Un enfoque para la integración de dispositivos IoT en el desarrollo de SIG en la nube

Daniel Rodríguez-López*, *Juan Rodríguez-López*, *Miguel Zúñiga-Prieto*, *Lizandro Solano-Quinde

Departamento de Ciencias de la Computación, Universidad de Cuenca, Av. 12 de Abril y Av. Loja, Cuenca, Ecuador, 01.01.168.

Autores para correspondencia: {daniel.rodriguez, juan.rodriguezl, miguel.zunigap}@ucuenca.edu.ec

Fecha de recepción: 30 de julio de 2017 - Fecha de aceptación: 15 de agosto de 2017

ABSTRACT

Cloud computing and Internet of Things (IoT) as technological support to the construction of Geographic Information Systems (GIS) is changing the way these systems are developed and employed. The technological infrastructure provided by cloud providers and used as deployment platform enables GIS to take advantage of its storage and processing capabilities, whereas using IoT devices automate the data collection process. However, current implementations of GIS, whose services are used on cloud environments and interact with IoT devices, are realized in an ad hoc manner, without providing solutions that ease their design and implementation. This work proposes a development approach, which from models that describe at a high abstraction level both the system architecture as well as the interaction among its services and IoT devices, guides the implementation and deployment on cloud environments activities. The feasibility of this approach has been illustrated by the design and implementation of a GIS application that analyzes spatial data collected by air quality sensors, with services deployed in the Google Cloud platform.

Keywords: GIS, cloud computing, SoaML, sensor, IoT.

RESUMEN

Cloud computing e Internet de las Cosas (IoT) como soporte tecnológico a la construcción de Sistemas de Información Geográfica (SIG) están cambiando la manera en la que estos sistemas son desarrollados y desplegados. La infraestructura tecnológica provista por proveedores cloud, y usada como plataforma de despliegue, permite a los SIG aprovechar sus altas capacidades de almacenamiento y procesamiento; mientras que el uso de dispositivos IoT permiten la automatización del proceso de recolección de datos. Sin embargo, las implementaciones actuales son realizadas de manera ad hoc, sin que se provean soluciones que faciliten su diseño, implementación y reutilización. En este trabajo proponemos un enfoque de desarrollo que, a partir de modelos que describen a un alto nivel de abstracción, tanto la arquitectura del sistema como la interacción entre sus servicios y dispositivos IoT, permite guiar las actividades de implementación y despliegue en entornos cloud. La factibilidad de esta propuesta ha sido ilustrada con el diseño e implementación de una aplicación SIG que brinda soporte al análisis de datos espaciales recolectados mediante sensores de calidad de aire y cuyos servicios fueron desplegados en la plataforma Google Cloud.

Palabras clave: SIG, computación en la nube, SoaML, sensores, IoT.

1. INTRODUCCIÓN

Una aplicación SIG (Sistema de Información Geográfica) está conformada por servicios que permiten recolectar, almacenar, analizar, administrar y visualizar información geográfica; pudiendo ser

desplegados en ambientes Web, adquiriendo así la capacidad de compartir información geográfica por medio de Internet (Yue, Zhou, Gong, & Hu, 2012). Los SIG requieren gran capacidad de almacenamiento y procesamiento, por lo que la tendencia actual es desplegarlos en plataformas provistas por proveedores cloud (Bhat & Ahmad, 2011). Estas plataformas no solo brindan alta capacidad de procesamiento y almacenamiento, sino que facilitan el uso dinámico de recursos (p. ej., hardware, software, redes, entorno de ejecución) de acuerdo con la demanda de la aplicación y el pago basado en métricas de consumo (pago-por-uso) (Liu, 2013). Por otra parte, los procesos de recolección de datos de los SIG siguen el paradigma IoT, cuyos principios facilitan su automatización mediante el uso sensores gestionados a través de Internet (Gubbi, Buyya, Marusic, & Palaniswami, 2013). El uso de cloud e IoT presenta desafíos relacionados con heterogeneidad tecnológica, desconocimiento de estándares, diseño arquitectónico e integración (Gubbi *et al*., 2013).

Existen propuestas que integran paradigmas cloud e IoT en el desarrollo de SIG; algunas sugieren arquitecturas de aplicaciones cuyos componentes son desplegados en la cloud (Yue *et al*., 2012; Bhat & Ahmad, 2011; Evangelidis, Ntouros, Makridis, & Papatheodorou, 2014), mientras otras hacen uso de sensores bajo el paradigma IoT para realizar la de recolección de datos (Gubbi *et al*., 2013; Bröring *et al*., 2011; Sagl, Lippautz, Resch, & Mittleboeck, 2011). Sin embargo, ninguna de estas propuestas soporta el diseño de la arquitectura e interacción entre los servicios del SIG, ni sugiere como abordar la implementación y despliegue. El estilo arquitectónico *Arquitectura Orientada a Servicios* (Service Oriented Architecture - SOA) brinda la capacidad de integrar servicios considerando la heterogeneidad de las tecnologías IoT, servicios Web de procesamiento y visualización de información (Chen, Guo & Bao, 2016), de información geográfica en este caso.

En este artículo proponemos un enfoque de desarrollo de aplicaciones SIG que, a partir de modelos arquitectónicos representados mediante SoaML (Service Oriented Architecture Modeling Language) (OMG, 2012) -un lenguaje especializado en la descripción de arquitecturas de servicios-, guía la implementación de artefactos de software que facilitan la integración e interacción de dispositivos IoT con servicios SIG desplegados en entornos cloud. La aplicabilidad de esta propuesta ha sido ilustrada con un ejemplo práctico de una aplicación SIG cuya arquitectura consistió en servicios desplegados en la plataforma Google Cloud y el proceso de recolección de datos fue soportado por sensores de calidad de aire bajo el paradigma IoT. El resto de este artículo está organizado en las siguientes secciones: la Sección 2 describe trabajos relacionados, la Sección 3 presenta el proceso de desarrollo sugerido, la Sección 4 presenta el ejemplo práctico que ilustra la propuesta, y finalmente la Sección 5 expone las conclusiones y trabajo futuro.

2. ESTADO DEL ARTE

Existen varios estudios que proponen la integración de tecnologías cloud y dispositivos IoT. (Ara, Gajkumar & Prabhakar, 2016) proponen un caso de estudio sobre la integración de sensores con servicios desplegados en plataformas cloud. Sin embargo, a pesar de que implementan un caso de estudio, no brindan detalles de dicha implementación ni proponen mecanismos que faciliten la integración entre servicios de la aplicación y dispositivos IoT. Por otro lado, Gubbi *et al*. (2013) proponen una arquitectura de aplicaciones que facilita la interacción entre dispositivos IoT y servicios de procesamiento y presentación de información desplegados en entornos cloud. A pesar de usar protocolos y estándares para la comunicación entre servicios, no brindan mecanismos que guíen las actividades de integración e interacción entre servicios de la aplicación SIG y dispositivos IoT.

En el caso de aplicaciones SIG, Sagl *et al.* (2011) proponen e implementan una arquitectura de aplicación cuyo proceso de adquisición de datos es realizado por sensores, mientras que los otros procesos son soportados por servicios Web especializados. Aunque proponen el uso de estándares del Open Geospatial Consortium (OGC), no proveen mecanismos para especificar la integración e interacción entre los servicios de la aplicación SIG y dispositivos IoT. Bröring *et al.* (2011) proponen una Infraestructura de Datos Espaciales (IDE) en donde mediciones tomadas por sensores son recolectadas para posteriormente ponerlas a disposición mediante servicios Web bajo el estándar SWE (Sensor Web Enablement) del OGC. Sin embargo, a pesar de que proponen una infraestructura

tecnológica que permite la implementación de SIGs que integran tecnologías cloud e IoT, esta propuesta no considera aspectos relacionados a la arquitectura de software.

Chen *et al*. (2016) proponen la integración de servicios Web y dispositivos IoT empleando el estilo arquitectónico SOA para diseñar la arquitectura de la aplicación. Proveen además un metamodelo que facilita el modelado de la arquitectura de la aplicación cuyos servicios serán desplegados en un ambiente cloud. Si bien no brindan soporte a la especificación e implementación de la interacción entre servicios, ni tampoco brindan soporte a la implementación de principios SOA. Zuñiga-Prieto, Gonzalez-Huerta, Abrahão, & Insfran (2009) y Zúñiga-Prieto, Insfran, & Abrahão (2016), proponen un proceso para la integración incremental de servicios web por medio de Lenguajes de Descripción de la Arquitectura orientadas a servicios. Este método tiene un enfoque genérico, es decir, puede ser implementado sobre servicios de cualquier naturaleza, por esta razón no se encuentra orientado a la integración de dispositivos IoT con servicio de una aplicación SIG.

A pesar de que algunos de los trabajos presentados en esta sección proponen arquitecturas que integran las tecnologías cloud e IoT en el diseño de SIG, estas no brindan mecanismos que soportan el diseño arquitectónico ni describen las actividades necesarias para su implementación y despliegue.

3. PROCESO PARA LA AUTOMATIZACIÓN DE LA INTEGRACIÓN DE SERVICIOS

En base al análisis realizado de los componentes involucrados en SIGs que combinan la tecnología IoT y cloud (Gubbi *et al.*, 2013; Sagl *et al*., 2011; Ara *et al*., 2016), se han identificado tres tipos de servicios (ver Fig. 1):

- Dispositivos IoT: Cumplen con las tareas de recolección de datos por sensores.
- Servicios de negocio: Tienen propósitos de integración de la información, procesamiento espacial o simplemente son clientes que consumen otros servicios. Incluye software desarrollado por terceros que puede ser desplegado en un ambiente cloud para ser integrado como parte de la arquitectura de la aplicación.
- Recursos cloud: Recursos tecnológicos ofrecidos como servicio por un proveedor cloud. Por ejemplo, servicios de base de datos espaciales. Pueden requerir la construcción de un servicio que funcione como intermediario para su acceso y gestión.

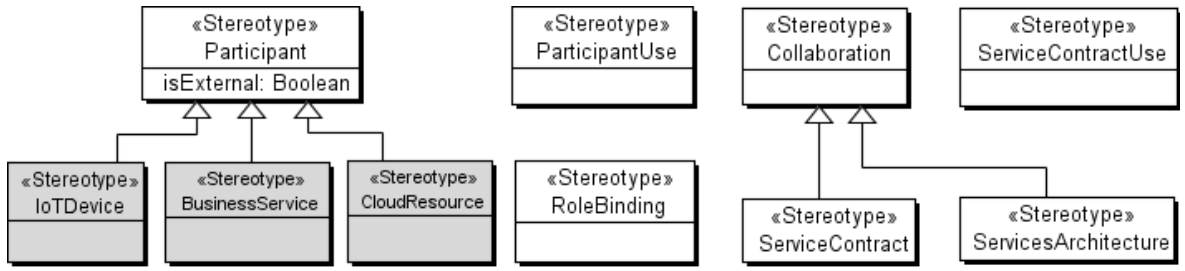

Figura 1. Perfil UML para la descripción de arquitecturas de servicios Cloud e IoT.

3.1. Especificación de la Arquitectura de la Aplicación

Para la creación de un SIG desplegado en la Web, en esta actividad, proponemos la aplicación de principios del estilo arquitectónico SOA (Chen *et al*., 2016). Estos principios permitirán implementar servicios débilmente acoplados; minimizando la dependencia entre ellos mediante la encapsulación de las particularidades de la implementación, requiriéndose únicamente una interfaz bien definida para la comunicación entre servicios. SoaML es un lenguaje especializado para describir arquitecturas basadas en servicios y lo utilizamos para especificar la arquitectura de un SIG. Sus conceptos relevantes son:

- *Participantes:* aquellos elementos arquitectónicos que consumen y/o proveen servicios.
- *Contrato de Servicio:* son acuerdos de definición de cómo se dará la comunicación entre los participantes (orquestación). Sus elementos internos son: Roles que los Participantes

involucrados deben cumplir, Interfaces que deben ser implementadas por el Participante que cumple un Rol, Mensajes y Tipos de dato, y el Protocolo de Interacción entre los Participantes.

- *Arquitectura de Servicios:* define la interacción entre cada uno de los elementos arquitectónicos de la aplicación, propone una ideología en la que los Participantes no pueden interactuar directamente entre sí, se requiere siempre como intermediario un Contrato de Servicios.
- *Interacciones:* permiten modelar el Flujo de Trabajo de la aplicación, cada Contrato de Servicios implica una Interacción.
- *Role Binding:* Describe el Rol que cumple un Participante en la Arquitectura de Servicios.
- En este trabajo, creamos un perfil UML que extiende el Perfil SoaML, de tal manera que permita modelar los tipos servicios que hemos identificado. Los perfiles son mecanismos provistos por UML que amplían su notación semántica (OMG, 2012).

La Figura 1 muestra como la notación de Participant ha sido extendida de tal manera que podemos clasificar los servicios que conforman una aplicación en: dispositivo IoT (IoTDevice), servicio de negocio (BusinessService) y recurso cloud (CloudResource). En esta actividad, los arquitectos de software producen el Modelo de la Arquitectura de la Aplicación; para ello, antes de realizar el diseño, identifican los estándares de comunicación aplicables o las interfaces provistas y requeridas por los servicios que conformarán el sistema. Por ejemplo, el estándar SOS de OGC para recolección de mediciones a través de sensores, o el estándar SensorML de OGC para el manejo de metadatos de sensores. Posteriormente proceden a modelar la arquitectura de la aplicación de acuerdo con los lineamientos definidos por SoaML. Usando además los estereotipos <<IoTDevice>>, <<BusinessService>> y <<CloudResource>> para clasificar los Participantes de acuerdo con su naturaleza. En cuanto a la especificación de Contratos de Servicios, los arquitectos describen las Interfaces y Protocolo de Interacción de acuerdo con los Mensajes u operaciones identificados en los estándares.

3.2. Implementación del diseño de la aplicación

Una vez que el arquitecto de software ha diseñado la arquitectura de la aplicación (Modelo de la Arquitectura de la Aplicación), el personal encargado de la etapa de desarrollo procede a la fase de implementación del diseño. En primer lugar, los desarrolladores crean un proyecto por cada Contrato de Servicio; este proyecto incluye artefactos de software tales como clases en las que se definen las Interfaces, Mensajes y Tipos de dato descritos en el Contrato de Servicio. Adicionalmente incluye un artefacto de software con la implementación del Protocolo de Interacción descrito como diagrama de secuencia dentro de cada Contrato de Servicio. Los Contratos de Servicio son implementados como servicios Web que orquestarán la interacción entre los servicios ofrecidos por los Participantes involucrados.

Posteriormente, por cada Participante, los desarrolladores crean artefactos de software cuyo código fuente es la implementación de las Interfaces que le corresponden de acuerdo con su Rol, según el Contrato de Servicio respectivo. Este artefacto implementa la lógica de negocio del servicio que ofrecerá el participante. En caso de que el Participante, sea quién inicia la interacción, se incluirá una operación que invoca al servicio web que implementa el Contrato de Servicio del cual participa (servicio de orquestación). La lógica de negocio de los participantes será implementada como servicios web, salvo la correspondiente al servicio que inicia la interacción que podría formar parte de una aplicación cliente.

El código de implementación del Participante que inicia la interacción debe tener en cuenta que el primer método invocado debe proveer todos los parámetros de inicialización requeridos en las operaciones de los demás participantes. Por ejemplo, si se pretende la inserción de una observación, el Participante que inicie la interacción debe proveer datos como tipo de medición, unidades de medida, etc., para que el Participante de tipo IoTDevice pueda operar.

Existen otros detalles a tener en cuenta al momento de la implementación:

- La comunicación entre el Contrato de Servicio (servicio de orquestación) y los servicios de los Participantes se establece a través de los puntos de acceso expuestos por cada servicio.
- Para los Participantes de tipo CloudResource la implementación de las interfaces corresponde a la capa de acceso a datos; requiriéndose además el uso de un servicio cloud de base de datos espacial ofrecido por un proveedor cloud.
- Para los Participantes de tipo IoTDevice, los dispositivos deberán proveer el soporte de conexión bajo el protocolo SOAP para el envío de datos de los sensores.
- Para los Participantes de tipo BusinessService que usan software desarrollado por terceros (atributo isExternal=true) es necesario, ya sea desplegarlos si es que se dispone de su paquete de despliegue, o invocarlos a través de su punto de acceso si es que se trata de un servicio disponible en la Web.

Una vez desarrollado el código de implementación, el siguiente paso es la configuración de servicios. Los puntos de acceso deben ser asignados para que el Contrato de Servicios orqueste la interacción entre los servicios ofrecidos por los Participantes. Los Participantes que inician interacciones en el Contrato de Servicios también deben poder acceder al punto de acceso correspondiente. Para esto es necesario considerar aspectos técnicos de conexión y visibilidad. Por ejemplo, en el momento en el que el Contrato de Servicios se conecte al punto de acceso de un participante IoTDevice, se debe proveer mecanismos de acceso tales como VPN (Virtual Private Network), IP pública o nombre de dominio.

4. EJEMPLO PRÁCTICO: RECOLECCIÓN DE MEDICIONES DE POLUCIÓN

Para ilustrar la propuesta presentada se planteó el siguiente ejemplo práctico: Una empresa de control ambiental requiere implementar un sistema de análisis de datos espaciales de calidad del aire, para esto, desea automatizar el proceso de recolección de datos de niveles de contaminación de CO_2 mediante el uso de sensores geo-posicionados. Adicionalmente, requiere una infraestructura que le brinde altas capacidades de almacenamiento y procesamiento, y a su vez le permita gestionar sus datos a través de Internet. Con ese propósito, la empresa ha planteado como requerimiento que el sistema a construir satisfaga la lógica que se muestra en la Figura 2; en donde, los servicios que conforman la aplicación serán desplegados haciendo uso de infraestructuras provistas por proveedores cloud.

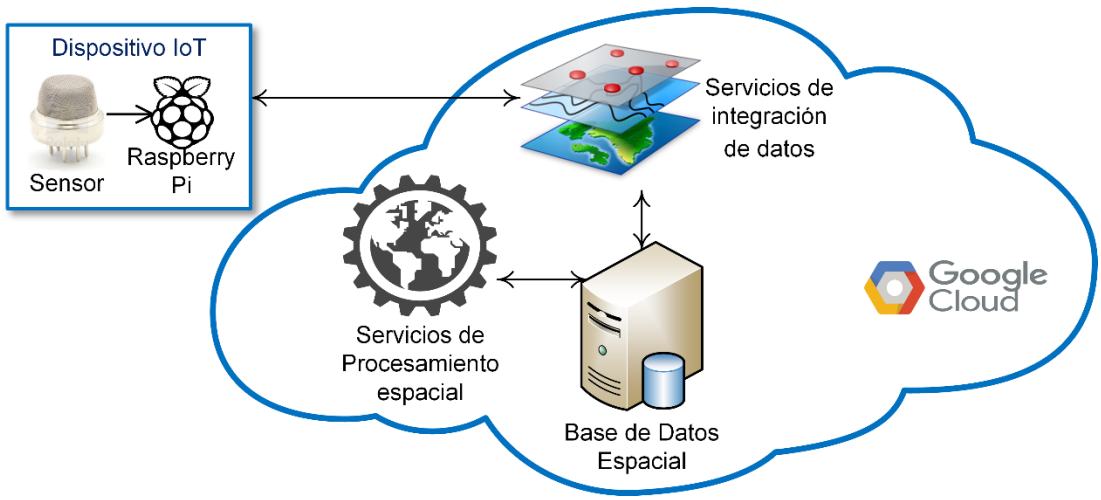

Figura 2. Ejemplo práctico.

4.1. Especificación de la arquitectura de la aplicación

Para el diseño arquitectónico de la aplicación se siguieron los estándares propuestos por la OGC. El

Modelo de la Arquitectura de la Aplicación (ver Figura 3) fue creado usando el Perfil UML definido en la sección 3.1 con el plug-in Papyrus del IDE Eclipse.

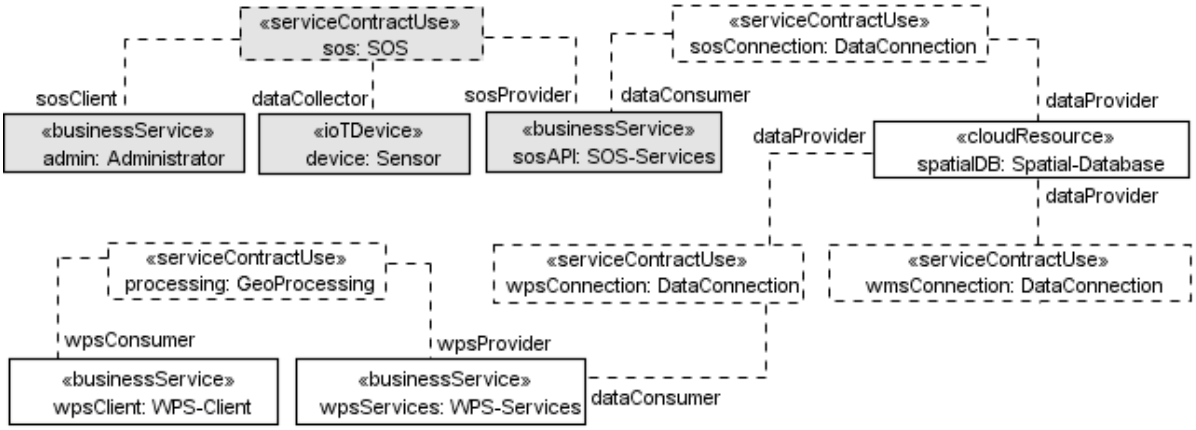

Figura 3. Modelo de la arquitectura de la aplicación.

En primer lugar, se identificaron los Participantes, o servicios, requeridos por el ejemplo práctico planteado, se los categorizó por los tipos, y se definió el estándar OGC que implementan: i) Bussiness Service, servicios bajo el estándar SOS para gestionar las observaciones recolectadas de sensores (sosServices), servicios bajo el estándar WPS para procesamiento espacial (wpsServices) y servicios bajo el estándar WMS para visualización de la información espacial resultante, ii) servicios Cloud Resource, tales como Servicios de Base de Datos Espacial (spatialDB), y iii) IoTDevices para la recolección de información mediante sensores (iotDevice). Posteriormente se especificaron los Contratos de Servicio con sus Roles y Conexiones como se puede apreciar en la Figura 4.

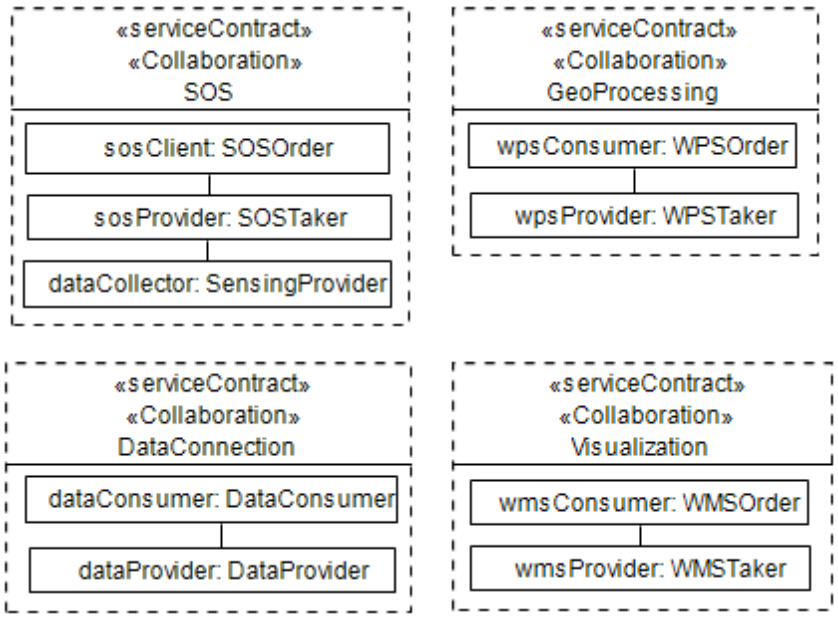

Figura 4. Modelado interno de los contratos de servicio.

La Figura 5 muestra el diseño del protocolo de interacción correspondiente al contrato de servicio SOS. En él se puede apreciar por ejemplo el rol sosProvider implementando la interfaz SOSTaker cuyos mensajes son los que define el estándar SOS (comparar Fig. 4 y Fig. 5 para comprender de mejor manera).

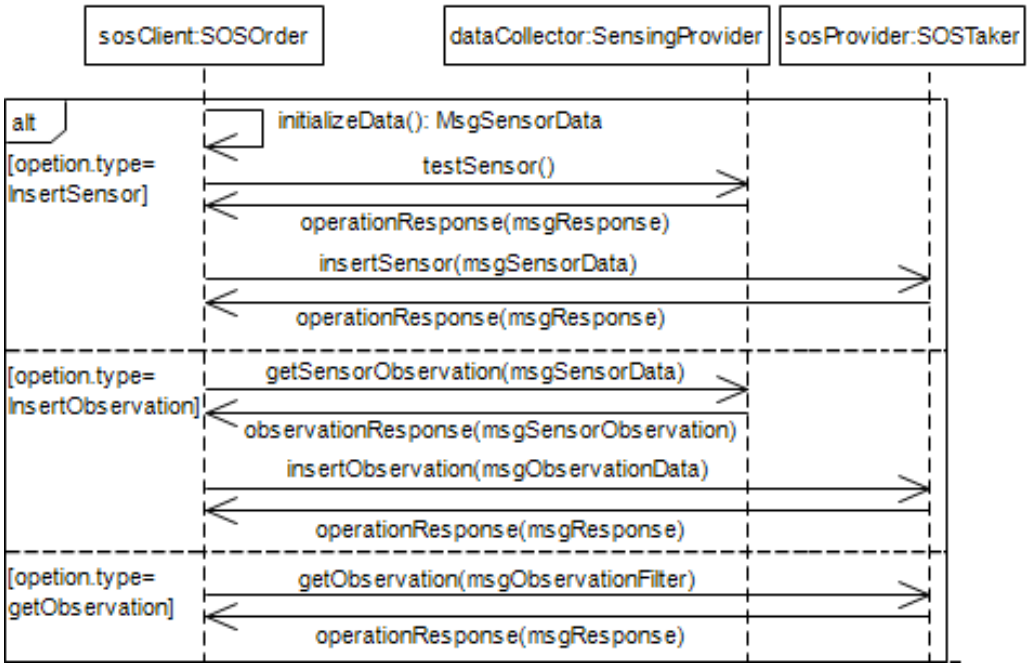

Figura 5. Extracto del protocolo de interacción del contrato de servicio SOS.

4.2. Creación del código de implementación

Para cada *Participante* de tipo *BusinessService* no externo (atributo *isExternal*=false) se creó el respectivo código de implementación. Para los participantes de tipo BusinessService que implementan software de terceros se utilizó los respectivos artefactos de despliegue provistos por terceros, por ejemplo, *52 North* para servicios de integración de información de sensores, *Geoserver* para servicios de procesamiento de información espacial.

Para los *Participantes* de tipo *Cloud Resource* se aprovisionaron servicios cloud de tipo *CloudSql* de la plataforma *Google Cloud*, implementándose y desplegándose además un servicio de acceso a datos. El dispositivo IoT utilizado es un Raspberry Pi 2 Modelo B+, en éste se instaló el sistema operativo *Raspbian* mismo que mediante el lenguaje de programación Python y las capacidades de conexión a internet del *Raspberry Pi* soporta una comunicación bajo protocolo SOAP codificando la información según el estándar SOS. Este dispositivo IoT recoleta la información del sensor MQ-135 conectado a los pines GPIO del mismo mediante un convertidor Analógico-Digital MCP3002, ya que dicho dispositivo no cuenta con entradas analógicas. Se implementó el código necesario (implementación de interfaces que le corresponden a su rol) para la comunicación con los servicios desplegados en *Google Cloud*.

En el caso de los *Contratos de Servicio*, se implementaron sus elementos arquitectónicos internos y se desplegaron como servicios Web de orquestación en la plataforma *Google Cloud*.

Este ejemplo práctico ha permitido ilustrar la aplicabilidad del lenguaje de descripción propuesto en este trabajo (basado en una extensión de SoaML mediante perfiles), aportando enormemente en las tareas de diseño, implementación e integración tanto de artefactos de software propios como artefactos de software de terceros.

5. CONCLUSIONES Y TRABAJOS FUTUROS

En la actualidad no existen propuestas que faciliten el diseño e implementación de sistemas de información geográfica (SIG) que integren los servicios de negocio con dispositivos bajo el paradigma IoT. En este trabajo se propone: un enfoque de desarrollo que, a partir de modelar a un alto nivel de abstracción, tanto la arquitectura del sistema como la integración e interacción entre servicios SIG y

dispositivos IoT, permita guiar las tareas de implementación y despliegue en entornos cloud.

Para la especificación de la arquitectura de la aplicación se creó un Perfil UML que extiende un lenguaje creado específicamente para la descripción de arquitecturas orientadas a servicios - SoaML. Además, se propone la aplicación de estándares *Sensor Web Enablement* formulados por *Open Geospatial Consortium* para el diseño de la interacción con dispositivos IoT. La aplicación de SoaML permitió describir la integración e interacción entre los diferentes componentes del sistema, y mediante las extensiones incluidas categorizamos los componentes en *Servicios de Negocio*, servicios ofrecidos por *Recursos Cloud* y *Dispositivos IoT* con los que se interactuará para soportar los servicios de negocio. Adicionalmente, se proponen las tareas a ejecutar durante las actividades de implementación y despliegue con la finalidad de que los principios de un diseño orientado a servicios se cumplan durante estas.

La aplicabilidad del proceso propuesto fue ilustrada con un ejemplo práctico de un SIG en donde los servicios que conforman su arquitectura fueron desplegados en la plataforma Google Cloud y su proceso de recolección de datos fue realizado por sensores de calidad de aire bajo el paradigma IoT. A pesar de que la propuesta se centra en un SIG, ésta se podría aplicar a otros dominios de aplicación en los que se requiera la integración de servicios desplegados en entornos cloud con dispositivos IoT.

Como trabajo futuro se pretende automatizar el proceso de desarrollo mediante una aproximación de Desarrollo Dirigido por Modelos (DSDM) en la cual, en base a transformaciones de modelo a código, se generen los diferentes artefactos de implementación y despliegue en entornos cloud. Además, se ha planificado la validación del enfoque propuesto en este trabajo mediante casos de estudio que involucren a aplicaciones propuestas por la industria.

AGRADECIMIENTOS

Investigación soportada por el proyecto DIUC_XIV_2016_038 de la Universidad de Cuenca.

REFERENCIAS

Ara, T., Gajkumar Shah, P., Prabhakar, M. (2016).. Internet of Things architecture and applications: A survey. *Indian Journal of Science & Technology*, *9*(45), 1-7. https://doi.org/10.17485/ijst/2016/v9i45/106507

Bhat, M. A., Ahmad, B. (2011). Cloud computing : A solution to Geographical Information Systems (GIS). *International Journal on Computer Science and Engineering*, *3*(2), 594-600.

Bröring, A., Echterhoff, J., Jirka, S., Simonis, I., Everding, T., Stasch, C., Liang, S., Lemmens, R. (2011). New generation sensor web enablement. *Sensors (Basel)*, *11*(3), 2652-99. https://doi.org/10.3390/s110302652

Chen, I. R., Guo, J., Bao, F. (2016). Trust management for SOA-based IoT and its application to service composition. IEEE Transactions on Services Computing, 9(3), 482-495. https://doi.org/10.1109/TSC.2014.2365797

Evangelidis, K., Ntouros, K., Makridis, S., Papatheodorou, C. (2014). Geospatial services in the cloud. *Computers & Geosciences*, *63*, 116-122. https://doi.org/10.1016/j.cageo.2013.10.007

Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, (7), 1645-1660. https://doi.org/10.1016/j.future.2013.01.010

Liu, Z. (2013). *Typical characteristics of cloud GIS and several key issues of cloud spatial decision support system*. 4th International Conference on Software Engineering and Service Science (ICSESS), pp. 668-671. https://doi.org/10.1109/ICSESS.2013.6615395

OMG (2012). *Service oriented architecture modeling language (SoaML)*. Object Management Group (OMG). Available at https://www.omg.org/spec/SoaML/About-SoaML/

Sagl, G., Lippautz, M., Resch, B., Mittleboeck, M. (2011). *Near real-time geo-analyses for emergency support: An radiation safety exercise*. 14th AGILE International Conference on Geographic Information Science, Utrecht, The Netherlands, pp. 1-8.

Yue, P., Zhou, H., Gong, J., Hu, L. (2012). Geoprocessing in cloud computing platforms - a comparative analysis. *International Journal of Digital Earth*, *6*(4), 404-425. https://doi.org/10.1080/17538947.2012.748847

Zuñiga-Prieto, M., Gonzalez-Huerta, J., Abrahão, S., Insfran, E. (2009). Dynamic reconfiguration of cloud application architectures. *Journal of Software: Practice and Experience*, *48*(2), 327-344. https://doi.org/10.1002/spe.2457

Zúñiga-Prieto, M., Insfran, E., Abrahão, S. (2016). Architecture description language for incremental integration of cloud services architectures. 10th International Symposium on Maintenance and Evolution of Service-Oriented and Cloud-Based Environments (MESOCA), pp. 16-23. https://doi.org/10.1109/MESOCA.2016.10