para configurar el equipo y las conexiones para la ejecución del procedimiento de prueba; agregue imágenes, fotografías o diagramas según sea necesario.

Este módulo es independiente, por lo tanto, no tiene una interfaz gráfica que permita interactuar directamente con este módulo, para que un usuario pueda ingresar mensajes es necesario hacerlo desde el código fuente.

1. Se debe definir una clave para cifrar y descifrar mensajes, esta clave se encuentra en el constructor de la clase y se genera automáticamente cuando se crea el objeto de la clase.

```
EncryptionAlgorithmFrame() throws Exception{
      // Generete a key of 128bits AES
      KeyGenerator keyGenerator = KeyGenerator.getInstance("AES");
      keyGenerator.init(128);
      key = keyGenerator.generateKey();
      key = new SecretKeySpec("This is mi key for encrypt and decrypt a message".getBytes(),  0, 16, "AES");
      aes = Cipher.getInstance("AES/ECB/PKCS5Padding");
}
```

2. Debe crear una clase que permita manipular los objetos de la clase EncryptionAlgorithmFrame, dentro de esta clase se invoca al método encryptFrame, este método espera un parámetro de tipo cadena que contenga el Frame sin encriptar.

```
//Invoke the method encryptString reciving a string as parameter the to
be encrypted
byte [] stringEncrypted  = objProgramAES.encryptFrame("This a String
   encrypted and decrypted by the algorithm AES128");
```

3. Una vez que la clave y el mensaje están definidos, estos serán encriptados:

```
El mensaje encriptado es: 7933b09492c946709b484bfdda8ca92c5e4fbc377f2066ca8b994a58f27e855144591897cac583b6c27cab39f76b6feefd1dc4367ee25fe7a68b59636c
[Finished in 2.1s]
```

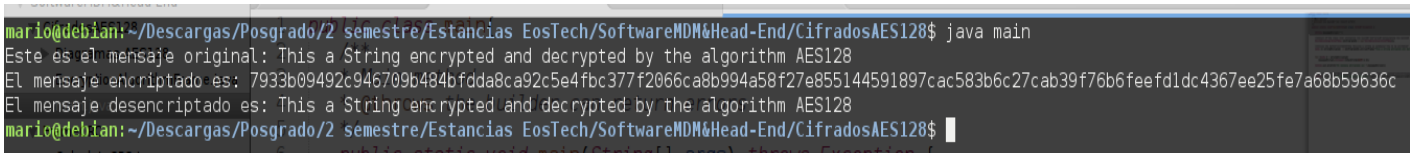
4. La misma clave es utilizada por el Frame para ser descifrado:

```
El mensaje desencriptado es: This a String encrypted and decrypted by the algorithm AES128
[Finished in 3.1s]
```

Caso de prueba TC001001

Descripción del caso de prueba: en este caso de prueba, es introducir una cadena de 61 caracteres que incluyen número, letras y espacios en blanco.

Criterios de aprobación/reprobación: la cadena introducida es igual al original de la secuencia, por lo tanto, esta prueba es correcta. La clave para cifrar y descifrar contiene 48 caracteres y solo son letras.



Método de prueba:

1. Definir la clave.
2. Introducir el mensaje.
3. Compilar y ejecutar el programa.

Observaciones:

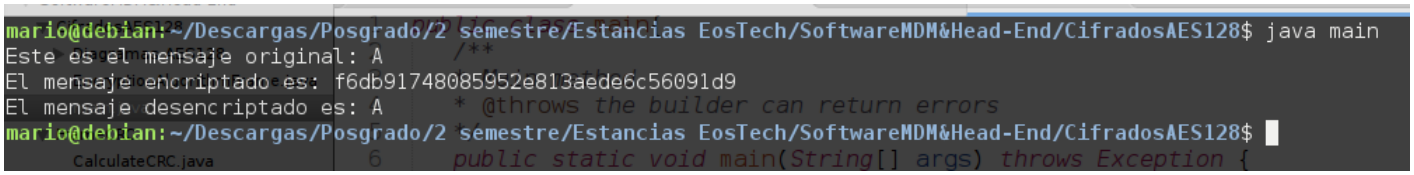
1. El mensaje no puede ser introducido desde una interfaz gráfica o línea de comandos, este mensaje debe ser introducido cuando se invoque la función como parámetro.
2. La clave no se puede introducir desde una interfaz gráfica o línea de comandos, la clave debe introducirse en el código fuente del programa.
3. La clave debe contener al menos 16 caracteres; de lo contrario, no será posible encriptar un mensaje.
4. La clave puede contener más de 500 caracteres, incluidos los espacios en blanco y caracteres especiales.

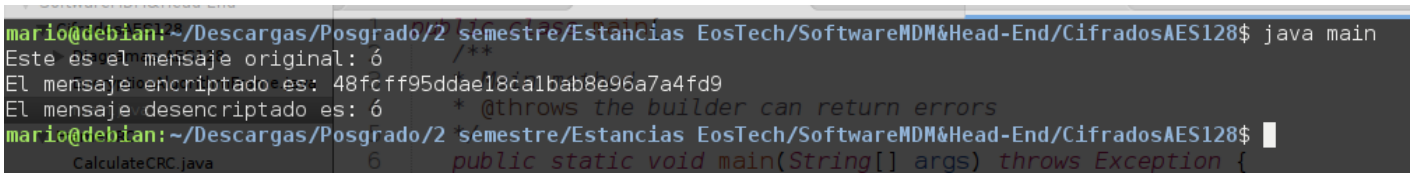
Resultado: Exitoso.

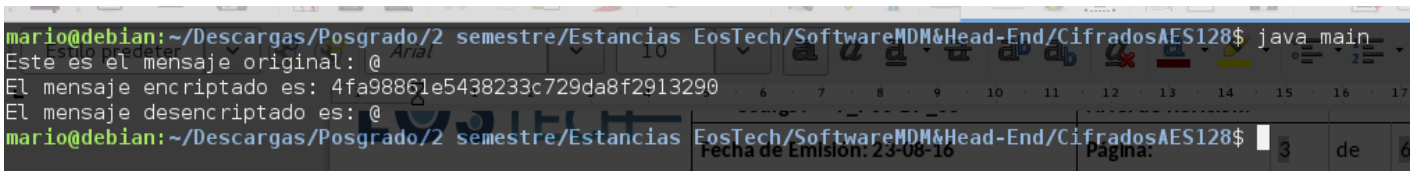
Caso de prueba TC001002

Descripción del caso de prueba: en este caso de prueba se introdujo una cadena de un solo carácter en mayúscula, minúscula, carácter especial y espacio en blanco.

Criterios de aprobación/reprobación: la cadena introducida es igual a la cadena original de la secuencia, por lo tanto, esta prueba es correcta. La clave para cifrar y descifrar contiene 28 caracteres en letras.







Método de prueba:

1. Definir la clave.
2. Introducir el mensaje.
3. Compilar y ejecutar el programa.

Observaciones:

1. El mensaje no puede ser introducido desde una interfaz gráfica o línea de comandos, este mensaje debe ser introducido cuando se invoque la función como parámetro.
2. La clave no se puede introducir desde una interfaz gráfica o línea de comandos, la clave debe introducirse en el código fuente del programa.
3. La clave debe contener al menos 16 caracteres; de lo contrario, no será posible encriptar un mensaje.
4. La clave puede contener más de 500 caracteres, incluidos los espacios en blanco y caracteres especiales.

Resultado: Exitoso.

Caso de prueba TC001003

Descripción del caso de prueba: en este caso de prueba se introdujo una cadena con más de 500 caracteres se utilizó una clave que contiene 48 caracteres, incluidas letras en mayúsculas, minúsculas y caracteres especiales.

Criterios de aprobación/reprobación: la cadena introducida es igual a la cadena original de la secuencia, por lo tanto, esta prueba es correcta. La clave para cifrar y descifrar contiene 48 caracteres que combinan letras en mayúsculas, minúsculas y caracteres especiales.

Método de prueba:

1. Definir la clave.
2. Introducir el mensaje.
3. Compilar y ejecutar el programa.

Observaciones:

1. El mensaje no puede ser introducido desde una interfaz gráfica o línea de comandos, este mensaje debe ser introducido cuando se invoque la función como parámetro.
2. La clave no se puede introducir desde una interfaz gráfica o línea de comandos, la clave debe introducirse en el código fuente del programa.
3. La clave debe contener al menos 16 caracteres; de lo contrario, no será posible encriptar un mensaje.
4. La clave puede contener más de 500 caracteres, incluidos los espacios en blanco y caracteres especiales.

Resultado: Exitoso.

Información de número de parte/Descripción

Producto/Descripción	Módulo de encriptación AES128
Producto	PR001339
Producto/Versión	1.0
Número de procedimiento de prueba	1
Módulo o Funcionalidad	Módulo de encriptación AES128
Fecha de la prueba	22-Marzo-2017
Responsable de la prueba	Ing. Mario Alberto Mendez Carmona
Resultado	Exitosa
Revisado por	M.C. Brigido Rodriguez Cruz

Lista de Resultados

Prueba Proc.	Módulo o Función	Caso de prueba	Descripción	Num. prueba	Pruebas fallidas	Pruebas exitosas	Resultado Exitoso/ Fallido
TP0001	***Módulo de encriptación AES128***	TC001001	Verificar si el mensaje introducido admite letras, espacios en blanco y caracteres especiales.	2	0	2	Exitoso
		TC001002	Verificar si el mensaje introducido puede encriptarse con un solo carácter en mayúsculas, minúsculas, acentos, un espacio en blanco y/o caracteres especiales.	5	0	5	Exitoso
		TC001003	Verificar la longitud máxima y mínima del mensaje y de la clave.	3	0	3	Exitoso

Clave de la revisión: 2017-04-27_1200
Clave del proyecto: SG1701
Nombre del proyecto: MDMS
Documento(s) a revisar con versión:

- **Modelo de datos (DER) de la base de datos operacional**

Link al documento:
10.42.130.133/eostech_archivos/Proyectos/SG1701-MDM/MDM/05_Implementacion/BasesDatos/DEROperacional.pdf
Fecha de revisión: 27 de Abril de 2017
Autor: Ing. Mario Alberto Méndez Carmona
Moderador: Ing. Carlos Domínguez Galván
Revisores: M.C. Brigido Rodríguez Cruz

- Ing. Carlos Domínguez Galván
- Ing Marlon Antonio Aguilar
- Ing. Gustavo Fuentes López

Comentarios:
Tiempo total de preparación: 110 min
Duración de la revisión: 60 min
Cantidad de material revisado: Todo el Diagrama
Resolución de la revisión:

	Completa (sin defectos).
X	Completa, resolución pendiente de defectos encontrados.
	Se requiere re-revisión.

Firma (o Nombre) del Moderador: Ing. Carlos Domínguez Galván

Verifico todos los que defectos encontrados se han resuelto adecuadamente.

Defectos sospechosos:

Número: 1

Severidad	
Ubicación	Diagrama Entidad-Relación
Descripción	
Responsable	Ing. Mario Alberto Méndez Carmona
Resolución	Implementar Catalogo “Status Meter”

Número: 2

Severidad	
Ubicación	Diagrama Entidad-Relación
Descripción	
Responsable	Ing. Mario Alberto Méndez Carmona
Resolución	Implementar Catalogo “Type Cabinete”

Número: 3

Severidad	
Ubicación	Diagrama Entidad-Relación
Descripción	
Responsable	Ing. Mario Alberto Méndez Carmona
Resolución	Delimitar Longitudes campos en el diseño de las tablas

Número: 4

Severidad	
Ubicación	Diagrama Entidad-Relación
Descripción	
Responsable	Ing. Mario Alberto Méndez Carmona
Resolución	Implementar Catalogo “Type of Meters”

Número: 5

Severidad	
Ubicación	Diagrama Entidad-Relación
Descripción	
Responsable	Ing. Mario Alberto Méndez Carmona
Resolución	Implementar Tabla de medidores asignados a un servicio

Número: 6

Severidad	
Ubicación	Diagrama Entidad-Relación
Descripción	
Responsable	Ing. Mario Alberto Méndez Carmona
Resolución	Eliminar campos “time_programmad_end” y “time_programmad_start” en tabla “schedule_Task”

Número: 7

Severidad	
Ubicación	Diagrama Entidad-Relación
Descripción	
Responsable	Ing. Mario Alberto Méndez CArmona
Resolución	Renombrar tabla “task_user” por “catalogue_task”

Apéndice B

Publicaciones

Diseño de la capa de datos de un MDMS para medidores inteligentes de energía eléctrica

Ing. Mario Alberto Méndez Carmona[1], MD. Higinio Nava Bautista[2],
M.C. José Juan Hernández Mora[3] y M.C. Guadalupe Medina Barrera[4].

***Resumen*-- Un Meter Data Management System (MDMS) es un software que permite administrar de manera remota y eficiente los datos de medidores inteligentes, dicho software se apoya de otras tecnologías previamente implementadas como una red Mesh encargada de interconectar dichos medidores y un Head-End System (HES) encargado de recolectar dichos datos, decodificarlos y finalmente entregarlos al MDMS. En el presente artículo se presenta la metodología utilizada para diseñar un Data Warehouse para la capa de datos del MDMS usando algoritmos ETL para la extracción, transformación y carga de los datos, también algoritmos VEE (Validación, edición y estimación), organizando la información en Data Marts, con el fin de homogeneizar, mantener, centralizar y disponer la información de manera consistente para mejorar la trazabilidad de consumo de los clientes.**

***Palabras clave*-- Algoritmos ETL, Algoritmos VEE, MDMS, Data Warehouse, Data Marts.**

Introducción:

Hoy en día vivimos en un mundo donde automatizar cada proceso se ha hecho indispensable, ya sea por comodidad, rapidez, ahorrar costos de operación o simplemente los procesos son muy complejos, tal es el caso de la empresas proveedoras de servicios de agua, luz y gas, estas empresa en conjunto con otras entidades pretenden desarrollar una infraestructura que permita administrar de manera más eficiente sus redes de dispositivos de medición.

Para automatizar los procesos de las empresas proveedoras de servicios se desarrollan infraestructuras de comunicación que involucran redes para interconectar dispositivos de medición utilizando tecnologías como: radiofrecuencia, Wi-Fi, Bluetooth, Internet, entre otras, para lograr una comunicación con diversos dispositivos. Esta red de dispositivos debe ser administrado por sistemas informáticos HES que monitoreen la comunicación y que recopilen información, (eventos, estatus, perfil de consumo, configuración de los dispositivos, etc) de acuerdo con su nivel de inteligencia de cada dispositivo.

La información que es recopilada por los sistemas HES son datos sin procesar que no permiten visualizar el contexto general del comportamiento de los dispositivos que proveen la información, tampoco permite analizar la información aplicando filtros (zonas geográficas, nivel de consumo, consumo promedio, demandas, zonas críticas con alarmas, etc) para ello se implementa un MDMS1 encargados de almacenar y procesar dicha información.

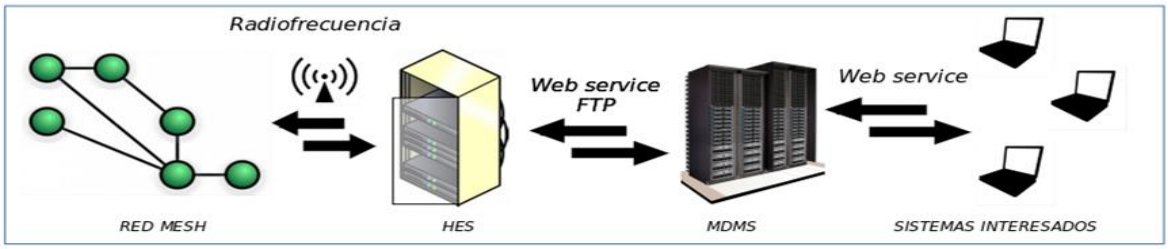


Figura 1. Descripción del sistema en general

[1] Ing. Mario Alberto Méndez Carmona es alumno de la Maestría en Sistemas Computacionales en el Instituto Tecnológico de Apizaco, México mario021793@gmail.com

[2]MD. Higinio Nava Bautista es profesor de la Maestría en Sistemas Computacionales en el Instituto Tecnológico de Apizaco, México higinionava@hotmail.com

[3] . M.C. José Juan Hernández Mora es profesor de la Maestría en Sistemas Computacionales en el Instituto Tecnológico de Apizaco, México jjhmora@itamail.itapizaco.edu.mx

[4]M.C. Guadalupe Medina Barrera es profesora de la Maestría en Sistemas Computacionales en el Instituto Tecnológico de Apizaco, México lupita_medina@hotmail.com.mx

Un MDMS es el encargado de recibir esos datos, procesarlos y guárdalos en una base de datos (Data Warehouse[2]) para posteriormente ser utilizados a conveniencia de la compañía proveedora de energía eléctrica.

El MDMS implementa múltiples algoritmos para su funcionamiento entre ellos se encuentra los algoritmos de ETL[3] (Extract, Transform, Load) y VEE (Validation, estimation, transformation), que permitirán entregar datos estadísticos, históricos, facturación, cortes y reconexiones entre otros, todo esto sin intervención humana simplemente se deben de entregar los datos apropiados para que el MDMS en conjunto con el HES y la red Mesh puedan realizar las tareas solicitadas.

Descripción del método

El MDMS cuenta con dos bases de datos relacionales una de seguridad donde se almacenan las cuentas de usuarios, contraseñas, roles y permisos para que los usuarios puedan acceder al sistema satisfactoriamente, la otra base de datos almacena información operacional para el sistema como: dispositivos, localizaciones, grupos, tareas, etc. esta información debe estar sincronizada con los sistemas de CFE y el HES de otra manera las operaciones solicitadas por alguno de ambos sistemas serian incompletas o no se podrían realizar. La base de datos operacional no solo sirve para el funcionamiento del MDMS sino que también proporciona información mediante minería de datos ya que de esta base de datos se obtiene información de los dispositivos y/o tareas entre otros datos de interés.

El Algoritmo ETL recibe datos de múltiples fuentes tal es así que el MDMS está diseñado para recibir información de múltiples HES o sistemas de datos que provean información operacional y de interés para el MDMS, la primera fase del método ETL es la extracción esta consiste en extraer la información proveniente de los diversos sistemas, esta información será depositada en un directorio dedicado a recibir la información de diversas fuentes en un formato homologo (XML) el algoritmo validara la información mediante archivos XSD puede darse el escenario de que la información sea incompleta o que venga en un formato diferente (por ejemplo, la información de una columna en la base de datos se almacene en KW/h y los datos sean proporcionados en W/h) es aquí donde la segunda fase entra en juego, la fase de transformación homologando dichos datos para que puedan ser persistidos en la base de datos del MDMS sin ningún inconveniente, estos algoritmos forzosamente deben hacer uso de metadatos que indique el tipo de información que se recibe es por eso que se utilizan los archivos con extensión XML y XSD ya que mediante sus etiquetas se pueden obtener dichos metadatos que permitan la transformación de la información correctamente.

Antes de pasar a la fase de carga del algoritmo ETL, es indispensable verificar la información ya que puede existir información faltante o que no esté homologada correctamente los algoritmos de transformación trabajan conjuntamente con otro tipo de algoritmos VEE, estos permiten hacer cálculos matemáticos que permiten estimar información faltante usando información contenida en el Data Warehouse previamente almacenada.

Una vez que la información es extraída y validada mediante los algoritmos descritos anteriormente la fase de carga persistirá los datos en una base de datos y más que una simple base de datos se trata de un Data Warehouse ya que contendrá una gran cantidad de registros de múltiples dispositivos esta información solo va a ser persistida y consultada difícilmente actualizada o eliminada aunque existen algoritmos que depuraran la base de datos cada cierto número de meses para evitar los grandes volúmenes de almacenamiento y procesamiento innecesario (ver Figura 2).

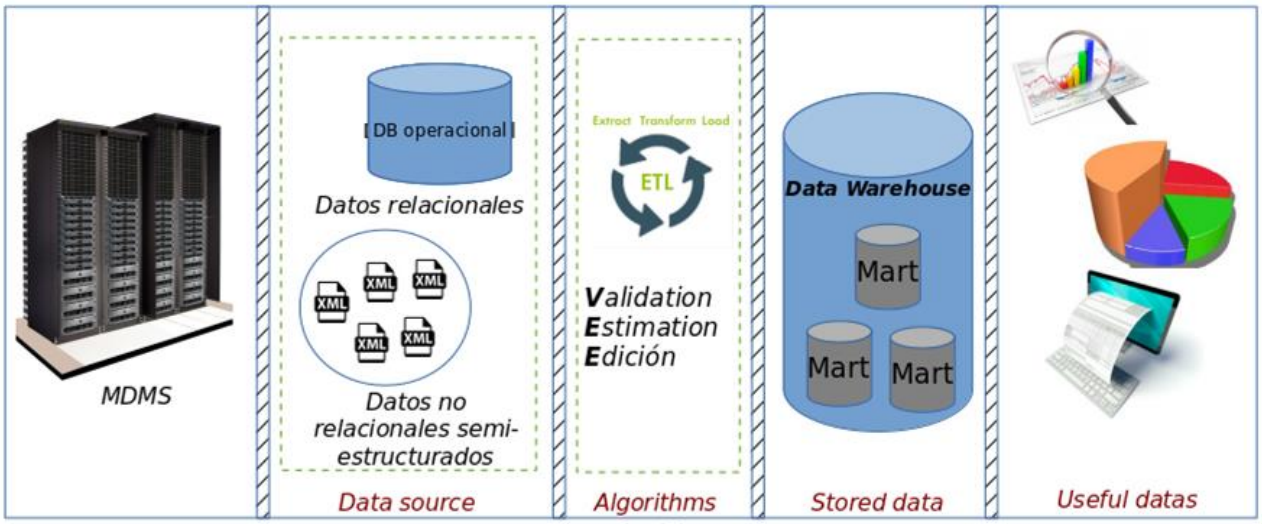


Figura 2. Arquitectura del MDMS

El Data Warehouse está diseñado para almacenar grandes volúmenes de información además cuenta con algoritmos que resumen la información todo esto con el fin de agilizar el cálculo de los datos y así entregar estadísticas rápidamente y evitar la sobre carga de los servidores. El procesos de persistencia consiste en utilizar tecnología ORM[4] (Object-Relational-Mapping) mecanismos que trabajan a base de objetos y no directamente con los lenguajes de base de datos permitiendo migrar de manera más rápida y eficiente los sistemas además de poder usar diversos motores de base de datos sin realizar modificaciones complejas en el código de programación del sistema.

Los Data Marts[5] son parte del Data Warehouse, lo que los caracteriza es que contiene información específica de un tema, de esta manera es más sencillo analizar la información que si tomáramos todos los registros del Data Warehouse para hacer análisis de datos ya que si multiplicamos la cantidad de registros que genera un dispositivo por el número total de dispositivos que administra el MDMS y estos a su vez los multiplicamos por el número de peticiones que los usuarios solicitan obtendríamos miles de registros y esto se traduce en grandes cantidades de cálculo y procesamiento por lo que disminuiría radicalmente el rendimiento del servidor incluso podría llegar a colapsar al no poder dar solicitud a todas las peticiones requeridas. Los Data Marts se agrupan en tres principales temas: eventos, perfiles y registros estos datos y otros más se encuentra registrados en cada dispositivo y están almacenados bajo las norma ANSI C12.18[6] y C12.19[7] estos datos serán recolectados periódicamente ya sea por un servicio o demonio ejecutándose en segundo plano o por petición del usuario, estos datos son recolectados y almacenados en el Data Warehouse ya que contienen información relevante para el cliente por ejemplo las tablas de eventos contiene alarmas y/o eventos que viene definidos o que son configurados por el cliente estas alarmas y/o eventos dan informe de algún suceso que se ha desencadenado, por ejemplo algún fallo en la energía eléctrica, alguna violación de seguridad del medidor, etc. La tabla de registros almacena el consumo y la entrega de energía del medidor con estos datos se pueden hacer los cálculos necesarios para la facturación de los servicios, por ultimo las tablas de perfiles almacenan un historial de los consumos y la demanda máxima que el medidor registra con esta información se puede obtener los históricos del medidor y se pueden realizar estadísticas para interés del cliente.

Comentarios finales:

Para esto se recomienda utilizar JPA[8] (*Java Persistence API*), un conjunto de anotaciones que permiten crear entidades, atributos y relaciones mediante clases POJO[9] (*Plain Old Java Object*) y con la ayuda de herramientas ORM como hibérnate que está disponible para Java se realiza la creación de dichas entidades y la persistencia en el data Warehouse principalmente además de otros datos de interés en otras bases de datos implicadas únicamente utilizando objetos. Con este tipo de arquitectura la integridad de los datos no es violada ya que al trabajar con objetos de Java se pueden evitar estos problemas, también se obtuvieron resultados satisfactorios al momento de migrar el sistema a otros servidores ya que no se tuvieron que realizar cambios complejos en el código de programación para configurar la conexión del sistema con las bases de datos utilizadas. Otro de los beneficios que aporto utilizar este tipo de arquitectura es evitar las consultas complejas que impliquen más de una entidad ya que al crear relaciones direccionales y bidireccionales en las clases POJO se pueden obtener los datos relacionados con alguna entidad en específico de esta manera se vuelve más amigable extraer datos sin necesidad de realizar consultas o joins complejos,

lo mismo sucede cuando tratamos de persistir datos en una entidad que contenga relaciones a otra entidad ya que se pueden ir construyendo los objetos necesarios que necesite una tupla y al último persistirlos de esta manera la integridad de los registros se mantiene, también las transacciones se realizan de forma más sencilla ya que si alguna sentencia falla cuando se intenta persistir un objeto las sentencias que se ejecutaron dentro del bloque que fallo serán destruidas así la integridad de la base de datos se mantiene.

Referencias:

1 Oracle "Overview: Oracle Master Data Management", Recuperado el 6 de Febrero de 2017 de http://www.oracle.com/us/products/applications/master-data-management/mdm-overview-1954202.pdf

2 Mendez, A., Mártire, A., Britos, P. Y Garcia-Martínez, R. "Fundamentos de Data Warehouse", Recuperado el 2 de Marzo de 2017 de http://artemisa.unicauca.edu.co/~ecaldon/docs/bd/fundamentosdedatawarehouse.pdf

3 PowerData, especialistas en gestión de datos, "Procesos ETL La Base de la Inteligencia de Negocio", Recuperado el 23 de Febrero de 2017 de http://cdn2.hubspot.net/hub/239039/file-44151143-pdf/docs/PowerData_-

4 Arvo Lipitsäinen. "ORM – Object Relational Mapping", Recuperado el 1 de Abril de 2017 de http://myy.haaga-helia.fi/~dbms/dbtechnet/labs/dae_lab/Orm.pdf

5 Joseph M. Firestone. "Data Warehouses, Data Marts, and Data Warehousing: New Definitions and New Conceptions", Recuperado el 20 de Marzo de 2017 de http://www.dkms.com/papers/dwdmed.pdf

6 American National Standard, "ANSI C12.18-2006", Recuperado el 10 de Febrero de 2017 de https://www.nema.org/Standards/ComplimentaryDocuments/ANSI-C12-18.pdf

7 American National Standard, "ANSI C12.19-2008", Recuperado el 23 de Marzo de 2017 de https://www.nema.org/Standards/ComplimentaryDocuments/ANSI-C1219-2008-contents-and-scope.pdf

8 Fernando Pech-May, Mario A. Gomez-Rodriguez, Luis A. de la Cruz-Diaz, Salvador U. Lara-Jeronimo. "Desarrollo de Aplicaciones web con JPA, EJB, JSF y PrimeFaces", Recuperado el 6 de Febrero de 2017 de http://www.tamps.cinvestav.mx/~fpech/sd/files/paper001.pdf

9 Bob McCune, "Exploring the Java Persistence API", Recuperado el 17 de Mayo de 2017 de https://www.intertech.com/resource/usergroup/Exploring%20the%20JPA.pdf

ACADEMIA JOURNALS
OPUS PRO SCIENTIA ET STUDIUM

SAN ANTONIO, TX
INFO@ACADEMIAJOURNALS.COM
ACADEMIAJOURNALS.COM

21 agosto 2017

AUTORES: Ing. Mario Alberto Méndez Carmona MD. Higinio Nava Bautista M.C. José Juan Hernandez Mora M.C. Guadalupe Medina Barrera

ARTÍCULO: **Diseño de la capa de datos de un MDMS para medidores inteligentes de energía eléctrica**

ARTÍCULO Núm: TX17-423

Estimados autores,
Con agrado les informamos que, con fecha de hoy, el artículo arriba citado ha sido aprobado para su presentación en el Congreso International de Investigación Academia Journals Tuxpan 2017. El congreso tendrá lugar los días 27 al 29 de septiembre del año en curso, en Tuxpan, Veracruz, México.

El artículo será incluido en las publicaciones del congreso, que incluyen modalidades ISSN e ISBN. (*Los trabajos presentados en el congreso serán publicados por medio de los siguientes vehículos: 1. Volumen Online con ISSN e indización en Fuente Académica Plus en EBSCO y 2. Libro digital ebook con ISBN online).*

Le recordamos que la política del congreso es que para recibir su reconocimiento e incluir su artículo en las publicaciones, es necesario presentar el mismo en el congreso. Le rogamos que utilice su número de artículo en toda correspondencia con Academia Journals.

Saludos cordiales.

Dr. Rafael Moras, P.E.
Editor
Academia Journals
Info@academiajournals.com

AcademiaJournals.com

Congreso Internacional de Investigación Academia Journals Tuxpan 2017
Ciencias y Sustentabilidad

Certificado

otorgado a

Ing. Mario Alberto Méndez Carmona
MD. Higinio Nava Bautista
M.C. José Juan Hernandez Mora
M.C. Guadalupe Medina Barrera

por su artículo intitulado

Diseño de la capa de datos de un MDMS para medidores inteligentes de energía eléctrica

Artículo No. TX17-423

El artículo fue presentado en el congreso llevado a cabo los días 27 al 29 de septiembre del año 2017 en Tuxpan, Veracruz, México y se publicó (1) en el portal de internet AcademiaJournals.com con ISSN 1946-5351 online e indizado por Fuente Académica Plus de EBSCO y (2) en el e-libro intitulado *Investigación en la Educación Superior: Eje de Competencias,* mismo que cuenta con versiones ISBN 978-1-939982-30-8 (CDROM) e ISBN 978-1-939982-29-2 (online).

FACULTAD DE CONTADURIA
ZONA POZA RICA-TUXPAN
DIRECCION

Dr. Edalid Álvarez Velázquez
Presidente de la Comisión Organizadora
Directora de la Facultad de Contaduría
Universidad Veracruzana Región Poza Rica-Tuxpan

Dr. Rafael Moras
Editor, Academia Journals
Profesor de Ing. Industrial
St. Mary´s University, San Antonio, TX, EEUU

Diseño de un motor de cálculo para datos estadísticos de un MDMS para una red de medidores inteligentes

Ing. Mario Alberto Méndez Carmona[1]

***Resumen*-- Un Meter Data Management System (MDMS) es un software que permite administrar una red inteligente de medidores de manera remota y centralizada. Los dispositivos conectados a este tipo de sistemas generan una gran cantidad de información, esta información es almacenada en un Data Warehouse y dividida en Data Marts para tener un mayor control sobre ella. En el presente artículo se describe la metodología usada para la optimización de algoritmos que trabajan en segundo plano, dichos algoritmos permiten generar cálculos sobre los registros de los Data Marts obteniendo datos estadísticos y de facturación que son de utilidad para sistemas de terceros interesados, los datos generados por los algoritmos son almacenarlos de manera resumida en una base de datos secundaria para optimizar el tiempo de respuesta y los recursos del sistema cuando exista una petición por uno o múltiples usuarios.**

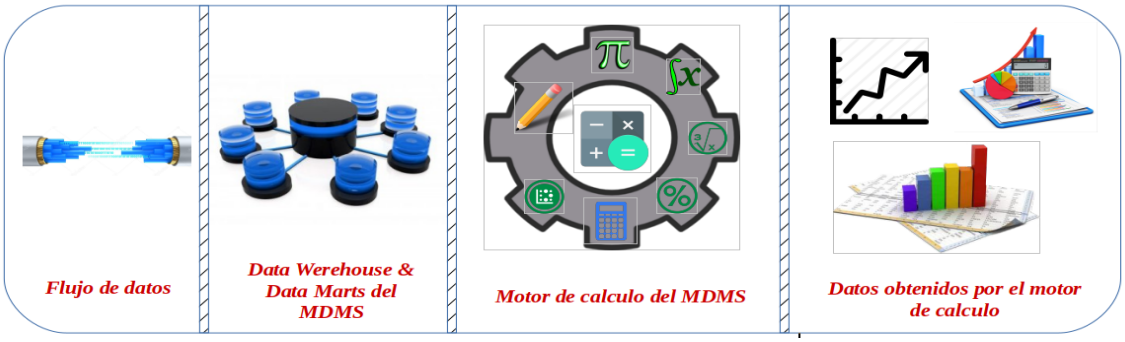
***Palabras clave* – Data Warehouse, Data Marts, Motor de cálculo, Motor de base de datos, demonio y/o servicio.**

Introducción:

Hoy en día vivimos en una era donde podemos encontrar dispositivos de hardware de alto desempeño desde un teléfono celular hasta una supercomputadora capaz de realizar millones de cálculos por segundo (dependiendo de las características del hardware) comparado con hace unos 15 años donde los recursos de hardware y software eran limitados, hoy en día el gran reto es sacarle el máximo provecho a dichos recursos de cómputo para que los procesos sean mucho más rápidos y eficientes ya que al usuario común espera resultados inmediatos, hablando de sistemas pequeños o que no necesitan de un gran poder de cálculo la tarea se facilita pero hablando de sistemas complejos, grandes y robustos la tarea se dificulta.

Empresas proveedoras de servicios (gas, agua, luz, etc.) junto con otras entidades pretenden desarrollar una infraestructura que permita agilizar la recolección de datos de medidores inteligentes para facturación, datos estadísticos, tener un mayor control sobre sus usuarios, etc., por esta razón se crean software como el MDMS[1] que permitan facilitar estas tareas, dichos software transportan miles de registros cada minuto, estos registros tienen que ser procesados y almacenados utilizando ciertos mecanismos.

El software MDMS cuenta con un Data Warehouse[2] que almacena los registros obtenidos de los dispositivos de medición, dicha información está dividida en Data Marts[3] para facilitar su extracción, el problema surge cuando se requieren datos estadísticos que involucran al Data Warehouse ya que este contiene miles y miles de registros en una línea de tiempo indefinida, por lo tanto el cálculo y los resultados son demorados tardando minutos e incluso horas, por esta razón se pretende desarrollar un motor de calculo que permita realizar una serie de operaciones (suma, resta, desviación estándar, media, moda, etc.) en segundo plano (también conocido como demonio y/o servicio) cada X intervalo de tiempo con una interfaz gráfica que permita algunas configuraciones básicas para dicho servicio tales como detener, iniciar o reiniciar el servicio, modificar el intervalo de tiempo con el cual se realizan dichas operaciones en segundo plano, así mismo se podrán agregar, modificar y/o eliminar operaciones del motor de cálculo según sean las necesidades (*Ver figura 1*).



Figu

1 Ing. Mario Alberto Méndez Carmona es alumno de la Maestría en Sistemas Computacionales del Instituto Tecnológico de Apizaco, México mario021793@gmail.com

ra 1. Diseño del motor de cálculo del MDMS.

Descripción del método

El servicio será desarrollado para el sistema operativo Windows Server aunque puede ser adaptado para diferentes plataformas como Unix o Linux ya que la idea es la mismas, además se pueden reutilizar las consultas que se ejecutaran en segundo plano ya que estarán estructuradas en lenguaje SQL[4] (*Structured Query Language*) de OracleDB[5], una de las principales ventajas de OracleDB es que este software de base de datos es multiplataforma por lo tanto puede ser instalado en casi cualquier sistema operativo, el motor de cálculo puede ser adaptado para trabajar en otro tipo de plataformas.

El software permitirá lanzar múltiples hilos[6] lo que hará posible realizar varias operaciones paralelamente optimizando así los recursos de hardware y obteniendo resultados en el menor tiempo posible, se utilizaran sentencias SQL para obtener la información de la base de datos, las sentencias SQL estarán estructuradas para obtener la mayor cantidad de operaciones posibles, es decir se utilizara el motor de base de datos para obtener el mayor número de resultados, las operaciones que no puedan ser realizadas o que sean muy complejas para el motor de base de datos se realizaran con el lenguaje de programación Java.

La persistencia de los datos en los Data Marts del Data Werehouse que proviene de los dispositivos de medición inteligentes se realiza mediante un ORM[7] (*Object-Relational-Mapping*) utilizando JPA[8] (*Java Persistence API*) e Hibernate[9], herramientas de Java que permiten almacenar información en una base de datos relacional mediante clases POJO[10] (*Plain Old Java Object*), estas herramientas son muy potentes y de gran utilidad para los programadores ya que permiten la integridad de los datos y la migración de los sistemas a otras plataformas y/o motores de bases de datos sin hacer grandes modificaciones en el código de programación, por desgracia este tipo de mecanismos son muy lentos al obtener grandes volúmenes de información por esta razón no son muy recomendables para consultas complejas o que involucren una gran cantidad de registros y/o tablas, por lo tanto no podemos hacer uso del ORM para extraer los datos de los Data Marts ya que eso implicaría un incremento de tiempo para obtener los registros necesarios utilizados por el motor de cálculo, sin embargo la mayoría de los motores de base de datos optimizan los recursos de hardware para obtener grandes cantidades de registros en muy poco tiempo, por esta razón el motor de cálculo se apoyara del motor de base de datos utilizando sentencias SQL nativas para la extractaron de los datos.

Las sentencias SQL nativas trabajara sobre tres principales Data Marts contenidos en el Data Warehouse (*Ver figura 2*):

- *Registros*: Este Data Mart almacena información sobre los servicios otorgados a los usuarios actualmente (agua, luz, gas, etc.), con esta información se pueden realizar operaciones aritméticas para obtener el consumo total entregado o recibido de un usuario en específico y así realizar facturación de los servicios al instante y masivamente.
- *Eventos*: Este Data Mart almacena disparadores, es decir alarmas y/o acciones desencadenadas por algún suceso ocurrido en los dispositivos de medición inteligentes, algunos de estos eventos vienen preconfigurados por las normas ANSI C12.18[11] y ANSI C12.19[12] aunque los administradores pueden definir eventos y alarmas propias. Estos datos almacenados en el Data Mart son de utilidad para conocer los estados de los medidores inteligentes y dar aviso de una falla o una violación de seguridad en alguno de los dispositivos.
- *Perfiles:* Por ultimo este Data Mart almacena información estadística e histórica sobre una línea de tiempo indefinida de los consumos y demandas máximas que registra los dispositivos de medición, esta información es de utilidad para realizar estudios de mercado, análisis de información e inteligencia de negocios.

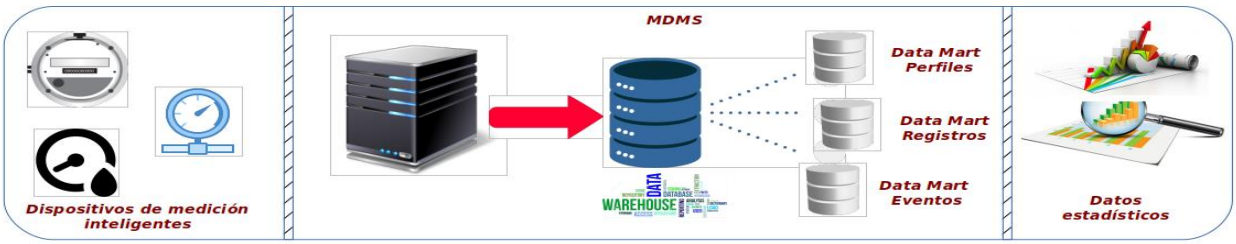


Figura 2. Data Marts contenidos en el Data Werehouse del MDMS.

Estos Data Marts almacenan información de interés para sistemas de terceros, aunque pueden ser creados más Data Marts con el fin de obtener ciertas ventajas o mejores resultados, esto depende de los requisitos y necesidades de las empresas o sistemas interesados en la información otorgada por los dispositivos de medición, el funcionamiento del motor de cálculo se divide la siguiente manera:

- Primero el motor de cálculo ejecutara una o varias(dependiendo de la configuración del motor de cálculo) sentencias SQL nativas en el motor de base de datos para obtener la información necesaria que cumpla con los criterios especificados en la sentencia SQL, cabe mencionar que estas consultas SQL son previamente analizadas y estructuradas por expertos en base de datos ya que algunas de ellas pueden ser grandes o complejas, una vez que la consulta se encuentra estructurada es ejecutada N veces ya sea desde la consola u otro medio que permita arrojar resultados del motor de base de datos, si la consulta arroja los resultados esperados esta puedes ser incorporada al motor de cálculo.
- Las consultas incorporadas al motor de cálculo serán ejecutadas periódicamente en el motor de base de datos cada X intervalo de tiempo (cada hora, al final del día, semanalmente, etc.), el administrador tomara los criterios necesarios para definir este intervalo de tiempo, con el fin de obtener nuevos resultados conforme aumente el número de registros obtenidos por el MDMS y los dispositivos de medición.
- Si los resultados obtenidos por el motor de base de datos necesitan cálculos adicionales para llegar a un resultado esperado, el motor de cálculo tomara los resultados arrojados por el motor de base de datos y aplicara las operaciones necesarias mediante el lenguaje de programación Java para llegar al resultado deseado antes de ser persistido.
- Cuando el motor de cálculo cuente con los resultados esperados estos pueden ser persistidos, los registros pueden ser persistidos de dos maneras, el primero método consiste en crear una nueva base de datos idéntica al Data Werehouse con la novedad de que dicha base de datos solo contendrá datos resumidos, por lo tanto, el número de registro se reduciría exponencialmente, el rendimiento y el tiempo de respuesta mejoraría además, permitiría tener otro servidor de base datos muy independiente que permita agilizar el rendimiento y almacenamiento del sistema, el segundo método consiste en almacenar los datos en el mismo Data Werehouse del MDMS junto con los registros ordinarios e identificarlos con algún campo que indique que es un resultado arrojado por el motor de cálculo (o un registro resumido), cuando estos registros sean requeridos pueden ser tomados solo los valores que han sido procesados por el motor de cálculo (datos resumidos) y no el total de registros que contiene el Data Werehouese, aunque este método no requiere crear una nueva base de datos no es muy recomendable ya que el motor de cálculo tomara todos los registros del Data Werehouse para ir discriminando los registros que no cumplan con los criterios especificados en la consulta SQL, esto implicaría que la respuesta se ejecute en un tiempo mayor al compararlo con el primer método.
  Por esta razón el MDMS se apoyará de una nueva base de datos que permita almacenar los registros arrojados por el motor de cálculo.
- Cuando exista una petición de uno o múltiples usuarios que requieran algún dato estadístico predefinido en la interfaz del MDMS el sistema buscara los resultados en la base de datos secundaria que almacena los datos resumidos y arrojara el resultado instantáneamente o al menos en un tiempo menor que si se utilizara cualquier otro método de almacenamiento o procesamiento ya que estos datos se encuentran preprocesados en la base de datos secundaria por lo tanto, no necesitan de la extracción de la fuente original y del procesamiento de la información, simplemente se arroja el resultado esperado optimizando así el tiempo de respuesta.

Esta es una vista general de como es el funcionamiento del motor de calculo que incorpora el MDMS con el fin de brindar un rendimiento mayor del sistema y así optimizar el tiempo de respuesta hacia los usuarios finales.

Comentarios finales:

Sin duda la automatización de los procesos cada día se hace una realidad, para estas automatizaciones se necesitan servidores de computo que permitan procesar y almacenar los grandes volúmenes de información proveniente de diversos sistemas embebidos, el software y hardware que administre los procesos automatizados debe ser altamente eficiente y reducido en costos, es por eso que se debe optimizar los recursos de hardware y de software al 100% construyendo sistemas informáticos altamente eficientes ya que los usuarios finales encargados de manejar y administrar dichos sistemas esperan resultados inmediatos y con un margen de error lo más nulo posible. El MDMS es un software que permitirá administrar de forma remota dispositivos de medición inteligentes, en este artículo se

describieron algunos de los métodos utilizados para optimizar el rendimiento del software que se va a ser ejecutado en los servidores principales pertenecientes al MDMS.

Para poder hacer el modelado del motor de cálculo se hizo un análisis de sistemas parecidos al MDMS con un aproximado mayor o menor de registros que procesa el MDMS por minuto llegando así a la conclusión de que un motor de calculo que trabaje en segundo dará solución al procesamiento masivo y liberar carga de los servidores, brindando así una mayor experiencia a los usuarios finales del MDMS.

Referencias:

1 Oracle "Overview: Oracle Master Data Management", Recuperado el 6 de Febrero de 2017 de http://www.oracle.com/us/products/applications/master-data-management/mdm-overview-1954202.pdf

2 Mendez, A., Mártire, A., Britos, P. Y Garcia-Martínez, R. "Fundamentos de Data Warehouse", Recuperado el 2 de Marzo de 2017 de http://artemisa.unicauca.edu.co/~ecaldon/docs/bd/fundamentosdedatawarehouse.pdf

3 Joseph M. Firestone. "Data Warehouses, Data Marts, and Data Warehousing: New Definitions and New Conceptions", Recuperado el 20 de Marzo de 2017 de http://www.dkms.com/papers/dwdmed.pdf

4 Hans-Petter Halvorsen. "Structured Query Language", Recuperado el 17 de Febrero de 2017 de http://home.hit.no/~hansha/documents/database/documents/Structured%20Query%20Language.pdf

5 Oracle® Database. "SQL Reference 10g Release 2 (10.2) B14200-02", Recuperado el 02 de Abril de 2017 de https://docs.oracle.com/cd/B19306_01/server.102/b14200.pdf

6 Rodrigo Santamaría. "Hilos en Java", Recuperado el 29 de Abril de 2017 de http://vis.usal.es/rodrigo/documentos/aso/JavaHilos.pdf

7 Arvo Lipitsäinen. "ORM – Object Relational Mapping", Recuperado el 1 de Abril de 2017 de http://myy.haaga-helia.fi/~dbms/dbtechnet/labs/dae_lab/Orm.pdf

8 Bob McCune, "Exploring the Java Persistence API", Recuperado el 17 de Mayo de 2017 de https://www.intertech.com/resource/usergroup/Exploring%20the%20JPA.pdf

9 Héctor Suárez González, "Manual Hibernate", Recuperado el 12 de Febrero de 2017 de http://index-of.es/Java/Manual%20-%20Hibernate.pdf

10 Chris Richardson, "Overview of POJO programming", Recuperado el 21 de Febrero de 2017 de http://www.chrisrichardson.net/presentations/OverviewOfPOJOs_BEA_JUG.pdf

11 American National Standard, "ANSI C12.18-2006", Recuperado el 10 de Febrero de 2017 de https://www.nema.org/Standards/ComplimentaryDocuments/ANSI-C12-18.pdf

12 American National Standard, "ANSI C12.19-2008", Recuperado el 23 de Marzo de 2017 de https://www.nema.org/Standards/ComplimentaryDocuments/ANSI-C1219-2008-contents-and-scope.pdf

ACADEMIA JOURNALS
OPUS PRO SCIENTIA ET STUDIUM

SAN ANTONIO, TX
INFO@ACADEMIAJOURNALS.COM
ACADEMIAJOURNALS.COM

13 septiembre 2017

AUTORES: Ing. Mario Alberto Méndez Carmona

ARTÍCULO: **Diseño de un motor de cálculo para datos estadísticos de un MDMS para una red de medidores inteligentes**

ARTÍCULO Núm: Celo778

Estimados autores,

Con agrado les informamos que, con fecha de hoy, el artículo arriba citado ha sido aprobado para su presentación en el Congreso International de Investigación Academia Journals Celaya 2017. El congreso tendrá lugar los días 08 al 10 de noviembre del año en curso, en Celaya, Guanajuato, México.

El artículo será incluido en las publicaciones del congreso, que incluyen modalidades ISSN e ISBN. Le recordamos que la política del congreso es que para recibir su reconocimiento e incluir su artículo en las publicaciones, es necesario presentar el mismo en el congreso. Le rogamos que utilice su número de artículo en toda correspondencia con Academia Journals.

Saludos cordiales.

Dr. Rafael Moras, P.E.
Editor
Academia Journals
Info@academiajournals.com

SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

ACADEMIA JOURNALS

CONGRESO INTERNACIONAL DE INVESTIGACIÓN DE ACADEMIA JOURNALS.COM, CELAYA 2017

OTORGAN EL PRESENTE

CERTIFICADO

A

ING. MARIO ALBERTO MÉNDEZ CARMONA

POR SU PARTICIPACIÓN CON LA PONENCIA TITULADA

DISEÑO DE UN MOTOR DE CÁLCULO PARA DATOS ESTADÍSTICOS DE UN MDMS PARA UNA RED DE MEDIDORES INTELIGENTES

PUBLICADA EN EL PORTAL DE INTERNET
CELAYA.ACADEMIAJOURNALS.COM
VOLUMEN ONLINE CON ISSN 1946-5351 VOL. 9, NO. 6, 2017
E INDIZACIÓN EN FUENTE ACADÉMICA PLUS (EBSCO) Y
LIBRO DIGITAL EBOOK CON ISBN 978-1-939982-32-2, ONLINE.
LA CUAL FUE PRESENTADA EN EL
TECNOLÓGICO NACIONAL DE MÉXICO EN CELAYA
LOS DÍAS 8, 9 Y 10 DE NOVIEMBRE DE 2017, CELAYA, GUANAJUATO, MÉXICO.

DR. RAFAEL MORAS
EDITOR, ACADEMIAJOURNALS.COM
PROFESOR DE INGENIERÍA INDUSTRIAL Y ADMINISTRATIVA
ST. MARY´S UNIVERSITY, SAN ANTONIO, TX. EEUU

ING. ALEJANDRO ÁLVAREZ BÁRCENAS
COORDINADOR GENERAL DEL
CONGRESO INTERNACIONAL DE INVESTIGACIÓN
ACADEMIA JOURNALS, CELAYA 2017

N° 1997

ANAB

Cel0778

Apéndice C

Documentos de estancias

EOS TECH

Querétaro, Querétaro a 19 de abril de 2017

ASUNTO: Carta de aceptación en Estancia Técnica

Mtro. Felipe Pascual Rosario Aguirre
Director - Instituto Tecnológico de Apizaco
Conurbado Apizaco - Tzompantepec s/n,
Col. Centro C.P.90300 Apizaco Tlaxcala
PRESENTE

Por este conducto me dirijo a usted para informarle la aceptación del C. Ing. Mario Alberto Méndez Carmona alumno de la Maestría en Sistemas Computacionales con número de control M11370834 del Instituto Tecnológico de Apizaco, para realizar una Estancia Técnica en la Empresa EOS TECH S.A. de C.V. ubicada en Calle Luis G Malvaez número 523, Colonia Reforma Agraria en Querétaro, Querétaro. El objetivo general del proyecto en cual trabajará es el desarrollo de un Sistema del tipo *Head End* y un *Meter Data Management* System (*MDMS*) sin embargo los objetivos específicos de la estudiante de Maestría se acotan a los siguientes entregables:

-Modelado de la base de datos de seguridad y usuarios del *MDMS*.
-Modelado de la base de datos operacional del *MDMS*
-Modelado de la base de datos de históricos del *MDMS*
-Operaciones de la base de datos de históricos de *MDMS*.
-Motor de cálculo sobre los datos contenidos en la base de datos de históricos del *MDMS;*
-Excepciones que pueda arrojar el motor de calculo del *MDMS*

Finalmente es importante indicar que el periodo que contempla la Estancia Técnica es del 07 de de febrero del 2017 al 07 de agosto del mismo año.
Sin otro particular, agradezco de antemano su atención. y quedo al pendiente para cualquier duda o aclaración.

Atentamente

JUAN GONZÁLEZ SALOMÓN
DIRECTOR GENERAL
EOS TECH S.A. de C.V.

Querétaro, Qro., a 02 de Agosto del 2017

Asunto: Constancia

MTRO. FELIPE PASCUAL ROSARIO AGUIRRE
DIRECTOR DEL INSTITUTO TECNOLÓGICO DE APIZACO
PRESENTE:

AT'N: DR. JOSÉ FEDERICO CASCO VÁZQUEZ
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

Por medio del presente le envío un cordial saludo y aprovecho para hacerle constar que el Ing. Mario Alberto Méndez Carmona, alumno de la Maestría en Sistemas Computacionales con número de control M11370834, de la Institución que usted destacadamente dirige, participó en el sub-proyecto:

"Modelado e implementación de un *data warehousing* para un *MDM* utilizando algoritmos ETL y VEE"

La etapa del proyecto realizada tuvo una duración de 6 meses y se desarrolló en las oficinas centrales de EOS TECH S.A. DE C.V. en la ciudad de Querétaro, Querétaro.

En virtud de que se han cubierto satisfactoriamente la fase del proyecto. Tenemos bien a dar constancia de que dicho trabajo realizado por el Ing. Mario Alberto Méndez Carmona, cubre y satisface las expectativas planteadas al inicio de la fase de este proyecto.

Agradeciendo sus atenciones a la presente quedo de ustedes.

ATENTAMENTE

Ing. Juan González Salomón
Director General
EOS TECH S.A. DE C.V.

EOS TECH

Querétaro, Qro., a 02 de Agosto del 2017

Asunto: Constancia de satisfacción

MTRO. FELIPE PASCUAL ROSARIO AGUIRRE
DIRECTOR DEL INSTITUTO TECNOLÓGICO DE APIZACO
PRESENTE:

AT'N: DR. JOSÉ FEDERICO CASCO VÁZQUEZ
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

Por medio del presente le envío un cordial saludo y aprovecho para hacerle constar que posterior a la recepción del sub-proyecto: **"Modelado e implementación de un *data warehousing* para un *MDM* utilizando algoritmos ETL y VEE"** realizado por el Ing. Mario Alberto Méndez Carmona con número de control M11370834, y dirigido por el M.C. Higinio Nava Bautista, alumno y profesor respectivos de la Maestría en Sistemas Computacionales, de la Institución que usted destacadamente dirige, **cubre y satisface las expectativas planteadas al inicio de la fase de este proyecto.**

Tenemos bien a dar constancia de que dicho trabajo realizado por el Ing. Mario Alberto Méndez Carmona. Agradeciendo sus atenciones quedo de ustedes.

ATENTAMENTE

Ing. Juan González Salomón
Director General
EOS TECH S.A. DE C.V.

EOS TECH S.A de C. V.
Luis G. Malvaez No. 523 Col. Reforma Agraria 2da Sección.
Querétaro, Qro., México. C.P. 76086
Tel.: (442) 243 1331

www.eostech.mx
"Mejoramos tu mundo con innovación"

EOS TECH

Rev. 03

EOS TECH

OTORGA EL SIGUIENTE

Reconocimiento

A: MARIO ALBERTO MENDEZ CARMONA

Por su participación en el curso de

INESA BASICO

Cristaneferty Cuamatzi
INSTRUCTOR

2017-02-17

Ing. Juan González Salomón
DIRECTOR GENERAL
EOS TECH

EOS TECH

Rev. 03

EOS TECH

OTORGA EL SIGUIENTE

Reconocimiento

A: MARIO ALBERTO MENDEZ CARMONA

Por su participación en el curso de

Building Blocks (POM) GRUPO 1

Agustín Sánchez y Mariano Centeno Camarena
INSTRUCTOR

2017-02-21

Ing. Juan González Salomón
DIRECTOR GENERAL
EOS TECH

EOS TECH

Rev. 03

EOS TECH

OTORGA EL SIGUIENTE

Reconocimiento

A: MARIO ALBERTO MENDEZ CARMONA

Por su participación en el curso de

Nexus Tech GRUPO 1

Brigido Rodríguez Cruz y Agustín Sánchez Atonal
INSTRUCTOR

2017-02-21

Ing. Juan González Salomón
DIRECTOR GENERAL
EOS TECH

Rev. 03

EOS TECH

OTORGA EL SIGUIENTE

Reconocimiento

A: MARIO ALBERTO MENDEZ CARMONA

Por su participación en el curso de

MTC GRUPO 1

Carlos Domínguez Galván y Jesus García Piña
INSTRUCTOR

2017-02-22

Ing. Juan González Salomón
DIRECTOR GENERAL
EOS TECH

EOS TECH

Rev. 03

EOS TECH

OTORGA EL SIGUIENTE

Reconocimiento

A: MARIO ALBERTO MENDEZ CARMONA

Por su participación en el curso de

GMMTools GRUPO 1

Agustín Sánchez y Gustavo Fuentes López
INSTRUCTOR

2017-02-22

Ing. Juan González Salomón
DIRECTOR GENERAL
EOS TECH

EOS TECH

Rev. 03

EOS TECH

OTORGA EL SIGUIENTE

Reconocimiento

A: MARIO ALBERTO MENDEZ CARMONA

Por su participación en el curso de

CFE G0100-05 GRUPO 1

Juan García Guzman y Mariano Centeno Camarena
INSTRUCTOR

2017-02-22

Ing. Juan González Salomón
DIRECTOR GENERAL
EOS TECH

EOS TECH

Rev. 03

EOS TECH

OTORGA EL SIGUIENTE

Reconocimiento

A: MARIO ALBERTO MENDEZ CARMONA

Por su participación en el curso de

Smart Grid Structure And concepts (RUM) GRUPO 1

Jeaneth E. Quíriz López / Mariano Centeno Camarena
INSTRUCTOR

2017-02-23

Ing. Juan González Salomón
DIRECTOR GENERAL
EOS TECH

EOS TECH

Rev. 03

EOS TECH

OTORGA EL SIGUIENTE

Reconocimiento

A: MARIO ALBERTO MENDEZ CARMONA

Por su participación en el curso de

Fundamentos ADDO Grupo 1

MARIANO CENTENO
INSTRUCTOR

2017-02-28

Ing. Juan González Salomón
DIRECTOR GENERAL
EOS TECH

EOS TECH

Rev. 03

EOS TECH

OTORGA EL SIGUIENTE

Reconocimiento

A: MARIO ALBERTO MENDEZ CARMONA

Por su participación en el curso de

INTRODUCCIÓN AL SGC ISO 9001:2015

MARIANO CENTENO
INSTRUCTOR

2017-03-01

Ing. Juan González Salomón
DIRECTOR GENERAL
EOS TECH

EOS TECH

Rev. 03

EOS TECH

OTORGA EL SIGUIENTE

Reconocimiento

A: MARIO ALBERTO MENDEZ CARMONA

Por su participación en el curso de

SCRUM GRUPO 1

Agustín Sánchez y Mariano Centeno Camarena
INSTRUCTOR

2017-03-20

Ing. Juan González Salomón
DIRECTOR GENERAL
EOS TECH

EOS TECH

"Mejoramos tu mundo con innovación"

EOS TECH OTORGA EL SIGUIENTE

Reconocimiento

A: **ING. MARIO ALBERTO MÉNDEZ CARMONA**

Por su participación en el desarrollo del Proyecto **Meter Data Management.**

Se extiende una felicitación por la conclusión satisfactoria
de su estancia profesional.

Ing. Juan González Salomón
DIRECTOR GENERAL
EOS TECH

Apéndice D

Glosario de acrónimos

Siglas

3G *Third Generation*–(Tercera Generación)

4G *Fourth Generation*–(Cuarta Generación)

ACID *Atomicity Consistency Isolation and Durability*–(Atomicidad Consistencia Aislamiento Durabilidad)

ADN *Deoxyribonucleic Acid*–(Ácido Desoxirribonucleico)

AJAX *Asynchronous JavaScript And XML*–(JavaScript Asíncrono y XML)

AMEF *Failure Mode Effect Analysis*–(Análisis de Modo y Efecto de la Falla)

AMI *Advanced Metering Infrastructure*–(Infraestructura Avanzada de Medición)

ARM *Automated Meter Reading*–(Lectura de Medición Automática)

ANSI *American National Standards Institute*–(Instituto Nacional Estadounidense de Estándares)

API *Application Programming Interface*–(Interfaz de Programación de Aplicaciones)

BDD *Distributed Database*–(Base de Datos Distribuida)

BI *Business Intelligence*–(Inteligencia de negocios)

CIM *Common Information Model*–(Modelo de Información Común)

CIS *Customer Information System*–(Sistema de Información del Cliente)

CRUD *Create Read Update and Delete*–(Crear Leer Actualizar y Borrar)

CSS *Cascading Style Sheets*–(Hoja de Estilo en Cascada)

CSV *Comma-Separated Values*–(Valores Separados por Comas)

CTQs *Critical To Quality*–(Características Criticas de Calidad)

CURP (Clave Única de Registro de Población)

DB2 *dBASE II*

DOC *Document*–(Documento)

DW *Data Warehouse*–(Almacén de Datos)

DWH *Data Warehousing*

EE *Enterprise Edition*–(Edición Empresarial)

ER *Entity-Relationship*–(Entidad-Relación)

ETL *Extract Transform and Load*–(Extracción Transformación y Carga)

FTP *File Transfer Protocol*–(Protocolo de Transferencia de Archivos)

GNU *GNU's Not Unix*–(GNU No es Unix)

GPS *Global Positioning System*–(Sistema de Posicionamiento Global)

GUI *Graphical User Interface*–(Interfaz Gráfica de Usuario)

HES *Head-End System*

HOLAP *Hybrid Online Analytical Processing*–(Procesamiento Analítico en Linea Híbrido)

HQL *Hibernate Query Languaje*–(Lenguaje de Consulta de Hibernate)

HTML *HyperText Markup Language*–(Lenguaje de Marcas de Hipertexto)

IBM *International Business Machines*–(Maquina de Negocios Internacionales)

ID *Identifier*–(Identificador)

IP *Internet Protocol*–(Protocolo de Internet)

ISP *Internet Service Provider*–(Proveedor de Servicio de Internet)

J2EE *Java Enterprise Edition*–(Edición de Java Empresarial)

JPA *Java Persistence API*–(API de Persistencia de Java)

JS *JavaScript*

JSON *JavaScript Object Notation*–(Notación de Objetos de JavaScript)

KVAh (Kilo Voltio-Amperio Hora)

KVARh (Kilo Voltio-Amperio Reactivo Hora)

KWh (Kilovatio Hora)

LAN *Local Area Network*–(Red de Área Local)

M2M *Machine to Machine*–(Maquina a Maquina)

MAC *Media Access Control*–(Control de Acceso al Medio)

MDM *Meter Data Management*–(Gestión de Datos de Medidores)

MDMS *Meter Data Management System*–(Sistema de Gestión de datos de medidores)

MOLAP *Multidimensional Online Analytical Processing*–(Procesamiento Analítico en Linea Multidimensional)

MPI *Message Passing Interface*–(Interfaz de Paso de Mensajes)

MRE (Módulo Remoto de Energía)

MVC *Model View Controller*–(Modelo Vista Controlador)

NFC *Near Field Communication*–(Comunicación de Campo Cercano)

noSQL *Non SQL*–(No sólo SQL)

NSS (Número de Seguridad Social)

OLAP *On-Line Analytical Processing*–(Procesamiento Analítico en Linea)

OLTP *OnLine Transaction Processing*–(Procesamiento de Transacciones en Linea)

ORDBMS *Object-Relational DataBase Management System*–(Sistema de Gestión de Base de Datos Objeto-Relacional)

ORM *Object-Relational Mapping*–(Mapeo Objeto Relacional)

PDF *Portable Document Format*–(Formato de Documento Portable)

PLC *Power Line Communications*–(Linea de Comunicación de Energía)

POD *Pay On Delivery*–(Identificación de Punto de Entrega)

POJO *Plain Old Java Object*–(Objeto Java Plano Antiguo)

POO *Object-Oriented Programming*–(Programación Orientada a Objetos)

PVM *Parallel Virtual Machine*–(Maquina Virtual Paralela)

RAID *Redundant Array of Independent Disks*–(Arreglo Redundante de Discos Independientes)

RAM *Random Access Memory*–(Memoria de Acceso Aleatorio)

RDBMS *Relational DataBase Management System*–(Sistema de Gestión de Base de Datos Relacional)

REI *Redes Eléctricas Inteligentes*–(Smart Grid)

RFC (Registro Federal de Contribuyentes)

RFID *Radio Frequency Identification*–(Identificación por Radiofrecuencia)

ROLAP *Relational Online Analytical Processing*–(Procesamiento Analítico en Linea Relacional)

RPN *Risk Priority Number*–(Número de Prioridad de Riesgo)

RPU (Registro Patronal Único)

SBV *Shared Business Vocabulary*–(Vocabulario Empresarial Compartido)

SE *Standard Edition*–(Edición Estándar)

SFTP *SSH File Transfer Protocol*–(Protocolo de Transferencia de Archivos Seguro)

SGBD *DataBase Management System*–(Sistema Gestor de Base de Datos)

SIMULINK *Plataforma de Simulación Multidominio*

SMS *Short Message Service*–(Servicio de Mensajes Cortos)

SQL *Structured Query Language*–(Lenguaje de Consulta Estructurado)

SSH *Secure SHell*–(Interprete de Ordenes Seguro)

TCP *Transmission Control Protocol*–(Protocolo de Control de Transmisión)

Telnet *Telecommunication Network*–(Red de Telecomunicación)

UML *Unified Modeling Language*–(Lenguaje Unificado de Modelado)

UPS *Uninterruptible Power Supply*–(Sistema de Alimentación Interrumpida)

VEE *Validation Estimation and Edition*–(Validación Estimación y Edición)

WAN *Wide Area Network*–(Red de Área Extensa)

WCC *Windows Computer Cluster*–(Cluster de Cómputo en Windows)

WI-FI *Wireless Fidelity*–(Fidelidad Inalámbrica)

XML *eXtensible Markup Language*–(Lenguaje de Marcado Extensible)