

# Incorporación del estándar CIM en sistemas de gestión de energía

Matias Antunez<sup>1</sup>, Leandro Fiaschetti<sup>2</sup>, Mariano Risso<sup>1</sup>, Gustavo Boroni<sup>1</sup> y Aldo Rubiales<sup>2</sup>

<sup>1</sup> PLADEMA, CONICET, Fac. Cs. Exactas - UNCPBA, Pinto 399 - CP 7000, Tandil, Argentina

<sup>2</sup> PLADEMA, CICPBA, Fac. Cs. Exactas - UNCPBA, Pinto 399 - CP 7000, Tandil, Argentina

**Resumen** Los sistemas de gestión de energía y más específicamente los sistemas de distribución, cuentan con numerosos módulos que requieren compartir información de un modelo de datos común. En general, estos módulos son desarrollados por separado y la información utilizada puede requerir de una adaptación. En este contexto, EPRI (*Electric Power Research Institute*) estableció un modelo de datos estándar para estos sistemas denominado CIM (*Common Information Model*).

CIM está descrito mediante diagramas de clases y paquetes, cuya composición arquitectónica ha sido ampliamente documentada. No obstante, no hay descripción clara de cómo instanciar el modelo. En este trabajo se propone el desarrollo de un módulo que administre un modelo CIM. Se describen diferentes alternativas de implementación y tecnologías a utilizar, detallando ventajas y desventajas. Finalmente, se propone una descomposición funcional en dos modelos: estático y dinámico, que permitan administrar por separado características físicas y lógicas de la red de energía.

**Keywords:** CIM, DMS, Redes de energía, Software de gestión, Interoperabilidad y sistemas de integración.

## 1 Introducción

### 1.1 Sistemas de gestión de energía

Las redes eléctricas se gestionan desde los centros de operaciones para garantizar el mantenimiento de un suministro de energía. Un centro de operaciones realiza y coordina diversas funciones, como por ejemplo el monitoreo de sistemas, el control y administración de cuadrillas, entre otros.

A medida que los sistemas de distribución de energía evolucionan, los centros de operaciones que los controlan también lo hacen para incorporar nuevas funciones. Los distintos sistemas tecnológicos utilizados se simplifican cada vez más, proporcionando una herramienta integrada de vigilancia y gestión para mejorar la toma de decisiones de los responsables.

En los últimos años, diversos factores externos han ayudado a acelerar el desarrollo y la expansión de los sistemas de gestión tendiente a generar redes inteligentes. Los factores del cambio comprenden a la sociedad, los gobiernos, y a las organizaciones encargadas de la distribución y la aplicación de tecnología. Estos se han dirigido hacia la modernización de la red. Los costos crecientes de la nueva generación y transmisión de energía, son factores que también influyen en el cambio tecnológico.

Claramente, los centros de control del área de distribución necesitan poder afrontar los nuevos desafíos del sector [5,7]. En este contexto, los DMS (*Distribution Management System*) son una pieza fundamental para el manejo inteligente de las redes eléctricas, permitiendo la integración de los sistemas de adquisición de información (normalmente SCADA) con aplicaciones orientadas a la operación de redes.

Un DMS integra una colección de aplicaciones diseñadas para monitorear y controlar toda la red de distribución de manera eficiente y confiable, buscando además mejorar la confiabilidad y calidad de servicio en términos de reducción de las interrupciones, como así también en minimizar el tiempo de interrupciones.

Con el fin de dar soporte en la toma de decisiones adecuadas, y en las actividades de operación y mantenimiento, los DMS suelen proveer una serie de aplicaciones con funcionalidades y herramientas básicas:

- **CAD Unifilar:** Una parte importante del DMS es la visualización de la red eléctrica con varios niveles de abstracción. Esta herramienta permite al operador del sistema ver el estado actual de la red de manera online, modificar la topología de la misma, y visualizar los resultados de los distintos algoritmos del DMS sobre el estado actual.
- **Sistema de Adquisición de Datos (SCADA):** Se encarga de la toma de mediciones de la red para generar datos que puedan ser manipulados por el software del DMS. Para ello deberá tomar un conjunto de señales físicas, convertirlas a unidades conocidas, y digitalizarlas de manera que se puedan procesar.
- **Flujo de Potencia:** Encargado del cálculo de condiciones de operación en régimen permanente. En estos cálculos se determina las tensiones en las barras del sistema, las potencias activa y reactiva (tanto de inyección en barra como flujo en líneas), y las pérdidas en los distintos componentes de la red.
- **Estimador de Estados:** Tiene como entrada el modelo de red, las medidas del SCADA y los datos históricos, con el objetivo de reconstruir el estado general del sistema. Además, el estimador de estados se utiliza para generar los datos de entrada para el resto de los algoritmos y/o aplicaciones.
- **Análisis de Conectividad de Red (NCA):** Permite determinar la topología dinámica basada en la información de las cargas y las fuentes, junto con la información de conectividad estática de los componentes de la red.

## 1.2 Interoperabilidad de las aplicaciones de un DMS

Como se mencionó anteriormente, los DMS están formados por distintas aplicaciones. Muchas de estas comparten información; por ejemplo, la mayor parte de las aplicaciones de un sistema de gestión deben importar la información acerca de la topología de la red eléctrica. El problema es que, en general, dichas aplicaciones pueden ser desarrolladas por distintos fabricantes. Esto implica que cada aplicación tiene su propia manera de organizar la información, y cuenta con un formato propio para intercambiarla. Por lo tanto, si dos aplicaciones deben intercambiar información es necesario un convertidor de formato, lo cual incrementa el costo y la complejidad de los DMS.

A finales de los 90, en vista de este problema, el principal organismo encargado de realizar investigaciones en relación a los sistemas eléctricos de potencia en los Estados Unidos de América, el EPRI (*Electric Power Research Institute*), comenzó a elaborar un modelo de información único para las aplicaciones encargadas de la gestión de las redes eléctricas. Este modelo se denominó CIM (*Common Information Model*). Posteriormente la IEC (*International Electrotechnical Commission*) lo adoptó como el modelo de información internacional estándar para la gestión de los sistemas eléctricos, con lo cual el modelo CIM propone la manera de organizar toda la información que pueda ser necesaria en las aplicaciones dedicadas a la gestión de las redes eléctricas.

El EPRI define CIM dentro de las series de normas IEC 61970 e IEC 61968. Las primeras, se centran en los sistemas de gestión de las redes de transporte, también llamados en la literatura EMS (*Energy Management System*); y las segundas extienden el paquete para soportar los sistemas de gestión de las redes de distribución o DMS[14]. En la Figura 1 puede verse un diagrama de la interacción de los paquetes. Además, las normas mencionadas describen una arquitectura de referencia para la integración entre las aplicaciones, los servicios de comunicación de acceso a la información, y formatos para el intercambio de información. Por ejemplo, CIM/XML [4,2] es un formato conocido para el intercambio de grandes cantidades de información acerca de la topología de las redes eléctricas a gestionar. También se utilizan otros formatos como CIM/XSD [4] o CIM/SVG [9].

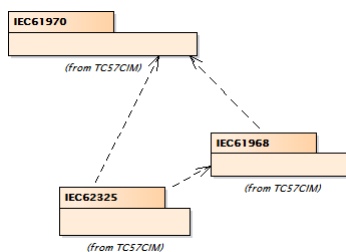


Figura 1. Paquetes CIM y dependencias [4].

CIM está descrito mediante diagramas de clases y de paquetes UML, y en este sentido hay mucho material de uso académico/profesional con detalles de la arquitectura de software [13]. En este artículo sólo se mencionan algunos de los paquetes de la norma IEC 61970, como por ejemplo el paquete *Wires*, donde se encuentran las clases que representan los equipos físicos que pueden formar parte de un sistema eléctrico de potencia, y el paquete *Topology*, el cual proporciona las clases necesarias para poder describir las conexiones entre los equipos.

## 2 Propuesta

Es claro que la utilización de CIM es fundamental para lograr la interoperabilidad entre distintas aplicaciones de un DMS. Por lo tanto, a la hora de desarrollar este tipo de sistemas es importante evaluar cuidadosamente su instanciación como modelo de datos, ya sea para obtener persistencia, validar y construir el modelo de manera confiable, y que las aplicaciones puedan utilizar los datos del modelo y actualizarlos.

Como se mencionó anteriormente, es muy escasa o casi nula la documentación de como instanciar CIM. En este sentido se pueden encontrar algunos trabajos que proponen una implementación parcial del modelo de datos [13], lo cual puede derivar en la selección errónea de la tecnología o software utilizado si no se consideran de manera conjunta todas las necesidades vinculadas a la interoperabilidad.

En el trabajo de Pradeep y colab.[12] se propone una extensión de CIM para la construcción del modelo de datos, descomponiendo la información lógica (modelo dinámico) y la información física (modelo estático) de la red, permitiendo que las distintas partes del sistema de gestión interactúen con el modelo correspondiente (por ejemplo, toda la información estructural estará asociada al modelo estático, y toda la información vinculada con las mediciones de campo impactará en el modelo dinámico).

Esta propuesta que a primera vista permite mejorar la interoperabilidad general del sistema, no brinda detalles en términos de eficiencia considerando el tamaño de la red o la cantidad de componentes. En este sentido, además de considerar una división del modelo de datos en estático y dinámico, es importante evaluar distintas tecnologías para implementar dichos modelos, analizando el impacto de las distintas funcionalidades y servicios involucrados, como así también de los requerimientos de tiempo real.

En este artículo, se propone una solución para la instanciación de CIM como un modelo de datos, sobre la base de su división en estático y dinámico, considerando los servicios comunes o habituales involucrados a dichos modelos y la interoperabilidad esperada en cada caso. En este sentido se presentan distintas soluciones tecnológicas, analizando las ventajas y desventajas de las más utilizadas. Finalmente, se propone una solución integral sobre un caso de estudio que permite realizar consultas, altas, bajas y modificaciones al modelo. Dicha solución considera además la necesidad de persistencia.

### 3 Desarrollo de propuesta

#### 3.1 Selección y uso de alternativas

Sobre la base de la utilización del modelo CIM como esquema de modelo de datos, se debe reflexionar sobre la manera de instanciarlo teniendo en cuenta la flexibilidad, posibilidad de persistencia, acceso a los datos, etc. Luego de analizar trabajos previos (mencionados anteriormente) que presentan un estudio parcial del problema de instanciación, se desprenden dos maneras de trabajar con el modelo de datos. En la primera, se considera la instanciación de las clases generadas con alguna herramienta partiendo del esquema de paquetes base [2]. Teniendo en cuenta la documentación propuesta por el estándar, se propone manejar la persistencia del modelo a través de representaciones RDF<sup>3</sup>. A partir de esta propuesta, surge como segunda alternativa trabajar con alguna herramienta que permita administrar el modelo de representación planteado. La comparación entre ambas alternativas se muestran en la Figura 2.

		Clases	RDF
Qué guarda?		Instancia de objetos.	Grafo que representa instancia de objetos
Complejidad	Entendimiento	Baja. Se utiliza directamente el esquema de clases	Media/Baja. Se debe determinar el mapeo, pero no requiere mucha complejidad
	Escalabilidad	Sube. Utilización de estructuras auxiliares	No varía
	Guardado	Alta. Se debe programar alternativa	Directo. Utilizando herramienta
	Consulta de datos	Alta. Utilización de estructuras auxiliares	Media. A través de lenguaje de consultas
Herramienta		Cualquier lenguaje orientado a objetos	Jena para Java. LinqToRdf para C#.

**Figura 2.** Alternativas para representar el modelo de datos.

Analizando las ventajas y desventajas de cada solución se optó por trabajar con un modelo RDF. Los mismos, se construyen bajo un esquema de clases que puede estar definido a través de un formato OWL (*Web Ontology Language*)<sup>4</sup> o RDFS (*Resource Description Framework Schema*)<sup>5</sup>[11,10]. Así, utilizando esta tecnología se provee de un medio para mapear un diseño de un modelo orientado a objetos, y representarlo en algún formato específico como por ejemplo XML. Además, existen herramientas que automáticamente generan esquemas RDF, y otras que permiten manejar instancias de modelos del mismo tipo [4,15].

Los modelos RDF están compuestos por un conjunto de recursos que contienen atributos, y relaciones entre estos recursos. Las herramientas que manejan

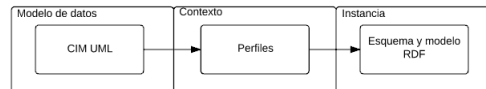
<sup>3</sup> Estándar definido *World Wide Web Consortium* (W3C) para representar modelo de datos y metadatos.

<sup>4</sup> Lenguaje diseñado para ser usado en aplicaciones que necesitan procesar el contenido de la información. OWL propone un mecanismo de interpretación de contenidos.

<sup>5</sup> RDFS es un lenguaje de propósito general para representar vocabularios RDF.

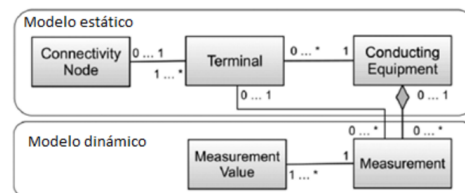
este tipo de modelos proponen diferentes maneras de persistir los datos, y pueden interpretar los datos guardados en los varios formatos de representación[8,3].

En cuanto a CIM, el estándar plantea como primer paso para utilizar el modelo, la definición del conjunto de clases a utilizar de acuerdo al contexto y tipo de sistema con el que se va a trabajar. Este conjunto de clases es denominado perfil. La serie de pasos definida se esquematiza en la Figura 3.



**Figura 3.** Definición de perfiles y esquema CIM.

Realizando un análisis del esquema de clases CIM y de acuerdo a las necesidades planteadas por el sistema a implementar, se realizó una descomposición funcional del modelo en dos módulos, respetando la propuesta del trabajo de Pradeep y colab. [12]. Uno de ellos esquematiza la topología de la red estática alojando los componentes y conexiones entre ellos. El otro define el estado de la misma en cuanto a las mediciones. Estos se denominaron modelo estático y dinámico respectivamente. Cada uno cuenta con un conjunto diferente de clases CIM como se plantea en la Figura 4.



**Figura 4.** Separación de modelo estático y modelo dinámico[12].

### 3.2 JENA

Una vez definido el formato de manejo de datos, se procedió a buscar una herramienta que maneje modelos RDF para poder instanciar y trabajar sobre la información. De acuerdo al entorno de desarrollo y, más específicamente el lenguaje utilizado, se determinó que JENA es una buena alternativa. La misma provee un *framework* para manejar y administrar datos representados con grafos RDF[3].

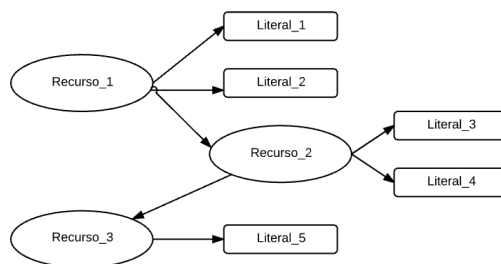
En cuanto a los conceptos básicos de este tipo de modelos, se pueden nombrar 3 tipos de componentes con los cuales se mapea el modelo de datos:

- **Recursos:** Representa una entidad de la cual se desean presentar características y relaciones. En CIM, se mapeará con una instancia de una clase.
- **Propiedades:** Refleja el nombre de una característica del recurso. En cuanto al uso del CIM, representa un atributo de una clase.
- **Literales:** Valores de una propiedad de un recursos. En CIM se corresponde con un valor de un atributo de una instancia de una clase.

JENA presenta una API (*Application Programming Interface*) para poder generar y administrar estos componentes. Este manejo se realiza a través del armado de un conjunto de *Statements*. Un *Statement* es una 3-tupla que representa un hecho en el modelo, la cual consta de los siguientes elementos:

- **Sujeto:** El recurso descrito.
- **Predicado:** La propiedad del recurso descrito.
- **Objeto:** El valor del predicado, y puede ser un literal si se trata de un atributo, un recurso, o una relación entre objetos.

El conjunto de estas 3-tuplas forman un grafo que representa un modelo RDF como se muestra en la Figura 5.



**Figura 5.** Ejemplo de instancia de modelo RDF con JENA.

A modo de ejemplo, el paquete *Wires* de la norma IEC 61970 tiene la clase *ACLineSegment* que representa una línea de transmisión. Esta clase tiene los atributos necesarios para definir este tipo de componente. Entre ellos se pueden nombrar: Susceptancia (bch), Conductancia (gch), Resistencia (r), etc. Una instancia de este tipo de clase puede ser mapeada a través de un recurso JENA que cuenta con literales que representan cada uno de esos parámetros. En la Figura 6 se representa este caso.

### 3.3 Persistencia

Una de las características a tener en cuenta al utilizar el modelo CIM, es elegir una buena manera de persistir los datos. Para ello, se investigaron diferentes alternativas de guardado y modos de acceso a la información. Los resultados comparativos de las soluciones más utilizadas son presentados en la Figura 7.

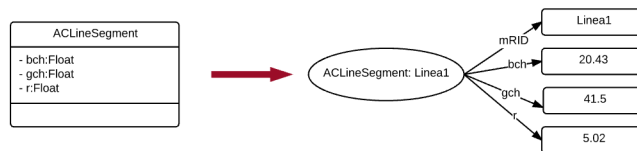


Figura 6. Ejemplo de instancia de línea de transmisión.

	Serializado	XML	Formato RDF	Json
Qué guarda?	Objetos en binario. Guarda clases java	Estructura de datos con atributos Guarda información serializada	Estructura, semántica y asociaciones. Ideal para guardar modelo de datos.	Sólo estructura
Complejidad	Baja	Media	Media/Alta	Media/Baja
Legibilidad	Nada	Media	Media	Media
Consultas	-	Si (XQL, XML QL)	Si (SPARQL)	Si (JSONiq)
Representación	-	Árbol (DOM)	Grafo	Árbol
Herramienta	Java	JAXB (Java)	Jena	JSONSimple (Java)

Figura 7. Comparación de alternativas para persistencia de información.

Inicialmente se descartó el formato binario serializado ya que, si bien su implementación es de baja complejidad, no tiene funcionalidad vinculada con el acceso de la información.

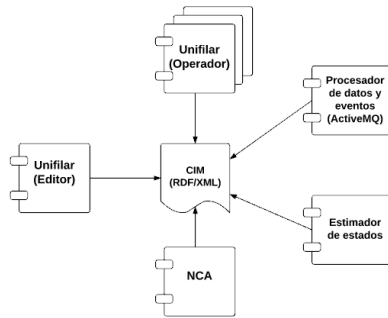
Para las soluciones restantes se concluyó que la más adecuada y directa era utilizar algún formato de representación RDF. Además, la herramienta utilizada para el armado del modelo cuenta con la funcionalidad de guardar y cargar los datos. En este sentido existen varias maneras de representar un modelo de este tipo, que permiten persistir la información de manera legible [3]. Entre las alternativas se puede nombrar Turtle, N-Triples, RDF/XML, JSON-LD, RDF/JSON, TriG, NQuads. Si bien, todas estas son válidas, el RDF/XML es el más estandarizado para el intercambio entre compañías. No obstante, se podría utilizar cualquier formato para el pasaje de datos interno entre aplicaciones.

Bajo un esquema de RDF se cuenta, además, con un lenguaje de consultas sobre los datos, que permite acceder a ellos de manera más ágil. Esta es una característica no menor para tener en cuenta a la hora de seleccionar una opción. Dicho lenguaje se denomina SPARQL<sup>6</sup>.

Definida la alternativa de guardado, la utilización de JENA soporta distintos tipos de formatos para persistir los datos en archivos. Esto funciona como una base de datos a la que el *framework* accede para consultar/leer o escribir. Así, es posible tener un manejo correcto de la interoperabilidad entre diferentes aplicaciones del DMS en cuanto a los datos de la topología de la red (Figura 8).

<sup>6</sup> SPARQL Protocol and RDF Query Language. Se trata de un lenguaje estandarizado para consultas en grafos RDF. Es una tecnología clave en el desarrollo de aplicaciones que utilizan este tipo de modelos



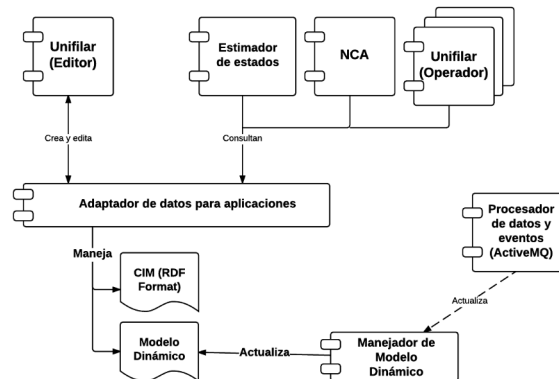


**Figura 8.** Interacción de los módulos del DMS con los datos del modelo persistidos.

### 3.4 Adaptador

Como se mencionó anteriormente, varias aplicaciones deben hacer uso de los datos de la topología de la red. Para desacoplar la implementación del modelo utilizando JENA y que cada módulo pueda consultar la información de los componentes, fue necesario realizar un adaptador de datos que se encargue de realizar la traducción necesaria para proveer una interfaz clara para el acceso a los datos.

Esta situación genera una agregación en la arquitectura de un módulo con las características referidas. En la Figura 9 se presenta el diagrama final de la arquitectura, con la separación del modelo estático y dinámico.



**Figura 9.** Interacción final de aplicaciones con el modelo de datos.

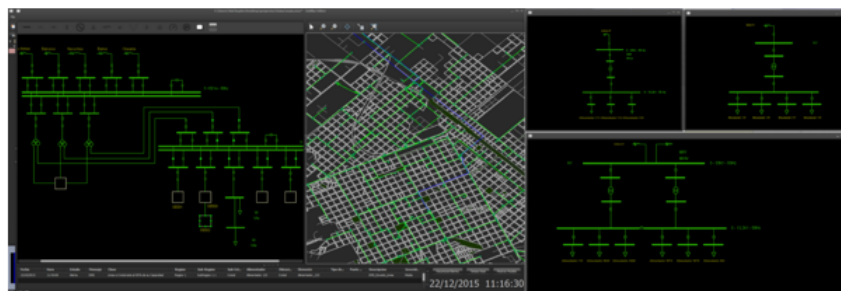
## 4 Caso de estudio e implementación

Como caso de estudio se desarrolló un módulo para la creación y administración de un modelo de datos CIM. El mismo se realizó en el marco del proyecto “TE-ENERGIA – Soluciones tecnológicas aplicadas a la distribución de redes eléctricas” financiado a través de FONARSEC - ANPCyT, Ministerio de Ciencia, Tecnología e Innovación Productiva, Argentina [6]. En este proyecto se propone el desarrollo completo de un DMS.

Las diferentes aplicaciones del DMS utilizarán este módulo para crear, consultar y editar el modelo CIM. La separación entre modelo dinámico y estático da flexibilidad para el manejo de datos en cuanto a su actualización y adquisición desde las diferentes aplicaciones.

Para el desarrollo se consideró la representación de modelo RDF, utilizando JENA para su implementación.

La creación del modelo estático se realiza a través de la aplicación del editor de la red, la cual cuenta con un diagrama unifilar que permite agregar componentes eléctricos y asignar parámetros con los correspondientes formularios. Un prototipo de esta funcionalidad se muestra en la Figura 10.



**Figura 10.** Interfaz de diagrama unifilar de prototipo desarrollado.

Para proveer de una interfaz clara necesaria para la agregación y actualización de los componentes eléctricos con sus respectivos parámetros, se optó por realizar una implementación utilizando el patrón de diseño *Factory Method*[16].

Las aplicaciones del DMS utilizarán los datos de los componentes creados sobre el modelo estático para poder visualizar, operar y correr algoritmos que permitan un manejo inteligente de la red eléctrica administrada. Además, deberán recibir información del modelo dinámico que es actualizado desde las mediciones realizadas. Para esto fue necesario tener un medio para acceder a la información de cada componente. La elección de RDF y más precisamente JENA, permitió contar con un lenguaje de consulta sobre los datos creados (SPARQL). Utilizando este lenguaje, se logró implementar una interfaz de acceso a la información de cada componente. De esta manera, es posible una obtención clara y adapta-

da para el manejo de los datos desde cada aplicación. Esta capa se denomina “Adaptador de datos para aplicaciones” en la Figura 9.

## 5 Resultados

Para validar la arquitectura propuesta se realizaron pruebas del módulo implementado interactuando con las diferentes aplicaciones del DMS en desarrollo. Como resultado de las pruebas se pudo constatar que las aplicaciones pueden acceder a la información solicitada de forma ágil y rápida, logrando una excelente interoperabilidad entre ellas. Para ello se creó una red característica desde el editor del DMS. La misma modela una parte de la red de distribución de la ciudad de Tandil y cuenta con una cantidad significativa de componentes (10 barras, 13 interruptores, 7 transformadores). Las pruebas realizadas consistieron en capturar mediciones de 3 puntos estratégicos de la distribuidora, de los cuales se tomaron datos de inyección y flujos de potencia activa y reactiva, ángulos de desfase y tensiones y corrientes por cada fase. Además se realizaron pruebas de estrés o carga extrapolando la información del sistema medido. Esto permitió validar los tiempos de sincronización del DMS según las funcionalidades mencionadas. En todos los casos se lograron tiempos aceptables, con actualización generales del sistema en el orden de los segundos.

Además, se realizaron verificaciones del proceso que exporta la topología a un archivo CIM/XML, y que a su vez pueda ser utilizado como entrada de otra herramienta de gestión. Para validar este concepto, se probaron importaciones de la red generada con otras herramientas (como OpenCIM[1]).

Finalmente, es importante mencionar que en el desarrollo realizado se buscó siempre mantener un diseño claro y flexible, que permitiera la incorporación de nuevos componentes de manera sencilla, lo cual es fundamental en la etapa de desarrollo de producto.

## 6 Conclusiones

En este trabajo se presentó un estudio sobre la utilización del estándar CIM para el modelo de datos, considerando la implementación de un sistema de gestión de energía. El mismo permitió describir las ventajas en la utilización de dicho modelo, principalmente en términos de persistencia, intercambio de información, etc. Además, permitió un mejor entendimiento del dominio en el que se está trabajando actualmente.

Para ello se desarrolló un módulo para la generación de una instancia de un modelo de datos CIM, que puede utilizarse dentro de las diferentes aplicaciones de un DMS. En este sentido se priorizó contar con una interfaz clara y transparente, que considere además contar con funcionalidad de consulta y guardado de datos de cada componente, logrando así mayor interoperabilidad entre los diferentes módulos.

Claramente, la elección de herramientas adecuadas en el proceso de desarrollo de este tipo de sistemas permite generar una solución modificable en el tiempo, y adaptable a la necesidades futuras que puedan surgir en el sector.

## Referencias

1. Open cim, <http://www.interpss.com/opencim>
2. Cim users group (2009-2016), <http://cimug.ucaiug.org/>
3. Apache jena (2011–2015), <https://jena.apache.org/index.html>
4. IntelliGrid Common Information Model Primer: Second Edition (2013)
5. Cammesa. informe anual correspondiente al año 2014 (2014), <http://portalweb.cammesa.com/Documentos%20compartidos/Informes/Informe%20Anual%202014v4.pdf>
6. Agencia - fonarsec (2015), <http://www.agencia.mincyt.gob.ar/frontend/agencia/fondo/fonarsec>
7. Se - secretaría de energía. ministerio de planificación federal, inversión pública y servicios (2015), <http://www.energia.gov.ar/contenidos/verpagina.php?idpagina=3691>
8. World wide web consortium (w3c), web standards developers (2016), <https://www.w3.org>
9. Chen, Y.Z., Sun, Y.Z., Wu, L.L., Feng, D., Zhou, Q., Xie, H.F.: Cim/svg based graphics-module integration power platform with secondary development of visio. In: Applied Mechanics and Materials. vol. 446, pp. 1636–1641. Trans Tech Publ (2014)
10. Espinosa Reza, A., Calleros Torres, T.M., Torres Espíndola, M.L., Alemán Cruz, N.A., García Mendoza, R., Sierra Rodríguez, B.: Interoperabilidad para la red eléctrica inteligente: Modelo cim y su proceso de adopción (2008)
11. McMorran, A.W., Ault, G.W., Elders, I.M., Foote, C.E., Burt, G.M., McDonald, J.R.: Translating cim xml power system data to a proprietary format for system simulation. Power Systems, IEEE Transactions on 19(1), 229–235 (2004)
12. Pradeep, Y., Seshuraju, P., Khaparde, S.A., Joshi, R.K.: Cim-based connectivity model for bus-branch topology extraction and exchange. Smart Grid, IEEE Transactions on 2(2), 244–253 (2011)
13. Shengye Lu, S.R.: Demonstration of using load flow and fault current calculation software tool with CIM interface and NIS database - Survey (2012)
14. Simmins, J.J.: The impact of pap 8 on the common information model (cim). In: Power Systems Conference and Exposition (PSCE), 2011 IEEE/PES. pp. 1–2. IEEE (2011)
15. Uslar, M., Specht, M., Rohjans, S., Trefke, J., González, J.M.: The Common Information Model CIM: IEC 61968/61970 and 62325-A practical introduction to the CIM. Springer Science & Business Media (2012)
16. Vlissides, J., Helm, R., Johnson, R., Gamma, E.: Design patterns: Elements of reusable object-oriented software. Reading: Addison-Wesley 49(120), 11 (1995)