

Una arquitectura de software para el desarrollo de un sistema de gestión de redes de distribución de energía eléctrica

Leandro Fiaschetti¹, Matias Antunez², Aldo Rubiales¹, Mariano Risso² y Gustavo Boroni²

¹ PLADEMA, CICPBA, Fac. Cs. Exactas - UNCPBA, Pinto 399 - CP 7000, Tandil, Argentina

² PLADEMA, CONICET, Fac. Cs. Exactas - UNCPBA, Pinto 399 - CP 7000, Tandil, Argentina

Resumen Los centros de control en distribución de energía necesitan un cambio de paradigma en su diseño arquitectónico para poder afrontar los nuevos desafíos del sector. 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 eléctricas. En este trabajo se presenta el desarrollo de una arquitectura novedosa que permite modelar, diseñar e implementar sistemas de gestión de redes de energía de distribución, que provee altos niveles de usabilidad, seguridad, extensibilidad, flexibilidad e interoperabilidad, en contraposición con las arquitecturas monolíticas utilizadas para este tipo de sistemas. En este caso, la arquitectura se desarrolló considerando funcionalidades de operación, tales como procesamiento de información, análisis de datos, flujo de carga, entre otros. Para ello se consideraron estándares abiertos como el modelo CIM (*Common Information Model*).

Keywords: arquitectura de sistemas, control de procesos y monitoreo, sistemas eléctricos, administración de sistemas de distribución, flujo de información, redes inteligentes.

1 Introducción

En los centros de control de los sistemas eléctricos se llevan a cabo todas las decisiones operativas de la red eléctrica que tiene asociada, y la actualidad del sector los obliga a repensar su diseño arquitectónico para poder afrontar los nuevos desafíos [5,7]. Aunque se pueden encontrar herramientas de apoyo a las decisiones tales como sistemas expertos, la mayoría están limitadas en su alcance, son costosas, y suelen ofrecer funcionalidades que no son adaptables a las realidades de la región [1]. Por otra parte, los operadores deben hacer un seguimiento de grandes volúmenes de datos en tiempo real, y abstraerse de los detalles para identificar los eventos de interés y las situaciones complejas que generan estos

eventos. Al trabajar sobre estas situaciones y en tiempos críticos, la responsabilidad recae directamente en los operadores para elegir selectivamente e inferir los eventos, tomar decisiones basadas en el conocimiento riguroso de la red, y poder actuar sobre ellos. Esto lleva a la necesidad de contar con una inteligencia extra, automatizada, que pueda procesar eventos complejos en tiempo real, abarcando una ventana de tiempo que va de minutos a horas, y al mismo tiempo, permita hacer análisis de tendencias y acontecimientos históricos similares. Sumado a lo anterior, dada las particularidades y características de los sistemas eléctricos, las herramientas de soporte deben brindar la posibilidad de agregar nuevas aplicaciones. Las herramientas convencionales aplicadas en los centros de control generalmente utilizan una arquitectura orientada a datos, que corresponden a sistemas pasivos y que son adecuadas para contar con información punto a punto. Sin embargo, estos sistemas carecen de la capacidad de detectar y responder de manera inteligente a los cambios del entorno. Por este motivo, los atributos de calidad tales como usabilidad, extensibilidad, interoperabilidad y flexibilidad deben ser tenidos en cuenta para el desarrollo de una arquitectura acorde [9,12,16]. En este documento se describen los detalles de una arquitectura para el desarrollo de un DMS (*Distribution Management System*) sobre la base de las restricciones de diseño necesarias para este tipo de sistemas. Asimismo se mostrarán las relaciones entre los distintos componentes y servicios junto con las responsabilidades de dichos componentes del sistema, como así también las tecnologías que pueden ser integradas para su implementación. Finalmente, se presentan los detalles de desarrollo de un prototipo correspondiente a una distribuidora de la región.

2 Restricciones de diseño

El principal obstáculo para lograr la interoperabilidad entre los sistemas SCADA (*Supervisory Control And Data Acquisition*) convencionales y arquitecturas de DMS, es la estrecha conexión entre los productores y consumidores de datos, que usualmente emplean formatos y protocolos de intercambio de información propietarios y/o cerrados, y que generalmente dificulta la incorporación de nuevas funcionalidades. Esta restricción se puede abordar a través de la utilización de estándares, donde el objetivo es describir un vocabulario común para el intercambio de los datos. En el área eléctrica, CIM (*Common Information Model*) [4,2] ha ganado un gran reconocimiento y aceptación como un estándar para la representación de datos del sistema de generación, transmisión y distribución [21]. Muchas empresas están adoptando CIM, y los productores de software están desarrollando productos que cumplen con dicho estándar. Teniendo en cuenta esta situación, su utilización se convierte en una opción atractiva para el desarrollo de una arquitectura abierta.

Por otra parte, el manejo de flujo de información en los centros de control se realiza a través de un modelo de extracción, donde el consumidor de datos inicia la comunicación para la transferencia de los mismos. Los sistemas orientados a datos reaccionan a las consultas realizadas por las aplicaciones que necesitan

datos, lo cual débilmente se adapta a la necesidad de conocer información en tiempo real. Por este motivo es necesaria una arquitectura sensible a los cambios del sistema, capaz de reaccionar ante la ocurrencia de diferentes eventos.

Otra situación a tener en cuenta está relacionada con que la mayoría de las aplicaciones utilizadas en los centros de control son parte de sistemas monolíticos y generalmente de carácter propietario. Sumado a esto, no son flexibles para incluir nuevas aplicaciones, cambios de funcionalidad, o cambios de plataforma.



Figura 1. *Smart Grid*

2.1 Partes del DMS

Un DMS integra una colección de aplicaciones diseñadas para monitorear y controlar toda la red de distribución de manera eficiente y confiable (Figura 1). Con dicho sistema se busca mejorar la calidad del 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 funcionalidades y herramientas básicas [14]. A continuación se enumeran algunas de ellas:

- **Herramienta de Visualización (CAD Unifilar)**. Una parte importante de todo DMS es la visualización de la red eléctrica (Unifilar de la red) con varios niveles de abstracción. Esta herramienta le 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.
- **Flujo de Potencia (LFA)**. Módulo encargado del cálculo de condiciones de operación en régimen permanente. En estos cálculos se determinan las tensiones en las barras, potencia activa y reactiva (tanto de inyección en barra como flujo en líneas), y pérdidas en los componentes de la red.

- **Estimador de Estados (SE)**. Esta herramienta tiene como entrada el modelo de red, medidas del SCADA y 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 de manera rápida y precisa para el resto de los algoritmos y/o aplicaciones.
- **Análisis de Conectividad de Red (NCA)**. El 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. Las principales salidas son: estado de los interruptores, zonas desenergizadas, detección de bucles, alimentación paralela, etc.
- **Sistema de Adquisición de Datos (DAS)**. El módulo DAS 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.
- **Integración del DMS**. Las actividades asociadas a la integración del sistema DMS implican todo aquello que permite unir los inputs/outputs del mismo. Además, debe permitir la posibilidad de realizar modificaciones y actualizaciones en las etapas de desarrollo y operación del DMS.

3 Arquitectura general propuesta

A partir de las restricciones de diseño y las principales funcionalidades de los DMS, se realizó una metodología ADD (*Attribute-Driven Design*) en el cual se priorizaron los principales atributos de calidad (usabilidad, extensibilidad, flexibilidad, interoperabilidad y seguridad). Durante la etapa de análisis se identificaron las diferentes tácticas arquitectónicas que satisfacen cada uno de los atributos, evaluando las ventajas y desventajas de cada alternativa. En este sentido, contar con una arquitectura dirigida por eventos EDA (*Event Driven Architecture*) [15] basada en un paradigma de producción-detección-consumo de eventos, se adapta a las necesidades actuales para este tipo de sistema de control. De esta manera, EDA permitirá monitorear, detectar, responder y controlar distintas situaciones de forma automatizada a través de múltiples componentes o procesadores de eventos. Los componentes de dicha arquitectura se encuentran desacoplados, permitiendo modificaciones de forma aislada, contribuyendo además a la flexibilidad y modificabilidad del sistema [11,10].

A su vez la arquitectura dirigida por eventos puede ser complementada por la arquitectura orientada a servicios SOA (*Service Oriented Architecture*), ya que los servicios pueden ser activados por disparadores que se encuentran en eventos entrantes. SOA posibilita que cada aplicación se acople libremente a la plataforma subyacente, donde los servicios se definen y modifican sin afectar a los otros, logrando así modularidad, flexibilidad, e independencia del proveedor y de la plataforma.

Por otra parte, la seguridad también es una preocupación importante de diseño, y puede garantizarse mediante mecanismos tradicionales que incluyan servicios de autenticación, autorización, perfiles, y permisos.

Finalmente, se propuso una arquitectura general para el desarrollo de sistemas de gestión de redes de energía (Figura 2).

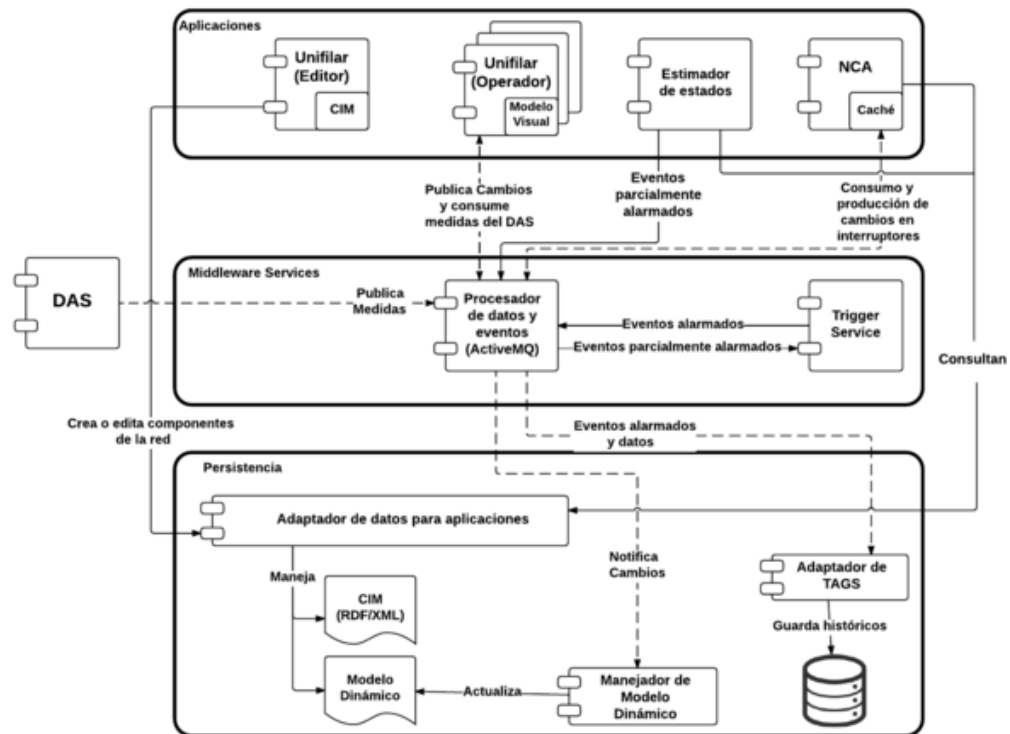


Figura 2. Arquitectura del DMS

4 Responsabilidades de los componentes

A continuación se detallan los componentes generales del sistema.

4.1 Data source

Este componente tiene vinculado la lectura/consulta/escritura de los datos de los sistemas SCADA y/o de los equipos de medición, y que a su vez pueden ser controlados (figura 3). El mismo permite adaptarse a distintas fuentes de información a través de un patrón de software *Adapter* [22] (en la implementación del prototipo se implementó la lectura a la base de datos del SCADA y el protocolo MODBUS).

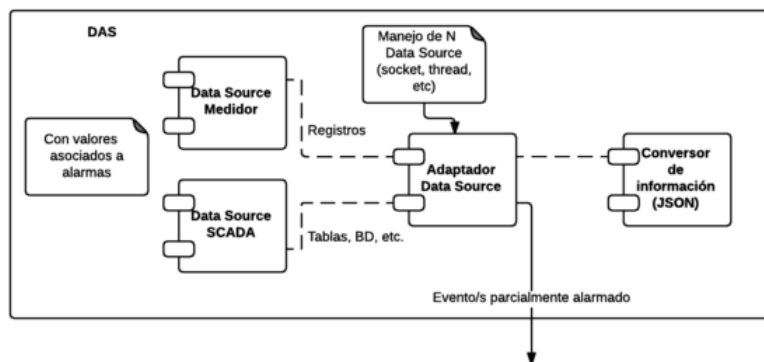


Figura 3. Componente DAS

4.2 Middleware Service

Trigger service: Integra distintos procesos a nivel de servicios que no son aplicaciones, sino que hacen operaciones sobre los datos que se reciben (figura 4). Por ejemplo, en función del valor actual de las tensiones se verifica si se encuentran dentro o fuera de un rango predefinido para establecer una posible alarma. Las funcionalidades más comunes que deben cumplir son: análisis de la topológico de la red, generación de alarmas, entre otros.

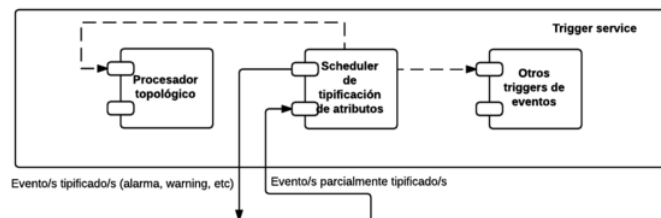


Figura 4. Módulo Trigger Service

Procesador de datos y eventos: Una característica importante a introducir en este tipo de sistemas es la capacidad del mismo a reaccionar ante la ocurrencia de diferentes eventos ya sean simples o complejos. El procesamiento de los eventos simples se refieren a los acontecimientos que están directamente relacionados con los cambios de una característica específica del sistema. Un ejemplo es la detección de una alarma proveniente de un valor de tensión fuera de rango. En cambio los eventos complejos se refiere a los acontecimientos que requieren una inteligencia extra para su detección. Los mismos necesitan un estado parcial

o total del sistema y pueden desencadenar en otros eventos. Un ejemplo es la detección de una falla a partir del análisis de las tensiones y corrientes en diferentes puntos del sistema. Teniendo en cuenta que en este tipo de sistemas existe una recolección de datos de forma continua, una estrategia para el tratamiento de los mismo es la utilización de *machine learning* a través de redes neuronales. Esto permitirá lograr un aprendizaje automático posibilitando al sistema detectar estos eventos y reaccionar ante los mismos.

El componente procesador de datos y eventos se encarga de manejar el *pull and push* de información de gran parte del sistema, brindando flexibilidad en lo que respecta a la producción, detección, y consumo de los acontecimientos del sistema (figura 5). Utilizando como base una arquitectura orientada a eventos, este componente tiene la responsabilidad de:

- Producir los eventos a partir de la información.
- Detectar eventos simples y eventos complejos.
- Consumir los eventos y reaccionar a los acontecimientos (generar nuevos eventos vinculados a las acciones a realizar).
- Suscribir nuevos tipos de eventos (alarmas, variaciones, etc).

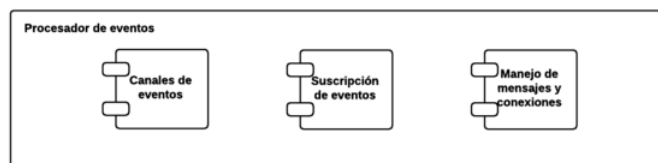


Figura 5. Módulo Procesador de eventos

4.3 Aplicaciones

Como se mencionó en la sección anterior un DMS cuenta con numerosas aplicaciones. Si bien en este artículo se muestran las principales (Unifilar, NCA y Estimador de Estados), agregar una nueva aplicación consiste simplemente en suscribirse al procesador de eventos para el intercambio de información, brindando interoperabilidad y modificabilidad al sistema [19,17,18].

4.4 Persistencia

Es el contenedor de todos los módulos que se encargan de manejar la persistencia de los datos para su posterior utilización. Se reconocen tres componentes que alojan información:

- CIM (RDF/XML): Modelo CIM estático el cual define la topología de la red en formato XML/RDF [20,4]. Contiene todos los parámetros de la red estática, es decir los valores normales y no los valores actuales.

- Modelo dinámico: Modelo acotado. Contiene el último estado de la topología de la red, como por ejemplo el estado actual de los interruptores, último valor de medidas, etc.
- Históricos: Contiene los datos de todos los estados de la red en cuanto a las medidas, estados de interruptores, alarmas, etc.

5 Implementación de un prototipo

En la presente sección se describe la implementación de un prototipo instanciado en una distribuidora de la republica Argentina, 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]. El proyecto propone desarrollar un DMS compuesto de dispositivos y software e ingeniería de comunicación, aplicado al monitoreo y control de las redes de distribución a nivel de Media Tensión.

5.1 Implementación del CAD Unifilar

Las clases del modelo definen un mecanismo de notificación de cambio utilizando el patrón Observer [22]. Esto permite que los componentes vista y controlador sean notificados por el modelo. Debido a que estos componentes se registran en el modelo y el modelo no tiene conocimiento de cualquier vista o controlador específico, esto no rompe la independencia del modelo. Este mecanismo de notificación es lo que proporciona la actualización inmediata.

En la figura 6 se puede ver una vista del diseño del CAD unifilar. La topología de la red es cargada desde un archivo XML/RDF estático en formato CIM. El modelo es actualizado a medida que surgen nuevos eventos, y este, por medio del patrón Observer, notifica a la vista para que se actualice. En caso de que se suscriba una nueva medida para ser tomada en tiempo real, se produce un evento desde la “vista al controlador” que actualiza el modelo para que este se prepare para tomar la medida desde un dispositivo de telemedición. En consecuencia, el modelo suscribe el evento en el procesador de datos y eventos, y se queda a la espera de que este publique nuevos eventos.

La Figura 7 muestra la interfaz del CAD Unifilar desarrollado. En la misma se puede ver una zona de edición de la red que se desea operar. Cada uno de los componentes que se permiten introducir en la interfaz gráfica, como por ejemplo interruptores, barras, cargas, etc, poseen un conjunto de puntos de conexión a partir de los cuales va a ser posible detectar y validar las diferentes conexiones en el modelo, creando de esta manera el modelo CIM con los componentes interconectados. Cabe aclarar que dicho modelo CIM creado durante la edición de una red se corresponde con un modelo interno que debe ser validado antes de crear el RDF/XML final, el cual será cargado posteriormente por los operadores del sistema en su interfaz unifilar. Asimismo, se podrá visualizar en tiempo real las medidas provenientes de cada uno de los equipos de medición configurados,

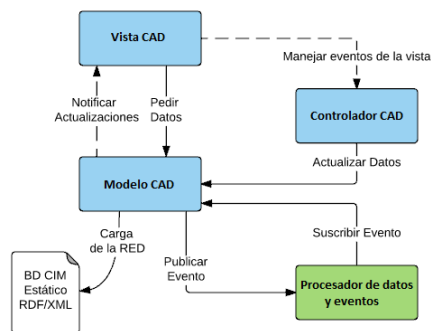


Figura 6. Esquema de actualizaciones y eventos del CAD Unifilar.

junto con las medidas provenientes del estimador. Por otra parte, en la zona central de la interfaz se puede observar el componente GIS, visualizando la trazabilidad entre el CAD Unifilar y la ubicación de cada uno de los componentes del mismo en un plano georeferenciado. Esto permite al usuario operador ubicar y posicionar una situación en el mapa de la región rápidamente. Por otra parte, en la zona inferior de la pantalla se presentan de forma jerárquica las diferentes alarmas provenientes del procesador de eventos, y que el usuario operador debe atender para realizar alguna operación que permita resolver la problemática. La forma jerárquica de visualización de alarmas da la posibilidad a los operadores del sistema de aplicar filtros de acuerdo a diferentes criterios, como por ejemplo el nivel de criticidad de la alarma.

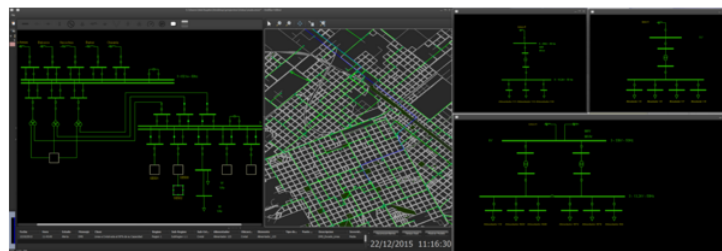


Figura 7. Zoom dispuesto en varias pantallas.

Otra funcionalidad implementada en el prototipo del sistema son las diferentes formas de hacer zoom, es decir diferentes formas de introducirse a cada una de las subestaciones existentes en la red. En este sentido se han diseñado e implementado dos modos de realizar esta tarea. Como primer alternativa el operador podrá abrir nuevas pantallas para las subestaciones que desee visuali-

lizar en tiempo real (Figura 7), permitiendo una perspectiva más detallada del sistema sobre de cada una de las subestaciones.

En segundo lugar, el operador podrá abrir una nueva subestación en la misma pantalla a medida que el mismo se acerca a una zona del unifilar. De esta manera se podrá navegar la red completa, y se logrará obtener mayor detalle de las subestaciones en esa zona.

5.2 Tecnologías utilizadas para la implementación

Para lograr el desarrollo de una arquitectura que permita modelar, diseñar e implementar sistemas de gestión de redes de energía, y que incluya atributos de calidad tales como usabilidad, flexibilidad y extensibilidad, se analizaron diferentes tecnologías capaces de cumplir con estas características y con los objetivos propuestos a lo largo del documento. Haciendo referencia al CAD Unifilar se estudiaron diferentes herramientas de edición de código libre, que permitiera extender su funcionalidad para ser adaptada específicamente a un editor y visualizador de diagramas unifilares. Entre ellas se destaca el framework JGraph [8], el cual se encuentra desarrollado en Java, a través de las librerías de Swing [13] y AWT [23]. El mismo posee una licencia BSD 3-Clause y proporciona funcionalidad para la visualización e interacción a través de nodos, incluyendo características como soporte de plantillas XML con el objetivo de crear nuevas figuras, y que resulta sumamente útil para crear los elementos de los diferentes estándares de visualización. Este framework permitió desarrollar componentes visuales adaptables a estándares característicos para redes de energía eléctrica, como por ejemplo mecanismos de conexión entre componentes bajo el modelo CIM. Con respecto al componente Middleware Service (sección 4), el mismo fue implementado con una arquitectura EDA, donde se utilizó el framework de intercambio de mensajes Apache ActiveMQ, que permite integrar aplicaciones orientadas a mensajes a través de múltiples lenguajes y plataformas. Para la comunicación entre los diferentes componentes que forman parte del intercambio de mensajes se utilizó la implementación del patrón Publish/Subscriber a través de Apache Camel, el cual permite configurar las reglas de ruteo y mediación. Dicha comunicación se realiza a través de la suscripción a diferentes canales de información llamados *Virtual topics*. Por último, y haciendo referencia al manejo del modelo CIM, se utilizó el framework JENA [3] para la construcción y manejo de modelos en formato RDF. Se compone de diferentes APIs que interactúan entre sí para procesar los datos en este formato.

6 Resultados

Para analizar el comportamiento de la arquitectura se realizaron diferentes casos de test sobre algunos atributos y características del sistema. En este sentido se modeló una parte de la red de distribución de la ciudad de Tandil - Argentina, en la cual se configuraron 10 barras, 3 transformadores de 3 devanados, 4 transformadores de 2 devanados y 3 subestaciones, junto con la configuración de los

dispositivos de maniobra como interruptores y seccionadores. Sobre dicha red se tomaron mediciones sobre 3 puntos de interés, junto con otros datos obtenidos desde el SCADA de la distribuidora. A partir de ello se pudieron generar eventos que permiten activar las distintas partes del DMS, como por ejemplo el DAS, CAD Unifilar, Alarmas. Todas las pruebas realizadas forman parte del cronograma de actividades establecido en el proyecto mencionado anteriormente. En cuanto a la visualización de datos, se verificó la instanciación de 1000 componentes eléctricos interconectados, cuyos valores son modificados y refrescados visualmente durante la actualización. Luego de realizada la prueba se pudo constatar que la interfaz no presenta dificultades de refresco (pérdida de valores, componentes u otras anomalías), y se comporta fluidamente antes los cambios. Con respecto al comportamiento del módulo del procesador de eventos y datos, se realizaron pruebas con un servidor corriendo ActiveMQ 5.7.0, una instancia de CAD Unifilar suscripto a un Virtual Topic del mismo recibiendo mensajes, y un productor de mensajes que publica de forma concurrente. Tanto el publicador como el suscriptor fueron implementados con Apache Camel. Para el proceso de test se utilizó un archivo de prueba que genera un Json con 998 tags para simular el DAS. Finalmente, y haciendo referencia al diseño arquitectónico, se analizó la agregación de diferentes módulos en la capa de aplicación. En este sentido, resulto simple la incorporación de nuevos componentes al sistema, lo que permitió entre otras cosas, guardar los datos históricos por día de las mediciones extraídas de los equipos. Los mismos se encuentran instanciados físicamente en la zona de la red eléctrica dispuesta por la distribuidora de la región. En ésta prueba se pudo verificar que la agregación de un nuevo componente se adapta fácilmente a la arquitectura propuesta, demostrando la capacidad de modificabilidad y extensibilidad del sistema. Si bien los componentes agregados implican una funcionalidad acotada, la incorporación de nuevas aplicaciones conlleva un esquema similar sin la necesidad de rediseñar la arquitectura.

7 Conclusión

En este trabajo se presentó una arquitectura de software para modelar e implementar un DMS. La funcionalidad implementada hasta el momento abarca un 40% respecto de la totalidad propuesta, y gran parte de la misma ha sido validada sobre una distribuidora de la región. Dada la importancia que tiene en la región el uso racional de la energía, principalmente en Argentina, la necesidad de contar con nuevas tecnologías que permitan mejorar el conocimiento del estado de la red a fin de tomar decisiones adecuadas, y su adaptación hacia la generación distribuida, genera la posibilidad de aportar soluciones creativas e innovadoras que asistan a los operadores de las distribuidoras. Es por ello que creemos que la solución propuesta en este artículo brindará un importante aporte en el área de distribución, que finalmente impactará en los clientes y en la sociedad mejorando la calidad del servicio.

Referencias

1. Oetec - observatorio de la energía, tecnología e infraestructura para el desarrollo, <http://www.oetec.org/informes/indicadoresenergeticos250114.pdf>
2. Cim users group (2009-2016), <http://cimug.ucaiug.org/>
3. Apache jena (2011)
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. Alder, G.: Design and implementation of the jgraph swing component. Technical Report 1(6) (2002)
9. Ambrosio, R., Widergren Sr, S.: A framework for addressing interoperability issues. In: Power Engineering Society General Meeting, 2007. IEEE. pp. 1–5. IEEE (2007)
10. Apostolov, A.: Requirements for automatic event analysis in substation automation systems. In: Power Engineering Society General Meeting, 2004. IEEE. pp. 1055–1060. IEEE (2004)
11. Britton, J.P.: Moving from data architecture to event architecture in the ems environment. In: Power Engineering Society General Meeting, 2005. IEEE. pp. 2740–2747. IEEE (2005)
12. D2.24, C.W.: Ems architectures of the 21st century. Technical presentation (2008)
13. Eckstein, R., Loy, M., Wood, D.: Java swing. O'Reilly & Associates, Inc. (1998)
14. Fan, J., Borlase, S.: The evolution of distribution. Power and Energy Magazine, IEEE 7(2), 63–68 (2009)
15. Michelson, B.M.: Event-driven architecture overview. Patricia Seybold Group 2 (2006)
16. Pradeep, Y., Medhekar, A., Maheshwari, P., Khaparde, S., Joshi, R.K.: Role of interoperability in the indian power sector. In: Proceedings of GridWise Grid-Interop Forum. pp. 117–1 (2007)
17. Risso, M., Rubiales, A.J., Andres Lotito, P.: Método híbrido para la estimación de estados de sistemas eléctricos de potencia. pp. 3513–352 (2012)
18. Risso, M., Rubiales, A.J., Andres Lotito, P.: Hybrid method for power system state estimation. Generation, Transmission & Distribution, IET 9(7), 636–643 (2015)
19. Risso, M., Rubiales, A.J., Andres Lotito, P., Mayorano, F.: Un estimador de estados basado en el common information model. pp. 1883 – 1893 (2010)
20. Santodomingo Berry, R.J., Rodríguez Mondéjar, J.A., Pilo De La Fuente, E.: Introducción al modelo cim de los sistemas de energía eléctrica (2009)
21. 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)
22. Vlissides, J., Helm, R., Johnson, R., Gamma, E.: Design patterns: Elements of reusable object-oriented software. Reading: Addison-Wesley 49(120), 11 (1995)
23. Zukowski, J.: Java AWT reference, vol. 3. O'Reilly (1997)