

Arquitectura de procesamiento de datos Operativos de activos industriales

Fredy Cabrera, Felipe Trevisan, Filipe Ventura Muggiati

ITAIPU BINACIONAL

Paraguay/Brasil

Resumen

El incremento de activos industriales con capacidad de generación de datos, referentes a su propio estado de operación, demanda la adopción de técnicas de procesamiento que permitan obtener información relevante y así poder generar un valor agregado a los datos en bruto que se generan. Este valor agregado permite tomar decisiones a corto, mediano y largo plazo, ya sea de manera holística o por cada activo, así como también mejorar la conciencia situacional de los procesos operativos. Entre las informaciones relevantes que pueden ser obtenidas se citan: índices de operación, comportamientos estadísticos de activos, detección de anomalías, entre otros. En este trabajo se presenta una arquitectura de procesamiento de datos operativos de un ambiente industrial, en el mismo se especifica el flujo de datos por etapas: desde su almacenamiento inicial, modelado, proceso de *ETL (Extract, Transform and Load)*, disposición final y visualización de los resultados del dato procesado. En todas las etapas son empleadas herramientas específicas, algunas propietarias y otras herramientas de la categoría *Open Source*.

La disposición inicial de los datos operativos de los activos industriales es normalmente en forma de conjunto de series temporales, estos son almacenados en una base de datos especializada para esta estructura de datos. En el motor de almacenamiento las series temporales son especificadas con tres componentes principales que se definen con una tripleta: valor, marca de tiempo y etiqueta, siendo la última la forma de identificar unívocamente a la serie. Debido a la simplicidad de esta forma de representación, es necesario un modelado que permita la estructuración de los datos para facilitar su interpretación. Para esta estructuración se utiliza una plataforma que permite realizar cálculos avanzados sobre las series temporales, generar visiones agregadas (periódicamente, por ejemplo) para los datos, organizar los datos en un árbol jerárquico y además agregar otras fuentes de datos que no necesariamente son del tipo serie temporal. Para facilitar el acceso y la visualización, se extraen los datos procesados en una base de datos relacional en un esquema de *DataWarehouse* a través de un proceso de *ETL*. Los datos en el *DataWarehouse* son consumidos por una herramienta de visualización para presentar las informaciones obtenidas en *dashboards* para los usuarios finales. Finalmente, en este trabajo se muestran ejemplos de aplicativos que utilizan la arquitectura mencionada y los beneficios obtenidos a través de su utilización.

Palabras clave

Procesamiento de datos operativos, DataWarehouse, Procesamiento de datos, Series temporales, ETL

XIV SEMINARIO DEL SECTOR ELECTRICO PARAGUAYO - CIGRE
23 y 24 de Junio 2022

1. INTRODUCCIÓN

La digitalización trae consigo un incremento continuo de datos, y el sector industrial también experimenta este crecimiento constante. Actualmente, los activos industriales cuentan con sensores empotrados que permiten la transmisión/recepción de datos tales como; parámetros de operación, estados, calidad de operación, entre otros [1]. La aparición de activos con las características mencionadas está dando origen a lo que se conoce como *Industrial Internet of Things* [2], esto presenta a las industrias nuevas oportunidades para mejora de sus procesos y productos, así como también el desafío de estar a la altura de poder aprovechar esta oportunidad [3]. Los datos *per se* no aportan un valor agregado, ellos necesitan ser interpretados para transformarlos en información y finalmente esta información necesita ser aprendida e incorporada para obtener como resultado un conocimiento [4].

En la Figura [1], presentada en [5], se tiene el modelo de madurez analítico de Gartner, en donde se describen 4 categorías de analítica de datos: a) Analítica descriptiva: ¿qué pasó? b) Analítica de diagnóstico: ¿por qué pasó? c) Analítica predictiva: ¿qué va a pasar? y d) Analítica Prescriptiva: ¿cómo hacer que pase? El valor agregado y la dificultad también aumentan dependiendo del tipo de analítica.

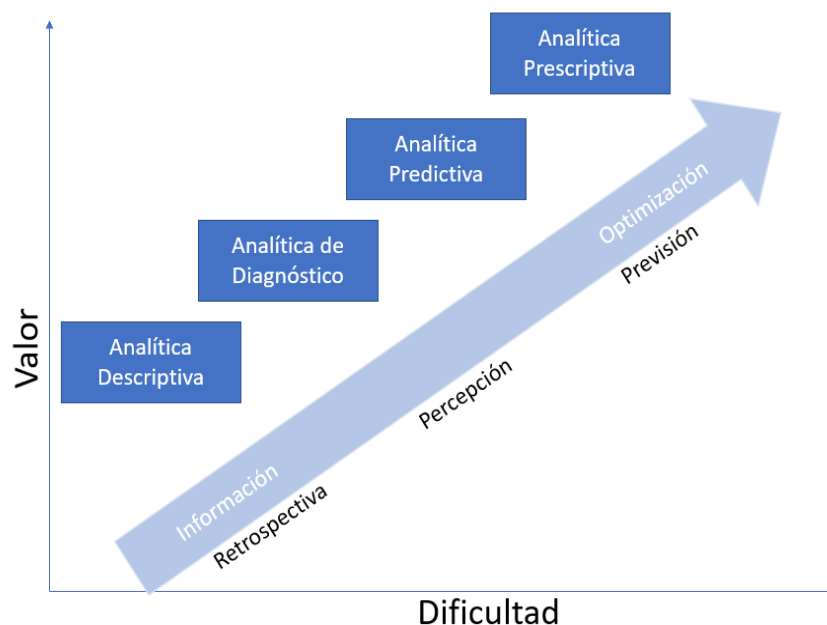


Figura 1: Modelo de ascendencia analítica de Gartner.

Desde el nivel descriptivo se requiere una infraestructura tecnológica de procesamiento de datos, que permita una forma ordenada y predecible el tratamiento de estos. Este trabajo se enfoca en la presentación de una arquitectura de procesamientos de datos operativos industriales. En las siguientes secciones se presentará la arquitectura, la descripción de cada componente, y finalmente un ejemplo de aplicación.

XIV SEMINARIO DEL SECTOR ELECTRICO PARAGUAYO - CIGRE
23 y 24 de Junio 2022

2. ARQUITECTURA

En la Figura [2] se muestra el esquema de la arquitectura implementada para el procesamiento de datos en ambiente industrial. Se muestra el *pipeline* desde la generación de datos, almacenamiento, modelado/procesamiento, proceso de Extracción-Transformación-Carga (*Extract Transform Load*), almacenamiento en esquema de *DataWareHouse* (DW) y finalmente la etapa de visualización. La arquitectura ha evolucionado y adaptado conforme a las tecnologías disponibles a lo largo de aproximadamente 10 años

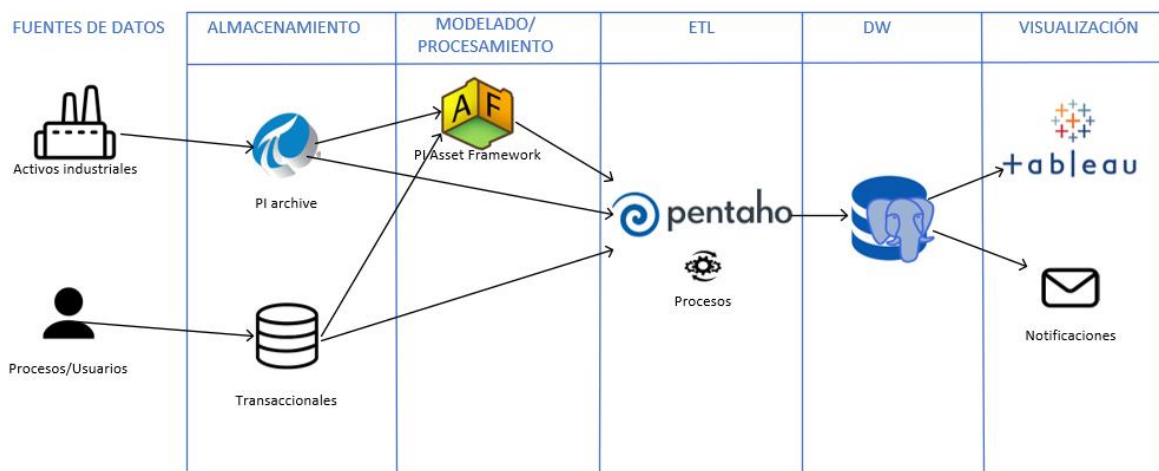


Figura 2: Arquitectura de procesamiento de datos operativos en ambiente industrial.

2.1 Fuentes de datos

Se tienen en esta etapa dos componentes, la principal que es la de los activos industriales. Los datos generados por los activos industriales son del tipo de serie temporal. Una serie de tiempo es “una secuencia de observaciones cronológicas sobre una variable” [6], los intervalos de observaciones pueden ser regulares o irregulares. Ejemplos de datos de series de tiempo son: potencia de generadores, estado de equipamientos, caudal de entrada en bombas, entre otros. El segundo conjunto de datos son las que tienen características relacionales originadas por usuarios y/o procesos de negocio tales como: procesos de solicitud de trabajos, liberación de trabajos, planificación de estado de equipamientos, registros de fallas/defectos, sistemas de gestión de activos, etc.

2.2 Almacenamiento

Debido a la naturaleza de los datos que fueron mencionados en la sección anterior, es necesario utilizar un tipo de almacenamiento conforme a la particularidad de estos. En el caso de los datos provenientes de activos industriales, se utiliza una base de datos de series de tiempo, TSDB por sus siglas en inglés, esta se define como un conjunto de sistemas que son especializados y optimizados para almacenar y retornar datos en forma de series de tiempo asociando pares de tiempo y valor [7], se caracterizan por un alto número de escrituras/inserciones y consultas, por lo general, en forma de agregación sobre un periodo de

XIV SEMINARIO DEL SECTOR ELECTRICO PARAGUAYO - CIGRE
23 y 24 de Junio 2022

tiempo [8]. Estas características hacen que la utilización de bases de datos relacionales tradicionales no sea ideal, ya que deben soportar transacciones ACID, lo que implica un mayor control y mecanismos sofisticados al momento de realizar operaciones de escritura [9]. El advenimiento de IoT/IIoT ha favorecido la aparición de varias bases de datos de series de tiempo, alguno de ellos *open-source* como: InfluxDB, TimescaleDB, KairosDB y OpenTSDB [10] y otras propietarias: kdb+, AVEVA/OSISoft PI, Aerospike, Databus entre otros [11]. Aunque la popularidad de las bases de datos de series de tiempo se haya incrementado solamente en los últimos años, estas existen desde hace algún tiempo atrás, especialmente en el ambiente industrial, en donde son conocidos como *data historians*. En el ambiente industrial donde se desarrolla esta arquitectura de procesamiento es utilizada el Sistema PI de AVEVA/OSISoft, este estuvo presente desde los inicios de la implementación del sistema SCADA. El sistema PI es un esquema completo de almacenamiento y visualización de series temporales, una vista simplificada de los componentes del sistema PI se muestra en la Figura [3], tal como se puede apreciar, actualmente es una solución de *end-to-end*, desde la fuente de los datos hasta el usuario final, sin embargo los componentes principales que son relevantes para la arquitectura presentada son los contenidos dentro del bloque PI-Server: *Data Archive* y *Asset Framework* (AF), el primero de ellos es conceptualmente el TSDB y como tal es el encargado de almacenar y retornar las series de tiempo. El *Asset Framework*, sin embargo, es una capa extra que permite el modelado de datos y la organización jerárquica de las series temporales, este se tratará en mayor detalle en la sección de Modelado/Procesamiento.

La estructura que se maneja en el *Data Archive* es bastante simple, una tripleta que incluye una estampa de tiempo, el valor y la etiqueta. La etiqueta es la forma unívoca de identificar a la serie, y por lo general debe reflejar la naturaleza de la variable que está representando.

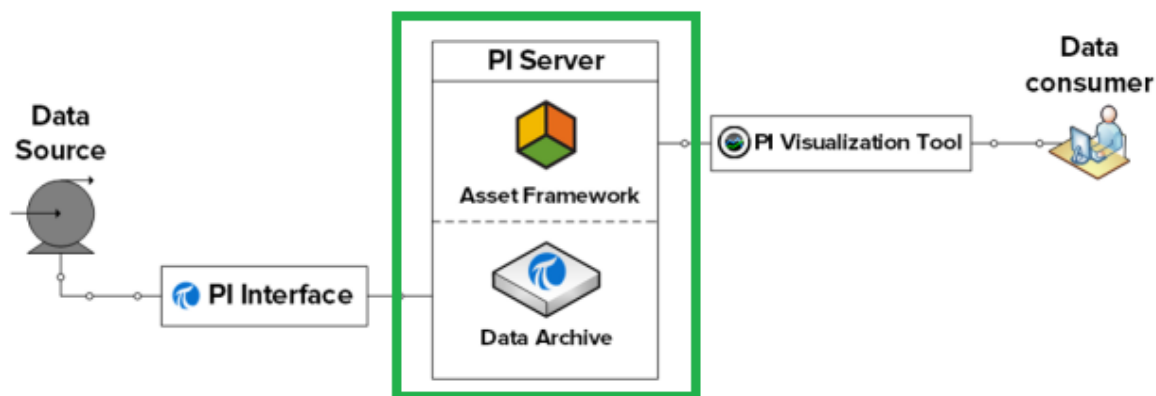


Figura [3]: Arquitectura simplificada del sistema PI.

Las consultas de datos de series temporales al PI pueden realizarse de varias maneras, ya sea utilizando las herramientas propias de visualización que se disponen, así como también las opciones de desarrollo en donde se pueden citar: el *PI-Software Development Kit*, que es una librería que se ejecuta en ambientes Windows que puede ser llamada desde lenguajes de programación que soportan *Component Object Model*, y el controlador JDBC (Java Database Connectivity), este último permite realizar consultas tipo SQL (*Structured Query Language*)

XIV SEMINARIO DEL SECTOR ELECTRICO PARAGUAYO - CIGRE
23 y 24 de Junio 2022

al *Data Archive* utilizando el lenguaje de programación Java o herramientas que utilicen este lenguaje como base. Es importante resaltar que cronológicamente estas herramientas fueron apareciendo según fueron citadas, inclusive en la actualidad se disponen otras opciones de consulta a través de una API *RESTful* y un nuevo kit de desarrollo que se conoce como PI-AF-SDK que reemplazará al SDK original que fue mencionado.

El segundo grupo de almacenamiento es la destinada para los sistemas de los procesos de negocio, en la arquitectura son llamadas de transaccionales y se refiere al conjunto de bases de datos relacionales tradicionales tales como: PostgreSQL, Oracle, MySQL, etc.

2.3 Modelado/Procesamiento

En esta etapa se puede destacar principalmente el *AF-Asset Framework*. El *Data Archive* solo permite consultas y agregaciones muy sencillas y además no cuenta de ningún mecanismo que permita relacionar series de tiempo. Para proveer estas funcionalidades OSISoft desarrolló el *Asset Framework* como un suplemento al *Data Archive*, se define como “una herramienta que permita modelar ya sea objetos físicos y lógicos de una manera que mejor se ajusta en la manera que se desea visualizar los activos y los datos asociados con los mismos” [12]. Se tiene una estructura jerárquica de árboles que permite organizar, relacionar y contextualizar los datos. Además, se tiene una capa de procesamiento analítico, que permite ejecutar y agendar funciones sobre la estructura del árbol que ha sido mencionada, los resultados de estas funciones/análisis pueden a su vez ser almacenadas nuevamente en el *Data Archive* o pueden ser calculadas sobre demanda cuando son consultadas. El AF permite agregar la etapa analítica a la arquitectura que se está presentando. Al igual que el *Data Archive* los resultados y estructuras del AF pueden ser consultadas programáticamente utilizando el conector JDBC de Java, con la API *RESTful* y con el AF-SDK. Cabe mencionar que esta herramienta es relativamente nueva, y anteriormente se suplió su funcionalidad con scripts y/o programas diseñados específicamente para realizar cálculos especiales sobre las series de tiempo.

AF es una de las herramientas que hace que la solución de OSISoft sea ampliamente utilizada en la industria, por dar un ejemplo: Shell tiene 7.5 millones de flujos de datos monitoreados y 100.000 cálculos por minuto, cifras que indican la potencialidad de la herramienta [12].

2.4 ETL y DataWarehouse

Estas dos etapas son expuestas juntas debido a la estrecha relación entre las mismas. Un *DataWareHouse* se define como “una copia de datos transaccionales, estructurado específicamente para consulta y análisis” [13], esta es la definición de uno de los pioneros del enfoque de *DataWareHouse*: Ralph Kimball, cuyo modelo es implementado. El mismo opta por un enfoque dimensional o de estrella, en este esquema se tiene una tabla principal en donde se van registrando los hechos y a esta tabla de hechos se le asignan dimensiones, se conocen cada tabla respectivamente como tablas de hechos y tablas de dimensiones. Las tablas de hechos contienen datos cuantitativos sobre el proceso de negocio y las tablas de dimensiones contienen datos descriptivos sobre los hechos [14]. En el caso de los datos provenientes del sistema PI por lo general las tablas de hechos son datos de un modelo en el *Asset Framework*,

XIV SEMINARIO DEL SECTOR ELECTRICO PARAGUAYO - CIGRE
23 y 24 de Junio 2022

cuyos campos son cálculos de los análisis y donde una dimensión obligatoria es la del tiempo, otras dimensiones son datos descriptivos provenientes de los sistemas transaccionales de negocio. Como infraestructura de almacenamiento se utiliza la base de datos PostgreSQL en donde por tipo de activo/sistema se tiene un esquema.

Ahora bien, debe existir un proceso que tenga como entrada las informaciones de la etapa de almacenamiento/procesamiento y como salida informaciones que tengan la estructura que maneja el Datawarehouse. Este proceso se conoce como ETL, y es el encargado de extraer la información de las etapas de almacenamiento y procesamientos, realizar filtros, limpieza, y la transformación para la estructura del DataWareHouse, y finalmente cargar estos datos en la base PostgreSQL. La herramienta utilizada para ETL es Pentaho. En esta herramienta se maneja el concepto de transformaciones, cada transformación o conjunto de transformaciones puede ser considerada un proceso de ETL. Una transformación es generada gráficamente a través de bloques, el componente (parte de Pentaho) utilizada para realizar estas ediciones es *Spoon*. Se posee también un motor de ejecución de transformaciones, este motor de transformaciones puede ser accedido ya sea por la línea de comandos o por peticiones HTTP, que son los componentes *Kitchen* y *Carte* respectivamente. Un agendador de procesos es utilizado para realizar ejecuciones periódicas de las transformaciones. Como la herramienta está basada en Java, se puede utilizar el controlador JDBC para extraer información tanto del *Data Archive*, PI-AF y de los transaccionales de proceso de negocio.

2.5 Visualización

La última etapa es la conexión con los usuarios finales que están realizando el análisis y tomarán decisiones conforme a las informaciones que están siendo presentadas. Existen innumerables herramientas de visualización, sin embargo, aprovechando el contexto de implantación en otras áreas de la empresa se optó por la utilización de la herramienta *Tableau* como principal método de visualización. Tableau además tiene funciones de agregación incorporadas que permiten al usuario final adentrarse más a las informaciones que le están siendo presentadas.

Además, son utilizadas como medios alternativos de visualización, notificaciones a correos electrónicos, visualizaciones empotradas en páginas webs, entre otros.

3. APLICACIONES PRÁCTICAS

Seguidamente se mostrará un ejemplo completo de aplicación de la arquitectura y se mencionan otras a modo de enriquecer la lista. En el contexto industrial en donde es implementada esta arquitectura existen varias UTRs (Unidad Terminal Remota) cuyos datos de estado son reportados y almacenados en el PI-*Data Archive*. El objetivo es obtener una estadística de indisponibilidad mensual que ayude a identificar equipamientos cuyo comportamiento pueda afectar la operación en tiempo real. La disposición de los datos en el *Data Archive* se muestra en la figura [4] (los complementos de los nombres de las UTRs aparecen tachados en negro por motivos de confidencialidad). El modelado de una UTR en el AF se aprecia en la figura [5], una de las características del AF es que permite la creación de *templates*, estos son un tipo de modelo genérico que pueden ser instanciados múltiples veces. En la sub-figura [5-a] se muestran los atributos del modelo y en la sub-figura [5-b] los análisis

XIV SEMINARIO DEL SECTOR ELÉCTRICO PARAGUAYO - CIGRE
23 y 24 de Junio 2022

implementados. Los análisis básicamente son cálculos que determinan el tiempo que la UTR se encuentra en un estado específico, este cálculo se realiza con una periodicidad de un día. Una porción del árbol jerárquico del modelo instanciado se muestra en la subfigura [5-c]. En el ETL de Pentaho existen bloques de conexión, uno al PI-AF a través del controlador JDBC del PI y otra al *Datawarehouse* para consulta de dimensiones compartidas, como por ejemplo la localización física de los activos. En el *Datawarehouse* se tiene una tabla de hechos cuyos campos son los datos calculados de periodos de tiempo en estados específicos de las UTRs y la cantidad de veces que estuvo en ese estado. Las tablas de dimensiones son el tiempo/fecha, la de localización de la UTR, la dimensión UTR que tiene el *id* y la descripción. La visualización final de se muestra en la figura [6], en ella se muestra un ranking de las UTRs que presentan mayor tiempo de indisponibilidad. Esta es solo una de las aplicaciones que utilizan la arquitectura mencionada, entre las demás aplicaciones podemos citar: cálculos de contabilización energética en tiempo real de la central hidroeléctrica, notificaciones de récords de suministros a la ANDE, índices de operación entre otros.

Name	Data Type	Value	Time Stamp
RTUERR.SE.UTR	Digital	In	10/4/2022 15:55:02
RTUERR.SE.UTR	Digital	In	10/4/2022 15:55:02
RTUERR.SE.UTR	Digital	In	10/4/2022 15:55:02
RTUERR.SE.UTR	Digital	In	10/4/2022 15:55:02
RTUERR.SE.UTR	Digital	In	10/4/2022 15:55:02
RTUERR.SE.UTR	Digital	In	13/4/2022 09:40:22
RTUERR.SE.UTR	Digital	In	10/4/2022 15:55:02
RTUERR.SE.UTR	Digital	In	10/4/2022 15:55:02

Figura [4]. Disposición inicial de datos de estado de UTRs en el Data Archive.

	<table border="1"> <thead> <tr> <th>Name</th> <th>Expression</th> </tr> </thead> <tbody> <tr> <td>Failed</td> <td>if(EventCount('MarkFail','*-1d','*')=0) then 0 else (TagMean('MarkFail','*-1d','*')*EventCount('MarkFail','*-1d','*'))</td> </tr> <tr> <td>Normal</td> <td>if(EventCount('MarkNormal','*-1d','*')=0) then 0 else (TagMean('MarkNormal','*-1d','*')*EventCount('MarkNormal','*-1d','*'))</td> </tr> <tr> <td>Marginal</td> <td>if(EventCount('MarkMarginal','*-1d','*')=0) then 0 else (TagMean('MarkMarginal','*-1d','*')*EventCount('MarkMarginal','*-1d','*'))</td> </tr> <tr> <td>Alarmed</td> <td>if(EventCount('MarkAlarm','*-1d','*')=0) then 0 else (TagMean('MarkAlarm','*-1d','*')*EventCount('MarkAlarm','*-1d','*'))</td> </tr> <tr> <td>OutOfService</td> <td>if(EventCount('MarkOutOfService','*-1d','*')=0) then 0 else (TagMean('MarkOutOfService','*-1d','*')*EventCount('MarkOutOfService','*-1d','*'))</td> </tr> </tbody> </table>	Name	Expression	Failed	if(EventCount('MarkFail','*-1d','*')=0) then 0 else (TagMean('MarkFail','*-1d','*')*EventCount('MarkFail','*-1d','*'))	Normal	if(EventCount('MarkNormal','*-1d','*')=0) then 0 else (TagMean('MarkNormal','*-1d','*')*EventCount('MarkNormal','*-1d','*'))	Marginal	if(EventCount('MarkMarginal','*-1d','*')=0) then 0 else (TagMean('MarkMarginal','*-1d','*')*EventCount('MarkMarginal','*-1d','*'))	Alarmed	if(EventCount('MarkAlarm','*-1d','*')=0) then 0 else (TagMean('MarkAlarm','*-1d','*')*EventCount('MarkAlarm','*-1d','*'))	OutOfService	if(EventCount('MarkOutOfService','*-1d','*')=0) then 0 else (TagMean('MarkOutOfService','*-1d','*')*EventCount('MarkOutOfService','*-1d','*'))	
Name	Expression													
Failed	if(EventCount('MarkFail','*-1d','*')=0) then 0 else (TagMean('MarkFail','*-1d','*')*EventCount('MarkFail','*-1d','*'))													
Normal	if(EventCount('MarkNormal','*-1d','*')=0) then 0 else (TagMean('MarkNormal','*-1d','*')*EventCount('MarkNormal','*-1d','*'))													
Marginal	if(EventCount('MarkMarginal','*-1d','*')=0) then 0 else (TagMean('MarkMarginal','*-1d','*')*EventCount('MarkMarginal','*-1d','*'))													
Alarmed	if(EventCount('MarkAlarm','*-1d','*')=0) then 0 else (TagMean('MarkAlarm','*-1d','*')*EventCount('MarkAlarm','*-1d','*'))													
OutOfService	if(EventCount('MarkOutOfService','*-1d','*')=0) then 0 else (TagMean('MarkOutOfService','*-1d','*')*EventCount('MarkOutOfService','*-1d','*'))													
(a)	(b)	(c)												

Figura 5: (a) Atributos del modelo de UTR (b) Análisis del modelo de UTR (c) Instanciación del modelo de UTR, vista de árbol jerárquico

XIV SEMINARIO DEL SECTOR ELECTRICO PARAGUAYO - CIGRE
23 y 24 de Junio 2022

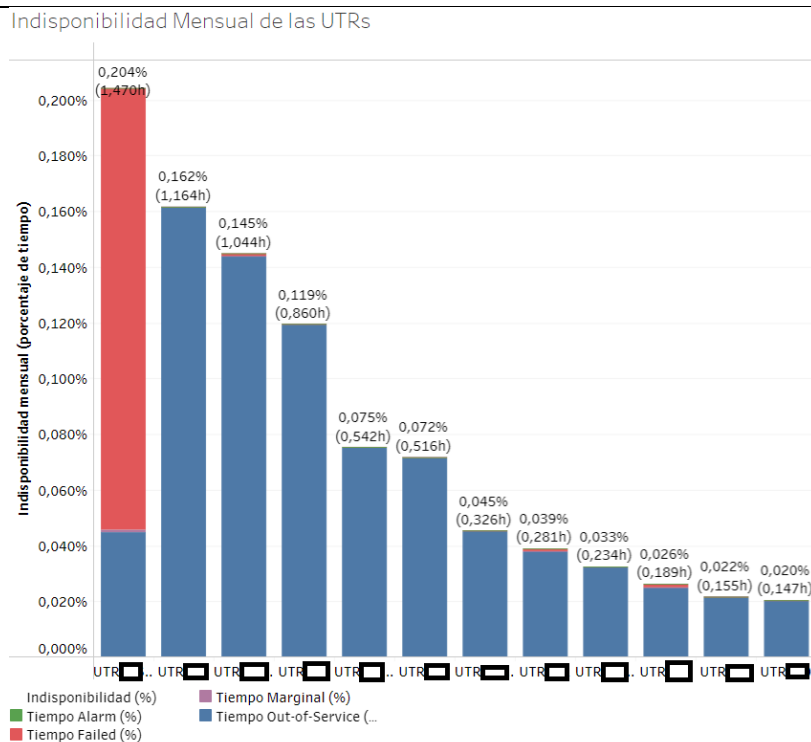


Figura 6: Visualización final de indisponibilidad de UTRs.

4. CONCLUSIÓN Y PROSPECCIONES FUTURAS

Se ha presentado una arquitectura de datos operativos en un ambiente industrial que está integrada por diferentes componentes funcionales y cuyo principal objetivo es la extracción de información que pueda ser convertida en conocimiento útil para los procesos operativos. La principal fuente de datos en el contexto del trabajo es del tipo serie temporal, motivo por el cual se utilizaron herramientas específicas para almacenar y procesar este tipo de datos; además, se tiene la capacidad de procesamiento de datos relacionales convencionales provenientes de aplicaciones de soporte para el ambiente operativo y de planificación: gestión de activos, solicitudes y autorizaciones de trabajo, etc. La arquitectura ha demostrado ser muy útil, prueba de esto es el interés creciente de desarrollar nuevas aplicaciones que utilizan la infraestructura, así como también agregar nuevas fuentes de datos a las ya existentes. Las aplicaciones a su vez han demostrado ser útiles al momento de tomar decisiones. La arquitectura se ha ido adaptando a lo largo de los años para adecuarse a los recursos tecnológicos disponibles y lo seguirá haciendo también para adaptarse al proceso evolutivo de Gartner presentado en la Figura [1]. Su estructura modular permite la inclusión y alteración de módulos/herramientas sin comprometer otros módulos existentes. La mayoría de las aplicaciones/visualizaciones son del tipo descriptivo y en menor medida de diagnóstico, llegar a los otros dos niveles de analítica implicaría probablemente cambios en la arquitectura, en el sentido de inclusión de nuevas herramientas. Un mayor flujo de datos en términos de volumen, variedad y velocidad también fomentarán la implementación de nuevos paradigmas de arquitecturas, como son las arquitecturas Lambda y Kappa [15], concebidas para ambientes de procesamiento en tiempo real y *big data*.

XIV SEMINARIO DEL SECTOR ELECTRICO PARAGUAYO - CIGRE
23 y 24 de Junio 2022

BIBLIOGRAFÍA

- [1] S. Groggert, M. Wenking, R. H. Schmitt and T. Friedli, "Status quo and future potential of manufacturing data analytics — An empirical study," 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 2017, pág. 779-783.
- [2] T. Li, W. -n. Song and J. -h. Zhou, "Research on Artificial Intelligence Industrial Big Data Platform for Industrial Internet Applications," 2021 IEEE 4th International Conference on Electronics Technology (ICET), 2021, pág. 955-959.
- [3] Jay Lee, Hossein Davari Ardakani, Shanhu Yang, Behrad Bagheri, "Industrial Big Data Analytics and Cyber-physical Systems for Future Maintenance & Service Innovation", Procedia CIRP, Volume 38, 2015, pág. 3-7.
- [4] Agnar Aamodt, Mads Nygård, "Different roles and mutual dependencies of data, information, and knowledge — An AI perspective on their integration", Data & Knowledge Engineering, Volume 16, 1995, pág. 191-222.
- [5] V. L. Uskov, J. P. Bakken, A. Shah, N. Hancher, C. McPartlin and K. Gayke, "Innovative InterLabs System for Smart Learning Analytics in Engineering Education," 2019 IEEE Global Engineering Education Conference (EDUCON), 2019, pág. 1363-1369.
- [6] Martin Charlton, Alberto Caimo, "Time series analysis", ESPON | Inspire Policy Making with Territorial Evidence, 2012, pág. 1.
- [7] H. F. Tayeb, M. Karabatak and C. Varol, "Time Series Database Preprocessing for Data Mining Using Python," 2020 8th International Symposium on Digital Forensics and Security (ISDFS), 2020, pág. 1-4
- [8] Grzesik P, Mrozek D., "Comparative Analysis of Time Series Databases in the Context of Edge Computing for Low Power Sensor Networks", Computational Science – ICCS 2020, 2020; pág. 371-383.
- [9] Y. Yang, Q. Cao and H. Jiang, "EdgeDB: An Efficient Time-Series Database for Edge Computing", IEEE Access, vol. 7, 2019, pág. 142295-142307.
- [10] Rui Liu and Jun Yuan, "Benchmarking Time Series Database with IoTDB-Benchmark for IoT Scenarios", arXiv, 2019.
- [11] Bader Andreas, "Comparison of time series databases", University of Stuttgart, 2016, pág. 127.
- [12] Altman, Jacklin. "Easy as Pi: Asset Framework (AF)", Osisoft, 2019, Disponible en: <https://www.osisoft.com/blog/easy-as-pi-asset-framework> [Consultado 7 marzo 2022].
- [13] Reddy, "Data Warehousing, Data Mining, OLAP and OLTP technologies are essential elements to support decision-making process in industries", International Journal on Computer Science and Engineering, 2010.
- [14] Kimball, Ralph and Ross, Margy, "The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling". John Wiley & Sons, Inc. USA, 2002.
- [15] A. Sanla and T. Numnonda, "A Comparative Performance of Real-time Big Data Analytic Architectures," 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), 2019, pág. 1-5