

# P3forI4: arquitectura de referencia para soporte de Industria 4.0 basada en tecnología big data

Dintén R., Yebeles J., López Martínez P., Zorrilla M. y Drake J.M.

<sup>1</sup> Grupo de Ingeniería Software y Tiempo Real (ISTR), Universidad de Cantabria  
[www.istr.unican.es]

**Resumen.** Se describe a alto nivel un modelo de referencia basado en tecnología big data para dar soporte computacional a la industria 4.0, que satisface sus requisitos de heterogeneidad, escalabilidad y tiempo real. La presentación desarrolla tres aspectos: describe los principios arquitecturales que constituyen el modelo de referencia P3forI4, propone un ejemplo de implementación del modelo de referencia construido con tecnología big data disponible actualmente como software abierto y analiza las nuevas prestaciones que se ha de requerir a estas tecnologías para su aplicación a la Industria 4.0.

**Keywords:** Industrie 4.0, Big data, Distributed system, and Performance metrics.

## 1 Industria 4.0 y plataformas de tercera generación.

El objetivo de la arquitectura de referencia P3forI4 es proporcionar criterios arquitecturales para una plataforma computacional distribuida destinada a dar soporte a entornos de industria 4.0. La arquitectura P3forI4 se formula con independencia de las tecnologías de implementación y establece un modelo arquitectural de las tareas de gestión y procesamiento de la información que se ejecutan en el entorno, de los recursos que constituyen la plataforma digital que le da soporte y de los agentes de monitorización que facilitan la supervisión y configuración de los recursos en fase operativa.

Si tomamos como punto de partida la definición de industria 4.0 [1] “*Estrategia de organización y control de los procesos de producción industrial que utiliza la información que se genera en todo el ciclo de vida de sus productos. El proceso de producción se basa en una infraestructura informática que interconecta todos sus sistemas cibernéticos de producción, logística, gestión y supervisión, para que cada uno de ellos tengan acceso en tiempo real a la información que generan los otros, y a través de asociaciones autónomas descentralizadas y auto-regulación, se optimice la producción y se incremente el valor añadido a los productos que se fabrican*”. Resultan para la plataforma informática que le da soporte los siguientes requisitos claves:

- **Diversidad de la información:** El entorno industrial es habitualmente extenso y diverso, por lo que la plataforma computacional de soporte debe tener capacidad gestionar datos con amplios rangos de volumen, variación y velocidad de generación y acceso, y que además, pueden cambiar dinámicamente en el tiempo.

- **Tiempo real y persistencia:** Los requisitos de accesibilidad a la información generada en el entorno es muy diversa. Hay información que requiere ser accedida en tiempo real con latencias acotadas, y otras deben ser persistida para que puedan ser accedidas por aplicaciones que no existían cuando se generó la información.
- **Dinamismo de los flujos de procesamiento:** La información que se gestiona se genera dinámicamente en el entorno, bien en base a fuentes de datos que existen en él, o bien en base al resultado de cadenas de procesamiento que se generan sobre ellos.
- **Interconectividad universal:** Las aplicaciones y componentes software deben tener capacidad de acceder a la información que generan cualquier otro componente del entorno. Las limitaciones de acceso no debe ser consecuencia de arquitectural, sino las establecidas por criterios dinámicos de funcionalidad y seguridad.
- **Heterogeneidad de los recursos:** La plataforma computacional que da soporte a un entorno de la Industria 4.0 es normalmente heterogénea. No se construye de forma planificada, sino que resulta de la integración dinámica de recursos requeridos o embebidos en equipos integrados en el entorno, servidores computacionales específicos agregados al entorno como recursos computacionales y recursos virtuales contratados ocasionalmente en la nube.
- **Escalabilidad de los recursos:** Los recursos computacionales que constituye la plataforma se reclutan o abandonan en fase operativa en base a la carga de trabajo que se produzca. Ocurrirá cuando el sistema requiera escalar horizontalmente modificar su capacidad, o verticalmente cambiar su funcionalidad.
- **Organización autónoma y descentralizada:** Para satisfacer el requisito de escalabilidad frente a la carga de trabajo, los componentes deben tener capacidad de establecer y cancelar interacciones con otros componentes de forma autónoma y descentralizada. Los componentes y aplicaciones debe poder organizarse en base a estrategias de búsqueda, negociación y asociación autónomas y descentralizadas de grupos funcionales que asuman las cargas y responsabilidades específicas que se produzcan.
- **Fiabilidad del sistema:** El entorno industrial tiene muchos aspectos críticos desde el punto de vista vital o económico, por lo que debe operar con estrategia que permitan establecer niveles de inmunidad frente a fallos de los elementos hardware y software declarados críticos.
- **Seguridad de la información:** La plataforma garantiza la seguridad (CIA) de la información a la que acceden o proporcionan los recursos, así como la información de configuración de la propia plataforma (incluida la que define la propia seguridad).

La Industria 4.0 comparten muchos de requisitos con los sistemas big data actualmente operativos en e-comercio y redes sociales, aunque también existen diferencias relevantes entre ellos, como consecuencia de la naturaleza del entorno en que opera y

de la heterogeneidad de los recursos hardware y software que le son propios. Actualmente [2][3] se consideran que las plataformas de tercera generación [4] son la solución en ambos campos, ya que hay que dar respuesta a:

- 1) La necesidad de disponer de alta capacidad de computación en cualquier entorno fijo o móvil (*Ubiquitous & Mobile Computing*),
- 2) La capacidad de gestionar información masiva en tiempo real (*Big Data*), sin necesidad de almacenarla para ser procesada en diferido.
- 3) La posibilidad de escalar dinámicamente la capacidad de procesamiento en base a su virtualización y deslocalización (*Cloud, Fog & Dew Computing*).
- 4) La interacción dinámica de las aplicaciones con entornos inteligentes (*IE & IoT*) y redes sociales (*Social Network*).
- 5) La generalización del uso de los nuevos paradigmas de computación basados en la inteligencia artificial (*AI*) y el aprendizaje automático (*Machine Learning*).

El objetivo de este proyecto es explorar el uso de plataformas de software abierto propias de sistemas de big data y que actualmente están plenamente operativas y probadas, para dar soporte a los entornos de Industria 4.0.

## 2 Claves de la arquitectura de referencia P3forI4

La arquitectura de referencia P3forI4 se propone con objeto de delimitar y jerarquizar las estrategias con las que se organiza, cualifica y asocia la información que se gestiona, las tareas que la procesan, los recursos de la plataforma que dan soporte a su transferencia, almacenamiento y procesamiento y los agentes de monitorización que permiten configurar y gestionar el sistema en su conjunto.

A fin de satisfacer los requisitos ya expuestos para las plataformas de soporte de la Industria 4.0, se propone una arquitectura de referencia basada en tres principios claves:

- **DaaS (Data as a Service):** Un entorno industrial se define al más alto nivel en base a la descripción y caracterización de la información que gestiona. Como se muestra en la Figura 1, el punto de partida en el diseño del sistema digital que da soporte al entorno es el análisis de los datos y la formalización de su gobernanza. De acuerdo con este criterio, los componentes software y los recursos hardware que constituyen la plataforma son sólo medios que se reclutan y escalan a fin de que generen, procesen, den soporte o consuman los datos gestionados.

La gobernanza de los datos es el reconocimiento de que en la Industria 4.0, los datos constituyen un activo estratégico, una ventaja competitiva y su gestión el objetivo de la plataforma informática. Del análisis y tomas de decisiones en la gobernanza de los datos resultan los requerimientos de su gestión que constituyen la base del diseño y de la configuración de la plataforma digital.

De la gobernanza emanan los metadata que en fase productiva están asociados a los datos y que sirven de base para que la plataforma opere de forma autónoma y descentralizada. Los metadata describen:

- La naturaleza, semántica y calidad de los datos que requieren los agentes que los procesan.
- La estimación del volumen, velocidad y variedad de los datos que han de ser gestionados.
- Los requisitos de seguridad (autenticación, integridad, confidencialidad y disponibilidad) frente a riesgos externos y de fiabilidad frente a fallos del sistema.
- Las métricas sobre productividad y utilización que han de ser evaluadas en fase de producción para su gestión dinámica.

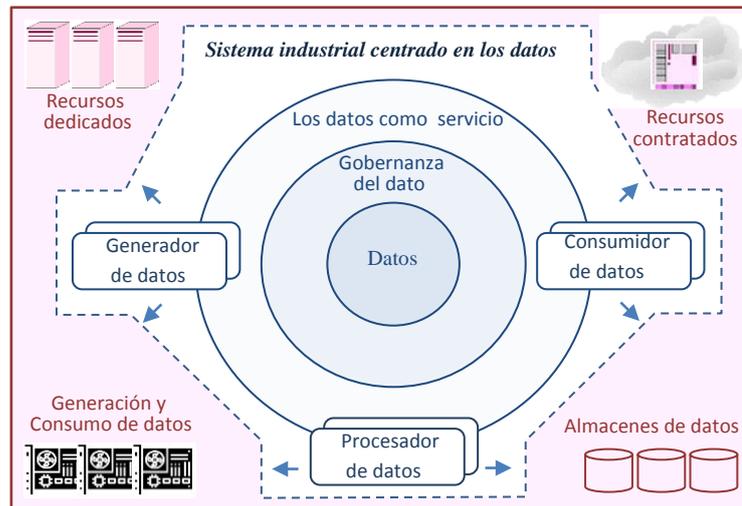


Figura 1.- Sistema industrial centrado en los datos.

Para gestionar la variabilidad de la información del entorno industrial, se organizan los datos por tópicos, que representan flujos de instancias de datos con un mismo tipo de información y que se gestiona con unos mismos criterios de persistencia, durabilidad, disponibilidad, seguridad, integridad, etc.. Los tópicos se registran en la plataforma de producción, y los metadatos asociados a ellos constituyen una información global que es consultada por todos los agentes que operan con sus instancias. Su disponibilidad para ser consultada y la capacidad de notificar sus actualizaciones son las que hacen posible la reconfiguración dinámica de los componentes y servicios de forma descentralizada. El conjunto de tópicos registrados en el sistema representa su dominio de operación.

Las instancias de datos de un t3pico se registran en la plataforma como series temporales inmutables, que alg3un productor escribe una 3nica vez, y son le3da tantas veces como requieran los consumidores que se suscriban a ellas. El ciclo de vida de las instancias est3 definido en el t3pico, y son eliminadas por el entorno en base a criterios de volumen o antig3edad. El almacenamiento de los datos se distribuye en base a particiones, las cuales constituyen el medio de escalar su almacenamiento por el volumen de datos o facilitar el nivel de concurrencia en su procesamiento para escalar su productividad. El n3mero de particiones, su ubicaci3n y los criterios de distribuci3n de las instancias entre ellas, se realiza en base a metadatos asignados al t3pico y utilizando criterios de distribuci3n equilibrada o de campos claves.

Las aplicaciones que incorporan la funcionalidad del entorno se conciben como workflows de tareas de procesamiento que operan en base a una estrategia reactiva. Un workflow se ejecuta en base a la ocurrencia de un determinado patr3n de instancias de t3picos declarados en el entorno.

El conjunto de tareas que constituyen un workflow se organizan en base a una relaci3n de flujo de control con estructura de grafo sin bucles (Figura 2). La ejecuci3n del workflow se inicia en una tarea ra3z que es activada con la ocurrencia del patr3n de activaci3n. Las activaciones de las restantes tareas de workflow se ejecutan en base a las dependencias de sincronizaci3n (*pipeline*, *branch*, *fork*, *merge* y *join*) establecidas en el workflow. Para implementar los mecanismos de transferencia de flujo de control entre tareas, el workflow puede crear y registrar t3picos privados cuyo ciclo de vida coincide con el del workflow que lo crea.

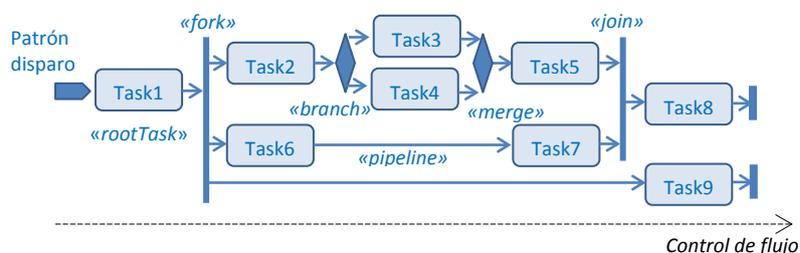


Figura 2.- Los workflows como gr3ficos ac3clicos de tareas.

Las tareas de procesamiento se implementan como componentes software desacoplados que se declaran subscriptoras de los t3picos que las activan y como publicadores de los t3picos que genera. Cada tarea de un workflow se puede instanciar de forma replicada en m3ltiples nodo de procesamiento, bajo el criterio de que el c3digo de la tarea se transfiere al nodo en que est3n los datos, y no se mueven los datos a donde se instancia el c3digo de la tarea. Las tareas de un workflow que corresponde a la ejecuci3n de una ocurrencia del patr3n de disparo se pueden ejecutar distribuidas por diferentes nodos de la plataforma, y as3 mismo, las ejecuciones correspondientes a la ocurrencia de diferentes patrones

de disparo se pueden ejecutar concurrentemente entre sí. El número de réplicas del código de cada tarea de procesamiento que se distribuye en la plataforma, y el nivel de concurrencia con el que se ejecutan las replicas, es función de la estrategia de particionado asignada al tópic y de mecanismo de planificación distribuida que se asigne.

- PaaS (La plataforma como servicio):** Las características claves de las plataformas computacionales de los entornos industriales son su heterogeneidad, su naturaleza distribuida y su necesidad de escalar. Una parte relevante de la capacidad computacional corresponde a sistemas embarcados (*dew computing*) (en sistemas ciberfísicos, aviones, trenes, satélites, terminales de venta, etc.) en los que su conectividad al hardware y/o las condiciones operativas requieren utilizar los recursos específicos establecidos por los fabricantes de los equipos. Otra gran parte de la capacidad computacional está compuesta por los recursos destinados específicamente al entorno industrial (*fog computing*) (data and dispatching centers, data warehouses, communication centers, etc.), y por último, cada día es mas frecuente contratar recursos públicos (*cloud computing*) que suplan bajo demanda la capacidad de computación que se necesita. En las plataformas digitales de la Industria 4.0 no sólo se necesita que los recursos estén interconectados y puedan intercambiar información, sino ofrecer capacidad de reconfiguración dinámica que permita disponer de su capacidad en base a las necesidades de cada momento. Si se reconoce que la heterogeneidad de los recursos existe porque se requiere para muchas de las funciones específicas del entorno industrial, y lo que queda es orquestar la capacidad de computación excedente, a través de un conjunto de servicios middleware que los haga utilizables por la plataforma bigdata que utiliza el entorno.

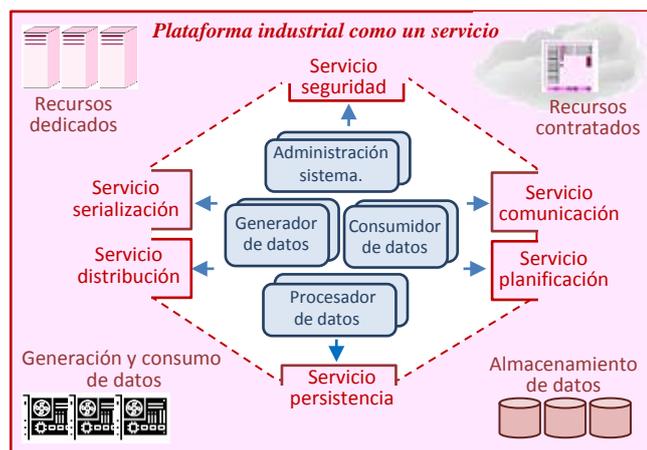


Figura 3.- Plataforma computacional como conjunto de servicios.

En la arquitectura de referencia, los recursos computacionales de la plataforma se organizan en nodos de procesamiento físicos o virtuales que se consideran las unidades de distribución de los datos y de despliegue de la tareas de procesamiento y en redes de comunicación que transfieren información entre ellos. Sin embargo, los recursos que constituyen la plataforma digital del entorno industrial cambian en el tiempo, bien por fallo de los equipos o por el escaldo de la plataforma en base a la carga de trabajo, y los servicios hacen referencia a agrupaciones de recursos (*clusters*) que definen los ámbito de gestión de los servicio con los que se acceden a ellos.

Como se muestra en la Figura 3, los agentes que gestionan los flujos de datos acceden a los recursos computacionales a través de las interfaces de los servicios que definen la plataforma de ejecución como un servicio. Así mismo, la configuración del servicio establece los modos de operación y la forma de uso de los recursos, y a través de ellos el comportamiento de la plataforma. Todos los servicios que se definen, son replicables, redundantes, distribuidos y escalables, esto es, se pueden definir versiones del servicio por ámbitos de tópicos, mantienen un número de replicas de la información que gestionan para garantizar su robustez frente caída de los recursos en que se almacena, permite instalan múltiples brocker de acceso al servicio y permiten definir por configuración los recursos que le dan soporte. En la arquitectura de referencia se definen la funcionalidad genérica y la responsabilidad que asume en el acceso a la plataforma, pero aspectos esenciales como la interfaz de acceso y los parámetros de configuración se han dejado que sean dependientes de la implementación del servicio que se utilice.

Los servicios que definen la plataforma son:

- **Servicio de serialización:** Resuelve la serialización de las instancias de los tópicos que se necesita para intercambiar información a través de una red o de un recurso de persistencia, con independencia de lenguaje de programación con el que se han codificado los agentes que acceden a él. El servicio de serialización define un lenguaje para formalizar la estructura de la información (esquema) y un formato binario para codificar la información serializada. El esquema se asocia como un metadato del tópico, y con ello resulta disponible para escribir o leer del entono cualquier instancia del entorno.
- **Servicio de distribución:** Resuelve el acceso seguro a una información global compartida en un sistema distribuido. En la plataforma digital proporciona dos servicios claves: es el medio de registro y consulta de los metadatos que cualifican los tópicos y de los datos de configuración y coordinación de los servicios distribuidos.
- **Servicio de comunicación:** Servicio distribuido que facilita el registro de las instancias de los tópicos de la plataforma y la transferencia de instancias entre agentes del entorno en base al paradigma publicador/subscritor. Este servicio gestiona el ciclo de vida de las instancias de los tópicos, escala la capacidad de gestión del volumen de datos en base a particiones y garantiza la distribución de los datos en base a la declaración de grupos de subscriptores.

Todos estos aspectos se configuran en base a metadata que se asocian a las descripciones de los tópicos.

- **Servicio planificación:** Servicios que planifican las ejecuciones de las tareas de procesamiento en los nodos en que están almacenados los datos. Bien por transferencia e invocación de los códigos de procesamiento, o por activación de los componentes software que están instanciados en los correspondientes nodos.
  - **Servicio de persistencia:** Servicio distribuido de almacenamiento persistente de instancias de los datos seleccionadas para que su ciclo de vida superen el nivel de persistencia establecido en el servicio comunicación general para las instancias del tópico. En una plataforma big data hay que balancear las características de consistencia, disponibilidad y particionado de los datos por los que hay que tener en consideración los diferentes paradigmas NoSQL.
  - **Servicio de seguridad:** Servicio distribuido que garantiza la autenticación de los agentes, y la integridad, confidencialidad y disponibilidad de la información. En una plataforma big data las reglas de seguridad deben ser dinámicas y estar orientada al dato, esto es, en los metadatos asociados a cada tópico debe incluir los criterios sobre quién y cómo puede acceder al dato y bajo qué limitaciones.
- **MaaS (La monitorización como un servicio):** En una plataforma descentralizada, se necesita un mecanismo que mantenga actualizada la información sobre el estado global del sistema, la disponibilidad de los datos y los niveles de utilización de los recursos. El servicio de monitorización mantiene una información global del sistema, que facilita la instanciación de los componentes software y de los servicios, así como la notificación de los cambios significativos del estado y de la utilización que los recursos de la plataforma para que se adapten dinámicamente.

A la monitorización de la plataforma contribuyen dos elementos: un servicio de monitorización distribuido que es mantenido por la plataforma y que está constituido por un conjunto de recursos activos instalados en sus nodos y un conjunto de agentes de monitorización que son asociados a los componente software y a los servicios que dinámicamente se instancien.

Los recursos del servicio de monitorización se configuran e instancian cuando los nodos en los que se instalan se incorporan a la plataforma digital. La estrategia de configuración y despliegue de los recursos de monitorización permiten balancear el impacto de la monitorización sobre el tráfico de datos en la red frente al requerimiento de memoria para el almacenamiento de la información en los nodos. Los agentes de monitorización se configuran e instancian como parte de la instanciación del componente o del servicio al que están asociado.

### 3 Ejemplo de implementación de la plataforma en base a la tecnología del proyecto Apache.

Como prueba de concepto se ha construido una plataforma experimental que sigue los criterios de la arquitectura de referencia propuesta y que permita cubrir los servicios requeridos, con tecnología de software abierto existente. Hay dos aspectos claves que deben ser elegidos cuando se implementa el sistema digital del entorno industrial:

- Tecnologías con la que se implementan los servicios middleware que constituyen de hecho la plataforma.
- Tecnologías de componentización de los módulos que implementan las tareas de procesamiento.

En la plataforma experimental, la mayoría de los servicios se han implementando usando tecnología desarrollada en el proyecto de software abierto Apache [5]. Sólo en el servicio de monitorización se ha utilizado la tecnología Prometheus ajena a ese proyecto. En la elección de la tecnología se han priorizado los aspectos de distribución y escalabilidad frente a la de predecibilidad temporal. En la tabla 1 se muestran los elementos que se han utilizado.

Tabla 1.- Tecnologías Plataforma computacional como conjunto de servicios.

Servicio	Producto	Suministrador	Doc.
Serialización	AVRO	Proy. APACHE	[6]
Distribución	Zookeeper	Proy. APACHE	[7]
Comunicación	Kafka	Proy. APACHE	[8]
Planificación	Spark	Proy. APACHE	[9]
Persistencia	Cassandra	Proy. APACHE	[10]
Seguridad	Pendiente	-	
Monitorización	Prometheus	PROMETHEUS	[11]

El servicio de seguridad es un aspecto clave en una plataforma distribuida descentralizada, en la que se busca que facilite estructuralmente la accesibilidad universal a los datos. Las restricciones de acceso sólo van a estar establecidas por las reglas de autenticación, confidencialidad e integridad que imponga el servicio de seguridad. Todos los servicios utilizados en la plataforma experimental son distribuidos y tienen incorporados mecanismos de seguridad propios, sin embargo, los mecanismos de seguridad que ofrecen están asociados a sus puertos de acceso, y no se corresponde con lo establecido en la arquitectura de referencia que es su asociación a los tópicos. El diseño que se ha previsto (aún no implementado), introduce un servicio de seguridad distribuido al que acceden los otros servicios de la plataforma en su fase de configuración, o tras notificación de cambio, para su auto configuración de acuerdo a las reglas de seguridad asociadas a los tópicos.

Los módulos software que constituyen las tareas de procesamiento de la información se han de desplegar dinámicamente por el servicio de planificación, y por tanto, han de estar diseñados de acuerdo con alguna tecnología de componentes común, que sea compatible con él. En la plataforma experimental se distribuyen los módulos como paquetes

JAR de código Java. Esta elección se ha realizado porque en todos los nodos se disponen de máquinas virtuales Java, con independencia de su sistema operativo con el que opere. Así mismo, estos módulos ofrecen opciones estándar para introducir agentes de monitorización en su instanciación, aun cuando sean componentes legados con código inaccesible.

#### 4 Adecuación de la tecnología Apache para un entorno industrial.

La tecnología Apache ha sido desarrollada principalmente para E-comercio y Redes sociales, que tienen características comunes de distribución, descentralización y escalabilidad con los entornos de la Industria 4.0, pero difiere en dos aspectos esenciales:

- *Requisitos de tiempo real*: El concepto de tiempo real de los sistemas big data convencionales, es relativo a la capacidad de procesar los flujos de datos sin necesidad de ser almacenados persistentemente para su posterior tratamiento en diferido. Para garantizar este requisito, utiliza estrategias de monitorización de los niveles de utilización de los recursos y escalado dinámico de los mismos. En el caso de los sistemas industriales el tiempo real es relativo a la acotación de las latencias dentro de un paradigma reactivo evento/respuesta, y para ello se utiliza la creación de threads y su planificación.  
Con independencia, de que la tecnología utilizada basada en sistemas operativos Linux, Window, ect. y lenguajes como Java, Python, etc. no permita sistemas estrictamente predecibles, parece que el volumen y la variabilidad de la información que se gestiona no es compatible con el cumplimiento de requisitos de tiempo real en base a la utilización de threads y su planificación. Habrá que evolucionar hacia nuevas estrategias no reactivas basadas en un escalado generoso de los recursos que garantice que el generador de un dato lo puede registrar en la plataforma en tiempo acotado, y cualquier consumidos puede acceder al dato que necesita en tiempo acotado. El tiempo de generación asociado al dato por el productor, es el que deberá ser analizado por el consumidor para determinar si le es útil o no.
- *Localización de las fuentes y consumidores de información*: En los sistemas big-data estándar, las fuentes y los consumidores de información están desacoplados desde el punto de vista geográfico, por lo que los datos se distribuyen a cualquier nudo, de acuerdo con un criterio de equilibrio de cargas, y en él, permanecen hasta que son procesados. En el caso de un entorno industrial, existe un acoplo relevante entre los puntos en que se generan los datos y los puntos en que se consumen, (los datos que resultan del procesado de datos generados en un equipo o planta industrial, son principalmente utilizados por los consumidores del mismo equipo o planta, respectos de otros puntos de la cadena de producción). Si se ha de mantener el principio de que los datos se ubican como inalterados en un recurso, y es el código el que se traslada al recurso para su procesamiento, se necesita que en los

procesos de escalado las particiones de los tópicos se ubique en recurso específicos de acuerdo con claves del contenido de sus instancias (por ejemplo, una clave de su ubicación geográfica).

Los servicios utilizados en la plataforma experimental permiten intervenir en el particionado de los datos en base a claves, pero requiere configuraciones complejas, que no siempre son compatibles con el escalado dinámico en fase de producción.

## 5 Resultados y líneas futuras.

Este trabajo presenta una arquitectura de referencia para dar soporte a un entorno industrial concebido de acuerdo con los requisitos que impone la Industria 4.0. Se ha utilizado una arquitectura próxima a la entornos big data convencionales, ya que comparte con ella, se satisfacían las características de distribución, desacoplo y volumen y variedad de datos. Sin embargo, las características de tiempo real difieren considerablemente entre ambos y requieren de nuevos planteamientos.

La implementación de prueba se ha construido en base a la tecnología del proyecto Apache y sobre ella se han realizado diferentes pruebas y valoraciones con un caso de uso real propuesto por una empresa de distribución eléctrica.

## Agradecimiento

El presente trabajo se ha realizado dentro del proyecto del Plan Nacional de investigación “Sistemas Informáticos predecibles y confiables para la Industria 4.0” (TIN2017-86520-C3-3\_R)

## Referencias

1. i-Scoop: "Industry 4.0: the fourth industrial revolution – guide to Industrie 4.0". <<https://www.i-scoop.eu/industry-4-0/>> (2016)
2. Lee J. Y., Yong J. S., Kim B.H.: "Big Data Analytics Platform for Smart Factories in Small and Medium-Sized Manufacturing Enterprises: An Empirical Case Study of a Die Casting Factory". Int. Journal of Precision Engineering and Manufacturing Vol. 18, No. 10, pp. 1353-1361 Oct. 2017.
3. Basanta-Val P.: "An Efficient Industrial Big-data Engine". IEEE Transactions on Industrial Informatics Vol. 14 , No: 4 , pp.1361 - 1369 , Apr. 2018.
4. Kwang K.: "Third platform' shift triggers enterprise software evolution". <<http://www.zdnet.com/article/third-platform-shift-triggers-enterprise-software-evolution/>> (2013).
5. The Apache Software Foundation <<http://www.apache.org/>> (2017).
6. The Apache Avro project: "A data serialization system. <<http://avro.apache.org/>>
7. Junqueira F. y Reed B.: "ZooKeeper: Distributed process Coordination". O,Reilly, (2014).

8. Apache Kafka project: "A distributed streaming platform". «<http://kafka.apache.org/>» (2017)
9. Apache Spark Project: "A fast and general engine for large-scale data processing". «<http://spark.apache.org/>» (2017)
10. Apache Cassandra Project: "Manage massive amounts of data, fast, without losing sleep". «<http://cassandra.apache.org/>» (2018)
11. The Linux Foundation: "From metrics to insight: Power your metrics and alerting with a leading open-source monitoring solution". «<https://prometheus.io>» (2019)