

Diseño y desarrollo de una arquitectura IoT en contexto con la plataforma FIWARE

Alicia Martínez¹, Haidée Onofre¹, Hugo Estrada², Daniel Torres¹,
Olivia Maquinay¹

¹Tecnológico Nacional de México/CENIDET, Cuernavaca, Morelos, México

²Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación
{amartinez, olivia}@cenidet.edu.mx, {haidee.onofreramirez,
torresestradadaniel}@gmail.com, hugo.estrada@infotec.mx

Resumen. En los últimos años, el surgimiento del concepto de Internet de las cosas ha impulsado la creación y el desarrollo de plataformas que permiten procesar grandes volúmenes de información de contexto, producida por redes de sensores que monitorean espacios o ambientes inteligentes. Estas plataformas procesan la información de contexto que producen las aplicaciones inteligentes a través de diferentes dispositivos y sensores conectados a Internet. FIWARE es una plataforma de código abierto que impulsa la creación de estándares para el desarrollo de aplicaciones y servicios inteligentes de diferentes dominios, en la nube. En este artículo se presenta el diseño y desarrollo de una arquitectura de Internet de las cosas, que hace uso de los estándares, componentes genéricos y APIs que ofrece la plataforma FIWARE. El artículo presenta también, la implementación de la arquitectura propuesta en una aplicación para el monitoreo y sensado de la conducción de automóviles en estacionamientos de instituciones y empresas.

Palabras clave: FIWARE, internet de las cosas, arquitecturas de internet de las cosas.

Design and Development of an Internet of Things Architecture in Context with the FIWARE Platform

Abstract. In the last few years, the emergence of Internet of Thing concept has boosted the creation and development of platforms that allows to process bigger volumes of context information produced by sensor networks monitoring smart environments. These platforms processing the context information that the smart applications produce through of different devices and sensors connected to Internet. FIWARE is an open source platform that impulses the standards creation for the development of smart applications and services of different domains, in the cloud. This paper presents the design and development of an Internet of Things Architecture, making use of the standards, generic enablers and APIs that offers the FIWARE platform. This paper also presents, the implementation of the architecture proposed, with an application for the monitoring of cars driving in the parking lots of institutions and companies.

Keywords: FIWARE, internet of things, internet of things architectures.

1. Introducción

Hoy en día, el Internet de las Cosas (IoT) y la Inteligencia Artificial (AI) se encontrarán entre las tecnologías transformadoras más importantes a lo largo de este año 2018 [1]. Estas tecnologías están aportando un gran valor a las empresas y las personas que utilizan estos nuevos servicios.

Actualmente, la automatización de procesos mediante técnicas de inteligencia artificial es una prioridad en las empresas que desean mejorar sus niveles de eficiencia. El IoT, por otro lado, ha sido un tema recurrente en los últimos años y su función es llevar el análisis en tiempo real a los procesos automatizados. El IoT genera un flujo de trabajo sofisticado para personalizar las experiencias que las empresas ofrecen a sus clientes y usuarios [2], utilizando sistemas que conectan el mundo físico con el mundo digital. En este sentido, IoT se ha definido como “un concepto que propone la conexión de los objetos cotidianos y de uso común a Internet” [3].

Mediante el Internet de las cosas, los objetos se reconocen a sí mismos y obtienen un comportamiento de inteligencia, al tomar decisiones relacionadas sobre la información que pueden comunicar sobre sí mismos [4]. De esta manera, los objetos inteligentes que intervienen dentro de una arquitectura IoT pueden producir información de contexto en grandes cantidades, es por esto que, actualmente, se requiere del uso de plataformas especializadas que cuenten con estándares y protocolos para el manejo y procesamiento de información de IoT a gran escala, con el objetivo de crear y desplegar aplicaciones de IoT inteligentes y administrables.

La plataforma FIWARE, desarrollada como resultado de una asociación pública-privada en Europa, permite desarrollar aplicaciones de software relacionadas con Internet del Futuro. Esta plataforma ofrece componentes para realizar análisis de grandes volúmenes de datos y la provisión de métricas en tiempo real, manipulación de información de contexto, análisis de eventos en tiempo real, recopilación de información de sensores y acción sobre actuadores. Para lograr esto, FIWARE presenta un conjunto de APIs genéricas estandarizadas que, facilitan la comunicación con Internet de las cosas y manipulan la información de contexto. Estos componentes pretenden impulsar la creación de estándares necesarios para desarrollar aplicaciones inteligentes en diferentes dominios.

En el presente artículo se presenta el diseño y desarrollo de una arquitectura para IoT que utiliza los estándares, componentes genéricos y APIs de FIWARE, y se combina con capas de servicio y de sensado para considerar un escenario completo de Internet de las cosas. Esta arquitectura fue utilizada para la implementación de una aplicación inteligente que monitorea la conducción de un usuario a través de su teléfono móvil. La aplicación realiza la detección de eventos de conducción de manera automática, los cuales pueden ser: conducción a una velocidad no autorizada, conducción en sentido contrario y paradas repentinas de un vehículo. Además, la aplicación envía alertas a los usuarios que se encuentran cerca del usuario que emitió la alerta y al guardia de seguridad de la empresa.

El artículo está organizado de la siguiente manera: la Sección 2 muestra los fundamentos teóricos utilizados en nuestra propuesta. La Sección 3 detalla el diseño y desarrollo de nuestra arquitectura de IoT utilizando la plataforma FIWARE. La Sección 4 muestra la implementación de la arquitectura propuesta y la aplicación IoT

generada a partir de esta arquitectura. Finalmente, la Sección 5 presenta las conclusiones y trabajos futuros.

2. Fundamentos teóricos y trabajos relacionados

2.1. Plataformas IoT

Las plataformas IoT pueden definirse como: “*servicios diseñados para facilitar la sociedad de la información, a través de la interconexión (física y digital) de cosas basadas en las tecnologías de la información, sobre un modelo de cómputo en la nube*” [5]. Esta definición señala el vínculo entre IoT y el concepto de cómputo en la nube; como plataformas ofrecidas por un proveedor de servicios, para el despliegue de aplicaciones de software que manejan a gran escala información de contexto proveniente de “*dispositivos o cosas*”.

Actualmente, el mercado de las plataformas IoT está en auge y en continua expansión. Hay encuestas que hacen mención de que más del 80% de las empresas cree que el campo del Internet de las Cosas es el más interesante para sus negocios. Algunas plataformas IoT son las Plataformas de conectividad/M2M, que se centran principalmente en establecer las condiciones que permitan a un dispositivo el intercambio de información de manera bidireccional con otro dispositivo, o con una aplicación de negocios, usando una red de telecomunicaciones [6].

Existen otras plataformas de software específicas de hardware. Una de estas plataformas es: PAVENET [7], plataforma para redes de sensores a partir del diseño de un nodo inalámbrico basado en Dual-CPU, denominado U3, como alternativa a los dispositivos disponibles, ya que proponen las dificultades de utilizar nodos basados en un solo núcleo de procesamiento para el desarrollo de aplicaciones que necesiten grandes recursos de procesamiento disponibles. Otro trabajo es LiteOS [8], plataforma que ha sido especialmente creada para programadores que deseen desarrollar aplicaciones para redes de sensores inalámbricas sin tener mayores conocimientos de la plataforma hardware, ya que se basa en la filosofía de trabajo UNIX, así como programación enfocada a hilos de ejecución, y programación orientada a eventos.

Otra plataforma es FIWARE, la cual proporciona capacidades en la nube basadas en OpenStack [9], y mejoradas junto con un conjunto de herramientas y librerías de valor añadido denominadas Generic Enablers (GEs). Estos GEs, ofrecen interfaces (APIs) estándar y abiertas que facilitan tareas como la integración de dispositivos de Internet de las Cosas (Internet of Things, IoT), analizar y procesar datos a media y gran escala (Big Data), o incorporar interfaces avanzadas para interactuar con los usuarios.

2.2. Ecosistema FIWARE

FIWARE es una plataforma de código abierto que impulsa la creación de estándares para el desarrollo de aplicaciones y servicios inteligentes de Internet del Futuro en diferentes dominios [10], que proporciona un conjunto de componentes genéricos e interfaces de programación de aplicaciones (APIs, por sus siglas en

inglés: *Application Programming Interface*), con el propósito de facilitar el desarrollo y despliegue de aplicaciones inteligentes en múltiples sectores y escenarios [11].

Algunos de los componentes genéricos y APIs que conforman el ecosistema FIWARE se describen a continuación.

Orion Context Broker. El Orion Context Broker es un componente genérico de FIWARE que permite manipular información de contexto y habilitar la suscripción de aplicaciones a estos datos de contexto, otorgando la oportunidad de disponer de estos datos de contexto en todo momento [12]. La información de contexto puede provenir de fuentes diferentes como: sistemas ya existentes, usuarios (a través de aplicaciones móviles) y redes de sensores.

El Orion ContextBroker (Fig. 1) permite modelar y obtener acceso a la información de contexto independientemente de la fuente de donde proviene esa información. Este componente está habilitado para manejar información de contexto a gran escala mediante la implementación del estándar API REST [13]. La manipulación de información de contexto en el Orion ContextBroker de FIWARE se realiza siguiendo la especificación de la API estándar: FIWARE NGSI.



Fig. 1. Esquema del Orion ContextBroker de FIWARE.

API FIWARE NGSI. La API estándar que propone FIWARE para recopilar, gestionar, publicar e informar sobre cambios en la información de contexto se denomina FIWARE NGSI (*Next Generation Service Interface*, por sus siglas en inglés). Esta API está diseñada para administrar todo el ciclo de vida de la información de contexto, incluyendo actualizaciones, consultas, registros y suscripciones [14].

Los elementos principales en el modelo de datos NGSI son: entidades de contexto, atributos y metadatos, como se muestran en la Fig. 2.

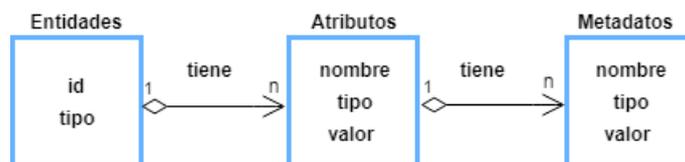


Fig. 2. Elementos del modelo de datos FIWARE NGSI.

Las entidades de contexto, o simplemente entidades, son el centro del modelo de datos NGSI de FIWARE. Una entidad representa una cosa. Por ejemplo, cualquier objeto físico o lógico (un sensor, una persona, una habitación, etc.). Los atributos de contexto son propiedades que describen las entidades de contexto y los metadatos son piezas de información que se encargan de describir características adicionales al atributo de contexto.

La API FIWARE NGSI es utilizada dentro del Orion ContextBroker para la manipulación de información de contexto de los modelos de datos NGSI. Actualmente la versión utilizada de esta API es FIWARE NGSI en su versión 2 (FIWARE NGSIv2).

API QuantumLeap. QuantumLeap es un API de FIWARE que soporta el almacenamiento de modelos de datos del estándar NGSI de FIWARE en una base de datos de series de tiempo [15]. Orion ContextBroker es el encargado de notificar a la API QuantumLeap los cambios en un modelo de datos, esto se realiza a través de la creación de una suscripción de un modelo de datos en el Orion ContextBroker, la cual tendrá como función notificar los cambios en el modelo de datos a la API QuantumLeap.

La API QuantumLeap recibe los modelos de datos en formato NGSI y los transporta al motor de base de datos correspondiente para su almacenamiento. Previo a esto, la API QuantumLeap realiza la conversión de cada atributo de los modelos de datos en formato NGSI, al que corresponda en el motor de base de datos, según el tipo de dato establecido para cada atributo en los modelos de datos.

3. Diseño y desarrollo de la arquitectura IoT

La arquitectura propuesta en este artículo tiene como objetivo utilizar los componentes de la plataforma FIWARE para diseñar e implementar aplicaciones inteligentes que interactúen con capas de servicio y de sensado de información de contexto. Con la finalidad de instanciar esta arquitectura en un caso específico, se diseñó e implementó una aplicación inteligente, denominada *DrivingApp*, que permite monitorear dentro de una institución o empresa el manejo de los vehículos que acceden a los estacionamientos de la empresa.

En caso de que un vehículo sea conducido de manera errónea, la aplicación enviará una alerta al guardia de seguridad, así como a los usuarios de la aplicación que se encuentren cerca de donde se generó la alerta.

Durante la etapa de desarrollo de la aplicación inteligente de IoT (*DrivingApp*), fueron identificados diversos inconvenientes en la manipulación del sensor GPS, dada la inexactitud que presenta en los valores de las coordenadas, lo cual depende directamente del tipo de sensor con el que cuenta el dispositivo, teniendo una eficacia aproximadamente del 70% en la mayoría de los casos; como respuesta a esta anomalía se adecuó una alternativa: que el dispositivo obtuviera su ubicación por medio de la red (wi-fi o de datos móviles), si es que el dispositivo estuviera conectado a una, en caso contrario esta se obtendría del GPS.

El comportamiento inteligente de las aplicaciones y servicios que integran la arquitectura de IoT propuesta, radica en la detección de eventos de conducción de manera automática, los cuales pueden ser: conducción a una velocidad no autorizada,

conducción en sentido contrario y paradas repentinas de un vehículo; además de la búsqueda y elección de los dispositivos móviles a los que se envía la alerta, según la configuración de envío de alertas establecida en la aplicación móvil por medio del usuario. La arquitectura desarrollada (ver Fig. 3) está compuesta por tres módulos: capa de aplicación y sensado, capa de servicios y el ecosistema FIWARE.

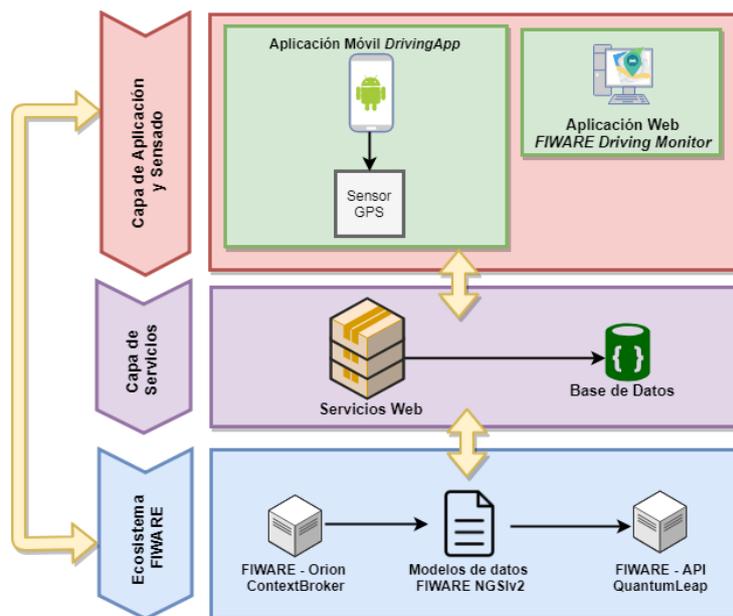


Fig. 3. Arquitectura IoT en contexto con el ecosistema FIWARE.

3.1. Capa de aplicación y sensado

La capa de aplicación y sensado es la encargada de recolectar y dar tratamiento a la información de contexto que se obtiene a través la aplicación móvil (*Driving App*) y la aplicación web (*FIWARE Driving Monitor*). La aplicación móvil (*DrivingApp*) procesa y manipula los datos recolectados del teléfono móvil, de las lecturas realizadas del sensor GPS. Una vez que la información recolectada es procesada, se detecta si sucede algún evento que pueda generar alguna alerta, en caso de que esto ocurra, la alerta es presentada en la interfaz gráfica del dispositivo móvil, así como también en la aplicación web.

La aplicación móvil consume los servicios web de la capa de servicios para extraer y modificar información de la base de datos. Además, la aplicación móvil establece comunicación directa con el Orion ContextBroker para el envío de modelos de datos FIWARE NGSI. El Orion ContextBroker almacena estos modelos de datos y los replica, ya sea a la API QuantumLeap o a la capa de servicios, según sea el caso.

La aplicación web de monitoreo (*FIWARE Driving Monitor*) es la encargada de mostrar la información de contexto recopilada por la aplicación móvil de IoT. Esta información de contexto generada por la aplicación móvil de IoT, la obtiene a través

de la capa de servicios, y la clasifica y representa a través diferentes elementos gráficos representativos.

3.2. Capa de servicios

La capa de servicios es un módulo que tiene como funcionalidad principal ser intermediario de comunicación entre el ecosistema FIWARE y la capa de aplicación.

La información de contexto generada por la aplicación móvil IoT es enviada a la capa de servicios, donde es manipulada a través de los módulos de servicios web, para su almacenamiento en la base de datos y en el Orion ContextBroker de FIWARE.

Los servicios web de la capa de servicios están divididos en 3 módulos, estos módulos proveen las siguientes funcionalidades:

- *API RESTful*: es un servicio que actúa como interfaz para el intercambio de información (inserción, actualización, eliminación y consulta de datos) entre las aplicaciones y la base de datos.
- *Servidor de Notificaciones*: es un servicio que tiene como objetivo replicar notificaciones de alertas emitidas por los usuarios de la aplicación móvil. Cada alerta emitida en la aplicación móvil es manipulada en este servicio, para ser replicada como notificación a los dispositivos móviles que hagan uso de la aplicación móvil DrivingApp. Adicional a la notificación de la alerta a los dispositivos móviles, el servidor de notificaciones se encarga de enviar las notificaciones de estas alertas a la aplicación web.
- Estas funcionalidades son posibles mediante a una conexión directa del servidor de notificaciones con el Orion ContextBroker de FIWARE, a través de la suscripción a la información de contexto del modelo de datos de alerta FIWARE NGSIv2.
- *Servicio de consultas de contexto*: es un servicio que tiene la capacidad de responder a las consultas de información de contexto específicas provenientes de la capa de aplicación. Por ejemplo, las solicitadas por la aplicación móvil o web.

3.3. Ecosistema FIWARE

El ecosistema FIWARE es la plataforma en la nube dentro de la arquitectura IoT propuesta en este trabajo. Este ecosistema se encarga de manipular y persistir la información de contexto proveniente de las aplicaciones, a través de los componentes que integran la plataforma. Se utilizan tres componentes claves del ecosistema FIWARE en este proyecto:

- *Orion ContextBroker*: este componente de FIWARE se utiliza para capturar los datos de contexto producidos por los sensores del teléfono inteligente, y representa estos datos en modelos de datos definidos que siguen el estándar FIWARE NGSIv2.
- *Modelos de datos FIWARE NGSIv2*: los modelos de datos de FIWARE utilizados son: *Alert*, *Device*, *DeviceModel*, *Road*, *RoadSegment*

OffStreetParking y *Building* categoría tipo “Zone” [16]. Estos modelos de datos sirven para estandarizar la información de contexto referente a: la detección de eventos de alerta emitidos a través de la aplicación móvil, las propiedades de los dispositivos móviles, los valores obtenidos por el sensor GPS del teléfono inteligente y, la información de las calles y estacionamientos que se encuentran dentro de la zona comprendida por la empresa o institución.

- *API QuantumLeap*: este componente se utiliza para almacenar los datos obtenidos del sensor GPS como series de tiempo, de esta forma será posible realizar visualizaciones de los datos a lo largo de cierto periodo de tiempo.

El aporte principal de este trabajo en comparación con otras arquitecturas y plataformas de IoT, es el diseño y desarrollo de la arquitectura descrita, en la que se hace uso del ecosistema FIWARE como plataforma de IoT, el cual interactúa con capas de servicio y de sensado de datos de contexto. Además, esta arquitectura no sólo se ha definido en lo abstracto, sino que ha sido implementada en una aplicación inteligente de IoT, lo cual ha permitido verificar el funcionamiento y operación de la arquitectura propuesta.

Al utilizar FIWARE como plataforma de IoT se facilita la conexión de dispositivos y almacenamiento de datos, integrando los protocolos de dispositivos y métodos de conectividad, para dar interpretación a la información relevante; asimismo permite aislar el procesamiento de datos, de las capas de servicio y de aplicación y sensado de la arquitectura IoT desarrollada.

4. Implementación de la arquitectura IoT propuesta

La implementación de la arquitectura propuesta se llevó a cabo a través del desarrollo de dos aplicaciones: *DrivingApp* y *FIWARE Driving Monitor*. Estas aplicaciones desempeñan sus funcionalidades en la capa de aplicación y sensado de la arquitectura IoT, y se comunican con la capa de servicios y con el ecosistema FIWARE respectivamente. Actualmente las aplicaciones se encuentran en la etapa final del desarrollo para posteriormente iniciar con la etapa de pruebas de software.

La implementación de la arquitectura IoT propuesta tiene como ventajas: la integración de servicios web que pueden adaptarse para ser consumidos por aplicaciones inteligentes de IoT con propósitos similares a la que se desarrolla en el presente trabajo; facilita el despliegue e implementación de una aplicación de monitoreo de conducción de automóviles que puede ser utilizada por empresas o instituciones que requieran de estos servicios, y que, además, busquen un entorno de desarrollo que puedan operar con un mínimo esfuerzo de administración, configuración y mantenimiento. La utilización de la plataforma FIWARE permitió el desarrollo de un sistema modular basado en componentes reusables, lo cual acelera el tiempo de construcción de sistemas. Si todos los componentes del sistema presentado en este artículo hubiesen sido desarrollados desde cero, se habría invertido una gran cantidad de tiempo únicamente para desarrollar el componente de manejo de datos de contexto, el cual ya es proporcionado por la plataforma FIWARE. Estas

características de modularidad también se presentan en la aplicación generada en este trabajo, lo cual facilita su reutilización.

En los resultados derivados de la experimentación e implementación de la arquitectura IoT (a través de las aplicaciones desarrolladas), se logró demostrar que el uso de la plataforma FIWARE facilita la conexión de dispositivos y manejo información de contexto; sin embargo la integración de FIWARE como la plataforma IoT dentro la arquitectura propuesta, fue uno de los principales retos en la etapa de diseño, ya que esto requirió de buscar la forma óptima de comunicación de los componentes de la plataforma FIWARE y el resto de las capas de arquitectura del sistema propuesto. Es importante destacar que la arquitectura es resultado de un uso novedoso de los componentes FIWARE unidos a capas de servicio y de sensado.

4.1. Aplicación móvil de IoT

DrivingApp es una aplicación móvil de IoT para dispositivos móviles Android. La Fig. 4 muestra una vista de la aplicación. La aplicación móvil tiene como objetivo detectar de manera automática la errónea conducción de un automóvil.

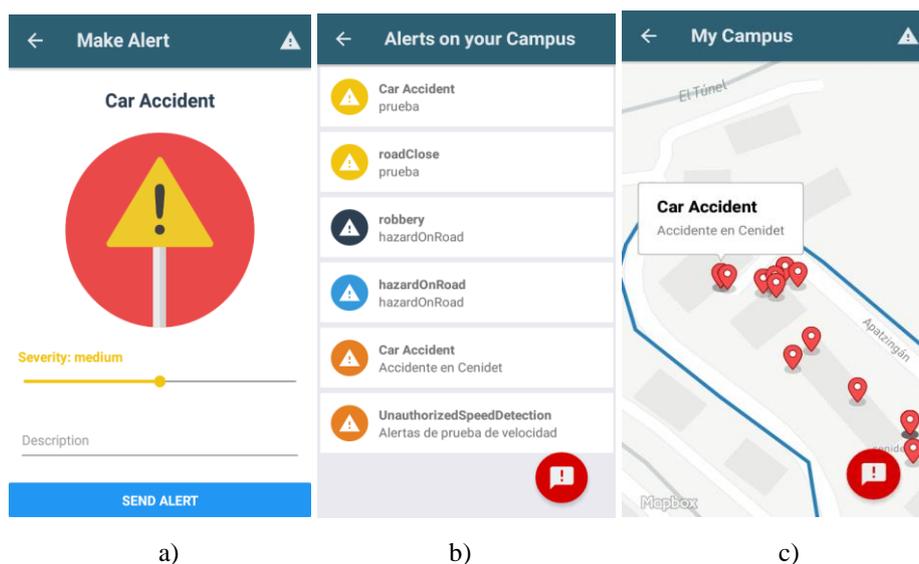


Fig. 4. Aplicación Móvil de IoT: *DrivingApp* a) Generación manual de alerta de accidente automovilístico, b) Listado de las alertas suscitadas más recientes, c) Mapa de la ubicación de las alertas suscitadas dentro de la institución.

La aplicación es configurada para monitorear a los usuarios de la aplicación dentro de los estacionamientos de una empresa o institución. Los eventos que detecta la aplicación móvil son: conducción a una velocidad no autorizada, conducción en sentido contrario y paradas repentinas de un vehículo.

El sensor utilizado para la detección automática de los eventos es el GPS, y además, se hace uso de los modelos de datos de FIWARE NGSIv2 mencionados con anterioridad dentro del ecosistema FIWARE, con el objetivo de estandarizar la

información recolectada. El teléfono se encuentra en constante recolección de la información del contexto; es decir, cada segundo se obtiene la posición GPS del teléfono y la velocidad a la que se encuentra. Esta información es procesada en el teléfono para determinar si sucede algún evento que pueda generar alguna alerta.

Cuando un evento de conducción errónea es detectado, se genera una alerta y se envía a todos los usuarios de la aplicación que se encuentren cerca del lugar en donde se generó dicha alerta. Además, también se envía la alerta a la aplicación web, que será utilizada por el guardia de seguridad de la empresa para notificarle que un usuario se encuentra conduciendo de manera errónea.

El sistema permite también generar alertas de manera manual, con el objetivo de alertar a los usuarios de eventos relacionados a tráfico, accidentes y/o seguridad, lo cuales sean observados en un área o lugar específico dentro de las instalaciones de la institución.

4.2. Aplicación web de monitoreo

FIWARE Driving Monitor es una aplicación web que tiene como objetivo gestionar la información de los usuarios de la aplicación móvil *DrivingApp* y la información de contexto generada por dicha aplicación. La Fig. 5 muestra una vista de la aplicación en la cual se muestra delimitado el mapa de la ubicación de la empresa.

Además, se muestran en el mapa las alertas que se han generado en diferentes puntos de la empresa. La aplicación provee las herramientas gráficas necesarias para realizar la delimitación geográfica en mapas, de las sucursales o campus de la empresa y de las zonas que comprenden los estacionamientos de la empresa.

El usuario de esta aplicación puede ser el guardia de seguridad de la empresa. Por lo que, este usuario podrá consultar el control estadístico de la información de contexto referente a alertas de eventos ocurridos dentro de los campus y zonas de la institución.

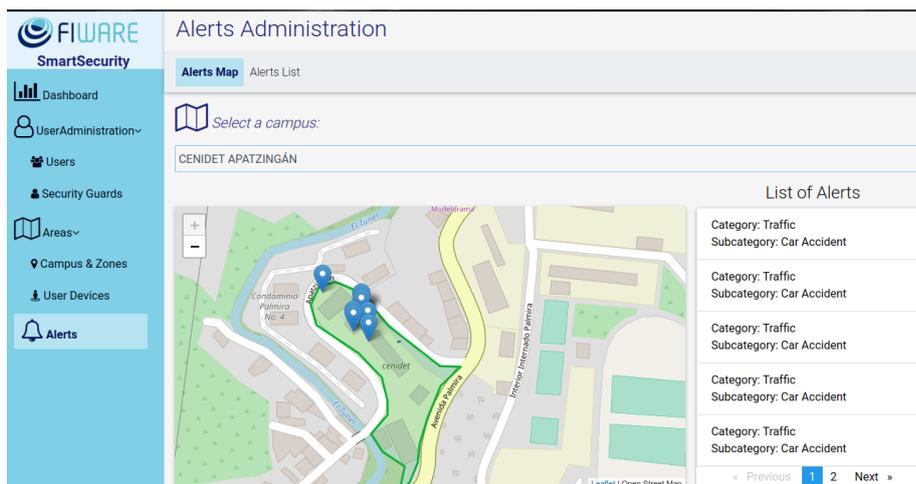


Fig. 5. Aplicación Web de Monitoreo: *FIWARE Driving Monitor*.

La información antes mencionada se muestra a través de gráficas y tablas en donde se visualizan los tipos de eventos ocurridos y el número de incidencias de cada uno. Esta aplicación web es la encargada de administrar el registro de información de contexto de los usuarios, como su localización. Esto con el fin de determinar si algún usuario se encuentra actualmente dentro de la empresa o bien, si estuvo en determinado momento dentro de esta, y esta información del usuario se muestra a través de tablas y marcadores en el mapa.

5. Conclusiones

La arquitectura de IoT propuesta en este artículo permite demostrar el uso de FIWARE como una plataforma que permite el desarrollo de aplicaciones inteligentes para diferentes dominios de aplicación.

La arquitectura desarrollada permite el despliegue de componentes y módulos de cada capa de forma independiente; esto demuestra el bajo acoplamiento de los módulos entre cada de capa y ofrece la posibilidad de implementar las funcionalidades de una capa sin necesidad de desplegar las demás capas.

La implementación de la arquitectura propuesta utiliza las capacidades de la plataforma FIWARE, para desarrollar una aplicación móvil que permite sensar la conducción de automóviles en estacionamientos de instituciones o empresas, a fin de detectar eventos de alerta durante la conducción del vehículo.

Es importante destacar que la arquitectura propuesta es novedosa porque permite unir componentes de propósito general de la plataforma FIWARE, con capas diseñadas con servicios web y con capas para el sensado de información de contexto. Una de las consecuencias directas del uso de los componentes FIWARE es la modularidad del sistema, lo cual facilita su adaptación y reuso, tanto a nivel de arquitectura para diversas aplicaciones IoT, como de la implementación específica que se realizó para validar la arquitectura, la cual puede ser reusada en entornos donde se requiera monitorear la actividad de los usuarios dentro de los estacionamientos y de las instalaciones de la empresa en general.

Como trabajos futuros, podría explorarse el diseño de una arquitectura de IoT configurable, de manera que esta pueda capturar información de diferentes tipos de sensores o dispositivos sin necesidad de adaptaciones al código. Por otra parte, sería útil analizar las posibilidades de desplegar otros componentes genéricos y servicios de la plataforma FIWARE, para su integración dentro de la arquitectura de IoT propuesta, con el propósito de enriquecer y ampliar las funcionalidades de la arquitectura actual; de manera que se pueda extender el uso de la arquitectura, implementándose en diversos escenarios de soluciones IoT con enfoques o dominios de aplicación distintos, y aplicables para un mismo espacio inteligente.

Agradecimientos. Esta investigación ha sido financiada por la Comisión Europea y CONACYT a través del proyecto SmartSDK.

Referencias

1. Portaltic Europa Press, <http://www.europapress.es/portaltic/sector/noticia-internet-cosas-inteligencia-artificial-pilares-transformacion-digital-2018-fujitsu-20180201124739.html>, (2018)

2. Diario del profesional Ti, <https://diarioti.com/cuando-la-inteligencia-artificial-el-internet-de-las-cosas-y-el-blockchain-convergen/105992> (2018)
3. Gubbi, J., Buyya, R., Marusic, S., Palaniswamia, M.: Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. *Future Generation Computer Systems Journal* 29(7), pp. 1645–1660 (2013)
4. Aldein Mohammed, Z., Ali Ahmed, E.: Internet of Things Applications, Challenges and Related Future Technologies. *World Scientific News Journal* 67(2), pp. 126–148 (2017)
5. Pratim Ray, P.: A survey of IoT cloud platforms. *Future Computing and Informatics Journal* 1(1-2), pp. 35–46 (2016)
6. Boswarthick, D., Elloumi, O., Hersent, O.: M2M Communications: A Systems Approach. John Wiley & Sons, Editor. West Sussex, United Kingdom (2012)
7. Saruwatari, S., Kashima, T., Minami, T., Morikawa, H., Aoyama, T.: PAVENET: A Hardware and Software Framework for Wireless Sensor Networks. *Trans. of the Society of Instrument and Control Engineers Vol. E-S-1, no.1*, pp. 74–84 (2005)
8. Cao, Q., Abdelzaher, T., Stankovic, J., He, T.: The LiteOS Operating System: Towards Unix Like Abstraction for Wireless Sensor Networks. In: *Proceedings of the 7th International Conference on Information Processing in Sensor Networks*, pp. 233. St. Louis, MO, USA (2008)
9. Kumar, R. *et al.*: Open Source Solution for Cloud Computing Platform Using OpenStack. *International Journal of Computer Science and Mobile Computing* 3(5), pp. 89 – 98 (2014)
10. Telefónica. FIWARE, el estándar que necesita el IoT (2018)
11. FIWARE: European Commission. About Us (2018)
12. FIWARE - ORION Documentation. <http://fiware-orion.readthedocs.io/en/stable/> (2018)
13. FIWARE - IOT Stack. Context Broker (Orion). http://fiware-iot-stack.readthedocs.io/en/latest/context_broker/#development-of-context-aware-applications (2018)
14. Telefónica I+D - FIWARE-NGSI v2 Specification, <http://fiware.github.io/specifications/ngsiv2/stable/>, último acceso 2018/03/22
15. QuantumLeap – Home, <https://smartsdk.github.io/ngsi-timeseries-api/>, último acceso 2018/03/30
16. FIWARE Data Models repository. <https://github.com/Fiware/dataModels>, último acceso 2018/05/16