ICT-UNPA-307-2022
ISSN: 1852-4516
Aprobado por Resolución N° 0782/22-R-UNPA
http://dx.doi.org/10.22305/ict-unpa.v14.n2.887

Especificación de requisitos de un sistema IoT con UML

Specification of requirements of an IoT system witch UML

Daniel Laguia, Karim Hallar, Osiris Sofía, Leonardo González, Esteban Gesto
{dlaguia, khallar, osofia, lgonzalez, egesto}@uarg.unpa.edu.ar

Unidad Académica Río Gallegos - Universidad Nacional de la Patagonia Austral
Avda. Gregores y Piloto Lero Rivera - Río Gallegos - Santa Cruz – Argentina

Recibido: 02/08/2022. Aceptado: 30/09/2022

RESUMEN

Actualmente la aplicación de tecnologías IoT se ha incrementado notoriamente, siendo utilizada en diferentes sectores como la agronomía, el transporte, la industria, la medicina y muchas otras áreas de aplicación. En los últimos años se ha avanzado mucho en las tecnologías necesarias que han permitido que esta práctica sea posible, tales como dispositivos de bajo costo, conectividad y plataformas middleware. Todas estas tecnologías ya han sido ampliamente estudiadas.
Un sistema IoT es el conjunto de sensores, actuadores y software que interactúan entre sí para lograr un propósito sin participación humana, lo que hace que el paradigma del desarrollo de software tradicional no sea suficiente. Es por ello que es necesario tener un nuevo enfoque sistemático para el desarrollo de software de sistemas IoT y especialmente para el modelado de requisitos funcionales. En este trabajo proponemos la aplicación e integración de distintos métodos recientemente propuestos para la obtención y especificación de requerimientos para este tipo de sistemas: IotReq, la arquitectura orientada a servicios (SOA), el Lenguaje Unificado de Modelado (UML) y Precise SOM. Se presenta un caso de estudio simple que toma como base el prototipo "Estación de monitoreo COVID" desarrollado durante la pandemia de COVID-19.

Palabras clave: IoT; Requerimientos; Especificaciones; IotReq; Ingeniería de Software

ABSTRACT

Nowadays, the application of IoT technologies has increased significantly, being used in different sectors such as agriculture, transportation, industry, medicine and many other application areas. In recent years, much progress has been made in the necessary technologies that have made this practice possible, such as low-cost devices, connectivity, and middleware platforms. All these technologies have already been extensively studied.
However, an IoT system is the set of sensors, actuators and software that interact with each other to achieve a purpose without human involvement, which makes the traditional software development paradigm not enough. That is why it is necessary to have a new systematic approach for the software development of IoT systems and especially for the modeling of functional requirements. In this work we propose the application and integration of different recently proposed methods for obtaining and elicit requirements for this type of systems: IotReq, service-oriented architecture (SOA), Unified Modeling Language (UML) and

ICT-UNPA-307-2022
ISSN: 1852-4516
Aprobado por Resolución N° 0782/22-R-UNPA
http://dx.doi.org/10.22305/ict-unpa.v14.n2.887

Precise SOM. A simple case is presented based on the "COVID Monitoring Station" prototype developed during the COVID-19 pandemic.

Key words: IoT; Requirements; Specs; IotReq; Software Engineering

1. INTRODUCCIÓN

Internet de las cosas (IoT), en general, consiste en una red de sensores y dispositivos que tiene la tarea de recolectar datos de su entorno con el propósito de tomar decisiones y posteriormente ejecutar acciones que mejoren dicho contexto (Ojo-Gonzalez 2021). Una delas principales características en los sistemas IoT es la interacción entre los dispositivos de la red sin la intervención humana, siendo esta una de las razones que ha llevado a adecuar el enfoque tradicional de desarrollo de software y tomar en consideración requerimientos específicos y característicos para estos sistemas.

En el siguiente trabajo se propone la aplicación de un conjunto de métodos y técnicas conocidas, integradas de forma novedosa, que aborden la tarea de describir claramente los requerimientos funcionales de un sistema IoT y que permitan representar el comportamiento dinámico y autónomo, característica tan particular de los sistemas IoT. A partir de una especificación textual de un sistema IoT simple, se busca encontrar, especificar y modelar las entidades propias del dominio en UML que permitan brindar entendimiento del problema.

Tras evaluar distintos enfoques como (Amsden 2010), (Dung 2019), (Elvesæter 2011), (Eterovic 2015), (Geller 2021), (Reggio 2017), (Reggio 2018), (Thramboulidis 2016), (Zambonelli 2015), (Da Silva 2019), para el caso de estudio elaborado en este trabajo se aplicó la propuesta preliminar de un método denominado IotReq formulado por (Reggio 2018) porque hace uso de las técnicas tradicionales para la elicitación de los requerimientos, a partir de la búsqueda de los objetivos deseados por parte de los usuarios continuando el modelado basado en el paradigma orientado a servicios utilizando casos de uso.

Para realizar el refinamiento de la especificación de estos casos de uso utilizamos Precise SOM, formulado por (Dung 2019), porque define un metamodelo que le da más especificidad al lenguaje SoaML permitiendo la construcción de modelos más refinados. SoaML proporciona una especificación general para el modelado de sistemas basados en SOA. La integración de IotReq y Precise SOM permitieron construir una especificación de requisitos funcionales para una sencilla aplicación que usa tecnología IoT.

En el modelado tradicional los dispositivos IoT se ven como un conjunto de sensores accesibles en la red que generan eventos y una serie de actuadores que influyen en el mundo real. En cambio, la arquitectura SOA se centra en la funcionalidad que estos dispositivos deben ofrecer. En un modelo de servicio, el objetivo es comenzar con la forma en que la aplicación IoT debe interactuar con el mundo exterior, en comparación con los aspectos técnicos de cómo funciona IoT. Por ejemplo, para saber la temperatura de una habitación, no leemos un sensor de temperatura, sino que solicitamos la temperatura ambiental, este pedido de servicio deriva en la lectura del sensor. Este servicio IoT se diseña para ofrecer funcionalidades que cumplan con los requisitos.

Para elaborar el modelado de dominio presentado en este trabajo se utilizó como base un ejercicio práctico, que desarrolló un alumno de la carrera de la Licenciatura en Sistemas,

ICT-UNPA-307-2022
ISSN: 1852-4516
Aprobado por Resolución N° 0782/22-R-UNPA
http://dx.doi.org/10.22305/ict-unpa.v14.n2.887

donde, a partir de un enunciado textual y sin contar con una especificación de requerimientos, el alumno construyo un prototipo “Estación de monitoreo COVID” aplicando tecnología IoT.

El documento se encuentra estructurado de la siguiente manera: la sección 2 introduce un marco teórico sobre la Arquitectura Orientada a Servicios (SOA), el método IotReq y una extensión del Lenguaje Unificado de Modelado (Precise SOM) para la representación del dominio para este tipo de arquitectura. En la sección 3 se explica la metodología aplicada en el caso práctico. En la sección 4 se presentan los resultados de aplicar la metodología al caso de estudio para el sistema IoT formulado. En la sección 5 y 6 se presentan las conclusiones y se propone el trabajo futuro.

2. CONTEXTO

Internet de las Cosas (IoT por sus siglas en inglés: “Internet of Things”) nace de la fusión de dos dominios completamente separados hasta hace no mucho tiempo: los sistemas embebidos,e Internet. Los sistemas embebidos, en auge actualmente dentro de la comunidad informática, no son nuevos: nacieron con los primeros microprocesadores desarrollados a principios de los años ’70. Internet es una red de redes que consta de millones de redes privadas, públicas, académicas, empresariales y gubernamentales, de alcance local a global, que están vinculadas por una amplia gama de tecnologías de redes electrónicas, inalámbricas y ópticas (Madakam 2015).

El concepto de computación ubicua (ubiquitous computing) introducido en 1991 por Mark Weiser en "The Computer of the 21st Century" (Weiser 1991) prefigura la revolución que IoT produciría en años posteriores. En su artículo Weiser bosquejó un futuro en el que multitud de dispositivos interconectados, por medios convencionales o inalámbricos, resultaría habitual y omnipresente en las sociedades avanzadas; para mejorar la vida de las personas no se requeriría así de una revolución en el campo de la inteligencia artificial, sino sólo de una multitud de pequeñas computadoras y sensores incorporados al entorno habitual de los usuarios.

El término Internet of the Things (IoT) es posterior y se debe, presumiblemente, a Kevin Ashton quien en 1999 habría defendido la idea de asociar las etiquetas RFID para identificación de productos, con la floreciente tecnología de Internet, en beneficio de las cadenas de suministro de Procter & Gamble (Ashton 2009).

2.1 Marco teórico

Los objetos IoT recolectan ingentes cantidades de datos que difunden automáticamente hacia otros objetos, aplicaciones, servicios y sistemas para mejorar, optimizar y controlar los procesos cotidianos. Con el término Sistema IoT generalmente nos referimos al conjunto general de dispositivos IoT y al software asociado dedicado a administrar sus redes e interacciones.

Encontramos tres componentes o elementos básicos en un sistema IoT que interactúan entre sí: a) el hardware, como sensores, actuadores y dispositivos de comunicación; b) la plataforma de middleware, software responsable de la gestión del servicio y que permite el intercambio de información entre las aplicaciones y c) aplicaciones que permiten la

ICT-UNPA-307-2022
ISSN: 1852-4516
Aprobado por Resolución N° 0782/22-R-UNPA
http://dx.doi.org/10.22305/ict-unpa.v14.n2.887

visualización e interpretación de los datos (Sofia 2020).

Por encima de un sistema IoT se puede implementar un software específico para coordinar las actividades del sistema para proporcionar servicios y aplicaciones de propósito más general. Podemos pensar en sistemas IoT en los que los servicios pueden existir sólo confinados dentro del contexto de alguna aplicación general, pero también en escenarios en los que hay servicios que se pueden implementar como software independiente (Zambonelli 2015).

Dado un sistema IoT, implementado por una infraestructura de software concreta, se puede diseñar e implementar una gran variedad de sistemas y aplicaciones. Para asegurar que el diseño y desarrollo de servicios y aplicaciones IoT cumpla con las expectativas, es necesario un enfoque disciplinado y riguroso para el análisis, diseño y desarrollo de software (Zambonelli 2015).

2.1.1 Lenguaje Unificado de Modelado

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema software (Booch 2006). El objetivo de UML es proporcionar a los arquitectos de sistemas, ingenieros de software y desarrolladores de software herramientas para el análisis, diseño e implementación de sistemas basados en software, así como para modelar negocios y procesos similares (OMG 2008).

UML fue adoptado como estándar en 1997 por el Object Management Group (OMG) para representar sistemas orientados a objetos. El UML define una notación y una semántica para la construcción de diagramas para describir el comportamiento estructural, el comportamiento dinámico, la gestión del modelo y proporciona construcciones que aportan una capacidad de extensión al UML.

2.1.2 Arquitectura Orientada a Servicios

La Arquitectura Orientada a Servicios (SOA) es un tipo de arquitectura que resulta de aplicar la orientación a servicios. Es una forma de describir y comprender organizaciones, comunidades y sistemas para maximizar la agilidad, la escala y la interoperabilidad. El enfoque SOA es simple: las personas, las organizaciones y los sistemas se prestan servicios unos a otros. Estos servicios nos permiten hacer algo sin hacerlo nosotros mismos o incluso sin saber cómo hacerlo, permitiéndonos ser más eficientes y ágiles. Los servicios también nospermiten ofrecer nuestras capacidades a otros a cambio de algún valor, estableciendo así una comunidad, un proceso o un mercado (OMG 2008).

Un servicio es un valor entregado por un participante que actúa de proveedor a otro a travésde una interfaz bien definida (contrato) y disponible para una comunidad en particular. Un servicio da como resultado un trabajo proporcionado de un participante a otro.

SOA trata un sistema complejo como un conjunto de objetos simples o subsistemas bien definidos. Esos objetos o subsistemas se pueden reutilizar y se mantienen individualmente;por lo tanto, los componentes de software y hardware en un sistema IoT se pueden reutilizar yactualizar de manera eficiente (Gokhale 2018).

ICT-UNPA-307-2022
ISSN: 1852-4516
Aprobado por Resolución N° 0782/22-R-UNPA
http://dx.doi.org/10.22305/ict-unpa.v14.n2.887

2.1.3 Terminología SoaML

SoaML proporciona una forma estándar de diseñar y modelar con UML, utiliza los mecanismos de extensión para definir conceptos SOA en términos de conceptos UML existentes.

Participantes: los participantes son entidades específicas o tipos de entidades que proporcionan o utilizan servicios. Los participantes pueden proporcionar o consumir cualquier número de servicios. Los participantes se representan mediante clases estereotipadas como
<<Participant>> (Fig. 1).[1]

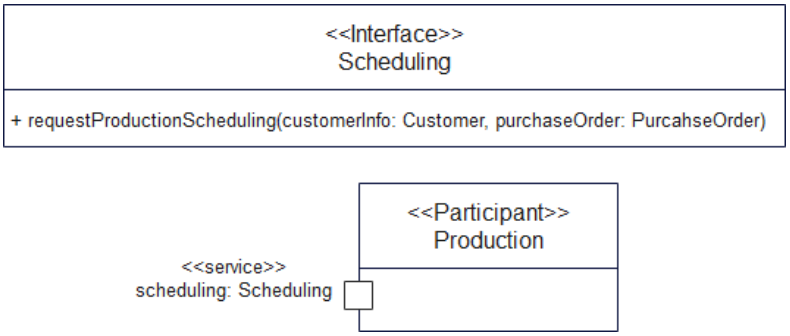

Fig. 1 Participantes y Servicios del Participante

Puertos: los participantes prestan o consumen servicios a través de los puertos (Port). Un puerto es la parte o característica de un participante que es el punto de interacción de un servicio donde se proporciona o se consume. Un puerto es donde se ofrece un servicio es designado como puerto de servicio (<<Service>>) y el puerto donde se consume un servicioes designado como puerto de solicitud (<<Request>>) (Fig. 2).

Fig. 2 Participantes. Servicio y uso del servicio. Extraída de (UML 2007)

Descripción del servicio: descripción de cómo interactúa el participante para proporcionar o utilizar un servicio. Hay tres formas de especificar una interacción de servicio: una interfaz UML (<<Interface>>), una interfaz de servicio (<<Service>>), y un contrato de servicio (<<ServiceContract>>). Cada caso da como resultado interfaces y comportamientos que definen cómo el participante proporcionará o utilizará un servicio a través de los puertos. Una especificación de servicio describe cómo se espera que los consumidores y los proveedores interactúen a través de sus puertos para implementar un servicio, pero no cómo lo hacen. Puede representarse con cualquier diagrama de comportamiento en UML, generalmente el diagrama de secuencia.

ICT-UNPA-307-2022
ISSN: 1852-4516
Aprobado por Resolución N° 0782/22-R-UNPA
http://dx.doi.org/10.22305/ict-unpa.v14.n2.887

Capacidades: los participantes que brindan un servicio deben tener la capacidad para brindarlo, pero diferentes proveedores pueden tener diferentes maneras de realizar el mismo servicio; algunos incluso pueden "subcontratar" o delegar la implementación del servicio a través de una solicitud de servicios de otros.

2.2 Materiales y métodos

En un sistema IoT las cosas en la red deben estar interconectadas. La arquitectura del sistema debe garantizar las operaciones de IoT. Al diseñar la arquitectura de IoT, se debe tener en cuenta la extensibilidad, escalabilidad e interoperabilidad entre dispositivos heterogéneos y sus modelos. El modelado se ha convertido en una parte integral de los enfoques de ingeniería de software. Es común modelar un sistema desde diferentes puntos de vista o perspectivas, como su estructura y comportamiento. Para ello se requieren nuevas metodologías y lenguajescon los conceptos necesarios para el modelado.

2.2.1 Método para la Obtención de requisitos

Para la obtención y especificación de los requisitos funcionales de un sistema IoT se utilizó el método propuesto por (Reggio 2018) llamado IotReq. La primera tarea sugerida en IotReq es modelar el dominio utilizando UML siguiendo el método orientado a servicios, luego se obtienen y especifican los objetivos del sistema IoT a construir también usando UML produciendo una especificación de los requisitos funcionales. Combina el modelado conceptual UML más una especificación de requisitos expresada mediante casos de uso, especificada nuevamente con UML (Reggio 2017) (Fig. 3).

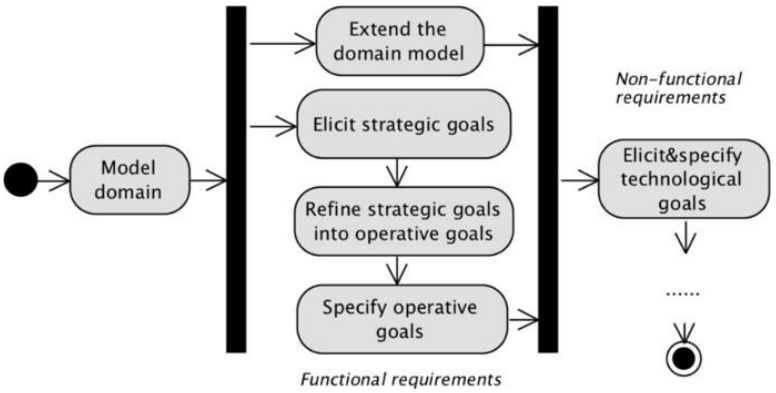

Fig. 3. Descripción general de IotReq (Reggio 2018)

2.2.2 Lenguaje de Modelado para Arquitecturas de Servicios

SoaML es el lenguaje de modelado para arquitecturas orientadas a servicios y define extensiones de UML para admitir la gama de requisitos de modelado para esta arquitectura, incluida la especificación de sistemas de servicios, la especificación de interfaces de servicios individuales y la especificación de implementaciones de servicios. La versión actual del estándar SoaML es 1.0.1. (OMG 2008).

Precise SOM es un método de modelado para el diseño de sistemas orientado a servicios

ICT-UNPA-307-2022
ISSN: 1852-4516
Aprobado por Resolución N° 0782/22-R-UNPA
http://dx.doi.org/10.22305/ict-unpa.v14.n2.887

basado en SoaML este método propuesto por (Dung 2019) determina un metamodelo que define de manera precisa la estructura del modelo del sistema de servicios, provisto con restricciones formales para guiar el uso de la notación (Fig. 4).

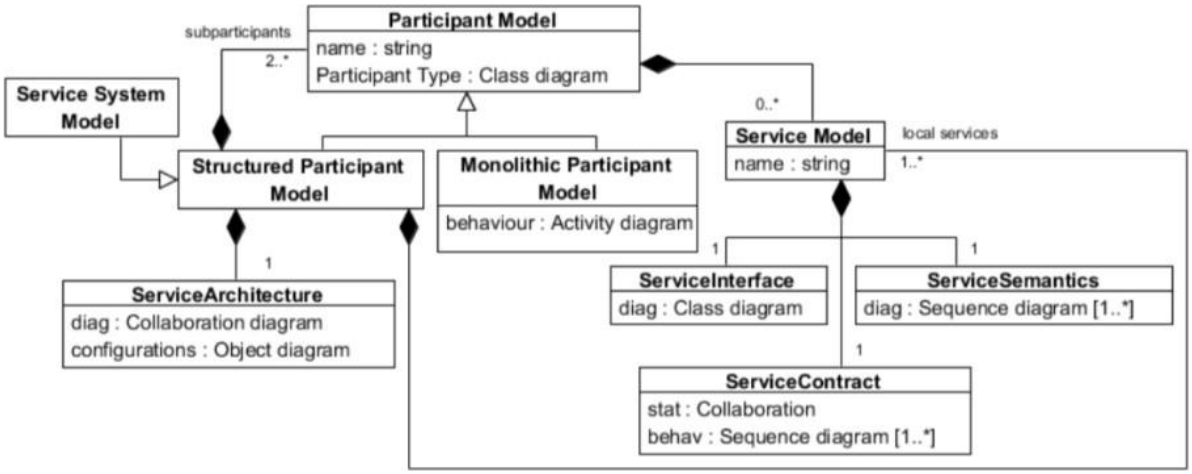

Fig. 4: Metamodelo Precise SOM. Extraída de (Dung 2019)

En Precise SOM hay dos tipos de participantes: estructurados y monolíticos. Un participante monolítico se caracteriza por: un nombre, los objetos que manipula (datos), los servicios que ofrece o utiliza (por ejemplo, una persona); un participante estructurado se describe como un subconjunto de participantes que interactúan intercambiando servicios ofrecidos y utilizados.

Un servicio se caracteriza por una interfaz, un contrato y una semántica. La Interfaz de Servicio contiene la información para acceder al servicio (conjunto de mensajes de entrada y salida). Un mensaje tiene un nombre único y el listado de parámetros. Los mensajes deentrada son para solicitar servicios, los mensajes de salida se utilizan para responder a las solicitudes. Un Contrato de Servicio especifica las secuencias de interacciones permitidas entre el proveedor de servicio y quien los utiliza. Un valor retornado por un servicio puede modificar el estado del dominio, la semántica del servicio expresa esto, se modela refinando los contratos agregando especificaciones que representan estas modificaciones.

Cada Participante, tendrá su diagrama de clases (vista estática), donde se define una clase estereotipada <<participant>> nombrada con el nombre del participante. La clase participante establece un tipo de participante, que podrá utilizarse para instanciar participantes específicos,tal como se muestra en la (Fig. 5), generada para el caso de estudio con el software Visual Paradigm.

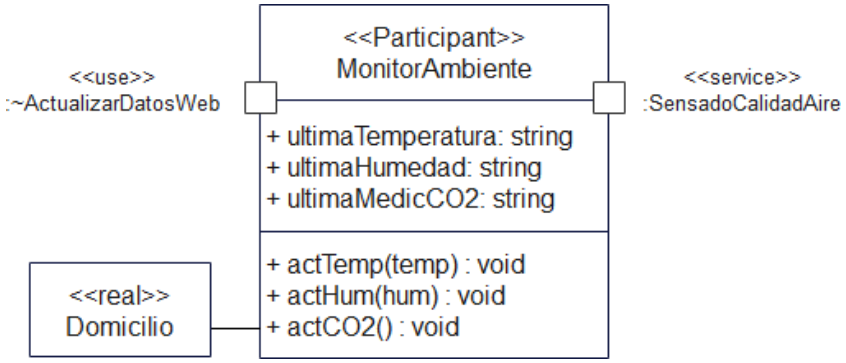

Fig. 5. Vista Estática de un Participante.

ICT-UNPA-307-2022
ISSN: 1852-4516
Aprobado por Resolución N° 0782/22-R-UNPA
http://dx.doi.org/10.22305/ict-unpa.v14.n2.887

Los puertos en la clase participantes que se definen son dos, un puerto de servicio estereotipado por <<service>> donde se ofrecen los servicios del participante y otro puerto de uso estereotipado por <<use>> donde otro participante hace uso de los servicios ofrecidos del participante. Un puerto de servicio tiene el tipo de la interfaz de servicio proporcionado y un puerto de uso tiene el tipo conjugado de la interfaz del servicio requerido. El nombre de la interfaz de servicio (conjugado) tiene el nombre del servicio con el prefijo "~".

Cada clase Participante tiene asociado un diagrama de objetos (vista de objetos), las clases están estereotipadas como objetos <<object>> y describen los tipos de objetos que un participante maneja.
Una clase Participante tiene una vista de reino que está modelada por un diagrama de clases que incluye la clase nombrada como el reino, estereotipada como reino <<realm>> y toda otraclase y tipo de datos necesarios para definirlo.

El modelo de servicio (vista de servicio) contiene los modelos de todos los servicios proporcionados y utilizados de un participante. Es una clase estereotipada <<service>> y con el mismo nombre del servicio. Una clase <<service>> tiene interfaces, contratos y una semántica. Tiene dos interfaces, una de entrada <<in>> y otra de salida <<out>> para los mensajes de entrada/salida, estas operaciones se denominan <<servicio_IN>> y <<servicio_OUT>> respectivamente (Fig. 10).

Los contratos se representan por un conjunto de diagramas de secuencia UML (vista de comportamiento), donde se especifican todas las maneras de utilizar el servicio (Fig. 9). Tiene una línea de vida para cada una de las interfaces del servicio (IN y OUT), los mensajes que intercambian deben ser operaciones internas a cada interfaz. Las guardas en cada operación en el diagrama de secuencia se utilizan para detallar las restricciones del contrato.

La semántica se especifica detallando cómo los mensajes entrantes modifican el estado del dominio y cómo este estado influye en los mensajes de salida. Esta semántica se detalla ampliando las restricciones de los contratos en los diagramas de secuencia.
Para modelar la arquitectura de un servicio de un participante estructurado se utiliza el diagrama de estructura o componentes de UML (Fig. 18).

3. METODOLOGÍA

Para la especificación de requerimientos funcionales del sistema IoT que se presenta en este trabajo se toma como base la especificación textual utilizada por un alumno de la carrera Licenciatura en Sistemas en la asignatura Seminario de Sistemas quién desarrolló un prototipo funcional de una estación de monitoreo Covid, utilizando como hardware un microcontrolador NodeMCU, sensores de temperatura, humedad, pulso cardíaco, saturación de oxígeno y monóxido de carbono, registrando los datos en la plataforma IoT Thingspeak.

A partir de este simple prototipo se plantea realizar la especificación de los requisitos en UML siguiendo el método IotReq, propuesto por (Reggio 2018), donde define un conjunto de subprocesos a realizar para la definición de los objetivos estratégicos y operativos.

Siguiendo lo propuesto por (Reggio 2018), la tarea inicial en la construcción del modelo de dominio fue detectar y especificar los objetivos estratégicos y operativos (requerimientos funcionales), representándolos mediante un diagrama de casos de uso.

ICT-UNPA-307-2022
ISSN: 1852-4516
Aprobado por Resolución N° 0782/22-R-UNPA
http://dx.doi.org/10.22305/ict-unpa.v14.n2.887

Para completar esta especificación inicial, se elige una arquitectura orientada a servicios (SOA), la elaboración de las diferentes vistas necesarias para completar el modelado se aplica el lenguaje de modelado Precise SOM formulado por (Dung 2019), una adaptación de SoaML específica para SOA, que permite mostrar los aspectos estructurales y dinámicos del sistema IoT.

4. RESULTADOS

En esta sección se presentan las actividades desarrolladas para la elaboración del modelo de dominio para los requisitos funcionales aplicando IotReq, SOA y Precise SOM.

4.1 Especificación de los requisitos del sistema con IotReq

En la etapa de relevamiento los interesados pueden expresar estos objetivos en distintos niveles de detalle, cuando los requisitos expresados detectados son de alto nivel suelen enunciar los motivos del sistema a desarrollar, lo clasificamos como objetivos estratégicos. Cada objetivo estratégico es analizado para encontrar todas las tareas a realizar para cumplir con este objetivo, estas tareas son los requisitos funcionales del sistema y los denominamos objetivos operativos.

Para el caso práctico los objetivos estratégicos y operativos son:

Objetivo estratégico: Desarrollar una estación de monitoreo de pacientes con COVID no hospitalizados informando a una página web los datos relevados de cada paciente.

Refinando este objetivo podemos encontrar tres objetivos operativos:

- Controlar los signos vitales al paciente: Se toma como signos vitales el pulso cardíaco y la saturación de oxígeno.
- Controlar el entorno ambiental: Se toma como variables ambientales la temperatura, humedad y dióxido de carbono en el aire.
- Actualizar página web: Panel de control que se actualizará cada determinada frecuencia de tiempo con los últimos valores disponibles tanto de los pacientes como del ambiente.

Las tareas de cada uno de los objetivos operativos son:

- Controlar los signos vitales al paciente:
    - Sensar pulso cardíaco.
    - Sensar saturación de oxígeno.
    - Enviar datos a base de datos.
- Controlar el entorno ambiental del domicilio del paciente:
    - Sensar temperatura.
    - Sensar humedad.
    - Sensar dióxido de carbono.
    - Enviar datos a base de datos.

Para modelar los objetivos del sistema se utilizó un diagrama de casos de uso sin actores (Fig. 6), se determinó un caso de uso para el objetivo estratégico “Desarrollar estación de monitoreo pacientes COVID no hospitalizados” y los casos de uso para los objetivos operacionales, “Controlar signos vitales paciente”, “Controlar entorno ambiental paciente” y

ICT-UNPA-307-2022
ISSN: 1852-4516
Aprobado por Resolución N° 0782/22-R-UNPA
http://dx.doi.org/10.22305/ict-unpa.v14.n2.887

"Actualizar Página Web" conectados con líneas de flecha punteadas con el caso de uso de objetivo estratégico. Cada objetivo operativo lo descomponemos en otros casos de uso que describen cada una de las tareas que ejecutarán para cumplir con el objetivo (Fig. 7) y (Fig. 8).

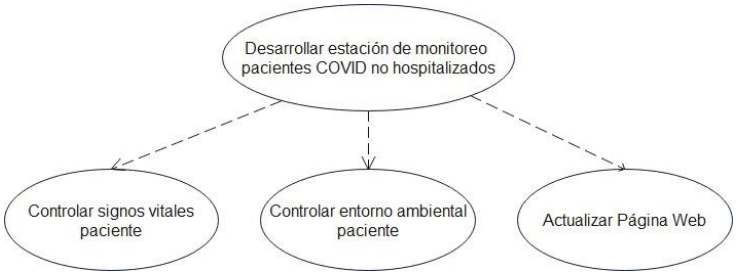

Fig. 6. Diagrama de casos de uso "Desarrollar estación de monitoreo pacientes COVID no hospitalizados"

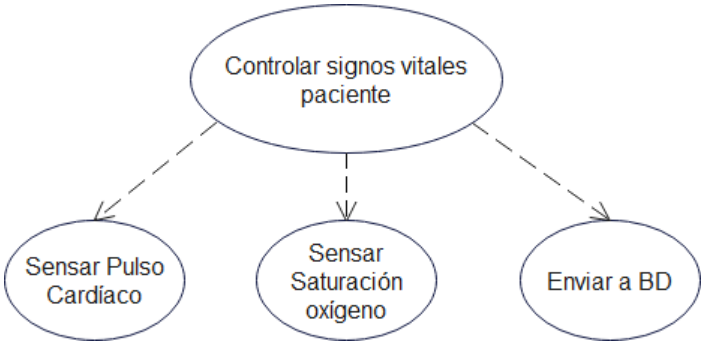

Fig. 7. Diagrama de casos de uso "Controlar signos vitales paciente"

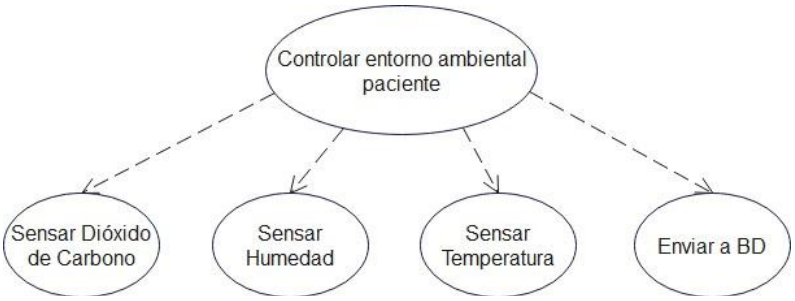

Fig. 8. Diagrama de casos de uso "Controlar entorno ambiental paciente"

4.2 Modelado orientado a servicios con Precise SOM

En esta sección se presentan las distintas vistas desarrolladas para completar el modelo de dominio elaborado en el punto anterior.

Vista Estática

El modelado de la vista estática lo hacemos mediante un diagrama de clases, donde se definen las clases participantes estereotipadas como <<Participant>> y las clases de los objetos que manipula cada participante estereotipada como <<object>>. Para el ejemplo de

ICT-UNPA-307-2022
ISSN: 1852-4516
Aprobado por Resolución N° 0782/22-R-UNPA
http://dx.doi.org/10.22305/ict-unpa.v14.n2.887

este trabajo, podemos describir en el modelo las clases participantes: EstaciónCOVID, MonitorPaciente, MonitorAmbiente y PáginaWeb, y a los objetos que estas clases manipulan. Estos participantes interactúan utilizando los servicios: SensadoSignosVitales, SensadoCalidadAire y ActualizarDatosWeb. EstaciónCOVID utiliza los servicios ofrecidos por MonitorPaciente, MonitorAmbiente y PáginaWeb (Fig. 9).

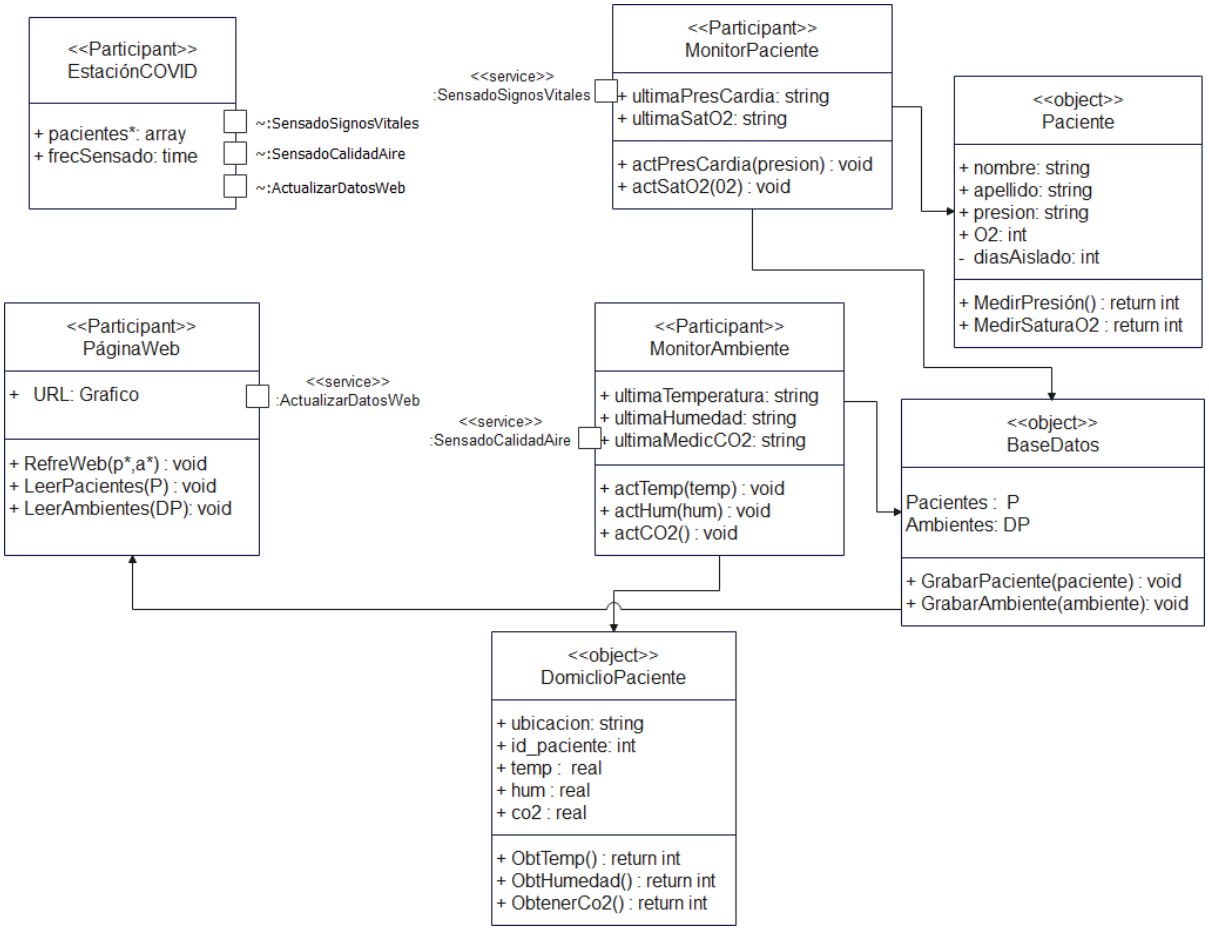

Fig. 9. Vista Estática: Modelo de dominio del sistema.

Vista de Servicios

Para el caso de uso “Controlar entorno ambiental paciente” se realizó la especificación de la interfaz del servicio “SensadoCalidadAire” del Participante MonitorAmbiente. Esta interfaz se utiliza para solicitar los valores de temperatura, humedad, y dióxido de carbono del ambiente donde está aislado el paciente con el propósito de monitorear la calidad del aire. En la (Fig. 10) se describe la interfaz del servicio SensadoCalidadAire del participante MonitorAmbiente. Las operaciones de SensadoCalidadAire_IN corresponde a los mensajes de entrada para solicitar los valores de temperatura, humedad y dióxido de carbono, mientras SensadoCalidadAire_OUT son las operaciones que representan los mensajes de salida, devolviendo los valores del ambiente censado. Por ej. El mensaje SensarTemp(DP) recibe como parámetro un objeto del tipo DomicilioPaciente.

ICT-UNPA-307-2022
ISSN: 1852-4516
Aprobado por Resolución N° 0782/22-R-UNPA
http://dx.doi.org/10.22305/ict-unpa.v14.n2.887

Fig. 10. Interfaz del servicio “SensadoCalidadAire”.

La interfaz de contratos del servicio la especificamos mediante un diagrama de secuencia donde se establecen dos líneas de vida, una para la interfaz IN y otra para la interfaz OUT del servicio SensadoCalidadAire. Este diagrama expresa todas las posibilidades de uso que tiene el servicio. En la (Fig. 11) observamos que se recibe una solicitud en SensarCalidadAire(DP) para obtener los valores de temperatura, humedad y dióxido de carbono del ambiente, este usodel servicio se hace a través de las operaciones de su interfaz de entrada (IN), a lo que se responde con el mensaje ParametrosSensados() que representa la respuesta de la solicitud, operación que corresponde a la salida (OUT) de la interfaz del servicio. La condición OPT se utiliza para expresar restricciones del contrato entre la entrada y la salida.

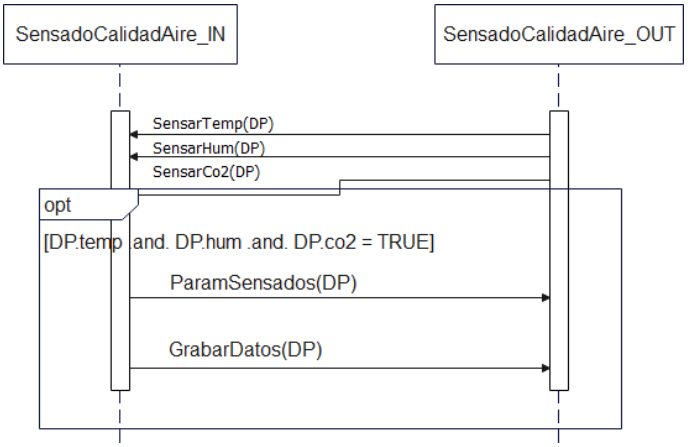

Fig. 11. Diagrama de secuencia Contratos del servicio “SensadoCalidadAire”.

El modelado de la interfaz de semántica también la describimos mediante un diagrama de secuencia donde se agregan los estados de los dominios y protecciones necesarias en las operaciones y parámetros (Fig. 12).

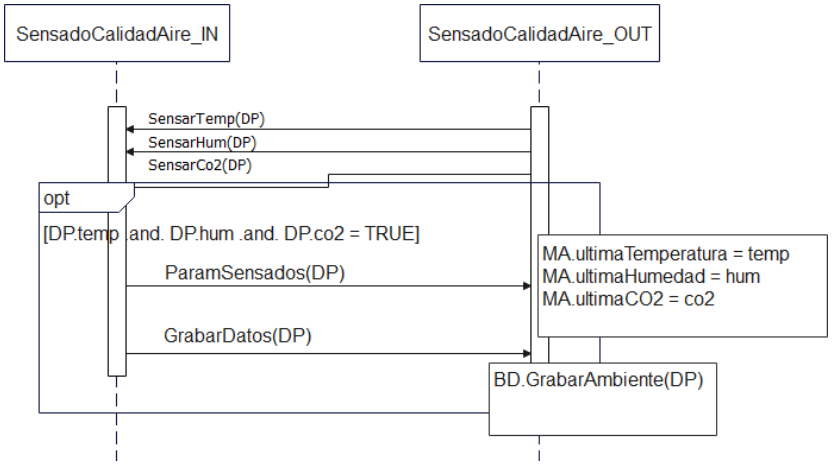

Fig. 12. Semántica del contrato.

ICT-UNPA-307-2022
ISSN: 1852-4516
Aprobado por Resolución N° 0782/22-R-UNPA
http://dx.doi.org/10.22305/ict-unpa.v14.n2.887

Para el caso de uso “Controlar signos vitales paciente” se especifica la Interfaz del Servicio SensadoSignosVitales correspondiente al participante MonitorPaciente (Fig. 13), (Fig. 14) y (Fig. 15).

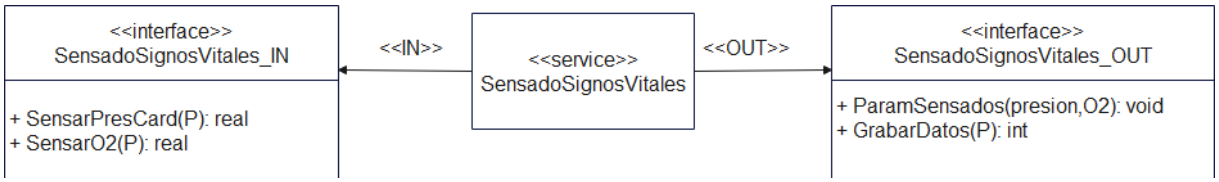

Fig. 13. Interfaz del servicio “SensadoSignosVitales”.

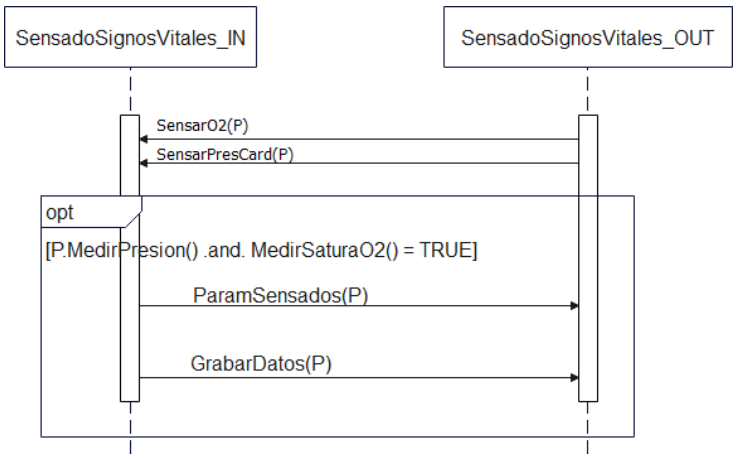

Fig. 14. Diagrama de secuencia Contratos del servicio “SensadoSignosVitales”.

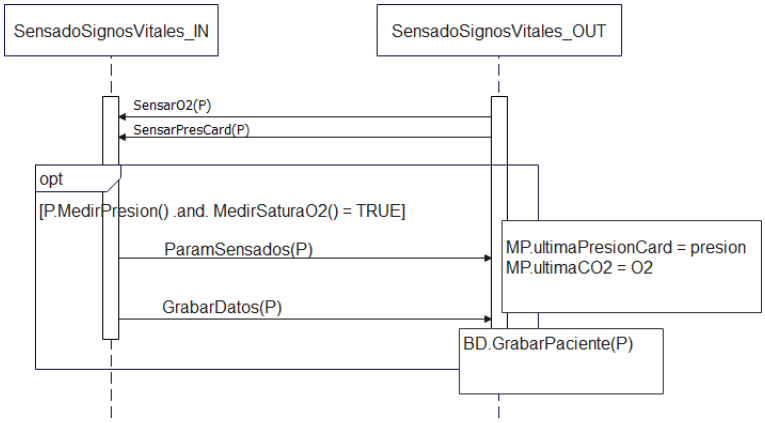

Fig. 15. Semántica del contrato.

Para el caso de uso “Actualizar página web” se especifica la Interfaz del Servicio “ActualizarDatosWeb” correspondiente al participante PaginaWeb (Fig. 16) y (Fig. 17).

Fig. 16. Interfaz del servicio “ActualizarPaginaWeb”.

ICT-UNPA-307-2022
ISSN: 1852-4516
Aprobado por Resolución N° 0782/22-R-UNPA
http://dx.doi.org/10.22305/ict-unpa.v14.n2.887

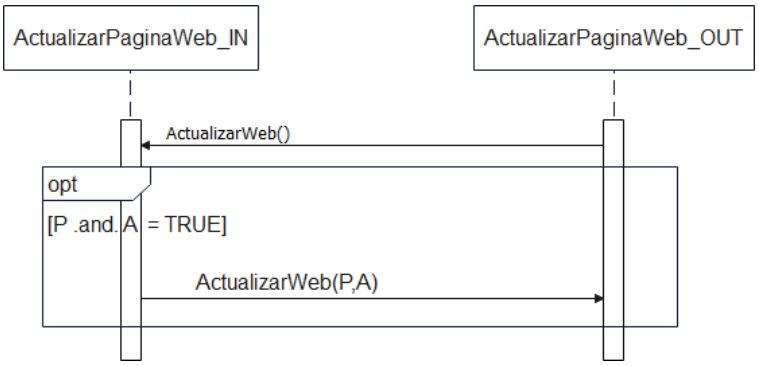

Fig. 17. Diagrama de secuencia Contratos del servicio “ActualizarPaginaWeb”.

Vista Arquitectura de Servicios

La arquitectura de servicios se modela mediante un diagrama de estructura compuesta, laclase estructurada es el sistema IoT y las entidades que forman el dominio son las clases Participantes. Cada entidad tendrá un nombre, un puerto escrito de una entidad debe estar conectado a otra entidad que utiliza dicho puerto mediante una asociación. Todos los servicios proporcionados deben ser utilizados por otros participantes y todos los puertos de servicios utilizados, debe ser conectado a otro participante (Fig. 18).

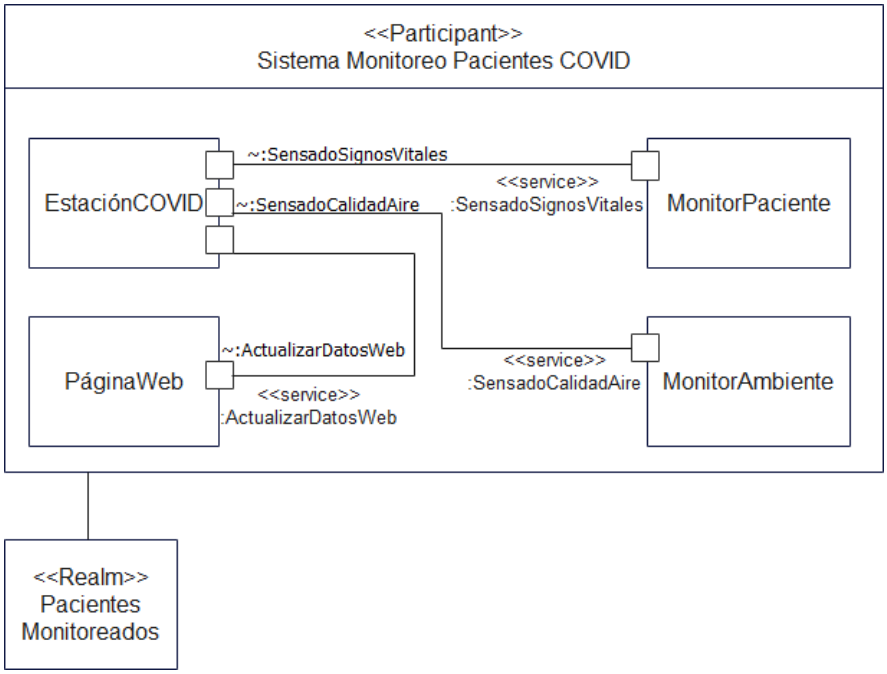

Fig. 18. Arquitectura de Servicios del sistema

5. CONCLUSIONES

En el caso práctico presentado los requisitos parten de un enunciado textual entregado a un alumno para la construcción del prototipo en base al marco teórico de tecnologías IoT de la asignatura Seminario de Sistemas. Para la búsqueda de los requerimientos funcionales, el método IotReq aplicado permitió encontrar objetivos estratégicos y operativos (requerimientos funcionales) y modelarlos con casos de uso. A partir de estos objetivos operativos, se pudo especificar de manera precisa tanto los aspectos estáticos (vista estática)

ICT-UNPA-307-2022
ISSN: 1852-4516
Aprobado por Resolución N° 0782/22-R-UNPA
http://dx.doi.org/10.22305/ict-unpa.v14.n2.887

como el comportamiento (vista de servicios) de cada requerimiento funcional del sistema utilizando Precise SOM.

IotReq y Precise SOM, en conjunto, permitieron realizar una especificación de requisitos funcionales para una sencilla aplicación que usa tecnología IoT separando aquellas características funcionales requeridas de los aspectos tecnológicos pudiendo expresar estos requerimientos con una extensión de UML claramente entendible por aquellos que ya conocen el lenguaje.

Tanto IotReq como Precise SOM están basadas en técnicas tradicionales de la ingeniería de software, por lo que los modelos que se producen, podrán ser utilizados en etapas posteriores del desarrollo del software. También permitió el uso de herramientas de software actualmente disponibles como, por ejemplo, Visual Paradigm que cuentan con extensiones actualizadas de UML.

Aunque IotReq define que para especificar los requisitos no funciones se debe realizar una descomposición adicional de cada caso de uso operativo definiendo objetivos tecnológicos y modelándolos como nuevos casos de uso con dependencia a los casos de uso operativos, no abordamos esta tarea ya que consideramos que es insuficiente modelar sólo con casos de uso los requisitos no funcionales, dada la relevancia que tienen éstos en los sistemas IoT. Se requieren de técnicas que capturen y especifiquen con mayor nivel de detalle estas características tecnológicas.

6. TRABAJO FUTURO

Siguiendo los paradigmas de la Ingeniería del software entendemos a la ingeniería de requerimientos como la disciplina que tiene por objetivo capturar, especificar y gestionartodos los requisitos de un sistema y una especificación completa debe incluir todas lasactividades y artefactos asociados. Para próximos trabajos nos planteamos la necesidad de aplicar estas propuestas a sistemas más complejos para validar los resultados obtenidos. Además, es necesario complementar esta especificación con la incorporación de técnicas que permitan describir los aspectos tecnológicos (requerimientos no funcionales).

7. BIBLIOGRAFÍA

AMSDEN, J. (2010). «Modeling with SoaML, the Service-Oriented Architecture Modeling Language: Part 5. Service implementation. ». IBM DeveloperWorks.

ASHTON, K. (2009). «"Internet of Things" Thing: In the Real World Things Matter More than Ideas» RFID Journal.

BOOCH, G., RUMBAUGH, J., y JACOBSON, I. (2006). «El lenguaje unificado de modelado.» Addison - Wesley. ISBN 8478290761.

DA SILVA, D. y GONÇALVES, T. y ROCHA, A. (2019). «A Requirements Engineering Process for IoT Systems» SBQS'19: Proceedings of the XVIII Brazilian Symposium on Software Quality. 204-209. https://doi.org/10.1145/3364641.3364664

DUNG, T., QUAN, D. y TRUNG, D. (2019). «A Precise Method for Modelling Service Systems using UML» International Journal of Engineering Research & Technology

(IJERT). ISSN: 2278-0181. Vol 8. Issue 07. https://doi.org/10.17577/IJERTV8IS070241

ELVESATER, B. y CARREZ, C. (2011). «Model-driven Service Engineering with SoaML» Service Engineering. ISBN 978-3-7091-0414-9.

ETEROVIC, T., KALJIC, E., DONKO, D., SALIHBEGOVIC, A. y RIBIC, S. (2015). «An Internet of Things Visual Domain Specific Modeling Language based on UML» https://doi.org/10.1109/ICAT.2015.7340537

GELLER, M. y MENESES, A. (2021). «Modelling IoT Systems with UML: A Case Study for Monitoring and Predicting Power Consumption» American Journal of Engineering and Applied Sciences, 14(1), 81-93. https://doi.org/10.3844/ajeassp.2021.81.93

GOKHALE, P., BHAT, O. y BHAT, S. (2018). «Introduction to IOT» International Advanced Research Journal in Science, Engineering and Technology ISO 3297:2007 Certified. Vol. 5, Issue 1. DOI: 10.17148/IARJSET.2018.517.

MADAKAM, S., RAMASWAMY, R. y TRIPATHI, S. (2015). «Internet of Things (IoT): A Literature Review» Journal of Computer and Communications, Vol. 3 No. 5, 164-173. https://doi.org/10.4236/jcc.2015.35021

OJO-GONZALEZ, K. Y BONILLA-MORALES B. (2021). «Requerimientos no funcionales para sistemas basados en el Internet de las cosas (IoT).» Revista de I+D Tecnológico VOL 17 No 2. https://doi.org/10.33412/idt.v17.2.3303

OMG (2008). «Service oriented architecture Modeling Language» (SoaML) Specification. Versión 1.1.0.

REGGIO, G., LEOTTA M., CLERISSI D. y RICCA FILIPPO. (2017). « Service-Oriented Domain and Business Process Modelling» ACM, 2017. pages 751-758. https://doi.org/10.1145/3019612.3019621

REGGIO, G. (2018). «A UML-based Proposal for IoT System Requirements Specification» ACM/IEEE 10th International Workshop on Modelling in Software Engineering. ISBN 2575-4475. https://doi.org/10.1145/3193954.3193956

SOFIA, O., HALLAR, K., GESTO, E., LAGUIA, D. y González, L. (2020). «Internet del futuro: aplicaciones de IoT en la Patagonia Austral.» WIIC 2019. ISBN 978-987-3714-82-5. 390-384.

THRAMBOULIDIS, K. y CHRISTOULAKIS, F. (2016). «UML4IoT - A UML profile to exploit IoT in cyber-physical manufacturing systems» Computers in Industry Volume 82, Pages 259-272. https://doi.org/10.1016/j.compind.2016.05.010

WEISER, M. (1991). «The Computer for the 21st Century. Scientific American» Scientific American. https://doi.org/10.1038/scientificamerican0991-94

ZAMBONELLI, F. (2015). «Towards a General Software Engineering Methodology for the Internet of Things» Modena, Italia: Dipartimento di Scienze e Metodi dell'Ingegneria.