

# Diseño e implementación de un banco de pruebas para evaluar plataformas middleware de internet de las cosas: un estudio acerca de FI-WARE en el ámbito de la agricultura de precisión

(Recibido: 02/05/2016; Aceptado: 11/07/2016)

Martínez, R.; Pastor, J.A.; Álvarez, B.; Iborra, A.

División de Sistemas e Ingeniería Electrónica (DSIE).

Universidad Politécnica de Cartagena, Campus Muralla del Mar s/n, 30202, Cartagena, Murcia, España.

Teléfono: 699662296

Email: rmc9@alu.upct.es

**Resumen.** Una de las tecnologías utilizadas en la agricultura de precisión son las redes de sensores inalámbricas. Durante los próximos años, se espera un incremento significativo en su despliegue in situ que provocará numerosos desafíos técnicos. Diferentes middleware allanan el camino para superar dichos desafíos. En este trabajo se comprueba si uno de estos middleware, FI-WARE, podrá escalar en la misma medida en la que lo harán las aplicaciones agrícolas. Para ello, hemos creado un banco de pruebas en el que se han simulado diferentes despliegues y condiciones de carga.

**Palabras clave.** Agricultura de Precisión; Internet de las Cosas; FI-WARE; Redes de Sensores Inalámbricas

**Abstract.** One of the technologies used in precision agriculture are the wireless sensor networks. During the next years, a meaningful increase is expected in its deployment in situ that will cause numerous technical challenges. Different middleware pave the way to overcome these challenges. In this work it is verified if one of these middleware, FI-WARE, will be able to scale in the same extent in which agricultural applications will do it. For this reason, we have created a test bench in which different deployments and load conditions have been simulated.

**Keywords.** Precision Agriculture; Internet of Things; FI-WARE; Wireless Sensor Networks

## 1. Introducción

La agricultura de precisión (AP) tiene como objetivo aumentar la productividad de las explotaciones agrícolas mediante la captura e interpretación de datos referentes al clima, al tiempo atmosférico, a las características del terreno y al estado de los cultivos.

El mayor reto con el que se enfrenta la AP es capturar suficientes datos con una calidad óptima para la toma de decisiones. En espacios abiertos, la mayor parte de los datos proviene de fotos aéreas realizadas por satélites y aviones así como de sensores y cámaras montados en la propia maquinaria agrícola. En consecuencia, la agricultura de precisión necesita utilizar una gran variedad de tecnologías. En este escenario, el procesamiento se basa frecuentemente en técnicas estadísticas y de visión artificial donde la latencia no es un parámetro de gran relevancia.

Un caso significativo de Internet de las Cosas (IoT) son las redes de sensores inalámbricas (WSNs), en el que nuestro grupo tiene una gran experiencia [1-3].

Actualmente, las WSNs in situ no están muy extendidas debido al elevado precio de los sensores y a los problemas que supone su alimentación en lugares donde no existe tendido eléctrico. En ausencia de fuentes de alimentación, la actividad de los sensores para capturar y transmitir datos debe limitarse lo máximo posible. Sin embargo, es previsible que estas limitaciones cambien muy significativamente en los próximos años, debido a los avances en miniaturización y alimentación, así como a la reducción de precio que supondrá la fabricación

masiva de sensores. Es posible, por tanto, pensar en escenarios densamente poblados por sensores, que miden parámetros del suelo (p.ej. humedad o concentración de nutrientes) o de la planta (p.ej. estrés hídrico).

Capturar, transmitir, almacenar y procesar este volumen de información procedente de las WSNs se traduce en una serie de desafíos, especialmente en cuanto a tecnologías de integración, comunicaciones, bases de datos y computación. Afortunadamente existen diferentes middleware que liberan a los técnicos de la realización de la mayor parte del trabajo, si bien sólo algunos de ellos están orientados hacia la IoT. Uno de estas tecnologías middleware es FI-WARE cuya arquitectura continúa en proceso de implementación y cuya evolución está siendo seguida de cerca por nuestro grupo.

La meta de este trabajo es comprobar que FI-WARE podrá escalar en la misma medida en la que lo harán las aplicaciones agrícolas. La respuesta es importante, porque el uso de una plataforma determinada supone una apuesta a largo plazo. Para conseguir tal fin, hemos creado un banco de pruebas.

El resto del documento se estructura como sigue. En la Sección 2 se introducen aspectos relativos al proyecto FI-WARE. En la Sección 3 se describe el banco de pruebas utilizado. En la Sección 4 se muestran los resultados. Finalmente, se presentan las conclusiones extraídas de este trabajo.

## 2. FI-WARE

### 2.1. Descripción

El proyecto FI-WARE [4] está siendo desarrollado como parte del programa *Future Internet Public Private Partnership* lanzado por la Comisión Europea en colaboración con la industria de las tecnologías de la información y comunicaciones.

El principal objetivo de FI-WARE es llegar a ser la plataforma central del *Future Internet* (FI). Por este motivo, su arquitectura es abierta y basada en elementos denominados *Generic Enablers* (GEs) que ofrecen funciones reutilizables y comúnmente compartidas por múltiples áreas de uso a través de diversos sectores. Además, dicha arquitectura es muy extensa estructurándose en seis capítulos técnicos. Por este motivo, su estudio queda fuera del ámbito de este trabajo exceptuando aquellos capítulos técnicos que proveen los GEs que facilitan el desarrollo de aplicaciones en el ámbito de la IoT.

### 2.2. Arquitectura IoT

La arquitectura IoT de FI-WARE es el resultado de la combinación de los GEs de los capítulos *IoT Services Enablement* y *Data/Context Management*. Se trata de una arquitectura compleja que permite el despliegue de diversos escenarios en función de factores tales como el número de dispositivos IoT, el tipo de protocolos de comunicación, etc.

La Figura 1 ilustra uno de los posibles escenarios caracterizado por el hecho de que los dispositivos limitados utilizan protocolos de comunicación estándar.

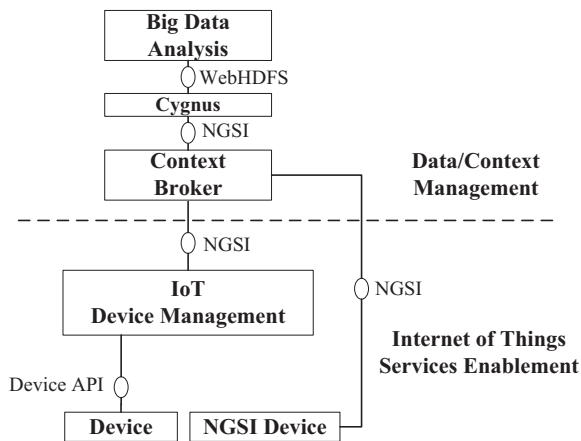


Fig. 1. Arquitectura IoT de FI-WARE.

En FI-WARE se distinguen dos tipos de dispositivos atendiendo a sus prestaciones *hardware* y *software*: el primer tipo abarca los dispositivos de recursos limitados que utilizan protocolos de optimización de energía mientras que el segundo tipo reúne todos aquellos que implementan el protocolo de comunicaciones OMA NGSI [5].

Los dispositivos tipo1 se comunican con el *GE IoT Device Management* que principalmente implementa las siguientes funciones: traducción de protocolos de comunicación estándar al protocolo NGSI, asignación

de entidades de contexto OMA NGSI a las cosas y características para la identificación de y comunicación con recursos IoT. Tanto el *GE IoT Device Management* como el dispositivo tipo2 se comunican con el *GE Context Broker* mediante el protocolo NGSI.

El *GE Context Broker* habilita la publicación de la información de contexto por parte de los productores de contexto (*IoT Device Management* o dispositivo tipo2), para que dicha información esté disponible para otros consumidores de contexto (*GE Big Data Analysis*). El principio fundamental soportado por el *GE Context Broker* es el de conseguir total disociación entre los productores de contexto y los consumidores de contexto. Esto es posible gracias a la implementación del patrón de diseño *Publish/Subscribe* [6].

Cygnus es un adaptador utiliza la API WebHDFS [7] para inyectar de forma automática en el *GE Big Data Analysis* los datos de contexto procedentes del *GE Context Broker*, creando una vista histórica de tales datos.

Finalmente, el *GE Big Data Analysis* está destinado a desplegar medios para analizar datos por lotes, con el fin de conseguir, al final, perspectivas sobre estos datos que revelan nueva información que estaba oculta. Los datos por lotes son almacenados por adelantado, y la latencia en este caso no es un parámetro extremadamente importante.

## 3. Banco de Pruebas

El banco de pruebas ha sido ideado con el objetivo de medir el rendimiento de la plataforma FI-WARE. En la fase de diseño se han seguido las siguientes premisas:

- Orientado inicialmente hacia FI-WARE, pero adaptable para su uso con otras plataformas, lo que permitirá estudios comparativos.
- Orientado a la medida del rendimiento y de la capacidad de almacenamiento de datos, pero extensible para incorporar otras propiedades.
- Parámetros de las pruebas en origen (generación de datos): número. de sensores, frecuencia de generación de datos y *payload*.
- Parámetros de las pruebas en el servidor: número y características de máquinas virtuales (VM) desplegadas, número y características de los GEs instanciados, parámetros del inyector Cygnus y capacidad de almacenamiento en la nube.
- Configuración y arranque automático de los tests contra instancias de FI-WARE.

### 3.1. Diseño

La Figura 2 ilustra el diseño del banco de pruebas, que está compuesto por los *Generic Enablers* descritos en el apartado 2.2. El sistema propuesto simula el comportamiento de una red sensores de tipo2 por medio de la implementación de las operaciones del protocolo de comunicaciones NGSI. Esta implementación se traduce en una aplicación cliente/servidor que se ejecuta en un nodo maestro y

12 nodos esclavos ubicados en los laboratorios del grupo de investigación División de Sistemas e Ingeniería Electrónica (DSIE) de la Universidad Politécnica de Cartagena.

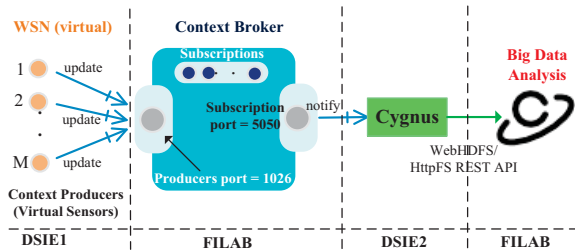


Fig. 2. Banco de pruebas.

A continuación, se describen los flujos de información que tienen lugar en el banco de pruebas. En primer lugar, un usuario del sistema configura la prueba a través de la interfaz gráfica de usuario del nodo maestro (DSIE2). Posteriormente, una trama de parámetros de prueba es generada y transmitida a los nodos esclavos (DSIE1). Cada nodo esclavo decodifica dicha trama, se sincroniza con el nodo maestro y ejecuta el método de prueba en un número determinado de clientes NGSI. Como resultado,  $M$  sensores virtuales envían información de contexto a una instancia del *Context Broker* desplegada en la nube de FIWARE (FILAB).

Cada vez que se produce la actualización de la información de contexto asociada a una entidad registrada en la base de datos del *Context Broker* (en nuestro caso cada entidad modela a un sensor virtual), se envía una notificación al *Big Data Analysis*. Esta notificación se realiza a través del inyector *Cygnus*. De esta forma, se genera un histórico de los datos del sistema.

Cuando finaliza la prueba, el nodo maestro recopila todas las medidas de los nodos esclavos realizando operaciones adicionales para obtener las estadísticas del experimento relativas al rendimiento. Finalmente, estas estadísticas son presentadas al usuario a través de la GUI y almacenadas en un fichero de texto formateado para ser cargado en el entorno de desarrollo *RStudio* [8].

### 3.2. Métodos de Generación de Datos

En cada prueba, la aplicación desarrollada ofrece al usuario la posibilidad de elegir entre dos métodos para la generación de tráfico NGSI.

La implementación del primer método (en lo sucesivo método bloqueante) es soportada por el paquete *java.net*, donde su funcionalidad está basada en la clase *HTTPURLConnection*. Esta clase solo permite conexiones bloqueantes. Así, en cada simulación se abren tantas conexiones persistentes HTTP como sensores simulados.

Respecto a la implementación del segundo método (método no bloqueante), se centra en los mecanismos que permiten generar tráfico concurrente. En este caso, se ha utilizado la clase *SocketChannel* de la librería *java.nio* para crear conexiones no

bloqueantes. Por este motivo, en cada simulación se abren tantas conexiones como datos generados.

Otro detalle de implementación a tener en cuenta es que cada cliente HTTP (bloqueante o no) simula el comportamiento de un sensor virtual en un hilo independiente.

### 3.3. Entorno de Experimentación

Todos los nodos (maestro y esclavos) tienen las mismas características *hardware*: HP *Compaq Business Desktop* dc7700p (procesador Intel *Core 2 Duo* E6600/2.4 GHz Dual-Core, memoria caché de 4MB por núcleo, memoria caché L2 de 4 MB y memoria RAM de 2GB).

Respecto a las características *software*, en cada nodo esclavo se ha instalado el sistema operativo *Windows 7 Home Premium*, un cliente *openVPN* y el entorno de desarrollo Eclipse Luna SR2 (4.4.2). Por otro lado, en el nodo maestro se ha instalado la versión 6.3 del sistema operativo *CentOS*, un servidor *openVPN* y la versión 0.11.0-1 del inyector *Cygnus*.

Las prestaciones de la VM desplegada en FILAB son: 1 VCPU, 2GB de RAM y 20 GB de disco. Esta VM hospeda una instancia del *Context Broker* 0.26.1 y una base de datos *mongoDB* 2.6.9. Por último, se ha creado una cuenta en la instancia global del *Big Data Analysis* de 5 GB de almacenamiento.

### 3.4. Plan de Pruebas

Atendiendo a la carga de trabajo o *workload* enviada al *Context Broker* en cada solicitud, hemos realizado 4 tipos de pruebas (1KB, 10KB, 100KB y 1MB). Cada una de estas pruebas se divide en 2 categorías en función del método de generación de datos seleccionado (apartado 3.2), siendo necesario lanzar 20 tests por categoría en diferentes días y franjas horarias. Por estos motivos, el plan de pruebas ha incluido más de 200 tests. Así, el siguiente apartado muestra un resumen de los resultados obtenidos.

## 4. Resultados

La Figura 3 muestra la representación gráfica del parámetro de *throughput* tras lanzar los tests correspondientes a la primera categoría de la prueba que mide el rendimiento de la plataforma FI-WARE cuando el *workload* es de 1 KB y la tasa mínima de envío de datos es de 20 ms.

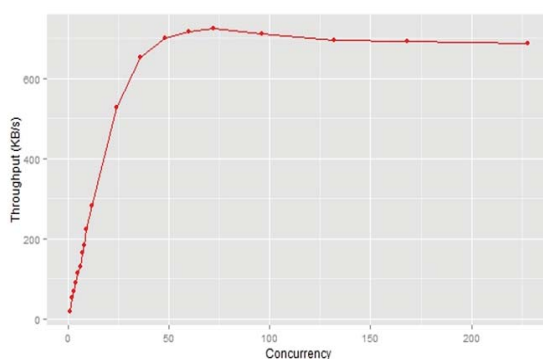


Fig. 3. Throughput. Prueba 1KB,  $T_{min}=20$  ms, categoría 1.

Al finalizar esta prueba, El *throughput* máximo obtenido es de 724,2304 KB/s o de 724,2304 solicitudes/s debido a que el *workload* es de 1KB. Para este valor, el *Round Trip Time* es de 98,8077 ms. Estas medidas son el resultado de lanzar el test que simula 6 sensores virtuales en los 12 nodos esclavos (conurrencia de 72).

En estas pruebas, se ha utilizado un método bloqueante para la generación y envío de datos. En dicho método cada cliente envía una nueva solicitud a FI-WARE solo después de que la anterior haya sido atendida. Como además, el banco de pruebas sólo soporta un número limitado de clientes, los clientes siempre están en sintonía con el servidor. Es decir, la tasa de generación de solicitudes nunca excede la capacidad del servidor Web del *Context Broker*.

Por otro lado, la Figura 4 muestra la representación gráfica del parámetro de *throughput* tras lanzar los tests correspondientes a la segunda categoría de la prueba que mide la capacidad de la plataforma FI-WARE cuando el *workload* es de 1 KB y la tasa de envío es de 50 ms.

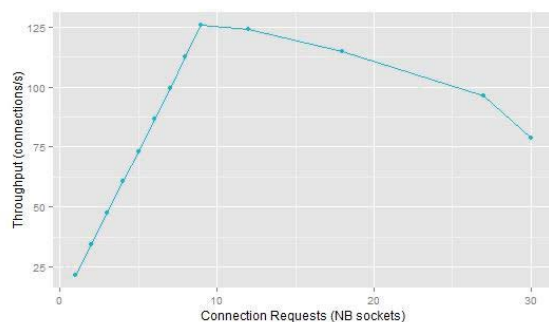


Fig. 4. Prueba 1KB, T=50 ms, categoría 2.

En este caso, el *throughput* máximo obtenido es de 125,9185 conexiones/s ya que se emplea el método no bloqueante. Este valor es alcanzado cuando se simula un sensor virtual en 9 nodos esclavos. Así, cuando se incrementa este número de clientes, el servidor del *Context Broker* entra en sobrecarga. Estos resultados se producen gracias a la implementación de sockets no bloqueantes lo que permite establecer una tasa de generación de datos constante.

## 5. Conclusiones

Los resultados obtenidos en este trabajo ponen de manifiesto que la plataforma FI-WARE satisface los requisitos de las aplicaciones agrícolas actuales, que conectan un número reducido de dispositivos, incluso bajas tasas de generación de datos (datos cada 10 minutos) y pocas demandas de ancho de banda. Sin embargo, es previsible que las aplicaciones futuras sean más complejas, impliquen mayor tráfico de datos y estén distribuidas a lo largo de un gran número de redes conectando *gateways* y nodos IoT. Para implementar estas aplicaciones, es necesario el despliegue de nuevos componentes que permitan reducir el tráfico de red e introduzcan mecanismos

avanzados de composición y descubrimiento. Estos *Generic Enablers* estarán disponibles al final de la versión 5 de FI-WARE.

## Referencias

- [1] López, J.A., Soto, F., Sánchez, P., Iborra, A., Longares, J., Vera, J.A., (2009). "Development of a Sensor Node for Precision Horticulture". *Sensors*, pp: 3240-3255, vol. 9.
- [2] López, J.A., Soto, F., Suardíaz, J., Iborra, A., Longares, J., Vera, J.A., (2009). "Wireless Sensor Networks for Precision Horticulture in Southern Spain". *Computer and Electronics in Agriculture*, pp: 25-35, vol. 68 (1).
- [3] López, J.A., García, A., Soto, F., García, F., (2011). "Design and Validation of a Wireless Sensor Network Architecture for Precision Agriculture Applications". *Precision Agriculture*, pp: 280-295, vol. 12(2).
- [4] Proyecto FI-WARE: <https://www.fiware.org/>
- [5] Protocolo OMA NGSI: <http://technical.openmobilealliance.org/Technical/technical-information/release-program/current-releases/ngsi-v1-0>
- [6] Eugster, P., Felber, P., Guerraoui, R., Kermarrec, A., (2003), "The many faces of Publish/Subscribe". *ACM Computing Surveys*, pp: 114-131, vol. 35 (2).
- [7] API REST HDFS: <https://hadoop.apache.org/docs/r1.0.4/webhdfs.html>
- [8] Software RStudio: <https://www.rstudio.com>